



Chapter 12

Cryptographic Hash Functions

Copyright © The McGraw-Hill Companies, Inc. Permission required for reproduction or display.

Objectives

- To introduce general ideas behind cryptographic hash functions
- To discuss the Merkle-Damgard scheme as the basis for iterated hash functions
- To distinguish between two categories of hash functions:
- To discuss the structure of SHA-512.
- To discuss the structure of Whirlpool.

12-1 INTRODUCTION

A *cryptographic hash function takes a message of arbitrary length and creates a message digest of fixed length. The ultimate goal of this chapter is to discuss the details of the two most promising cryptographic hash algorithms – SHA-512 and Whirlpool.*

Topics discussed in this section:

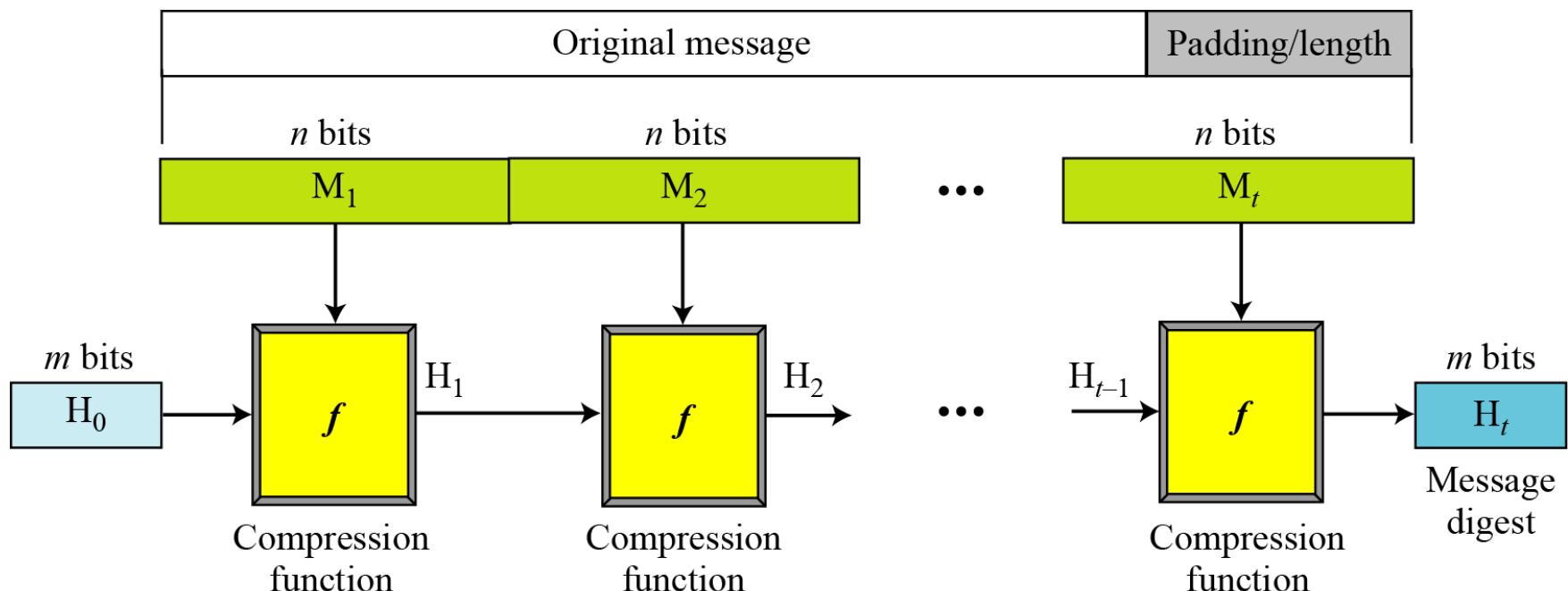
12.1.1 Iterated Hash Function

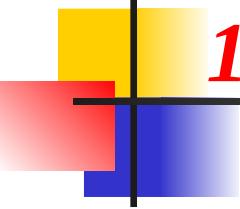
12.1.2 Two Groups of Compression Functions

12.1.1 Iterated Hash Function

Merkle-Damgard Scheme

Figure 12.1 Merkle-Damgard scheme





12.1.2 Two Groups of Compression Functions

1. *The compression function is made from scratch.*

Message Digest (MD)

2. *A symmetric-key block cipher serves as a compression function.*

Whirlpool

Table 12.8 A Comparison of MD5, SHA-1, and RIPEMD-160

	MD5	SHA-1	RIPEMD-160
Digest length	128 bits	160 bits	160 bits
Basic unit of processing	512 bits	512 bits	512 bits
Number of steps	64 (4 rounds of 16)	80 (4 rounds of 20)	160 (5 paired rounds of 16)
Maximum message size	∞	$2^{64} - 1$ bits	$2^{64} - 1$ bits
Primitive logical functions	4	4	5
Additive constants used	64	4	9
Endianness	Little-endian	Big-endian	Little-endian

**Table 12.9 Relative Performance of Several Hash Functions
(coded in C++ on a 850 MHz Celeron)**

Algorithm	MBps
MD5	26
SHA-1	48
RIPEMD-160	31

Note: Coded by Wei Dai; results are posted at <http://www.eskimo.com/~weidai/benchmarks.html>

12.1.2 *Continued*

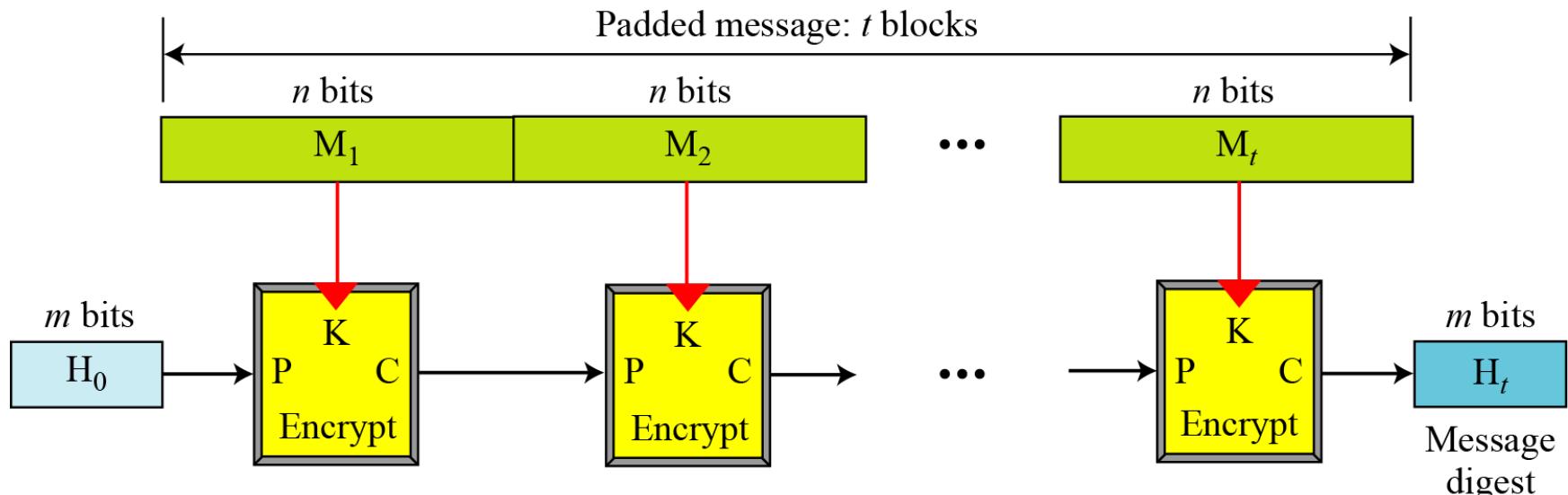
Table 12.1 *Characteristics of Secure Hash Algorithms (SHAs)*

<i>Characteristics</i>	<i>SHA-1</i>	<i>SHA-224</i>	<i>SHA-256</i>	<i>SHA-384</i>	<i>SHA-512</i>
Maximum Message size	$2^{64} - 1$	$2^{64} - 1$	$2^{64} - 1$	$2^{128} - 1$	$2^{128} - 1$
Block size	512	512	512	1024	1024
Message digest size	160	224	256	384	512
Number of rounds	80	64	64	80	80
Word size	32	32	32	64	64

12.1.2 Continued

Rabin Scheme

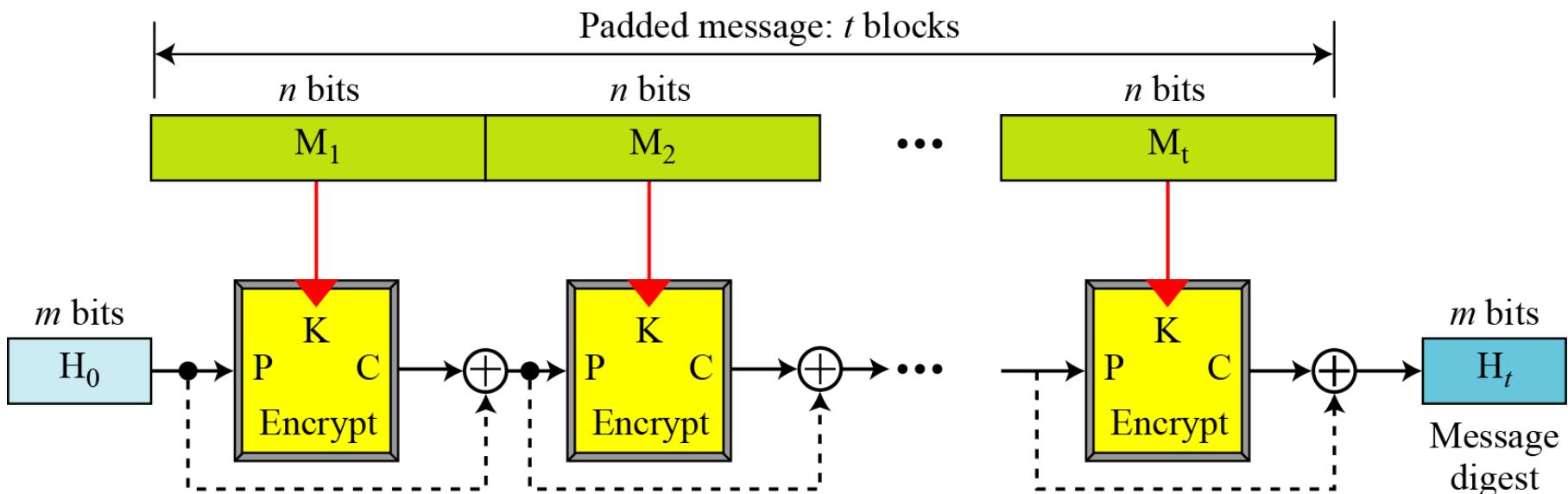
Figure 12.2 Rabin scheme



12.1.2 Continued

Davies-Meyer Scheme

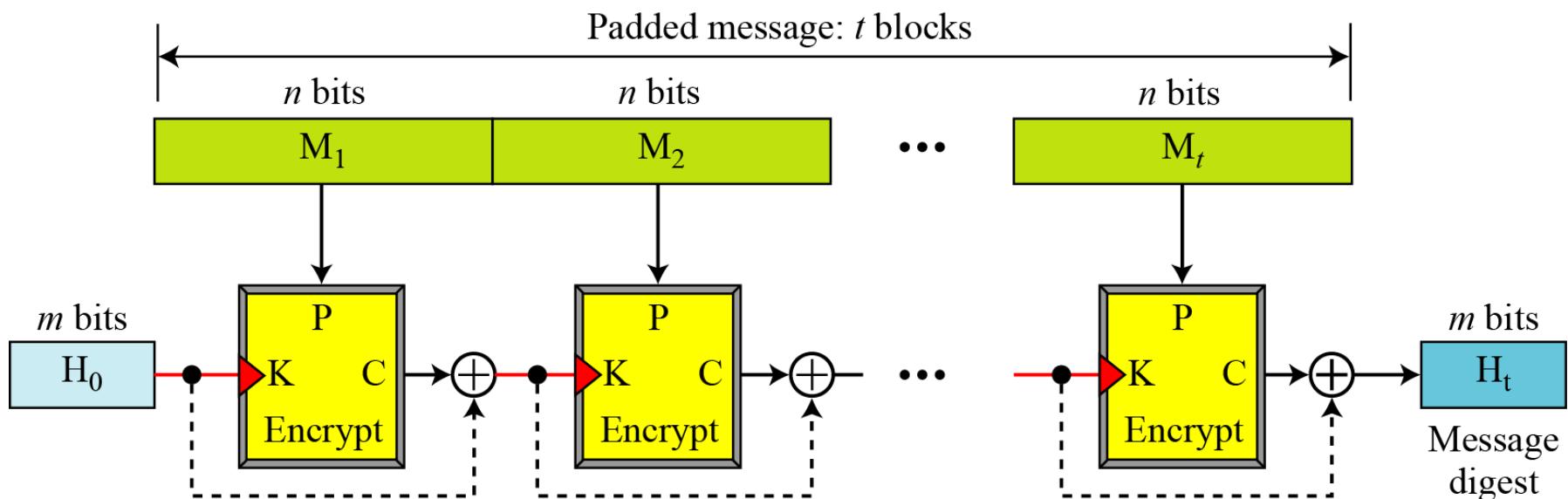
Figure 12.3 Davies-Meyer scheme



12.1.2 Continued

Matyas-Meyer-Oseas Scheme

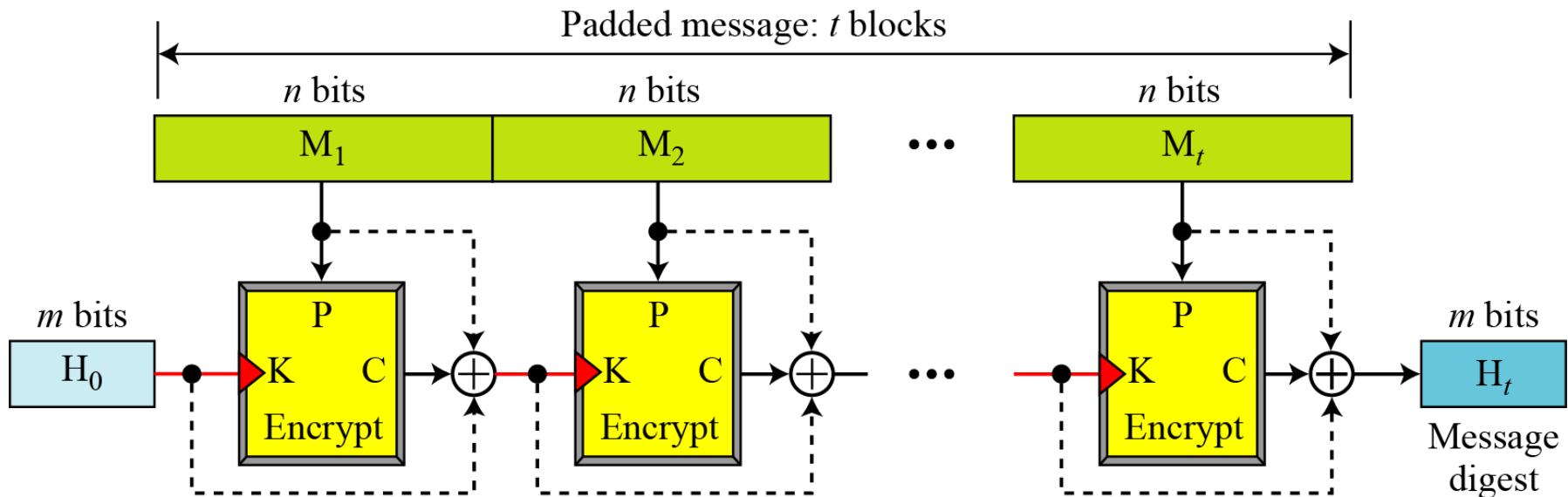
Figure 12.4 Matyas-Meyer-Oseas scheme



12.1.2 Continued

Miyaguchi-Preneel Scheme

Figure 12.5 Miyaguchi-Preneel scheme



12-2 SHA-512

SHA-512 is the version of SHA with a 512-bit message digest. This version, like the others in the SHA family of algorithms, is based on the Merkle-Damgard scheme.

Topics discussed in this section:

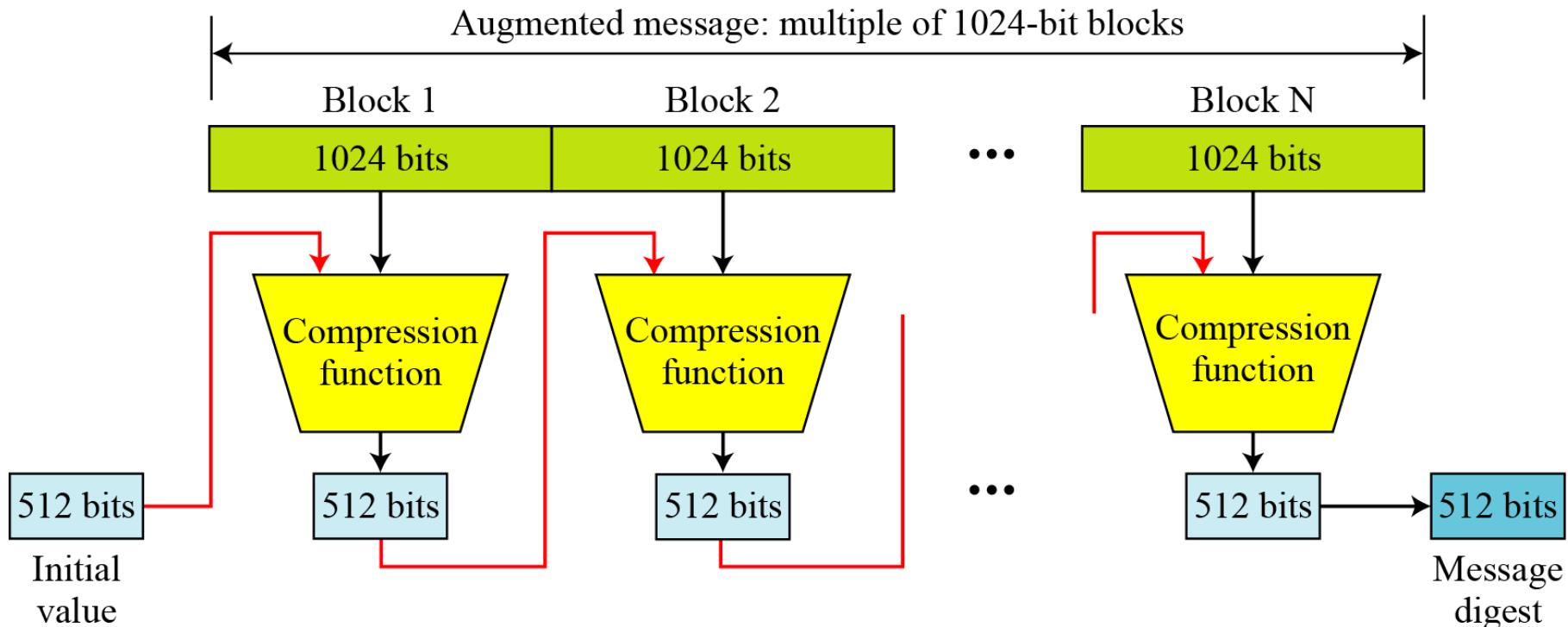
12.2.1 Introduction

12.2.2 Compression Function

12.2.3 Analysis

12.2.1 Introduction

Figure 12.6 Message digest creation SHA-512



12-3 WHIRLPOOL

Whirlpool is an iterated cryptographic hash function, based on the Miyaguchi-Preneel scheme, that uses a symmetric-key block cipher in place of the compression function. The block cipher is a modified AES cipher that has been tailored for this purpose.

Topics discussed in this section:

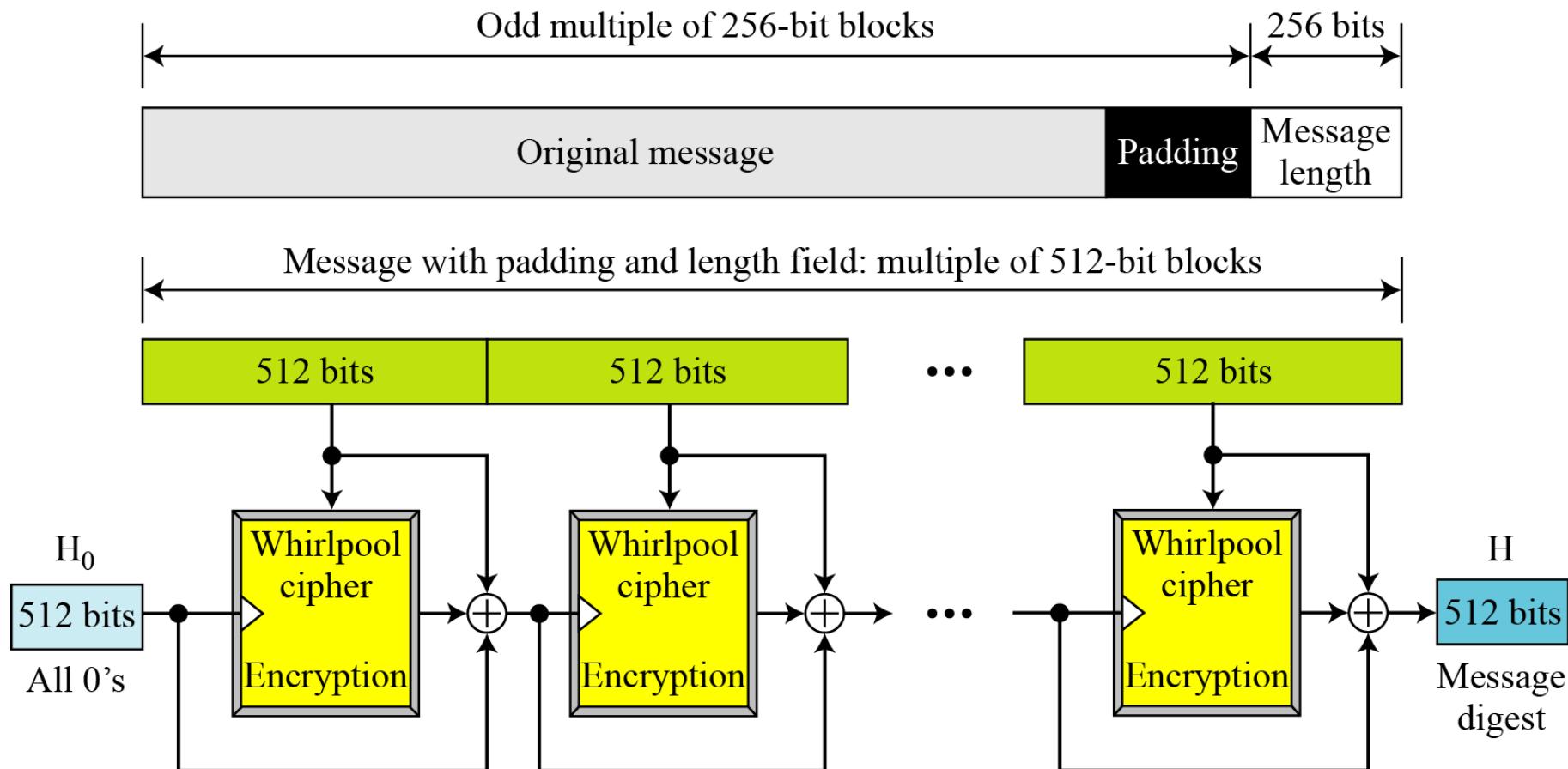
12.3.1 Whirlpool Cipher

12.3.2 Summary

12.3.3 Analysis

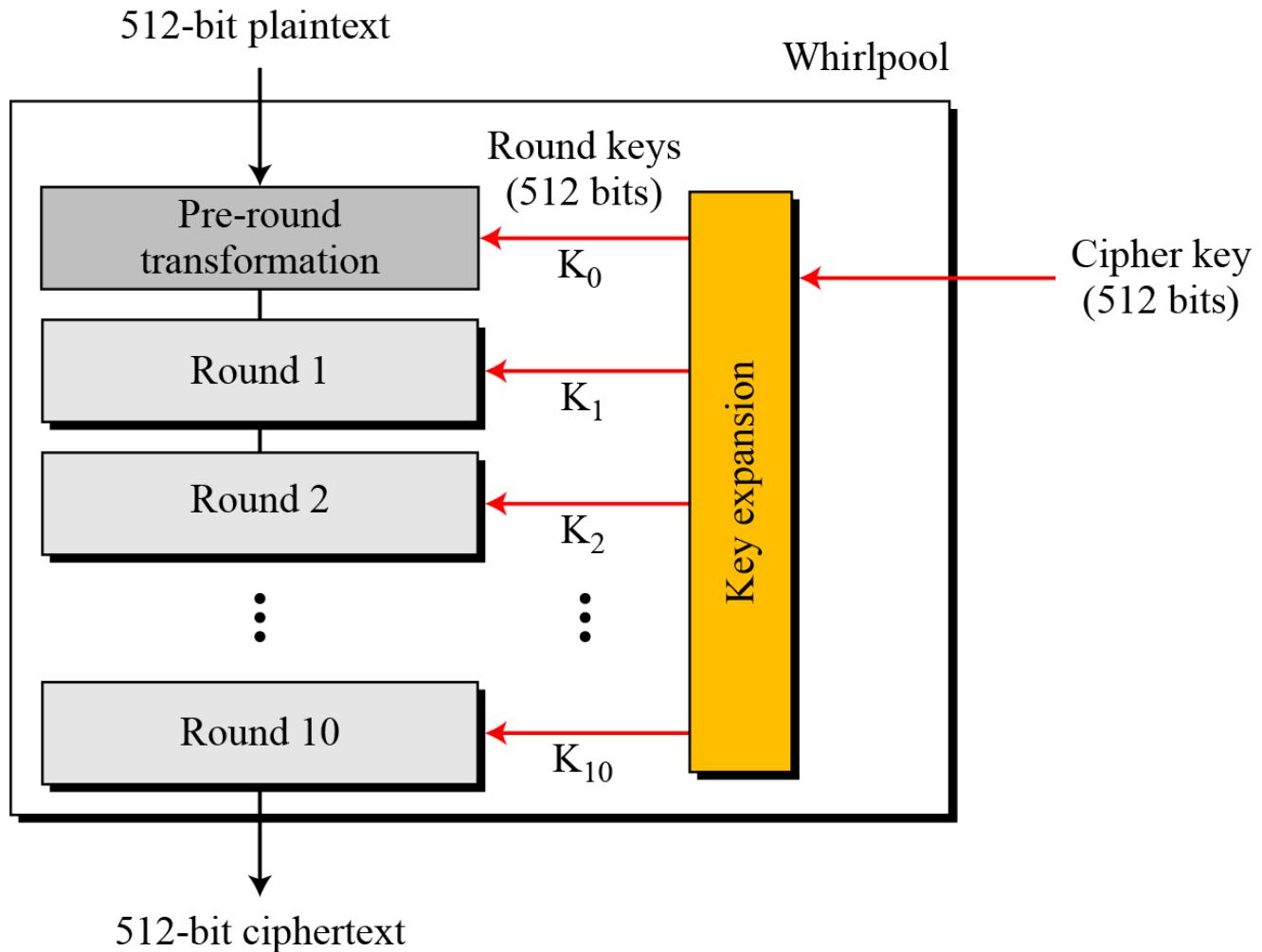
12-3 Continued

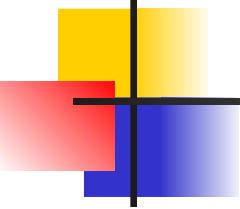
Figure 12.12 Whirlpool hash function



12.3.1 Whirlpool Cipher

Figure 12.13 General idea of the Whirlpool cipher

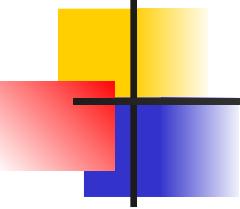




12.3.2 Summary

Table 12.5 *Main characteristics of the Whirlpool cipher*

Block size: 512 bits
Cipher key size: 512 bits
Number of rounds: 10
Key expansion: using the cipher itself with round constants as round keys
Substitution: SubBytes transformation
Permutation: ShiftColumns transformation
Mixing: MixRows transformation
Round Constant: cubic roots of the first eighty prime numbers



12.3.3 Analysis

Although Whirlpool has not been extensively studied or tested, it is based on a robust scheme (Miyaguchi-Preneel), and for a compression function uses a cipher that is based on AES, a cryptosystem that has been proved very resistant to attacks. In addition, the size of the message digest is the same as for SHA-512. Therefore it is expected to be a very strong cryptographic hash function.



Chapter 13

Digital Signature

Objectives

- To define a digital signature**
- To define security services provided by a digital signature**
- To define attacks on digital signatures**
- To discuss some digital signature schemes, including RSA, ElGamal,**
- Schnorr, DSS, and elliptic curve**
- To describe some applications of digital signatures**

13-2 PROCESS

Figure 13.1 shows the digital signature process. The sender uses a signing algorithm to sign the message. The message and the signature are sent to the receiver. The receiver receives the message and the signature and applies the verifying algorithm to the combination. If the result is true, the message is accepted; otherwise, it is rejected.

Topics discussed in this section:

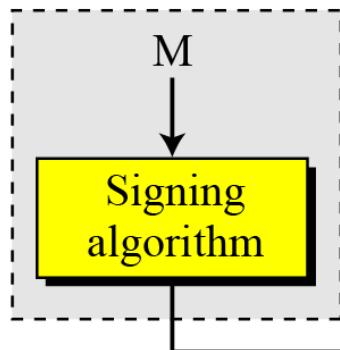
13.2.1 Need for Keys

13.2.2 Signing the Digest

13-2 Continued

Figure 13.1 *Digital signature process*

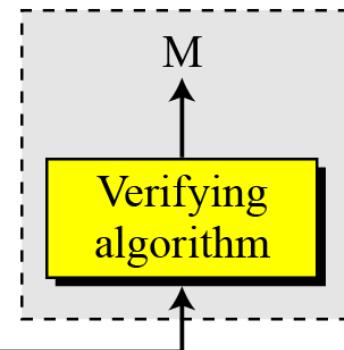
Alice



M: Message
S: Signature

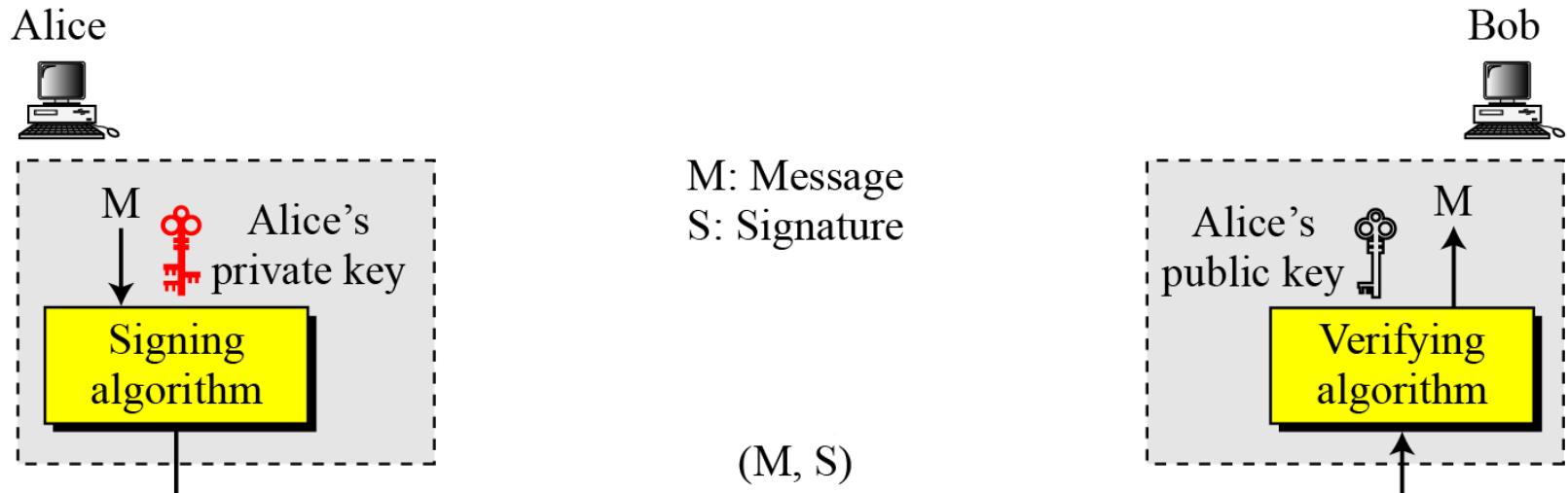
(M, S)

Bob



13.2.1 Need for Keys

Figure 13.2 Adding key to the digital signature process

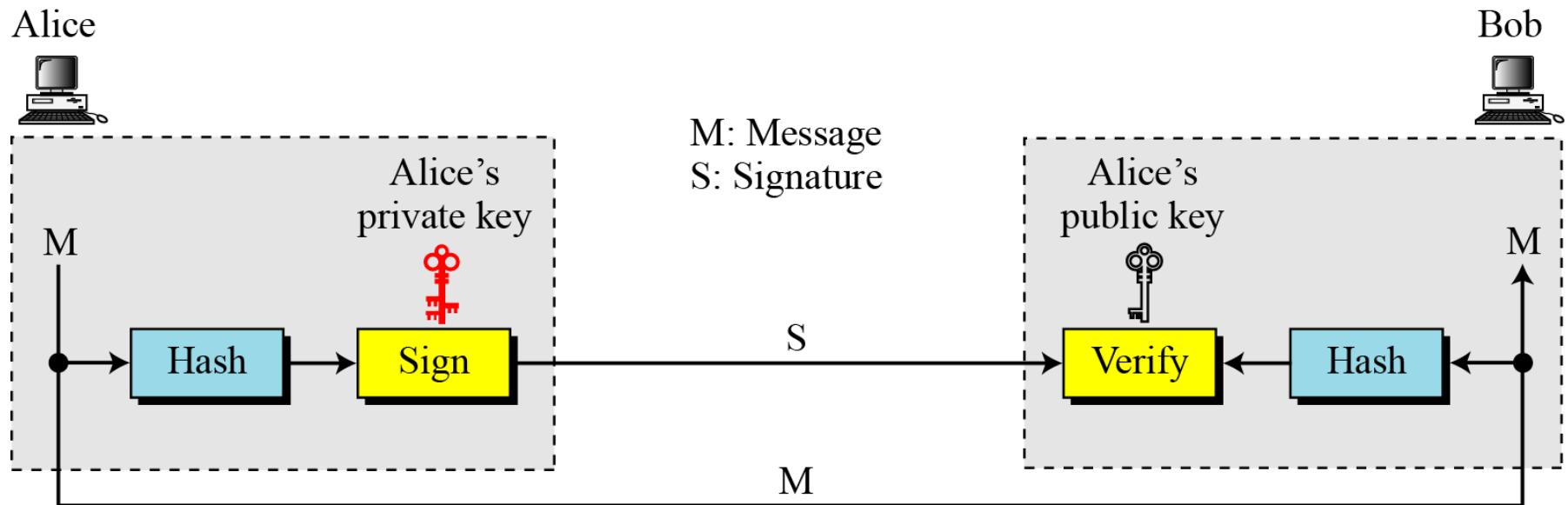


Note

**A digital signature needs a public-key system.
The signer signs with her private key; the verifier
verifies with the signer's public key.**

13.2.2 Signing the Digest

Figure 13.3 Signing the digest



13-3 SERVICES

We discussed several security services in Chapter 1 including message confidentiality, message authentication, message integrity, and nonrepudiation. A digital signature can directly provide the last three; for message confidentiality we still need encryption/decryption.

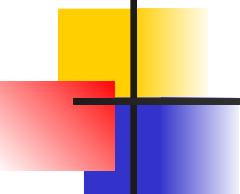
Topics discussed in this section:

13.3.1 Message Authentication

13.3.2 Message Integrity

13.3.3 Nonrepudiation

13.3.4 Confidentiality

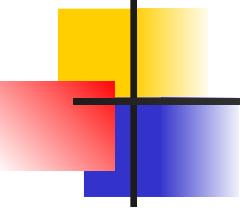


13.3.1 Message Authentication

A secure digital signature scheme, like a secure conventional signature can provide message authentication.

Note

A digital signature provides message authentication.



13.3.2 Message Integrity

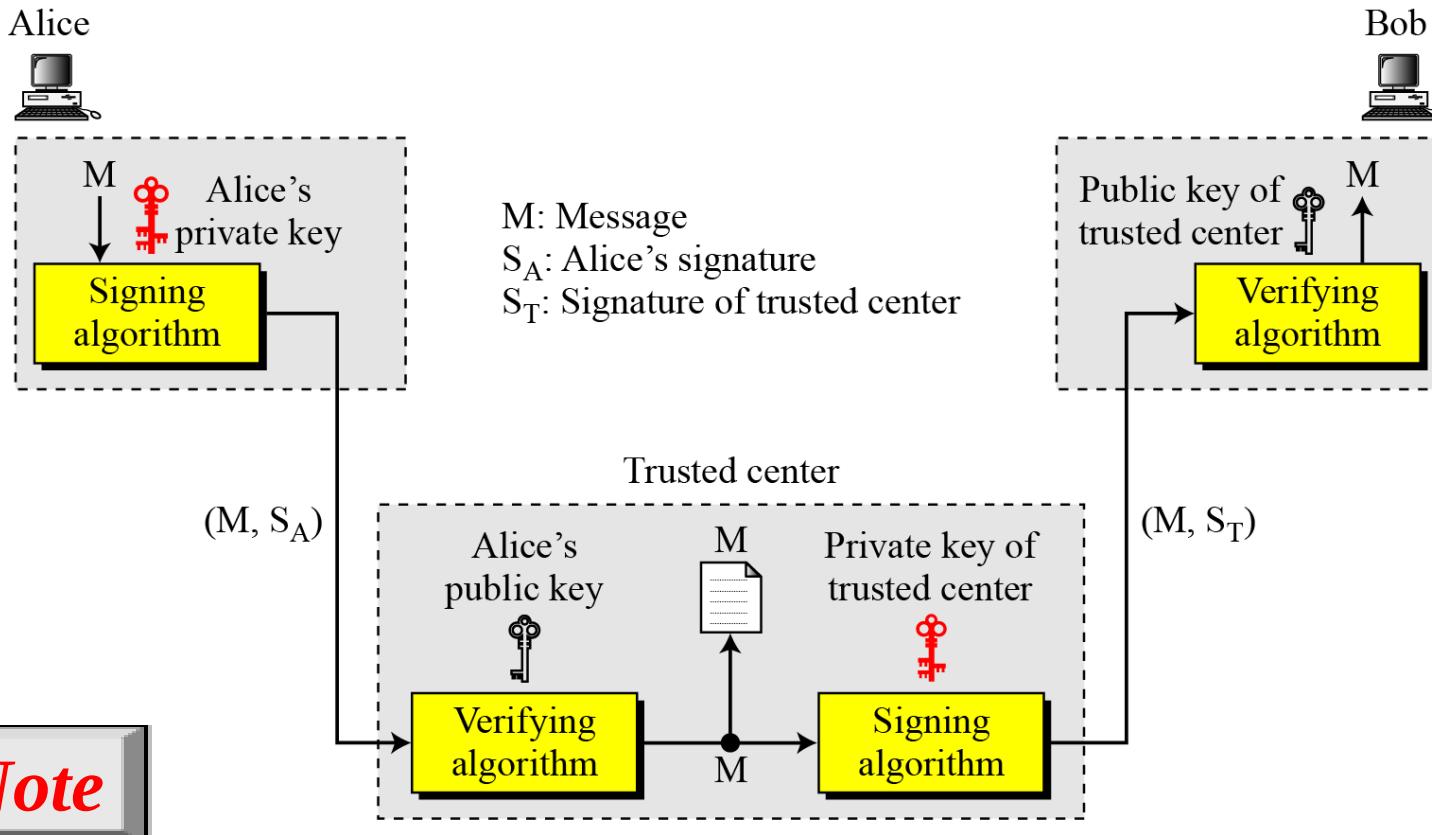
The integrity of the message is preserved even if we sign the whole message because we cannot get the same signature if the message is changed.

Note

A digital signature provides message integrity.

13.3.3 Nonrepudiation

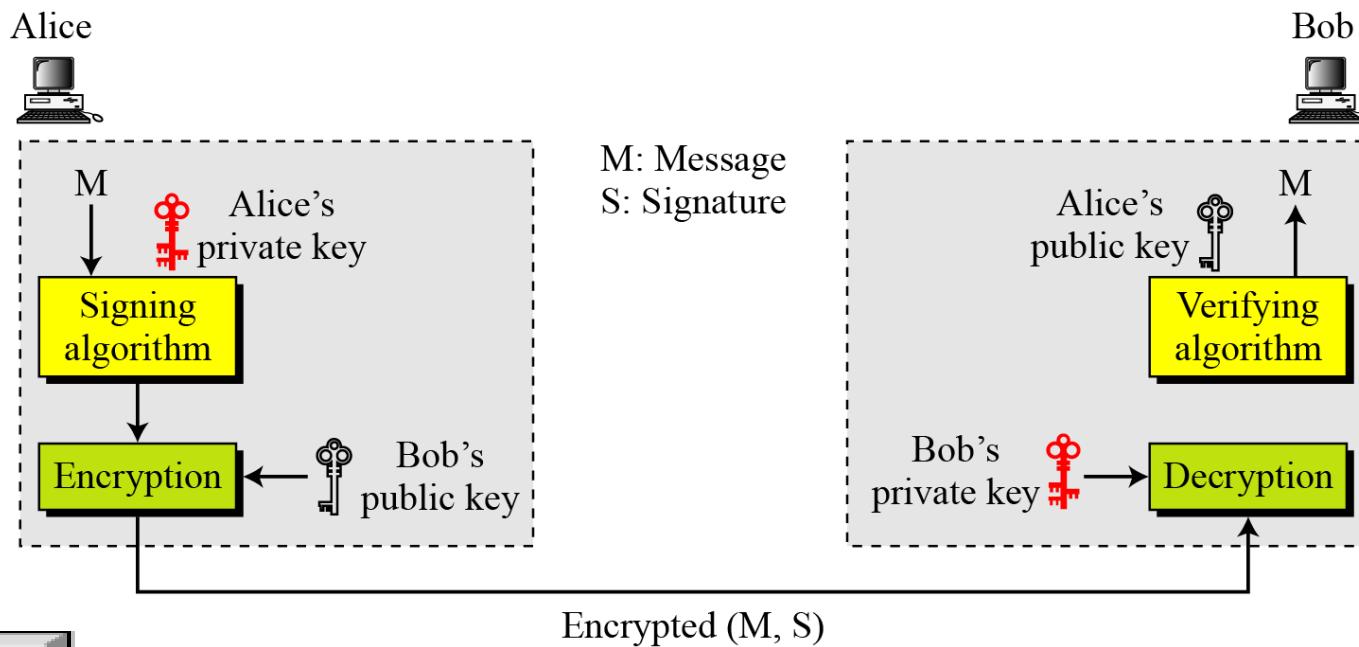
Figure 13.4 Using a trusted center for nonrepudiation



Nonrepudiation can be provided using a trusted party.

13.3.4 Confidentiality

Figure 13.5 Adding confidentiality to a digital signature scheme



Note

A digital signature does not provide privacy.
If there is a need for privacy, another layer of encryption/decryption must be applied.

13-5 DIGITAL SIGNATURE SCHEMES

Several digital signature schemes have evolved during the last few decades. Some of them have been implemented.

Topics discussed in this section:

13.5.1 RSA Digital Signature Scheme

13.5.2 ElGamal Digital Signature Scheme

13.5.3 Schnorr Digital Signature Scheme

13.5.4 Digital Signature Standard (DSS)

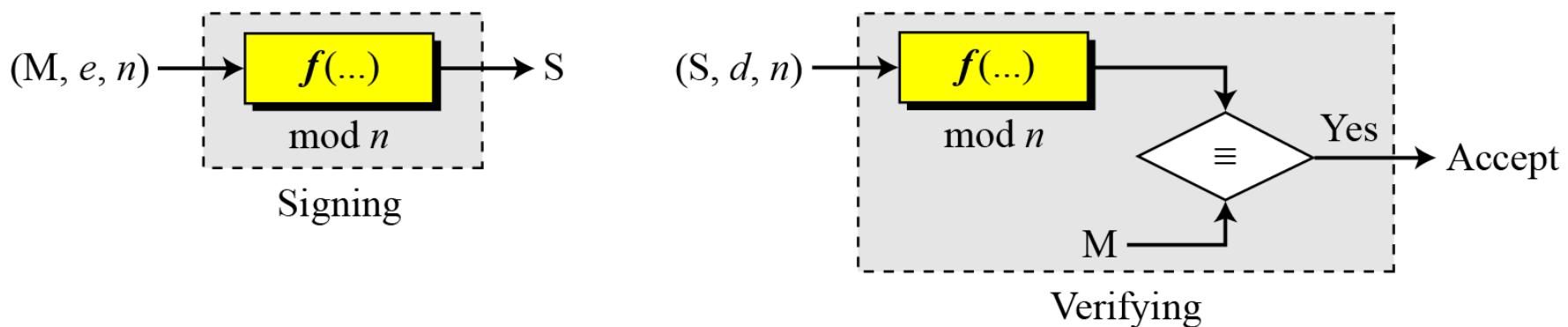
13.5.5 Elliptic Curve Digital Signature Scheme

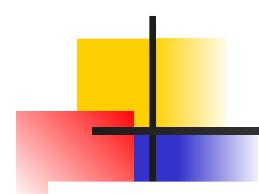
13.5.1 RSA Digital Signature Scheme

Figure 13.6 General idea behind the RSA digital signature scheme

M: Message
S: Signature

(e, n) : Alice's public key
 d : Alice's private key





13.5.1 *Continued*

Key Generation

Key generation in the RSA digital signature scheme is exactly the same as key generation in the RSA

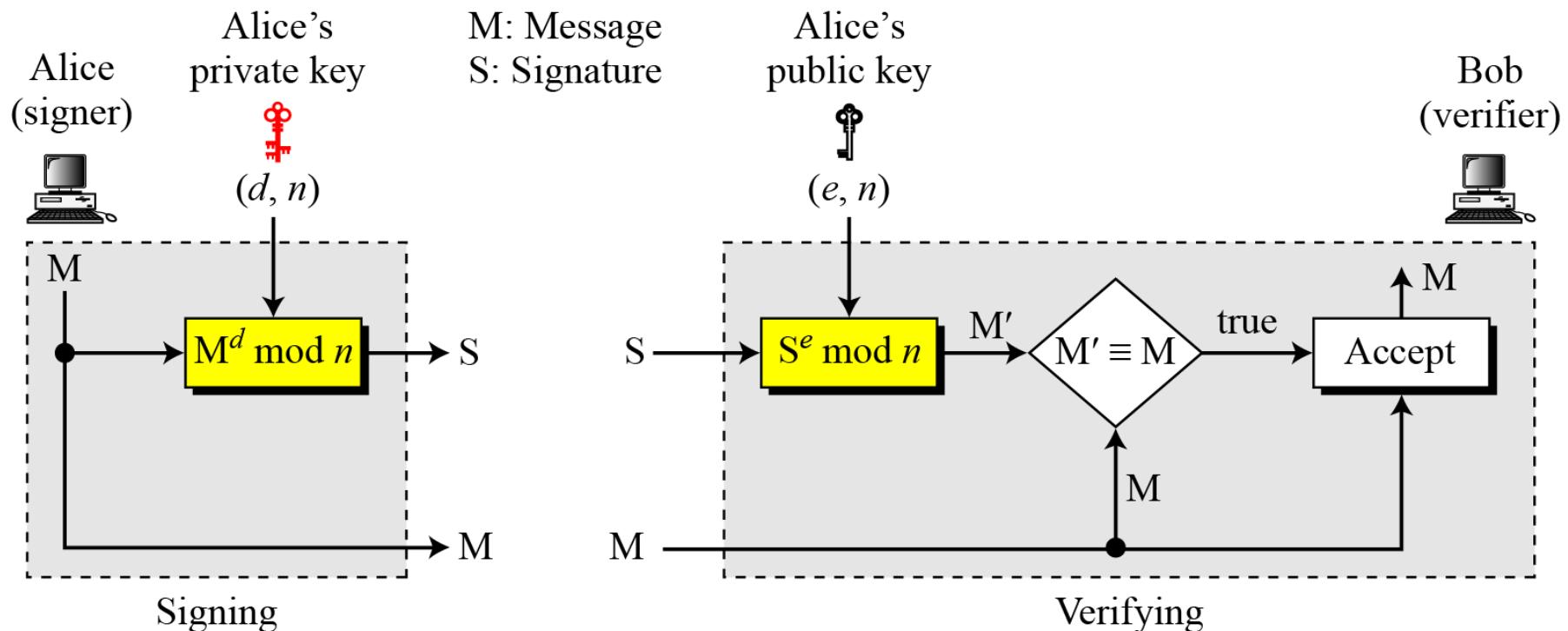
Note

In the RSA digital signature scheme, d is private; e and n are public.

13.5.1 Continued

Signing and Verifying

Figure 13.7 RSA digital signature scheme



13.5.1 *Continued*

Example 13.1

As a trivial example, suppose that Alice chooses $p = 823$ and $q = 953$, and calculates $n = 784319$. The value of $\phi(n)$ is 782544 . Now she chooses $e = 313$ and calculates $d = 160009$. At this point key generation is complete. Now imagine that Alice wants to send a message with the value of $M = 19070$ to Bob. She uses her private exponent, 160009 , to sign the message:

$$M: 19070 \rightarrow S = (19070^{160009}) \bmod 784319 = 210625 \bmod 784319$$

Alice sends the message and the signature to Bob. Bob receives the message and the signature. He calculates

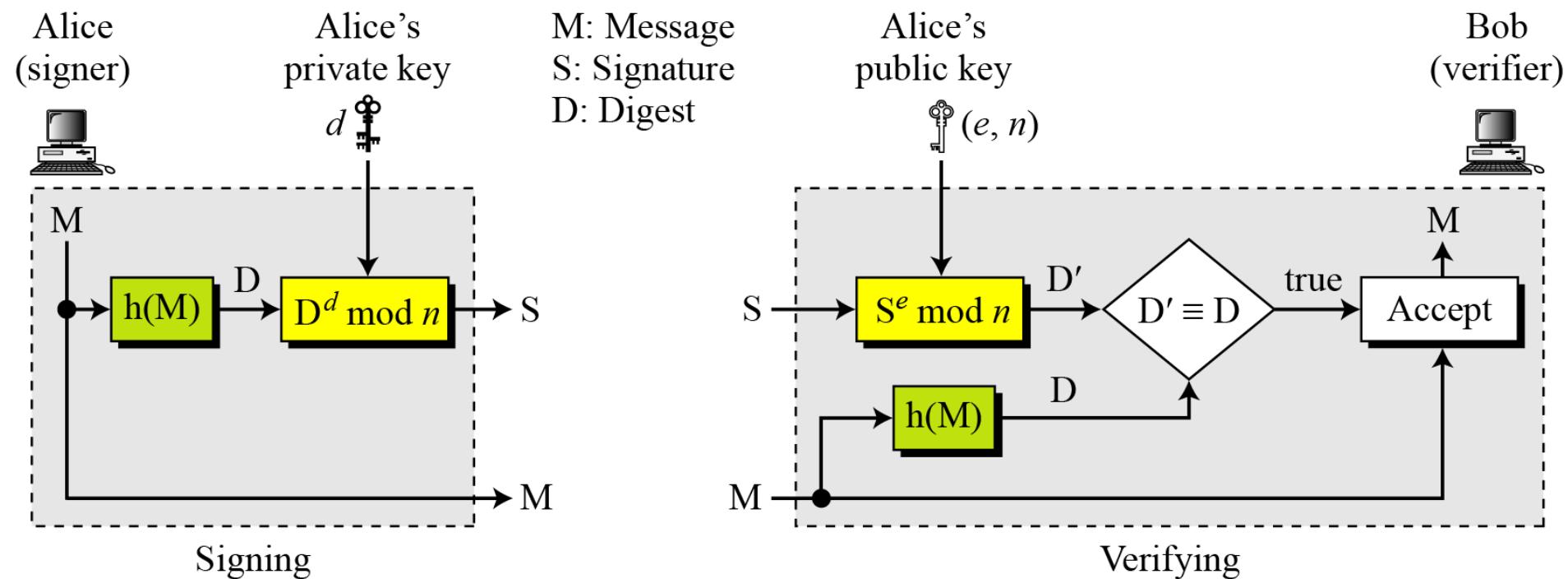
$$M' = 210625^{313} \bmod 784319 = 19070 \bmod 784319 \rightarrow M \equiv M' \bmod n$$

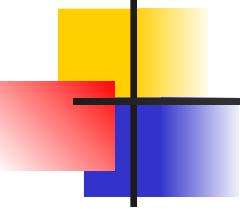
Bob accepts the message because he has verified Alice's signature.

13.5.1 Continued

RSA Signature on the Message Digest

Figure 13.8 The RSA signature on the message digest





13.5.1 Continued

Note

When the digest is signed instead of the message itself, the susceptibility of the RSA digital signature scheme depends on the strength of the hash algorithm.

13.5.2 ElGamal Digital Signature Scheme

Figure 13.9 General idea behind the ElGamal digital signature scheme

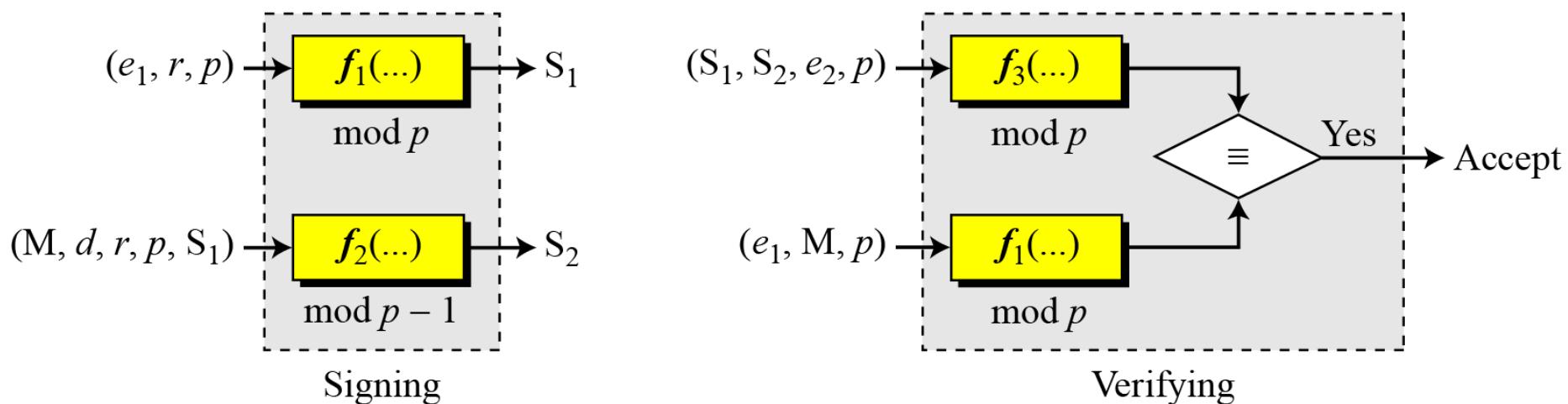
S_1, S_2 : Signatures

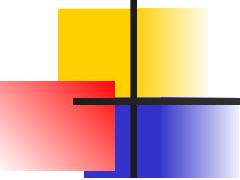
M: Message

(e_1, e_2, p) : Alice's public key

d : Alice's private key

r: Random secret





13.5.2 Continued

Key Generation

The key generation procedure here is exactly the same as the one used in the cryptosystem.

Note

In ElGamal digital signature scheme, (e_1, e_2, p) is Alice's public key; d is her private key.

13.5.2 Continued

Verifying and Signing

Figure 13.10 ElGamal digital signature scheme

M: Message r : Random secret

S_1, S_2 : Signatures

d : Alice's private key

V_1, V_2 : Verifications

(e_1, e_2, p) : Alice's public key

Alice
(signer)



d

r

M

$$e_1^r \bmod p$$

$$(M - dS_1)r^{-1} \bmod (p - 1)$$

S_1

M

(e_1, e_2, p)

Bob
(verifier)



M

$$e_2^{S_1} S_1 S_2 \bmod p$$

$$e_1^M \bmod p$$

V_2

V_1

Accept

true

Signing

Verifying

13.5.1 *Continued*

Example 13.2

Here is a trivial example. Alice chooses $p = 3119$, $e_1 = 2$, $d = 127$ and calculates $e_2 = 2^{127} \bmod 3119 = 1702$. She also chooses r to be 307. She announces e_1 , e_2 , and p publicly; she keeps d secret. The following shows how Alice can sign a message.

$$M = 320$$

$$S_1 = e_1^r = 2^{307} = 2083 \bmod 3119$$

$$S_2 = (M - d \times S_1) \times r^{-1} = (320 - 127 \times 2083) \times 307^{-1} = 2105 \bmod 3118$$

Alice sends M , S_1 , and S_2 to Bob. Bob uses the public key to calculate V_1 and V_2 .

$$V_1 = e_1^M = 2^{320} = 3006 \bmod 3119$$

$$V_2 = d^{S_1} \times S_1^{S_2} = 1702^{2083} \times 2083^{2105} = 3006 \bmod 3119$$

13.5.1 *Continued*

Example 13.3

Now imagine that Alice wants to send another message, $M = 3000$, to Ted. She chooses a new r , 107. Alice sends M , S_1 , and S_2 to Ted. Ted uses the public keys to calculate V_1 and V_2 .

$$M = 3000$$

$$S_1 = e_1^r = 2^{107} = 2732 \bmod 3119$$

$$S_2 = (M - d \times S_1) r^{-1} = (3000 - 127 \times 2083) \times 107^{-1} = 2526 \bmod 3118$$

$$V_1 = e_1^M = 2^{3000} = 704 \bmod 3119$$

$$V_2 = d^{S_1} \times S_1^S = 1702^{2732} \times 2083^{2526} = 704 \bmod 3119$$

13.5.3 Schnorr Digital Signature Scheme

Figure 13.11 General idea behind the Schnorr digital signature scheme

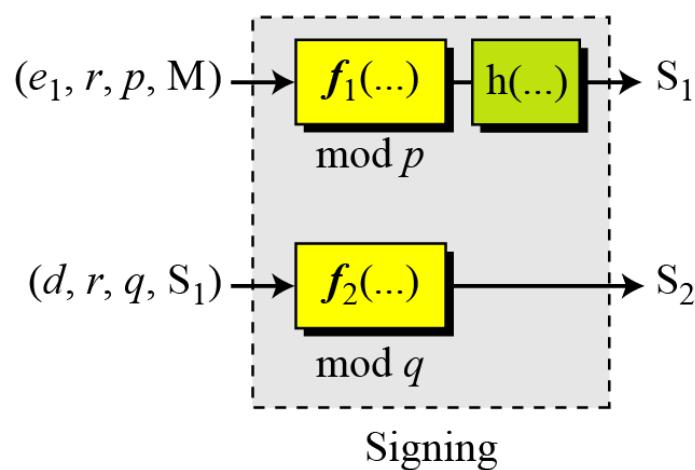
S_1, S_2 : Signatures

M: Message

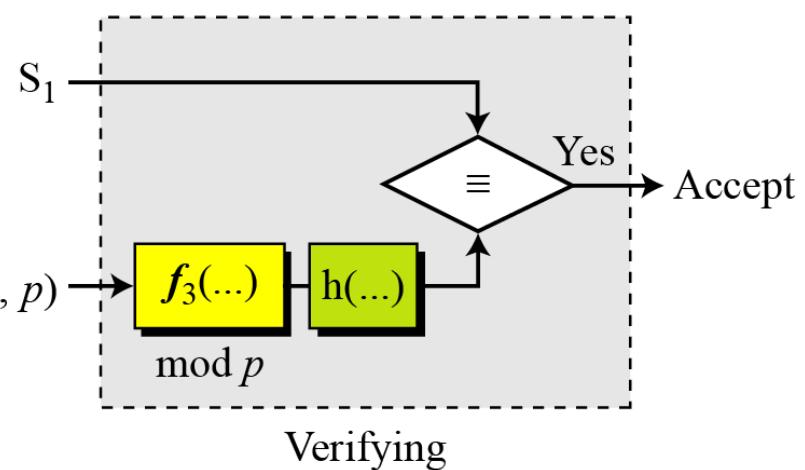
(e_1, e_2, p, q) : Alice's public key

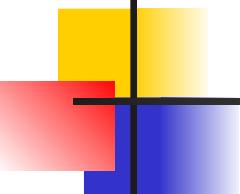
(d) : Alice's private key

r: Random secret



$(S_1, S_2, M, e_1, e_2, p)$





13.5.3 Continued

Key Generation

- 1) Alice selects a prime p , which is usually 1024 bits in length.
- 2) Alice selects another prime q .
- 3) Alice chooses e_1 to be the q th root of 1 modulo p .
- 4) Alice chooses an integer, d , as her private key.
- 5) Alice calculates $e_2 = e_1^d \text{ mod } p$.
- 6) Alice's public key is (e_1, e_2, p, q) ; her private key is (d) .

Note

In the Schnorr digital signature scheme, Alice's public key is (e_1, e_2, p, q) ; her private key (d) .

13.5.3 Continued

Signing and Verifying

Figure 13.12 Schnorr digital signature scheme

M: Message

S₁, S₂: Signatures

V: Verification

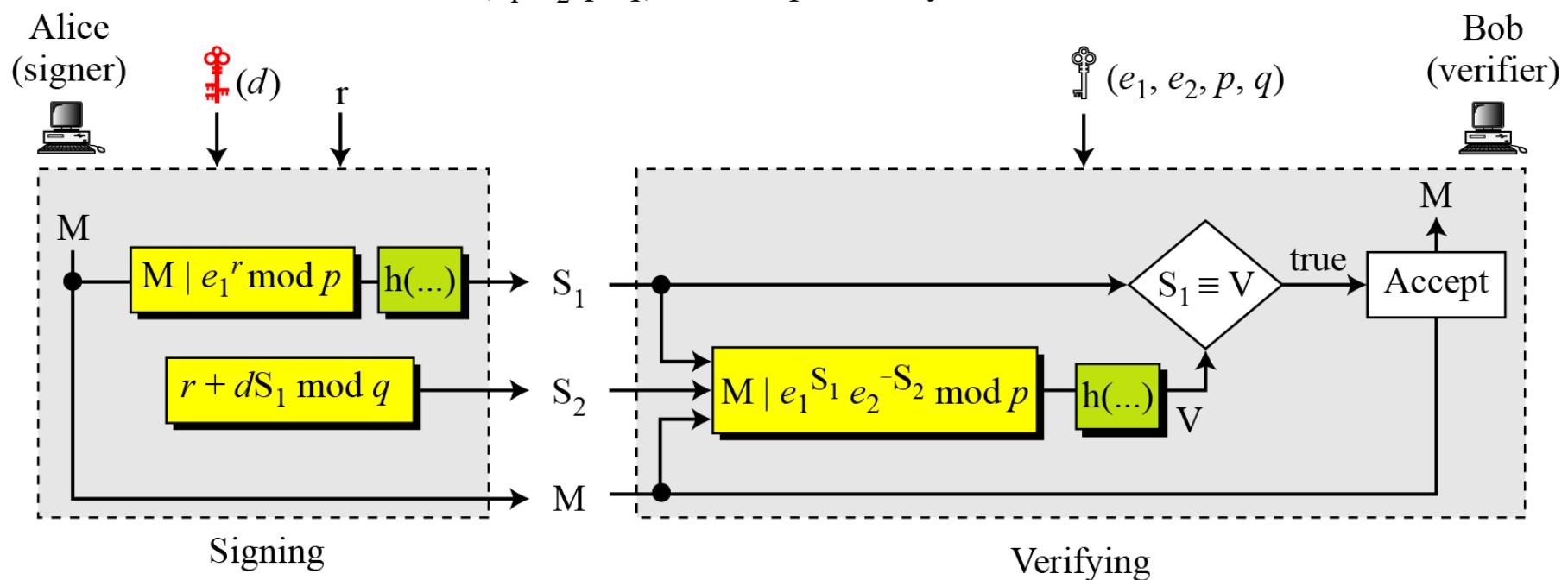
r: Random secret

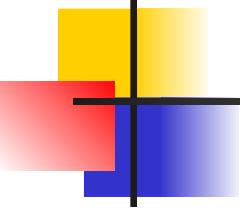
(d): Alice's private key

(e₁, e₂, p, q): Alice's public key

| : Concatenation

h(...): Hash algorithm





13.5.3 Continued

Signing

1. *Alice chooses a random number r.*
2. *Alice calculates $S_1 = h(M|e_1^r \bmod p)$.*
3. *Alice calculates $S_2 = r + d \times S_1 \bmod q$.*
4. *Alice sends M, S_1 , and S_2 .*

Verifying Message

1. *Bob calculates $V = h(M | e_1^{S_2} e_2^{-S_1} \bmod p)$.*
2. *If S_1 is congruent to V modulo p, the message is accepted;*

13.5.1 *Continued*

Example 13.4

Here is a trivial example. Suppose we choose $q = 103$ and $p = 2267$. Note that $p = 22 \times q + 1$. We choose $e_0 = 2$, which is a primitive in \mathbb{Z}_{2267}^* . Then $(p - 1) / q = 22$, so we have $e_1 = 2^{22} \bmod 2267 = 354$. We choose $d = 30$, so $e_2 = 354^{30} \bmod 2267 = 1206$. Alice's private key is now (d) ; her public key is (e_1, e_2, p, q) .

Alice wants to send a message M . She chooses $r = 11$ and calculates $e_2^r = 354^{11} = 630 \bmod 2267$. Assume that the message is 1000 and concatenation means 1000630. Also assume that the hash of this value gives the digest $h(1000630) = 200$. This means $S_1 = 200$. Alice calculates $S_2 = r + d \times S_1 \bmod q = 11 + 1026 \times 200 \bmod 103 = 35$. Alice sends the message $M = 1000$, $S_1 = 200$, and $S_2 = 35$. The verification is left as an exercise.

13.5.4 Digital Signature Standard (DSS)

Figure 13.13 General idea behind DSS scheme

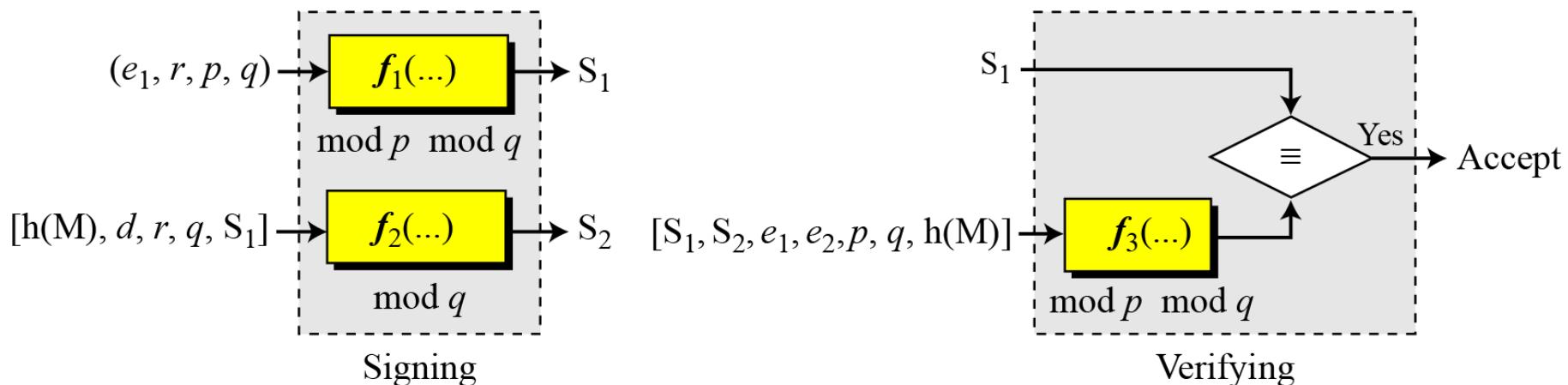
S_1, S_2 : Signatures

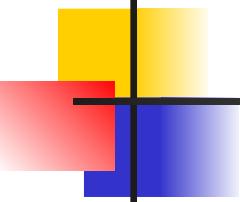
d : Alice's private key

M : Message

r : Random secret

(e_1, e_2, p, q) : Alice's public key





13.5.4 Continued

Key Generation.

- 1) *Alice chooses primes p and q .*
- 2) *Alice uses $\langle \mathbb{Z}_p^*, \times \rangle$ and $\langle \mathbb{Z}_q^*, \times \rangle$.*
- 3) *Alice creates e_1 to be the q th root of 1 modulo p .*
- 4) *Alice chooses d and calculates $e_2 = e_1^d$.*
- 5) *Alice's public key is (e_1, e_2, p, q) ; her private key is (d) .*

13.5.4 Continued

Verifying and Signing

Figure 13.14 DSS scheme

M: Message

S_1, S_2 : Signatures

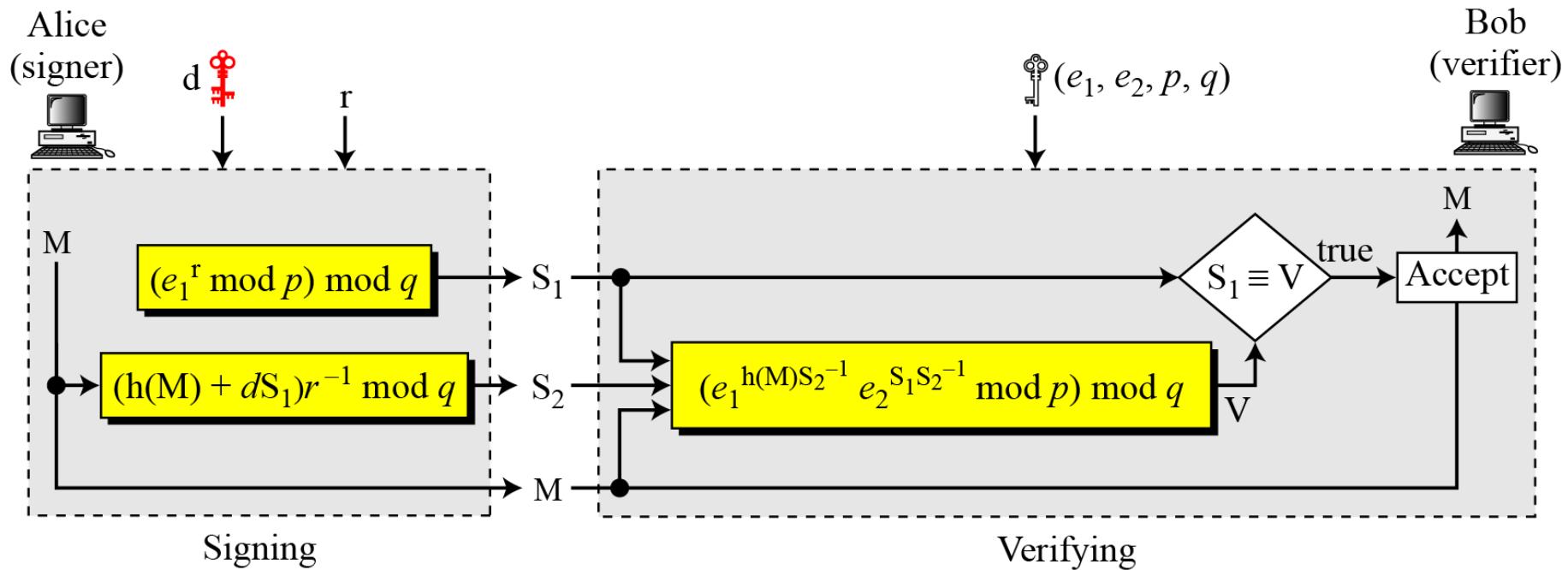
V: Verification

r: Random secret

d: Alice's private key

(e_1, e_2, p, q) : Alice's public key

$h(M)$: Message digest



13.5.1 *Continued*

Example 13.5

Alice chooses $q = 101$ and $p = 8081$. Alice selects $e_0 = 3$ and calculates $e^1 = e_0^{(p-1)/q} \bmod p = 6968$. Alice chooses $d = 61$ as the private key and calculates $e_2 = e_1^d \bmod p = 2038$. Now Alice can send a message to Bob. Assume that $h(M) = 5000$ and Alice chooses $r = 61$:

$$h(M) = 5000 \quad r = 61$$

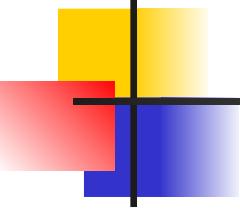
$$S_1 = (e_1^r \bmod p) \bmod q = 54$$

$$S_2 = ((h(M) + d S_1) r^{-1}) \bmod q = 40$$

Alice sends M , S_1 , and S_2 to Bob. Bob uses the public keys to calculate V .

$$S_2^{-1} = 48 \bmod 101$$

$$V = [(6968^{5000 \times 48} \times 2038^{54 \times 48}) \bmod 8081] \bmod 101 = 54$$



13.5.4 Continued

DSS Versus RSA

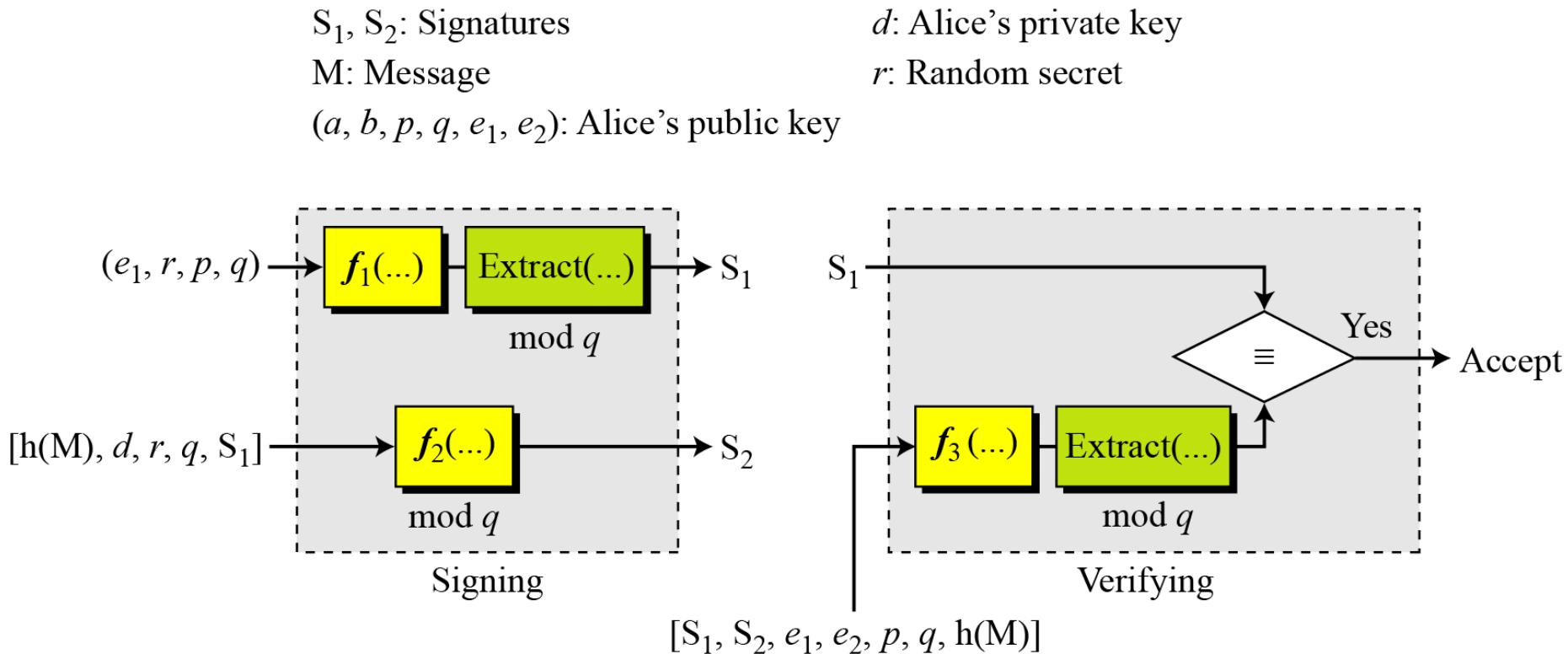
Computation of DSS signatures is faster than computation of RSA signatures when using the same p.

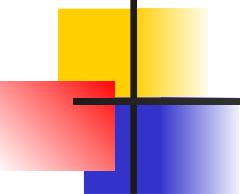
DSS Versus ElGamal

DSS signatures are smaller than ElGamal signatures because q is smaller than p.

13.5.5 Elliptic Curve Digital Signature Scheme

Figure 13.15 General idea behind the ECDSS scheme





13.5.5 Continued

Key Generation

Key generation follows these steps:

- 1) *Alice chooses an elliptic curve $E_p(a, b)$.*
- 2) *Alice chooses another prime q the private key d .*
- 3) *Alice chooses $e_1(\dots, \dots)$, a point on the curve.*
- 4) *Alice calculates $e_2(\dots, \dots) = d \times e_1(\dots, \dots)$.*
- 5) *Alice's public key is $(a, b, p, q, e1, e2)$; her private key is d .*

13.5.5 Continued

Signing and Verifying

Figure 13.16 The ECDSS scheme

M: Message

S_1, S_2 : Signatures

V: Verification

r: Random secret

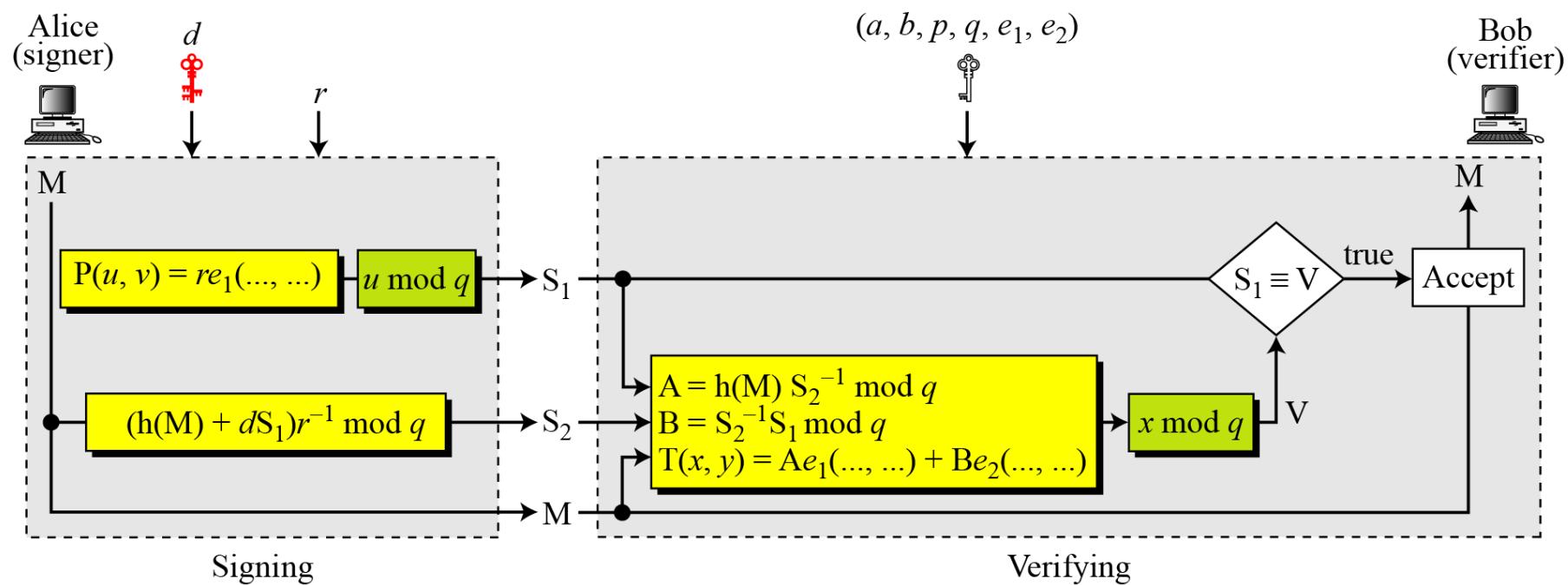
d: Alice's private key

(a, b, p, q, e_1, e_2): Alice's public key

$P(u, v), T(x, y)$: Points on the curve

$h(M)$: Message digest

A, B: Intermediate results





Data Communications and Networking

Fourth Edition

Forouzan

Chapter 31

Network Security

31-1 SECURITY SERVICES

Network security can provide five services. Four of these services are related to the message exchanged using the network. The fifth service provides entity authentication or identification.

Topics discussed in this section:

Message Confidentiality

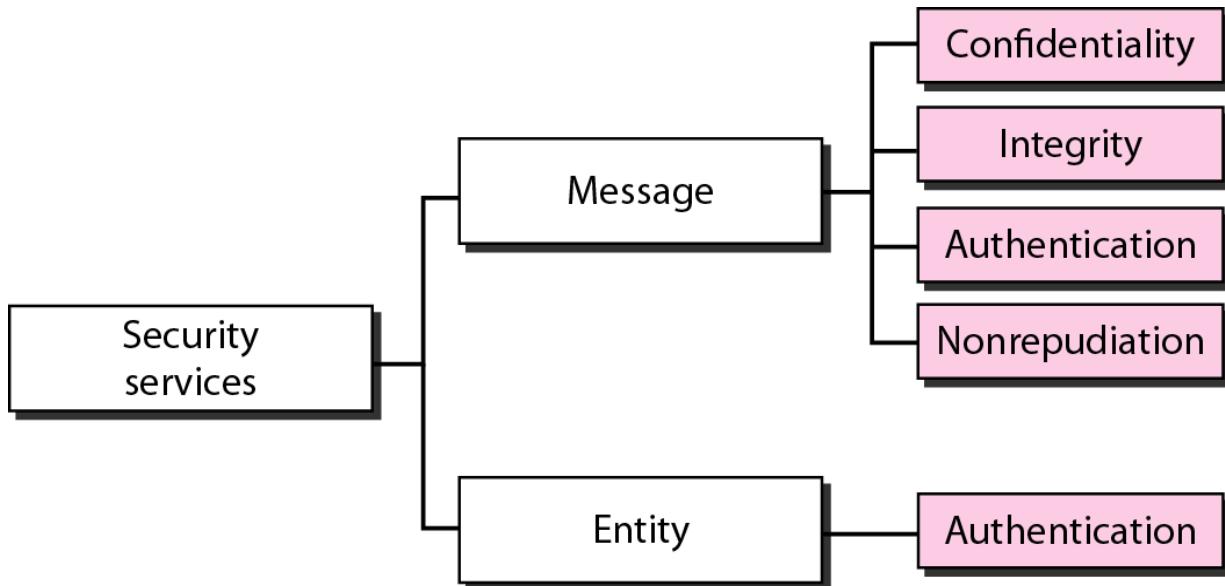
Message Integrity

Message Authentication

Message Nonrepudiation

Entity Authentication

Figure 31.1 *Security services related to the message or entity*



31-3 MESSAGE INTEGRITY

Encryption and decryption provide secrecy, or confidentiality, but not integrity. However, on occasion we may not even need secrecy, but instead must have integrity.

Topics discussed in this section:

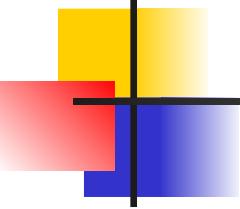
Document and Fingerprint

Message and Message Digest

Creating and Checking the Digest

Hash Function Criteria

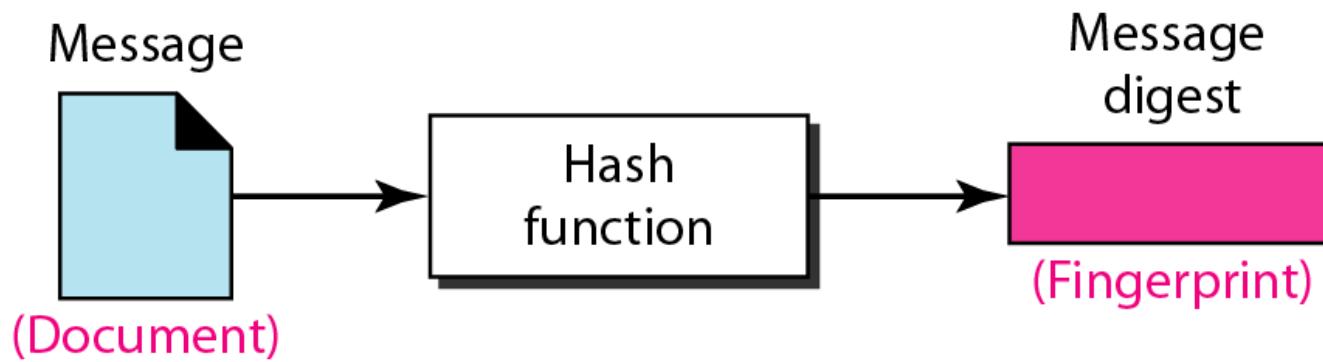
Hash Algorithms: SHA-1



Note

**To preserve the integrity of a document,
both the document and the fingerprint
are needed.**

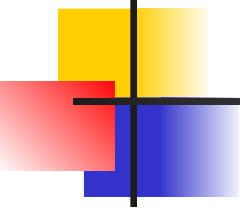
Figure 31.4 *Message and message digest*



Notations:

m: message

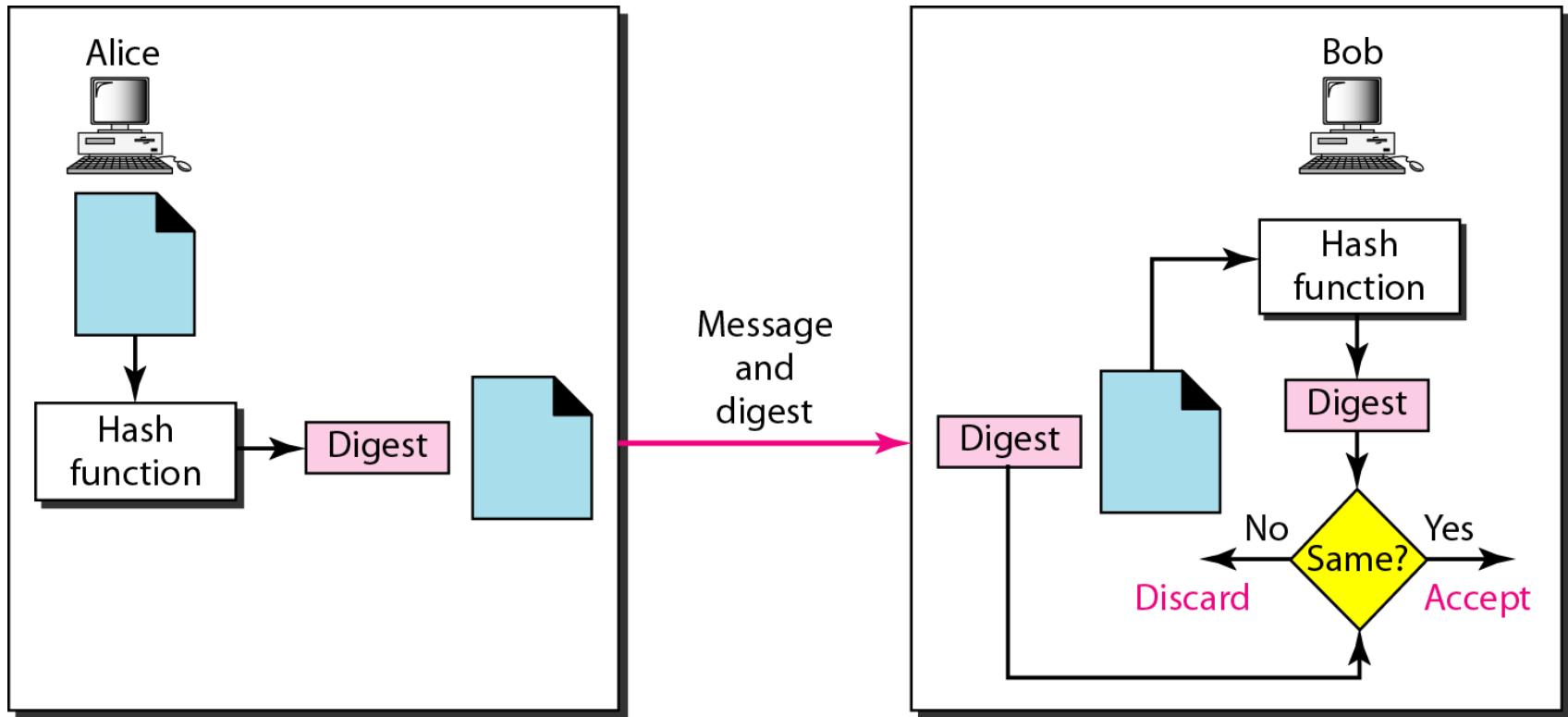
H(m): message digest of m by using hash function H()



Note

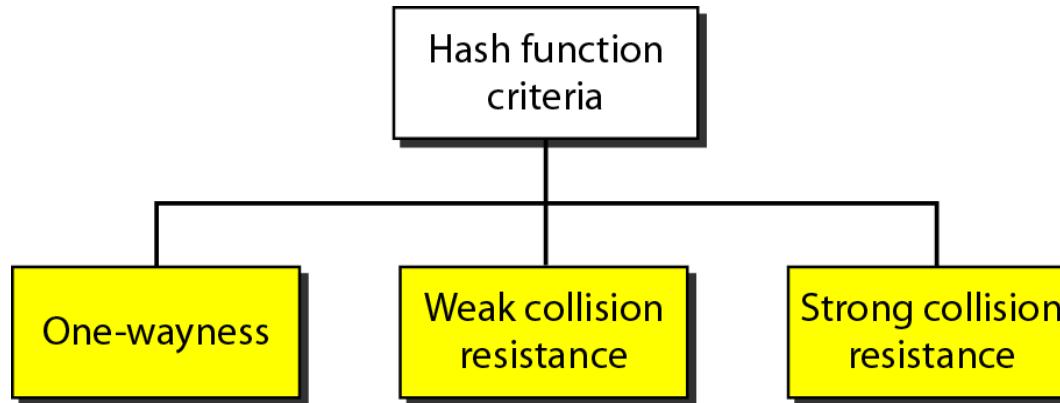
The message digest needs to be kept secret, or unalterable by others.

Figure 31.5 *Checking integrity*

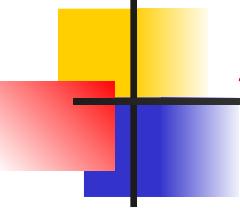


Notes: We need to make sure the digest cannot be altered by attacker

Figure 31.6 Criteria of a hash function



- **One-wayness:**
 - *Cannot recover message m given its digest $H(m)$*
- **Weak collision resistance:**
 - *Given message m , cannot generate another message m' such that $H(m')=H(m)$* → ensure integrity
- **Strong collision resistance:** (a stronger requirement than above one)
 - *Sender cannot generate two messages m and m' such that $H(m)=H(m')$* → ensure nonrepudiation

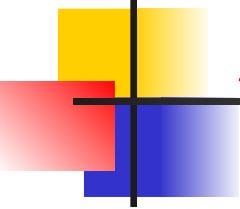


Example 31.1

Can we use a conventional lossless compression method as a hashing function?

Solution

We cannot. A lossless compression method creates a compressed message that is reversible. You can uncompress the compressed message to get the original one.



Example 31.2

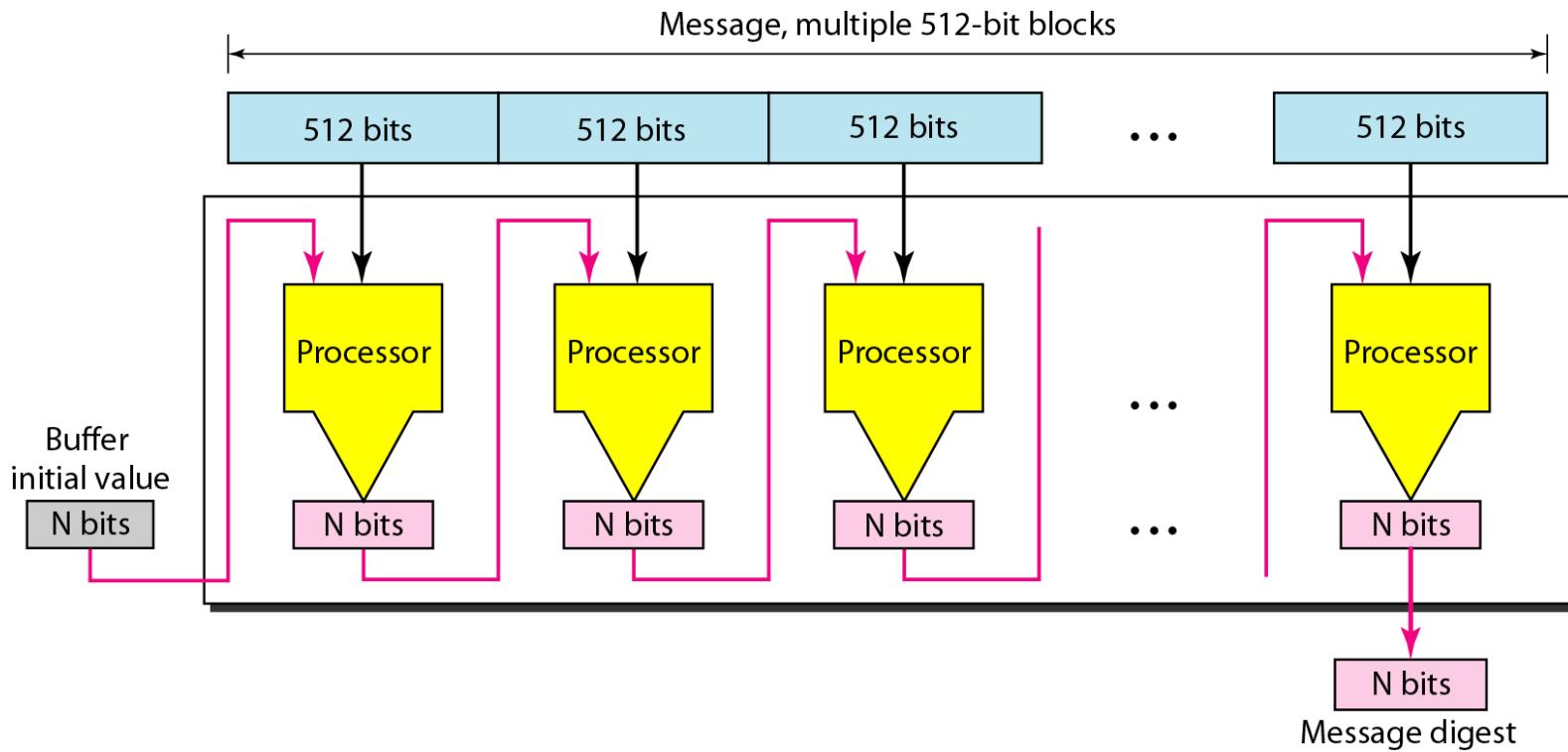
Can we use a checksum method as a hashing function?

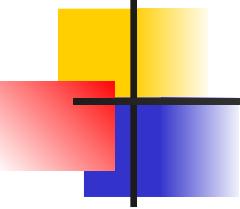
Solution

We cannot. A checksum function is not reversible; it meets the first criterion.

However, it does not meet the other criteria. That is to say, an attacker can easily modify a message without being detected.

Figure 31.7 Message digest creation





Note

SHA-1 hash algorithms create an N-bit message digest out of a message of 512-bit blocks. SHA-1 has a message digest of 160 bits.

Another popular hash algorithm is MD5 (message digest algorithm 5). It is an older generation than SHA-1.

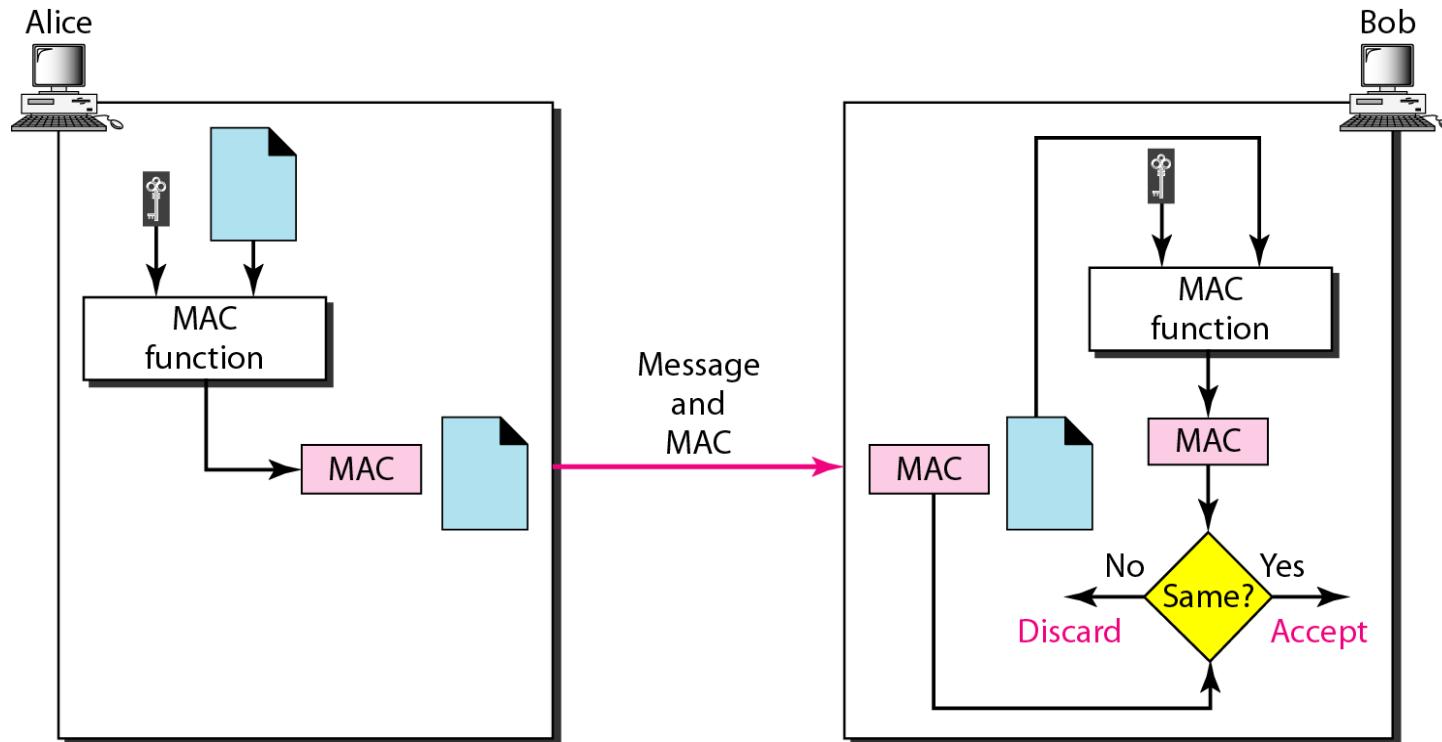
31-4 MESSAGE AUTHENTICATION

*A hash function per se cannot provide authentication.
The digest created by a hash function can detect any
modification in the message, but not authentication.*

Topics discussed in this section:

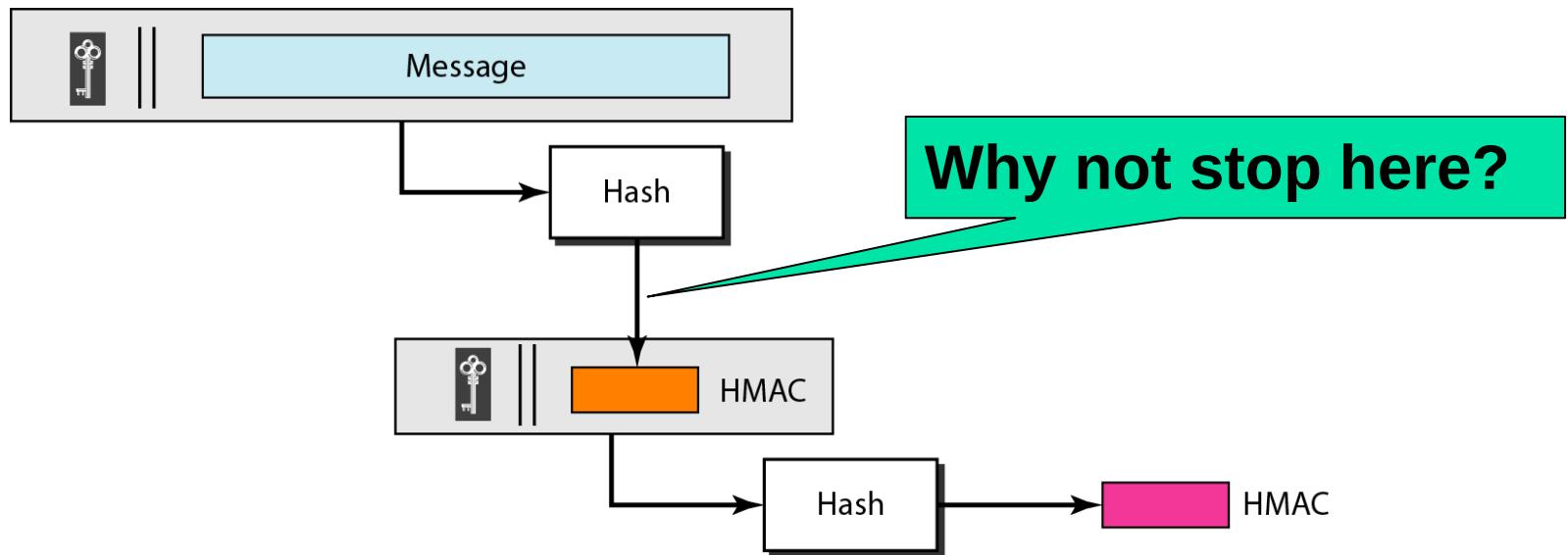
MAC (message authentication code): can be used to ensure both integrity and authentication

Figure 31.9 MAC, created by Alice and checked by Bob



Message itself is unencrypted

Figure 31.10 HMAC (Hashed MAC): uses keyless hash function



Reason: we can directly use mature keyless hash function such as SHA-1 or MD5

31-5 DIGITAL SIGNATURE

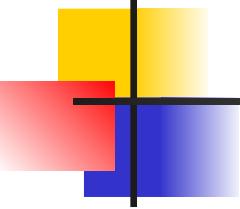
When Alice sends a message to Bob, Bob needs to check the authenticity of the sender; he needs to be sure that the message comes from Alice and not Eve. Bob can ask Alice to sign the message electronically. In other words, an electronic signature can prove the authenticity of Alice as the sender of the message. We refer to this type of signature as a digital signature.

Topics discussed in this section:

Comparison

Need for Keys

Process



Note

A digital signature needs a public-key system.

Notations:

m : message

$H(m)$: message digest of m by using hash function $H()$

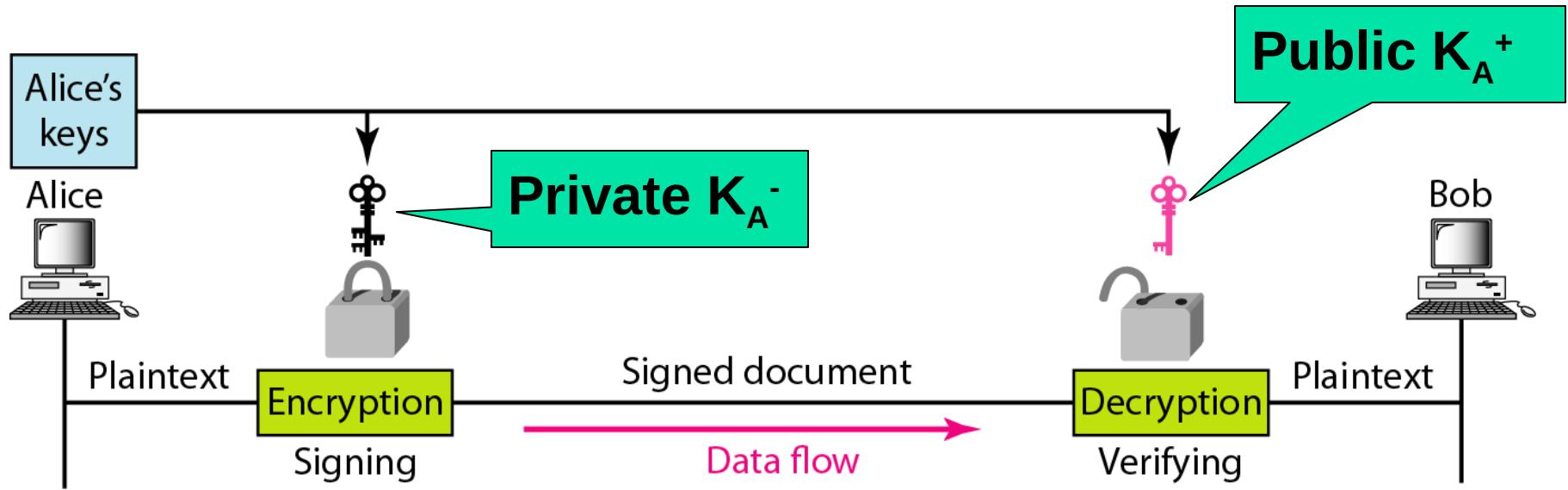
K_A^- : Private key of user A

K_A^+ : Public key of user A

K_{AB} : Symmetric key between A and B

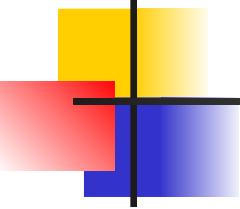
$K(m)$: ciphertext of message m by using encryption key K

Figure 31.11 *Signing the message itself in digital signature*



Provide no confidentiality (message is not secret)

Problem: *Too expensive to sign message itself using public key system*

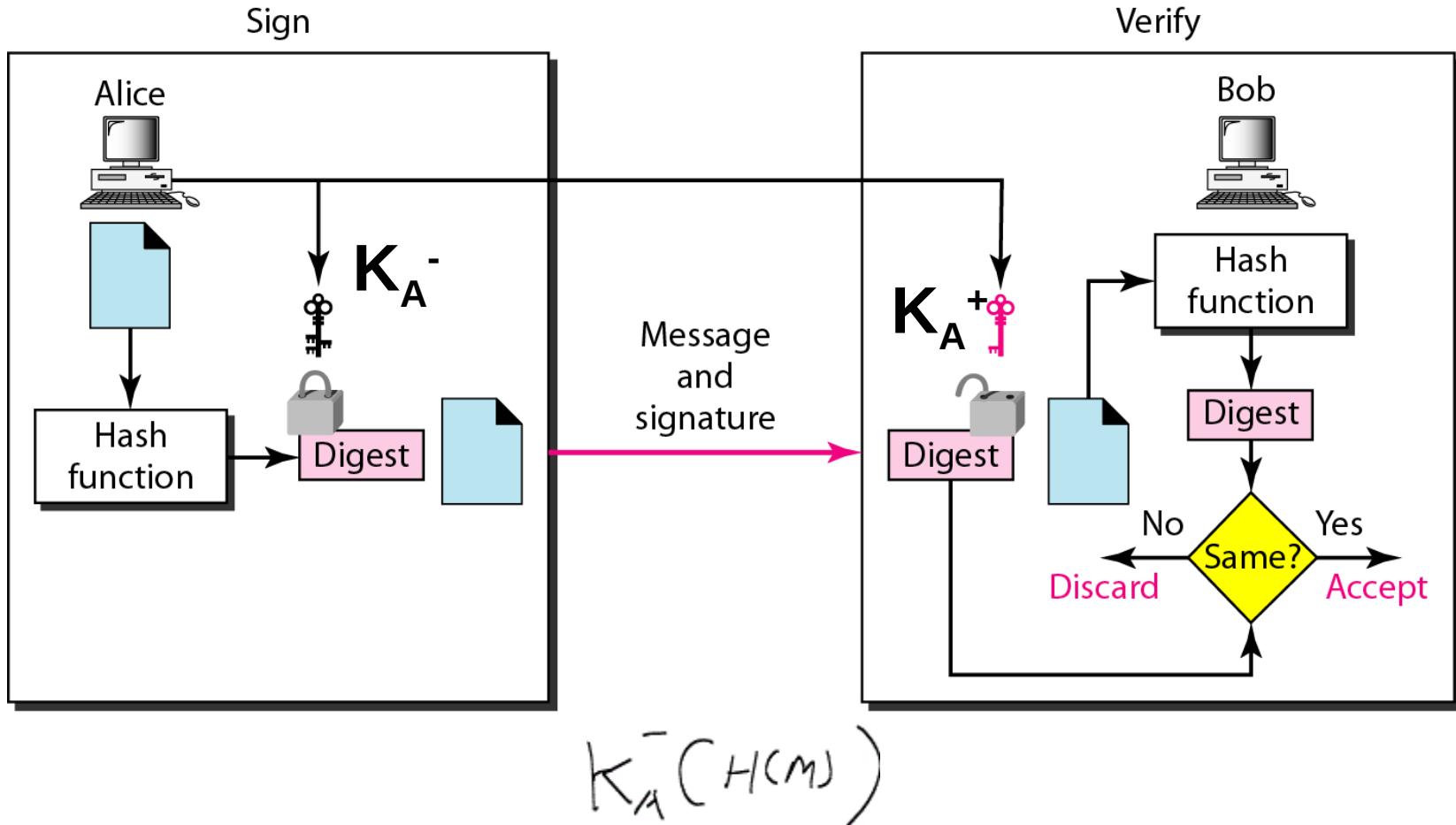


Note

For message confidentiality, we use the private and public keys of the receiver;

In digital signature (integrity, authentication, nonrepudiation), we use the private and public keys of the sender.

Figure 31.12 Signing the digest in a digital signature



Digital signature provides three out of the five services we mentioned for security systems

- ✓ Integrity
- ✓ Authentication
- ✓ Nonrepudiation

31-7 KEY MANAGEMENT

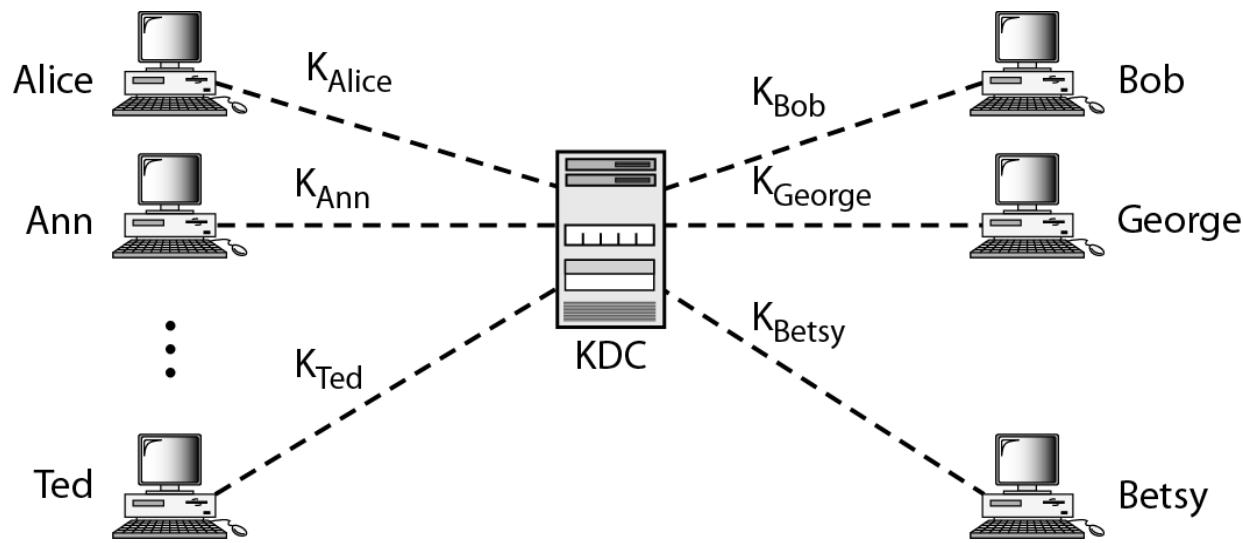
We never discussed how secret keys in symmetric-key cryptography and how public keys in asymmetric-key cryptography are distributed and maintained. In this section, we touch on these two issues. We first discuss the distribution of symmetric keys; we then discuss the distribution of asymmetric keys.

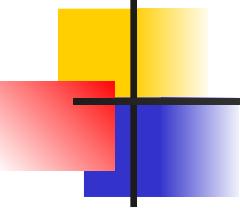
Topics discussed in this section:

Symmetric-Key Distribution

Public-Key Distribution

Figure 31.19 KDC (key distribution center)

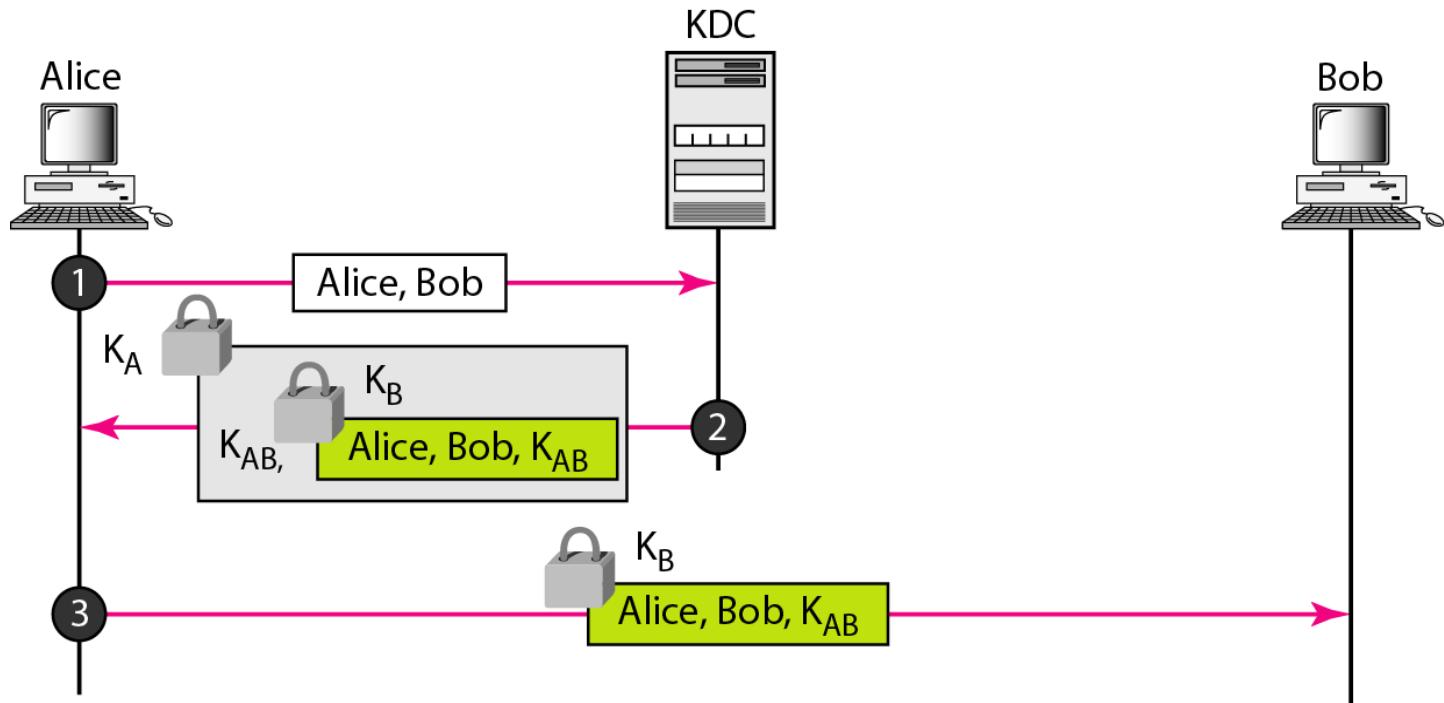


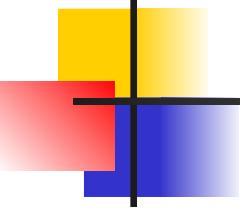


Note

A session symmetric key between two parties is used only once.

Figure 31.30 *Creating a session key between Alice and Bob using KDC*

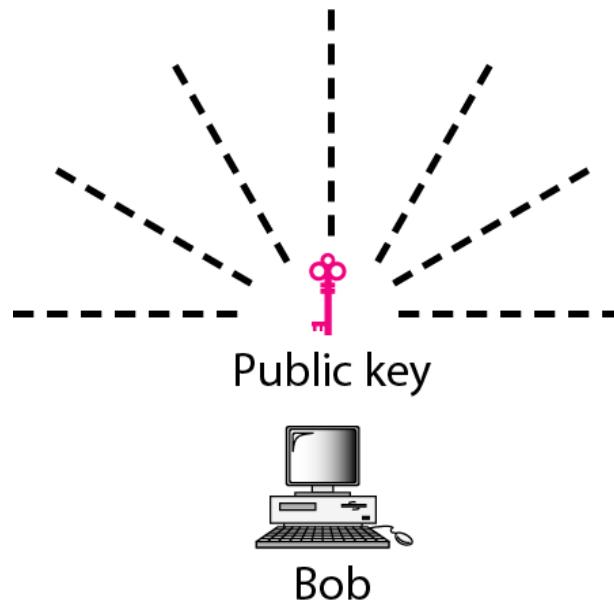




Note

In public-key cryptography, everyone has access to everyone's public key; public keys are available to the public.

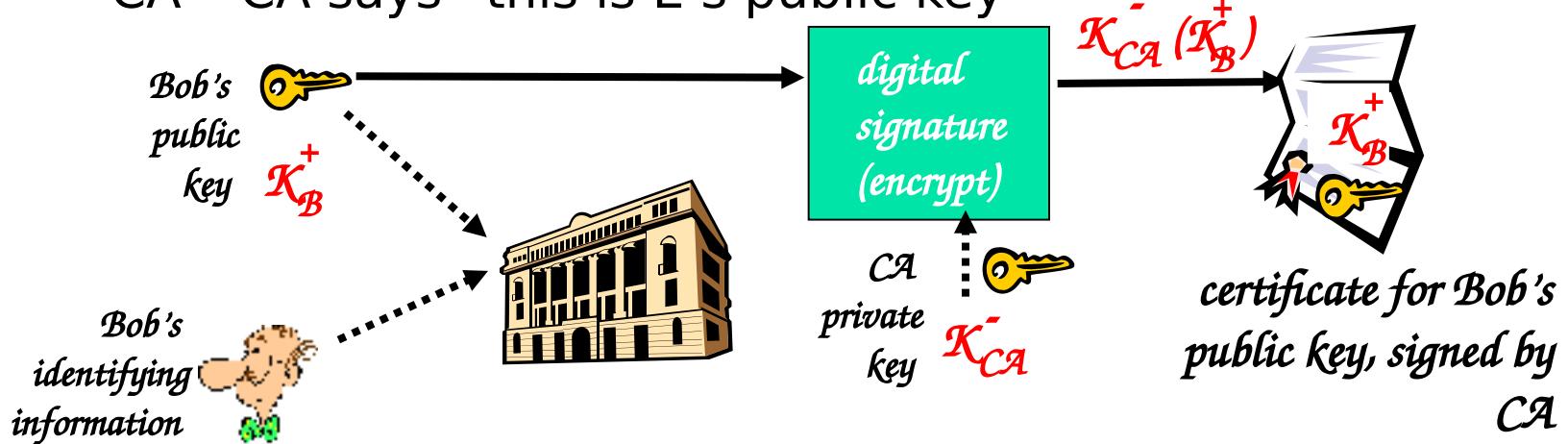
Figure 31.23 *Announcing a public key*



Problem: How can you know what you get is really Bob's public key?

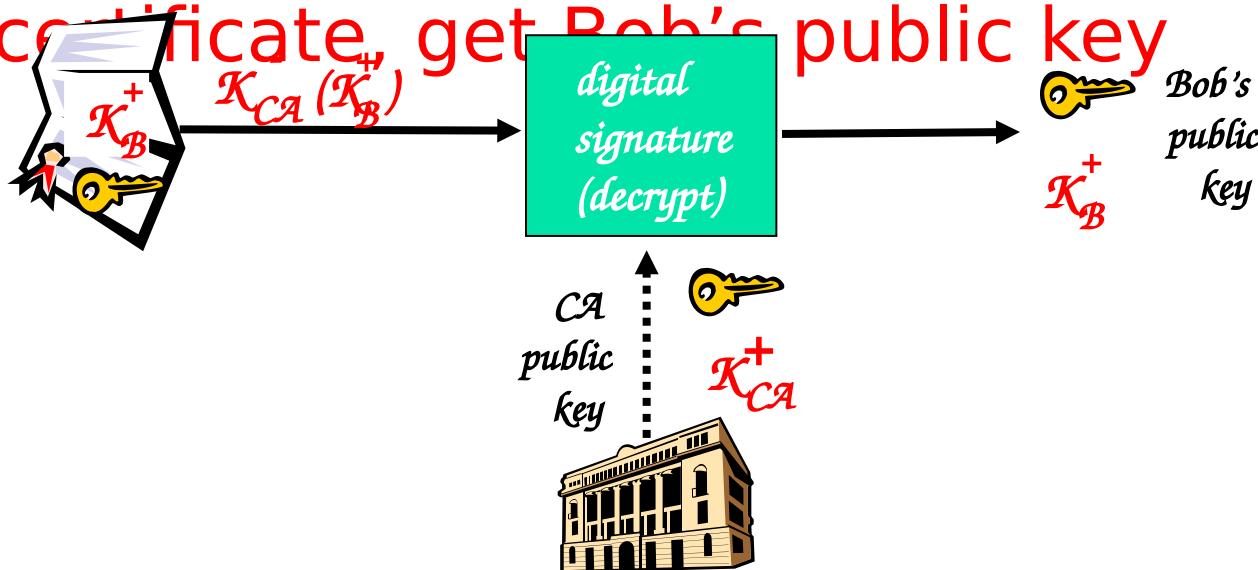
Certification Authorities

- Certification authority (CA): binds public key to particular entity, E.
- E (person, router) registers its public key with CA.
 - E provides “proof of identity” to CA.
 - CA creates certificate binding E to its public key.
 - certificate containing E’s public key digitally signed by CA – CA says “this is E’s public key”



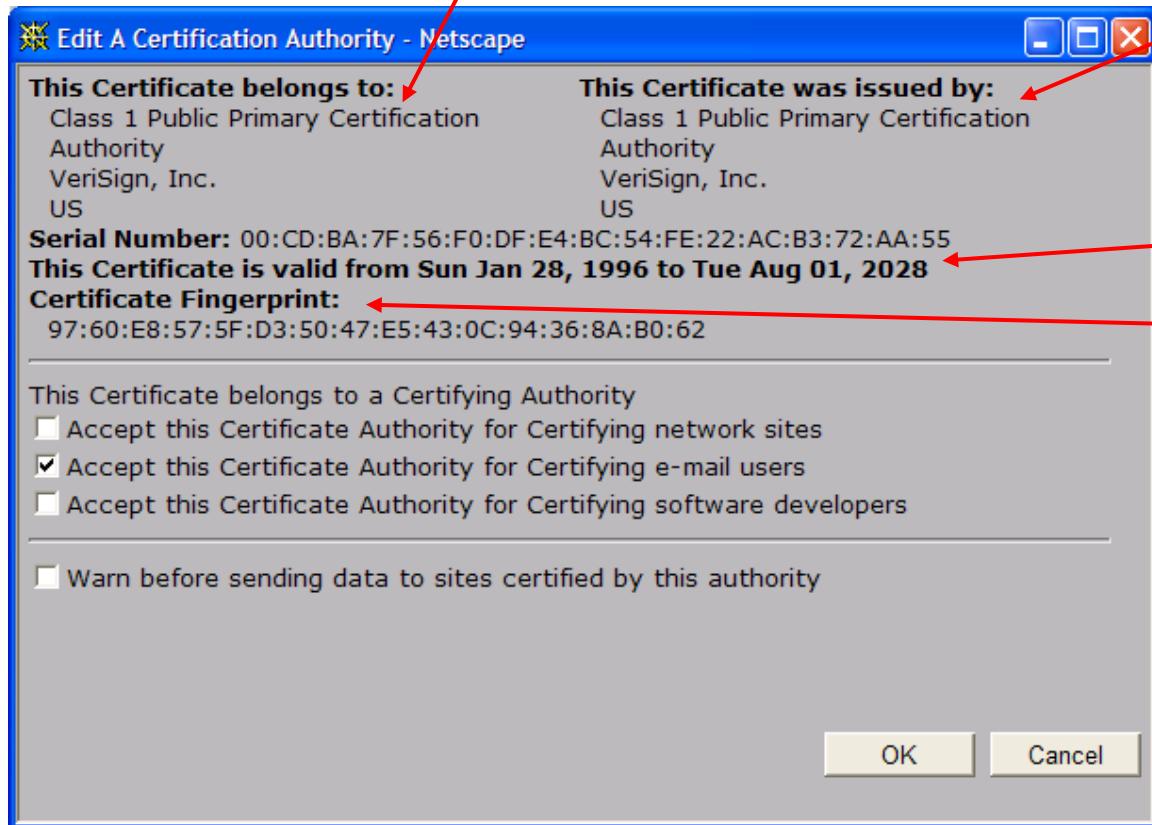
Certification Authorities

- When Alice wants Bob's public key:
 - gets Bob's certificate (Bob or elsewhere).
 - apply CA's public key to Bob's certificate, get Bob's public key



A certificate contains:

- Serial number (unique to issuer)
- info about **certificate owner**, including algorithm and key value itself (not shown)

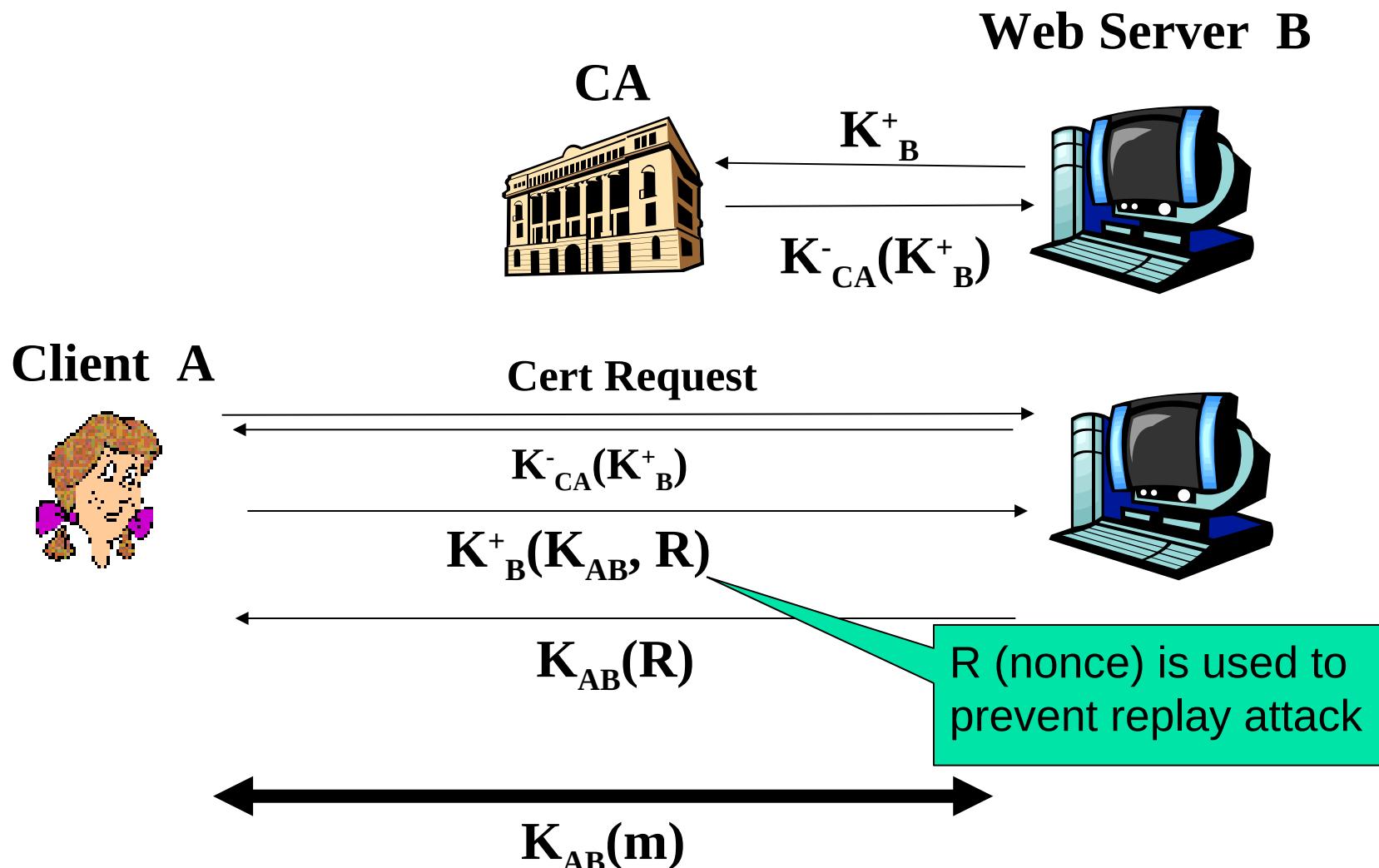


*info about
certificate issuer*

valid dates

*digital signature
by issuer*

Internet Web Security Architecture



Internet Web Security Conditions

- Clients' web browsers have built-in CAs.
- CAs are trustable
- Web servers have certificates in CAs.

- Q: What if a server has no certificate?
 - Example: SSH servers

CO-INS:Information and Network Security

Mathematics

Soma Saha (PhD)

Department of Computer Engineering
SGSITS Indore, India

March 30, 2021

Set of Integers

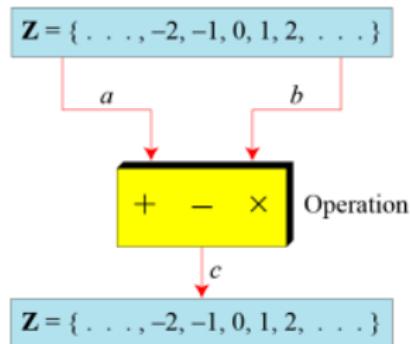
- The set of integers consists of zero (0), the positive natural numbers (1, 2, 3, ...), also called whole numbers or counting numbers, and their additive inverses (the negative integers, i.e., -1, -2, -3, ...).

$$\mathbf{Z} = \{ \dots, -2, -1, 0, 1, 2, \dots \}$$

Binary Operations on Set of Integers

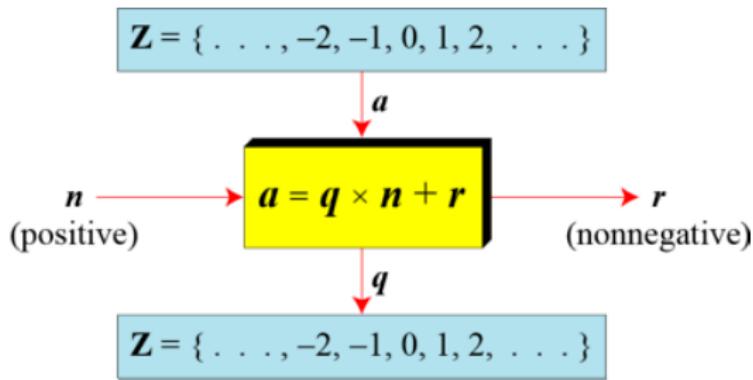
- A binary operation takes two inputs and creates one output.

Figure 1: Three Binary Operations for the set of integers:



Division Operation on Set of Integers

Figure 2: Division Algorithm for Integers.



Division Operation on Set of Integers: Example

- How can we add restriction that remainder r will always be **positive**?

$$-255 = (-23 \times 11) + (-2) \quad \leftrightarrow \quad -255 = (-24 \times 11) + 9$$

Modular Arithmetic

- The division relationship ($a = q \times n + r$) has two inputs (a and n) and two outputs (q and r). In modular arithmetic, we are interested in only one of the outputs, the remainder r.
 1. Modular Operator
 2. Set of Residues
 3. Congruence
 4. Operations in Z_n
 5. Addition and Multiplication Tables
 6. Different Sets

Modulo Operator

- The modulo operator is shown as **mod**.
- The second input (n) is called the modulus.
- The output r is called the residue.

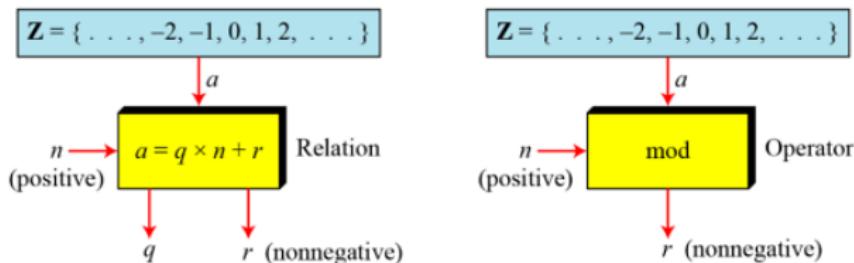


Figure 3: Division Algorithm and Modulo Operator.

Modulo Operator: Examples

- Find the result of the following operations:

- a. $27 \bmod 5$
- b. $36 \bmod 12$
- c. $-18 \bmod 14$
- d. $-7 \bmod 10$
- e. $-36 \bmod 5$
- f. $-27 \bmod 12$

Set of Residues: Z_n

- The result of the modulo operation with modulus n is always an integer between 0 and $n - 1$.
- The modulo opeartion creates a set, which in modular arithmetic is referred to as the **set of least residues modulo n**, or Z_n .

$$Z_n = \{ 0, 1, 2, 3, \dots, (n - 1) \}$$

$$Z_2 = \{ 0, 1 \}$$

$$Z_6 = \{ 0, 1, 2, 3, 4, 5 \}$$

$$Z_{11} = \{ 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 \}$$

Figure 4: Some Z_n sets

Congruence

- To show that two integers are congruent, we use the congruence operator (\equiv). For example, we write:

$$2 \equiv 12 \pmod{10}$$

$$3 \equiv 8 \pmod{5}$$

$$13 \equiv 23 \pmod{10}$$

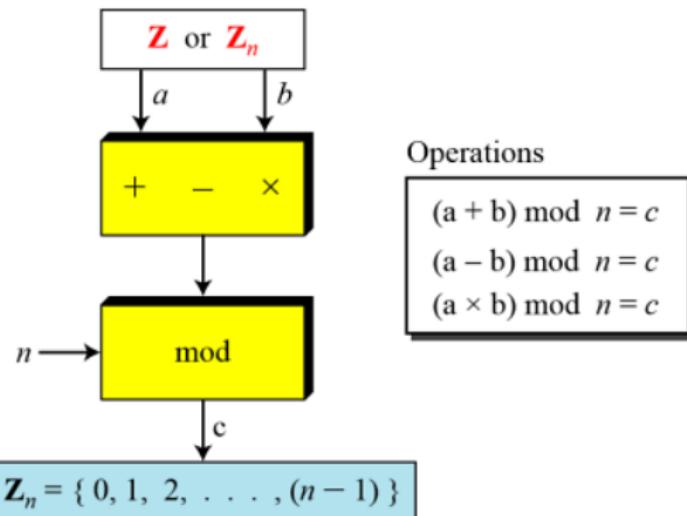
$$8 \equiv 13 \pmod{5}$$

- $2 \bmod 10 = 2$,
- $12 \bmod 10 = 2$,
- $22 \bmod 10 = 2$,

In modular arithmetic, 2, 12, 22 are called congruent mod 10.

Operations in Z_n

- the three binary operations that we used for the set Z , can also be defined for the set Z_n .
- The result may need to be mapped to Z_n using the mod operator.



Operations in Z_n : Examples..1

- Perform the following operations (the inputs come from Z_n):
 - a. Add 7 to 14 in Z_{15} .
 - b. Subtract 11 from 7 in Z_{13} .
 - c. Multiply 11 by 7 in Z_{20} .

Operations in Z_n : Examples..1

- Perform the following operations (the inputs come from Z_n):
 - a. Add 7 to 14 in Z_{15} .
 - b. Subtract 11 from 7 in Z_{13} .
 - c. Multiply 11 by 7 in Z_{20} .

$$(14 + 7) \text{ mod } 15 \rightarrow (21) \text{ mod } 15 = 6$$

$$(7 - 11) \text{ mod } 13 \rightarrow (-4) \text{ mod } 13 = 9$$

$$(7 \times 11) \text{ mod } 20 \rightarrow (77) \text{ mod } 20 = 17$$

Operations in Z_n : Examples..2

- Perform the following operations (the inputs come from either Z or Z_n):
 - a. Add 17 to 27 in Z_{14} .
 - b. Subtract 34 from 12 in Z_{13} .
 - c. Multiply 123 by -10 in Z_{19} .

Operations in Z_n : Examples..2

- Perform the following operations (the inputs come from either Z or Z_n):

- a. Add 17 to 27 in Z_{14} .
 - b. Subtract 34 from 12 in Z_{13} .
 - c. Multiply 123 by -10 in Z_{19} .
-

a. $(17 + 27) \text{ mod } 14 \rightarrow (44) \text{ mod } 14 = 2$

b. $(12 - 43) \text{ mod } 13 \rightarrow (-31) \text{ mod } 13 = 8$

c. $(123 \times (-10)) \text{ mod } 19 \rightarrow (-1230) \text{ mod } 19 = 5$

Properties of mod operations for Z_n

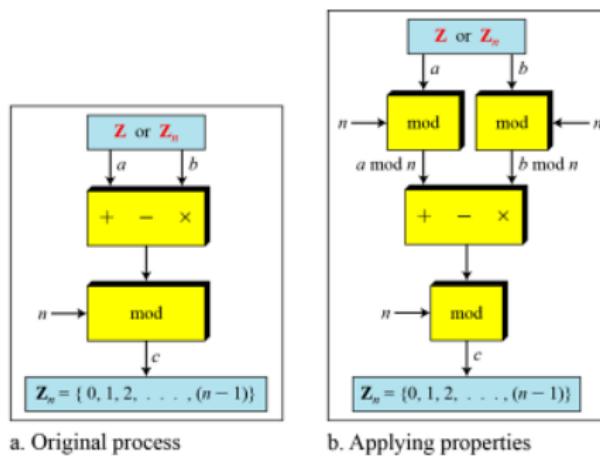
- The following properties allow us to first map the two inputs to Z_n (if they are coming from \mathbb{Z}) before applying the three binary operations (+, -, ×).

First Property: $(a + b) \text{ mod } n = [(a \text{ mod } n) + (b \text{ mod } n)] \text{ mod } n$

Second Property: $(a - b) \text{ mod } n = [(a \text{ mod } n) - (b \text{ mod } n)] \text{ mod } n$

Third Property: $(a \times b) \text{ mod } n = [(a \text{ mod } n) \times (b \text{ mod } n)] \text{ mod } n$

Properties of mod operator



- The properties allow us to work with smaller/reduced numbers.

Application of mentioned properties:

$$(1,723,345 + 2,124,945) \bmod 11 = (8 + 9) \bmod 11 = 6$$

$$(1,723,345 - 2,124,945) \bmod 16 = (8 - 9) \bmod 11 = 10$$

$$(1,723,345 \times 2,124,945) \bmod 16 = (8 \times 9) \bmod 11 = 6$$

Inverses

- When we are working with modular arithmetic, we often need to find the inverse of a number relative to an operation.
 - Additive Inverse (relative to addition operation).
 - Multiplicative Inverse (relative to multiplication operation)

Additive Inverse

- In \mathbb{Z}_n , two numbers a and b are additive inverses of each other if

$$a + b \equiv 0 \pmod{n}$$

- In modular arithmetic, each integer has an additive inverse.
- The sum of an integer and its additive inverse is congruent to 0 modulo n.

Additive Inverse: Example

- Find all additive inverse pairs in Z_{10} .

Additive Inverse: Example

- Find all additive inverse pairs in Z_{10} .
- Six pairs: (0, 0), (1, 9), (2, 8), (3, 7), (4, 6), (5, 5).

Multiplicative Inverse

- In \mathbb{Z}_n , two numbers a and b are multiplicative inverses of each other if

$$a \times b \equiv 1 \pmod{n}$$

- In modular arithmetic, each integer may or may not have a multiplicative inverse.
- When there exists multiplicative inverse for an integer, the product of the integer and the multiplicative inverse is congruent to 1 modulo n.

Multiplicative Inverse: Example

- 1 Find the multiplicative inverse of 8 in Z_{10} .

Multiplicative Inverse: Example

- 1 Find the multiplicative inverse of 8 in Z_{10} .
 - There is no multiplicative inverse, because $\gcd(10,8) = 2 \neq 1$. In other words, we cannot find a number between 0 and 9 such that when multiplied by 8, the result is congruent to 1.
2. Find the multiplicative inverses in Z_{10} .

Multiplicative Inverse: Example

- 1 Find the multiplicative inverse of 8 in Z_{10} .
 - There is no multiplicative inverse, because $\gcd(10,8) = 2 \neq 1$. In other words, we cannot find a number between 0 and 9 such that when multiplied by 8, the result is congruent to 1.
2. Find the multiplicative inverses in Z_{10} .
 - there are only 3 pairs: (1, 1), (3, 7), and (9, 9). The numbers 0,2,4,5,6, and 8 do not have a multiplicative inverse. We can see that,
 $(1 \times 1) \bmod 10 = 1$,
 $(3 \times 7) \bmod 10 = 1$,
 $(9 \times 9) \bmod 10 = 1$
 - **The integer a in Z_n has a multiplicative inverse if and only if $\gcd(n, a) \equiv 1 \pmod{n}$**

Addition and Multiplication Tables

	0	1	2	3	4	5	6	7	8	9
0	0	1	2	3	4	5	6	7	8	9
1	1	2	3	4	5	6	7	8	9	0
2	2	3	4	5	6	7	8	9	0	1
3	3	4	5	6	7	8	9	0	1	2
4	4	5	6	7	8	9	0	1	2	3
5	5	6	7	8	9	0	1	2	3	4
6	6	7	8	9	0	1	2	3	4	5
7	7	8	9	0	1	2	3	4	5	6
8	8	9	0	1	2	3	4	5	6	7
9	9	0	1	2	3	4	5	6	7	8

Addition Table in \mathbf{Z}_{10}

	0	1	2	3	4	5	6	7	8	9
0	0	0	0	0	0	0	0	0	0	0
1	0	1	2	3	4	5	6	7	8	9
2	0	2	4	6	8	0	2	4	6	8
3	0	3	6	9	2	5	8	1	4	7
4	0	4	8	2	6	0	4	8	2	6
5	0	5	0	5	0	5	0	5	0	5
6	0	6	2	8	4	0	6	2	8	4
7	0	7	4	1	8	0	2	9	6	3
8	0	8	6	4	2	0	8	6	4	2
9	0	9	8	7	6	5	4	3	2	1

Multiplication Table in \mathbf{Z}_{10}

Different Sets

- In cryptography, we often work with inverses.
- If the sender uses an integer (as the encryption key), the receiver uses the inverse of that integer (as the decryption key).
- If the operation (encryption/decryption algorithm) is addition, Z_n can be used as the set of possible keys because each integer in this set has an additive inverse.
- if the operation (encryption/decryption algorithm) is multiplication, Z_n cannot be the set of possible keys because only some members of this set have a multiplicative inverse.
- We need another set; the new set, which is a subset of Z_n , includes only integers in Z_n that have a unique multiplicative inverse. the set is called Z_n^* .

Different Sets: Example

Figure 6: Some Z_n and Z_n^* sets

$$\mathbf{Z}_6 = \{0, 1, 2, 3, 4, 5\}$$

$$\mathbf{Z}_6^* = \{1, 5\}$$

$$\mathbf{Z}_7 = \{0, 1, 2, 3, 4, 5, 6\}$$

$$\mathbf{Z}_7^* = \{1, 2, 3, 4, 5, 6\}$$

$$\mathbf{Z}_{10} = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

$$\mathbf{Z}_{10}^* = \{1, 3, 7, 9\}$$

The Index of Coincidence (IC)

(content courtesy: Debdeep Mukhopadhyay)

- Can be used to determine m as well as to confirm m , determined by *Kasiski test*
- Definition: suppose $x=x_1x_2,\dots,x_n$ is a string of length n .
- The *index of coincidence* of x , denoted by $I_c(x)$, is defined to be the probability that two random elements of x are identical.
 - Denoted the frequencies of A,B,...,Z in x by f_0,f_1,\dots,f_{25}

$$I_c(x) = \frac{\sum_{i=0}^{25} \binom{f_i}{2}}{\binom{n}{2}} = \frac{\sum_{i=0}^{25} f_i(f_i-1)}{n(n-1)}$$

The Index of Coincidence (IC)..cont..(content

courtesy: Debdeep Mukhopadhyay)

An Important Property:

Suppose x is a string of English text, denote the expected probability of occurrences of A,B,...,Z by p_0,p_1,\dots,p_{25} with values from the frequency graph, then:

- probability that two random elements both are A is p_0^2 , both are B is p_1^2,\dots
- then $I_c(x) \approx \sum p_i^2 = 0.082^2 + 0.015^2 + \dots + 0.001^2 = 0.065$

Question: if y is a ciphertext obtained by **shift cipher**, what is the $I_c(y)$?

Answer: should be 0.065, because the individual probabilities will be permuted, but the $\sum p_i^2$ will be unchanged. So, this is an Invariant.
This Property is used to determine the key.

The Index of Coincidence (IC)..cont..(content

courtesy: Debdeep Mukhopadhyay)

Therefore, suppose $y=y_1y_2\dots y_n$ is the ciphertext from Vigenere cipher.

For any given m , divide y into m substrings:

- | | |
|-------------------------------|--|
| $y_1=y_1y_{m+1}y_{2m+1}\dots$ | if m is indeed the keyword length,
then each y_i is a shift cipher, $I_c(y_i)$
is about 0.065. |
| $y_2=y_2y_{m+2}y_{2m+2}\dots$ | otherwise, $I_c(y_i) \approx 26(1/26)^2 = 0.038$. |
| ... | |
| $y_m=y_my_{2m}y_{3m}\dots$ | |

The Index of Coincidence (IC)..cont..(content

courtesy: Debdeep Mukhopadhyay)

For purpose of verify keyword length m , divide the ciphertext into m substrings, compute the index of coincidence by for each substring. If all IC values of the substrings are around 0.065, then m is the correct keyword length. Otherwise m is not the correct keyword length.

If want to use I_c to determine correct keyword length m , what to do?

Beginning from $m=2,3, \dots$ until an m , for which all substrings have IC value around 0.065.

Now, how to determine keyword $K=(k_1, k_2, \dots, k_m)$? Assume m is given.

Determine Keyword $K = (k_1, k_2, k_3, \dots, k_m)$

(content courtesy: Debdeep Mukhopadhyay)

- Suppose $x=x_1, x_2, \dots, x_n$ and $y=y_1, y_2, \dots, y_{n'}$ are strings of n and n' alphabetic characters respectively.
- The mutual index of coincidence of x and y , denoted by $MI_c(x, y)$, is the probability that a random element of x is equal to that of y .
- Let, the probabilities of A, B, ... be f_0, f_1, \dots, f_{25} and $f_0', f_1', \dots, f_{25}'$ respectively in x and y .

$$MI_c(x, y) = \frac{\sum_{i=0}^{26} f_i f_i'}{nn'}$$

Matrices

- In cryptography, we need to handle matrices.
 1. Definitions
 2. Operations and Relations
 3. Determinants
 4. Residue Matrices

Definitions

Figure 7: A matrix of size $l \times m$

m columns

Matrix A:

rows

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \vdots & \vdots & & \vdots \\ a_{l1} & a_{l2} & \dots & a_{lm} \end{bmatrix}$$

Examples of Matrices

Figure 8: Examples of Matrices

$$\begin{bmatrix} 2 & 1 & 5 & 11 \end{bmatrix}$$

Row matrix

$$\begin{bmatrix} 2 \\ 4 \\ 12 \end{bmatrix}$$

Column
matrix

$$\begin{bmatrix} 23 & 14 & 56 \\ 12 & 21 & 18 \\ 10 & 8 & 31 \end{bmatrix}$$

Square
matrix

$$\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix}$$

0

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

I

Addition and Subtraction on Matrices

Figure 9: Addition and subtraction of Matrices

$$\begin{bmatrix} 12 & 4 & 4 \\ 11 & 12 & 30 \end{bmatrix} = \begin{bmatrix} 5 & 2 & 1 \\ 3 & 2 & 10 \end{bmatrix} + \begin{bmatrix} 7 & 2 & 3 \\ 8 & 10 & 20 \end{bmatrix}$$

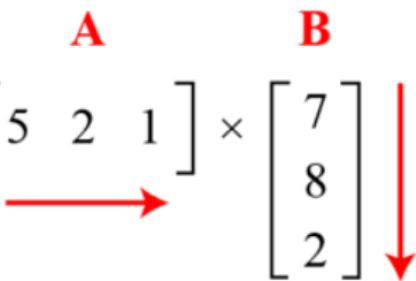
C = A + B

$$\begin{bmatrix} -2 & 0 & -2 \\ -5 & -8 & 10 \end{bmatrix} = \begin{bmatrix} 5 & 2 & 1 \\ 3 & 2 & 10 \end{bmatrix} - \begin{bmatrix} 7 & 2 & 3 \\ 8 & 10 & 20 \end{bmatrix}$$

D = A - B

Matrix Multiplication: Example 1

Figure 10: Multiplication of Matrices: The product of a row matrix (1×3) by a column matrix (3×1). The result is a matrix of size 1×1 .

$$\begin{matrix} \mathbf{C} & \mathbf{A} & \mathbf{B} \\ \left[\begin{matrix} 5 & 3 \end{matrix} \right] & = & \left[\begin{matrix} 5 & 2 & 1 \end{matrix} \right] \times \left[\begin{matrix} 7 \\ 8 \\ 2 \end{matrix} \right] \end{matrix}$$


In which:

$$53 = 5 \times 7 + 2 \times 8 + 1 \times 2$$

Matrix Multiplication: Example 2

Figure 11: Multiplication of Matrices: The product of a 2×3 matrix by a 3×4 matrix. The result is a 2×4 matrix..

$$\begin{matrix} & \textcolor{red}{C} \\ \left[\begin{matrix} 52 & 18 & 14 & 9 \\ 41 & 21 & 22 & 7 \end{matrix} \right] & = & \left[\begin{matrix} 5 & 2 & 1 \\ 3 & 2 & 4 \end{matrix} \right] & \times & \left[\begin{matrix} 7 & 3 & 2 & 1 \\ 8 & 0 & 0 & 2 \\ 1 & 3 & 4 & 0 \end{matrix} \right] & \textcolor{red}{B} \end{matrix}$$

Scalar Matrix Multiplication

Figure 12: Scalar Multiplication

$$\mathbf{B} = 3 \times \mathbf{A}$$
$$\begin{bmatrix} 15 & 6 & 3 \\ 9 & 6 & 12 \end{bmatrix} = 3 \times \begin{bmatrix} 5 & 2 & 1 \\ 3 & 2 & 4 \end{bmatrix}$$

Identity Matrix

(content courtesy: Prof. Kindred)

Definition The *identity matrix*, denoted I_n , is the $n \times n$ diagonal matrix with all ones on the diagonal.

$$I_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

If A is an $m \times n$ matrix, then

$$I_m A = A \quad \text{and} \quad A I_n = A.$$

If A is a square matrix, then

$$IA = A = AI.$$

Important property
of identity matrix

Determinant

Definition of the Determinant Let $A = [a_{ij}]_{n \times n}$ be an $n \times n$ matrix.

(1) If $n = 1$, that is $A = [a_{11}]$, then we define $\det(A) = a_{11}$.

(2) If $n > 1$, we define $\det(A) = \sum_{k=1}^n (-1)^{1+k} a_{1k} \det(A_{1k})$

Determinant: Example 1

Figure 13: The determinant of a 2×2 matrix based on the determinant of a 1×1 matrix.

$$\det \begin{bmatrix} 5 & 2 \\ 3 & 4 \end{bmatrix} = (-1)^{1+1} \times 5 \times \det[4] + (-1)^{1+2} \times 2 \times \det[3] \longrightarrow 5 \times 4 - 2 \times 3 = 14$$

or

$$\det \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = a_{11} \times a_{22} - a_{12} \times a_{21}$$

Determinant: Example 2

Figure 14: The calculation of the determinant of a 3×3 matrix.

$$\begin{aligned}\det \begin{bmatrix} 5 & 2 & 1 \\ 3 & 0 & -4 \\ 2 & 1 & 6 \end{bmatrix} &= (-1)^{1+1} \times 5 \times \det \begin{bmatrix} 0 & -4 \\ 1 & 6 \end{bmatrix} + (-1)^{1+2} \times 2 \times \det \begin{bmatrix} 3 & -4 \\ 2 & 6 \end{bmatrix} + (-1)^{1+3} \times 1 \times \det \begin{bmatrix} 3 & 0 \\ 2 & 1 \end{bmatrix} \\ &= (+1) \times 5 \times (+4) \quad + \quad (-1) \times 2 \times (24) \quad + \quad (+1) \times 1 \times (3) = -25\end{aligned}$$

Inverses

- Additive inverse: $A+B=0$
 $= -A$
- Multiplicative inverse: If a square matrix A is a square matrix B , such that
 $A \times B = B \times A = I$.
- Multiplicative inverse exists only if the $\det(A)$ has a multiplicative inverse in the corresponding set.

The notion of Inverses

(content courtesy: Prof. Kindred)

Exploration Consider the set of real numbers, and say that we have the equation

$$3x = 2$$

and we want to solve for x .

What do we do?

We multiply both sides of the equation by $\frac{1}{3}$ to obtain

$$\frac{1}{3}(3x) = \frac{1}{3}(2) \implies x = \frac{2}{3}.$$

multiplicative inverse
of 3 since $\frac{1}{3}(3) = 1$

Now, consider the linear system

$$\begin{aligned} 3x_1 - 5x_2 &= 6 \\ -2x_1 + 3x_2 &= -1 \end{aligned}$$

Notice that we can rewrite equations as

$$\underbrace{\begin{bmatrix} 3 & -5 \\ -2 & 3 \end{bmatrix}}_A \underbrace{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}}_{\vec{x}} = \underbrace{\begin{bmatrix} 6 \\ -1 \end{bmatrix}}_{\vec{b}}$$

How do we isolate the vector \vec{x} by itself on LHS?

The notion of Inverses..

(content courtesy: Prof. Kindred)

Now, consider the linear system

$$\begin{aligned}3x_1 - 5x_2 &= 6 \\-2x_1 + 3x_2 &= -1\end{aligned}$$

Notice that we can rewrite equations as

$$\underbrace{\begin{bmatrix} 3 & -5 \\ -2 & 3 \end{bmatrix}}_A \underbrace{\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}}_{\vec{x}} = \underbrace{\begin{bmatrix} 6 \\ -1 \end{bmatrix}}_{\vec{b}}$$

How do we isolate the vector \vec{x} by itself on LHS?

$$\underbrace{\left[\begin{array}{cc} ? & ? \end{array} \right]}_{\text{want this equal to identity matrix, } I} \left(\begin{bmatrix} 3 & -5 \\ -2 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \right) = \left[\begin{array}{cc} ? & ? \end{array} \right] \begin{bmatrix} 6 \\ -1 \end{bmatrix}$$

want this equal to identity matrix, I

$$\begin{bmatrix} -3 & -5 \\ -2 & -3 \end{bmatrix}$$

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} -3 & -5 \\ -2 & -3 \end{bmatrix} \begin{bmatrix} 6 \\ -1 \end{bmatrix} = \boxed{\begin{bmatrix} -13 \\ -9 \end{bmatrix}}$$

Matrix Inverses

Definition A square matrix A is *invertible* (or *nonsingular*) if \exists matrix B such that $AB = I$ and $BA = I$. (We say B is an *inverse* of A .)

Example

$A = \begin{bmatrix} 2 & 7 \\ 1 & 4 \end{bmatrix}$ is invertible because for $B = \begin{bmatrix} 4 & -7 \\ -1 & 2 \end{bmatrix}$,

$$\text{we have } AB = \begin{bmatrix} 2 & 7 \\ 1 & 4 \end{bmatrix} \begin{bmatrix} 4 & -7 \\ -1 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I$$

$$\text{and likewise } BA = \begin{bmatrix} 4 & -7 \\ -1 & 2 \end{bmatrix} \begin{bmatrix} 2 & 7 \\ 1 & 4 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I.$$

Inverse Matrix: Property

The notion of an inverse matrix only applies to square matrices.

Example Find the inverse of $A = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$. We have

$$\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \implies \begin{bmatrix} a+c & b+d \\ a+c & b+d \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$
$$\implies a+c = 1 \text{ and } a+c = 0 \quad \text{IMPOSSIBLE!}$$

Inverse of 2×2 Matrix: Method

(content courtesy: Prof. Kindred)

Inverse of a 2×2 matrix: Consider the special case where A is a 2×2 matrix with $A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$. If $ad - bc \neq 0$, then A is invertible and its inverse is

$$A^{-1} = \frac{1}{ad - bc} \begin{bmatrix} d & -b \\ -c & a \end{bmatrix}.$$

Inverse of 2 x 2 Matrix: Example

(content courtesy:
Prof. Kindred)

Example For $A = \begin{bmatrix} -2 & 1 \\ 3 & -3 \end{bmatrix}$, we have

$$A^{-1} = \frac{1}{3} \begin{bmatrix} -3 & -1 \\ -3 & -2 \end{bmatrix} = \begin{bmatrix} -1 & -\frac{1}{3} \\ -1 & -\frac{2}{3} \end{bmatrix}.$$

We can easily check that

$$AA^{-1} = \begin{bmatrix} -2 & 1 \\ 3 & -3 \end{bmatrix} \begin{bmatrix} -1 & -\frac{1}{3} \\ -1 & -\frac{2}{3} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

and

$$A^{-1}A = \begin{bmatrix} -1 & -\frac{1}{3} \\ -1 & -\frac{2}{3} \end{bmatrix} \begin{bmatrix} -2 & 1 \\ 3 & -3 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

Inverse of 3 x 3 Matrix: Example

(content courtesy:

Prof. Kindred)

Example: Find the inverse of the matrix $A = \begin{bmatrix} -1 & -3 & 1 \\ 3 & 6 & 0 \\ 1 & 0 & 1 \end{bmatrix}$.

$$\begin{array}{c} \left[\begin{array}{ccc|ccc} -1 & -3 & 1 & 1 & 0 & 0 \\ 3 & 6 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 & 1 \end{array} \right] \xrightarrow[R_2+3R_1]{R_3+R_1} \left[\begin{array}{ccc|ccc} -1 & -3 & 1 & 1 & 0 & 0 \\ 0 & -3 & 3 & 3 & 1 & 0 \\ 0 & -3 & 2 & 1 & 0 & 1 \end{array} \right] \\ \xrightarrow[-R_1]{R_3-R_2} \left[\begin{array}{ccc|ccc} 1 & 3 & -1 & -1 & 0 & 0 \\ 0 & -3 & 3 & 3 & 1 & 0 \\ 0 & 0 & -1 & -2 & -1 & 1 \end{array} \right] \\ \xrightarrow[R_1+R_2]{-R_3} \left[\begin{array}{ccc|ccc} 1 & 0 & 2 & 2 & 1 & 0 \\ 0 & -3 & 3 & 3 & 1 & 0 \\ 0 & 0 & 1 & 2 & 1 & -1 \end{array} \right] \\ \xrightarrow[-\frac{1}{3}R_2]{} \left[\begin{array}{ccc|ccc} 1 & 0 & 2 & 2 & 1 & 0 \\ 0 & 1 & -1 & -1 & -\frac{1}{3} & 0 \\ 0 & 0 & 1 & 2 & 1 & -1 \end{array} \right] \\ \xrightarrow[R_1+2R_3]{R_2+R_3} \left[\begin{array}{ccc|ccc} 1 & 0 & 0 & -2 & -1 & 2 \\ 0 & 1 & 0 & 1 & \frac{2}{3} & -1 \\ 0 & 0 & 1 & 2 & 1 & -1 \end{array} \right] \end{array}$$

Thus, A is invertible and its inverse is

$$A^{-1} = \begin{bmatrix} -2 & -1 & 2 \\ 1 & \frac{2}{3} & -1 \\ 2 & 1 & -1 \end{bmatrix}$$

Algebraic Structures

- Cryptography requires sets of integers and specific operations that are defined for those sets.
- The combination of the set and the operations that are applied to the elements of the set is called an **algebraic structure**.

Group, Ring, Field, GF

- Taught in class from Behrouz's book "Cryptography and Network Security".

Finding multiplicative Inverse in Galios Field (GF)

- For questing minds find the material from the following link:

[https:](https://engineering.purdue.edu/kak/compsec/NewLectures/Lecture5.pdf)

[//engineering.purdue.edu/kak/compsec/NewLectures/Lecture5.pdf](https://engineering.purdue.edu/kak/compsec/NewLectures/Lecture5.pdf)

Bibliography: Books and Resources

- Cryptography and Network Security: Principles and Practice by William Stallings
- Cryptography and Network Security by Behrouz A Forouzan and Debdeep Mukhopadhyay
- Principles of Information Security by Michael E. Whitman and Herbert J. Mattord.
- Cisco platform, and Internet.
- Published research papers, study materials from researchers of security domain.

CO-INS:Information and Network Security

UNIT-II (Part-I)

Course Instructors:

Soma Saha

Veerendra Srivastava

Soma Saha (PhD)

Department of Computer Engineering
SGSITS Indore, India

March 8, 2021

UNIT-II (Part-I): Learning Objectives

Upon completion of this unit, you should be able to

- LO1 Define the terms and concepts of symmetric-key ciphers
- LO2 Discuss the two broad categories of traditional symmetric-key ciphers with focus on different cipher cryptanalysis
- LO3 Show the idea behind stream ciphers and block ciphers

Cryptography: Symmetric-Key Cipher



Figure 1: Symmetric-Key Encipherment.

- $c = E_s(p, k)$
- $p = D_s(c, k)$
- D_s = Decryption function (symmetric)
- c = ciphertext
- p = plaintext
- k = secret key
- E_s = Encryption function (symmetric)

Symmetric-Key Encipherment: Message exchange Prove

- We can prove that the plaintext created by Bob is the same as the one originated by Alice. We assume that Bob creates p_1 ; we prove that $p_1 = p$:

Symmetric-Key Encipherment: Message exchange Prove

- We can prove that the plaintext created by Bob is the same as the one originated by Alice. We assume that Bob creates p_1 ; we prove that $p_1 = p$:
- Alice : $c = E_s(p, k)$
- Bob: $p_1 = D_s(c, k) = D_s(E_s(p, k), k) = p$

Symmetric-Key Encipherment: Message exchange Prove

- We can prove that the plaintext created by Bob is the same as the one originated by Alice. We assume that Bob creates p_1 ; we prove that $p_1 = p$:
 - Alice : $c = E_s(p, k)$
 - Bob: $p_1 = D_s(c, k) = D_s(E_s(p, k), k) = p$
- **Kerckhoff's Principle:** A cryptosystem should be secure even if everything about the system, except the key, is public knowledge.
- Kerckhoff's principle was reformulated (or possibly independently formulated) by American mathematician **Claude Shannon** as "the enemy knows the system", i.e., "one ought to design systems under the assumption that the enemy will immediately gain full familiarity with them". In that form, it is called **Shannon's maxim**.(Source: wikipedia)

Symmetric-Key Encipherment: How to compute #keys?

- If there are m people in a group who need to communicate with each other, how many keys are needed?

Symmetric-Key Encipherment: How to compute #keys?

- If there are m people in a group who need to communicate with each other, how many keys are needed?

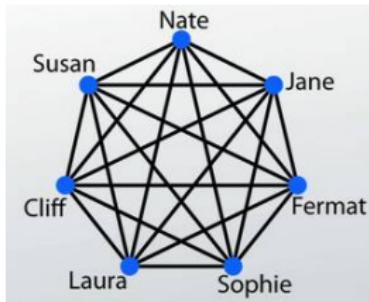


Figure 2: Example scenario.

Cryptography-Cryptanalysis-Cryptology

- Cryptography - Science and art of creating secret codes.
- Cryptanalysis - Science and art of breaking those codes.
- The study of cryptanalysis helps us create better secret codes.

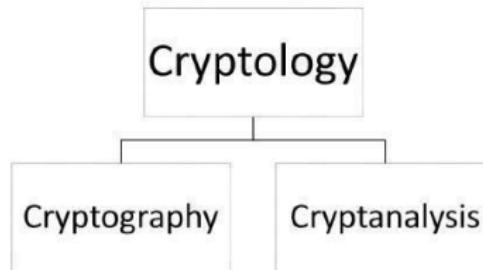


Figure 3: Cryptography-Cryptanalysis-Cryptology.

Cryptanalysis Attacks

- Classification based on encryption techniques:
 1. Ciphertext-only attack
 2. Known-plaintext attack
 3. Chosen-Plaintext attack
 4. Chosen-Ciphertext attack
 5. Chosen-text attack

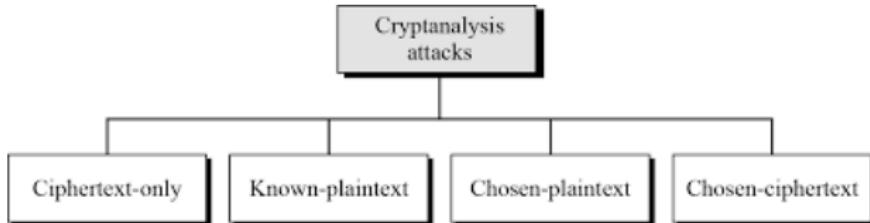


Figure 4: Classification of cryptanalysis attacks based on encryption techniques.

Ciphertext-Only Attack

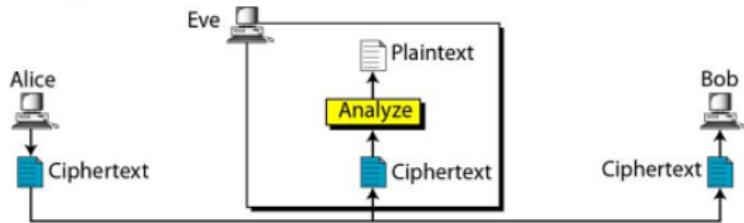


Figure 5: Ciphertext-only attack.

- Adversary, Eve/Darth has access to only some ciphertext.
- **Assumption:** Eve/Darth knows the algorithm and can intercept the ciphertext.

Ciphertext-Only Attack: various methods

- **Brute-Force Attack/Exhaustive-key-search Attack:** Eve/Darth tries to use all possible keys.
- **Assumption:**
 - Eve/Darth knows the algorithm.
 - Eve/Darth knows the key domain (the list of all possible keys).
- **Application:** Using the intercepted cipher, Eve/Darth decrypts the ciphertext with every possible key until the plaintext makes sense.
- **Prevention:** The number of possible keys must be very large.

Ciphertext-Only Attack: various methods..cont..1

- **Statistical Attack:** The cryptanalyst can benefit from some inherent characteristics of the plaintext language to launch a **statistical attack**.
- **Example:**
 - The letter E is the most frequently used letter in English text.
 - The cryptanalyst finds the mostly-used character in the ciphertext and assumes that the corresponding plaintext character is E.
 - After finding a few pairs, the analyst can find the key and use it to decrypt the message.
- **Prevention:** The cipher should hide the characteristics of the language.

Ciphertext-Only Attack: various methods..cont..2

- **Pattern Attack:** Some ciphers may hide characteristics of the language, but may create some patterns in the ciphertext. A cryptanalyst may use a **pattern attack** to break the cipher.
- **Prevention:** Important to use ciphers that make the ciphertext as random as possible.

Known-Plaintext Attack

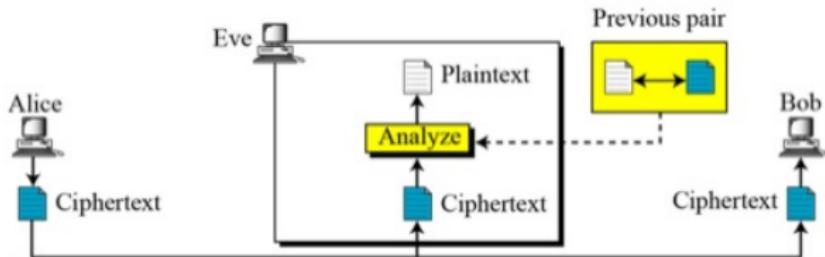


Figure 6: Known-Plaintext Attack.

- **Known-Plaintext Attack:** Eve/Darth has access to some plaintext/ciphertext pairs in addition to the intercepted ciphertext that she/he wants to break.
 - The plaintext/ciphertext pairs have been collected earlier.
 - This type of attacks are less likely to happen, because...

Chosen-Plaintext Attack

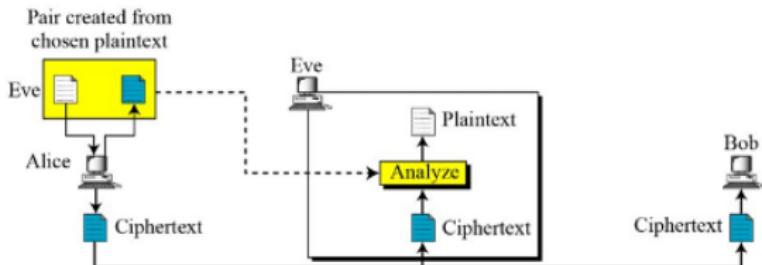


Figure 7: Chosen-Plaintext Attack.

- **Chosen-Plaintext Attack:** This attack is similar to the known-plaintext attack, but the plaintext/ciphertext pairs have been chosen by the attacker herself.
 - Example: If Eve/Darth has access to Alice's computer, she/he can choose some plaintext and intercept the captured ciphertext.
 - She/he does not have the key because the key is normally **embedded in the software** used by the sender.
 - Easy to implement, but less likely to happen, because..

Chosen-Ciphertext Attack

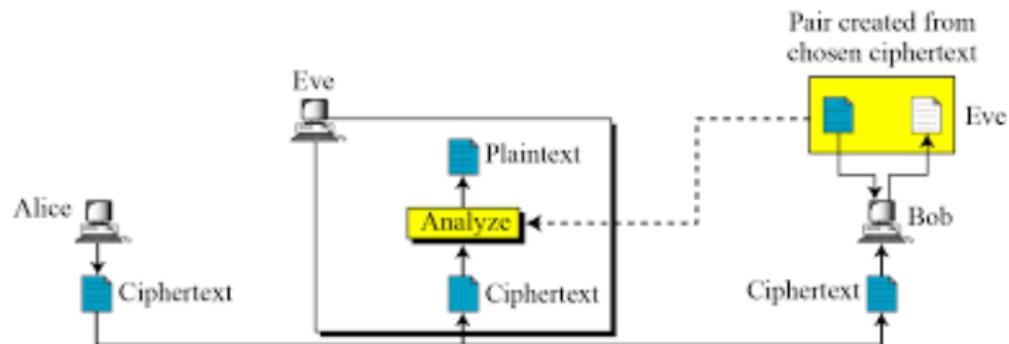


Figure 8: Chosen-Ciphertext Attack.

- **Chosen-ciphertext attack** is similar to the chosen-plaintext attack, except that Eve/Darth chooses some ciphertext and decrypts it to form a ciphertext/plaintext pair.
- This can happen if Eve/Darth has access to Bob's computer.

Hypothetical bad symmetric encryption algorithm: XOR

(Content courtesy: Tyler Bletsch, Duke Univ.)

- A lot of encryption algorithms rely on properties of XOR
 - Can think of $A \wedge B$ as "Flip a bit in A if corresponding bit in B is 1"
 - If you XOR by same thing twice, you get the data back
 - XORing by a random bit string yields NO info about original data
 - Each bit has a 50% chance of having been flipped
- Could consider XOR itself to be a symmetric encryption algorithm (but it seems dreadful at it!) - can be illustrative to explore
- Simple XOR encryption algorithm:
 - $E(p, k) = p \wedge k$ (keep repeating k as often as needed to cover p)
 - $D(c, k) = c \wedge k$ (same algorithm both ways!)

A	B	$A \wedge B$
0	0	0
0	1	1
1	0	1
1	1	0

```
>>> a=501  
>>> b=199  
>>> a ^= b  
>>> print a  
306  
>>> a ^= b  
>>> print a  
501
```

XOR “encryption” demo

(Content courtesy: Tyler Bletsch, Duke Univ.)

Plaintext: 'Hello'

Key : 'key'

H e l l o
Plaintext : 01001000 01100101 01101100 01101100 01101111
k e y Key repeats> k e

Key : 01101011 01100101 01111001 01101011 01100101

Ciphertext: 
^ XOR result

Ciphertext: 00100011 00000000 00010101 00000111 00001010

Key : 01101011 01100101 01111001 01101011 01100101

Decrypted : 
^ XOR result

Attacking XOR

(Content courtesy: Tyler Bletsch, Duke Univ.)

Figure 9: Known-Plaintext Attack:

```
Given plaintext : 01001000 01100101 01101100 01101100 01101111  
Given ciphertext : 00100011 00000000 00010101 00000111 00001010  
XOR result      : 01101011 01100101 01111001 01101011 01100101  
                  ^^ it's the key!!!
```

Figure 10: Chosen-Plaintext Attack:

```
Chosen plaintext : 00000000 00000000 00000000 00000000 00000000  
Given ciphertext : 01101011 01100101 01111001 01101011 01100101  
XOR result      : 01101011 01100101 01111001 01101011 01100101  
                  ^^ it's the key!!!
```

Attacking XOR..contd..

(Content courtesy: Tyler Bletsch, Duke Univ.)

Figure 11: Ciphertext-Only Attack:

Ciphertext: 00100011 00000000 00010101 00000111 00001010

- "I assume the plaintext had ASCII text with lowercase letters, and in all such letters bit 6 is 1, but none of the ciphertext has bit 6 set, so I bet the key is most/all lower case letters"
- "The second byte is all zeroes, which means the second byte of the key and plaintext are equal"
- etc...

**** Conclusion: XOR is a dreadful encryption algorithm**

Categories of Traditional Ciphers

- * Traditional Symmetric-Key Ciphers

- **Substitution Ciphers:** We replace one symbol in the ciphertext with another symbol.
- **Transposition Ciphers:** We reorder the position of symbols in the plaintext.

KEY	
Plaintext	ABCDEFGHIJKLMNOPQRSTUVWXYZ
Ciphertext	PQOWIEURYTLAKSJDHFGMZNXBCV
ENCRYPTION	
Plaintext	T H E M O N E Y I S I N T H E B A G
Ciphertext	M R I K J S I C Y G Y S M R I Q P U

Figure 12: Substitution Cipher.

1	2	3	4	5	6
M	E	E	T	M	E
A	F	T	E	R	P
A	R	T	Y		
4	2	1	6	3	5
T	E	M	E	E	M
E	F	A	P	T	R
Y	R	A			T

Plain Text: MEET ME AFTER PARTY

Key Used: 421635

Cipher Text: TEMEEMEFAPTRYRAT

Figure 13: Transposition Cipher.

Substitution Ciphers: Types

- **Substitution Cipher**

1. **Mono-alphabetic** : It only uses one alphabet to substitute.

- Additive/Shift/Caeser
- Multiplicative
- Affine
- Monoalphabetic Substitution Cipher

2. **Poly-alphabetic**: It may use two or more alphabets to substitute.

- Auto-Key Cipher
- Vigenere Cipher
- Playfair cipher
- Hill Cipher
- One Time Pad
- Rotor Cipher

Additive Cipher/Shift Cipher

- Each letter in the plaintext is replaced by a letter some fixed number of positions down the alphabet.
- For example, with a left shift of 3, D would be replaced by A, E would become B, and so on.
- **Caesar Cipher** is a particular case (for $k = 3$).
 - *Julius Caesar*, who used it in his private correspondence; he used fixed #3 for shifting.

Additive Cipher/Shift Cipher

- Each letter in the plaintext is replaced by a letter some fixed number of positions down the alphabet.
- For example, with a left shift of 3, D would be replaced by A, E would become B, and so on.
- **Caesar Cipher** is a particular case (for $k = 3$).
 - *Julius Caesar*, who used it in his private correspondence; he used fixed #3 for shifting.
- **Mathematically,**
 - * $Z_{26} = \{0, 1, 2, \dots, 24, 25\}$
 - * $P = C = K = Z_{26}$
 - * For $k \in K$,
 $E_k(x) = (x + k) \bmod 26$ for $x \in P$
 $D_k(y) = (y - k) \bmod 26$ for $y \in C$

Additive/Shift Cipher: Example

- The plaintext is ordinary English text.
- Correlation between alphabetic characters and integer:
 $A = 0, B = 1, \dots, Y = 24, Z = 25.$

Plaintext →	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
Ciphertext →	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Value →	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Additive/Shift Cipher: Working Principle

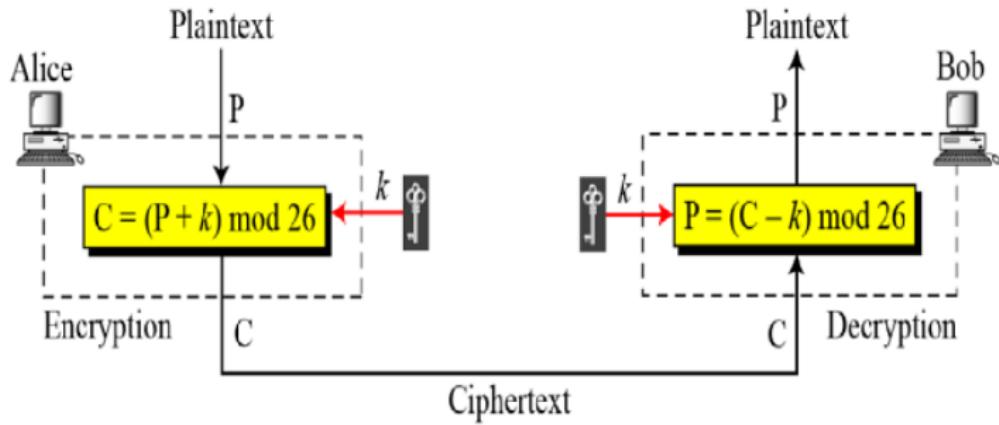


Figure 14: Additive/Shift Cipher (**Note: When the cipher is additive, the plaintext, ciphertext, and key are integers in Z_{26}**).

Additive/Shift Cipher: Encryption

- key $k = 15$
- Plaintext is “hello”
- Corresponding sequence of integers:
07 04 11 11 14
- we add 15 (key) to each value (reducing modulo 26):
22 19 00 00 03
- Convert the sequence of integers to alphabetic characters:
W T A A D

Plaintext: h → 07

Encryption: $(07 + 15) \text{ mod } 26$

Ciphertext: 22 → W

Plaintext: e → 04

Encryption: $(04 + 15) \text{ mod } 26$

Ciphertext: 19 → T

Plaintext: l → 11

Encryption: $(11 + 15) \text{ mod } 26$

Ciphertext: 00 → A

Plaintext: l → 11

Encryption: $(11 + 15) \text{ mod } 26$

Ciphertext: 00 → A

Plaintext: o → 14

Encryption: $(14 + 15) \text{ mod } 26$

Ciphertext: 03 → D

Additive/Shift Cipher: Decryption

- ciphertext : "WTAAD"
- convert the ciphertext to sequence of integers:
22 19 00 00 03
- subtract 15 from each value (reducing modulo 26):
07 04 11 11 14
- convert the sequence of integers to alphabetic characters: "hello"

Ciphertext: W → 22	Decryption: $(22 - 15) \text{ mod } 26$	Plaintext: 07 → h
Ciphertext: T → 19	Decryption: $(19 - 15) \text{ mod } 26$	Plaintext: 04 → e
Ciphertext: A → 00	Decryption: $(00 - 15) \text{ mod } 26$	Plaintext: 11 → l
Ciphertext: A → 00	Decryption: $(00 - 15) \text{ mod } 26$	Plaintext: 11 → l
Ciphertext: D → 03	Decryption: $(03 - 15) \text{ mod } 26$	Plaintext: 14 → o

Caesar Cipher

- **Caesar Cipher** is the earliest known (and the simplest).
- It involves replacing each letter of the alphabet with the letter standing three places further down. This is then wrapped around on itself when the end is reached.
- For example:

with K=3

	0	1	2	3	4	5	6	7	8	9	10	11	12
plaintext	a	b	c	d	e	f	g	h	i	j	k	l	m
ciphertext	D	E	F	G	H	I	J	K	L	M	N	O	P
	13	14	15	16	17	18	19	20	21	22	23	24	25
plaintext	n	o	p	q	r	s	t	u	v	w	x	y	z
ciphertext	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

attackatdawn → DWWDFNDWFDZQ

Additive/ Shift Cipher: Example

- Eve/Darth has intercepted the ciphertext "**UVACLYFZLJBYL**". Show how she/he can use a brute-force attack to break the cipher!!

Additive/ Shift Cipher: Example

- Eve/Darth has intercepted the ciphertext “**UVACLYFZLJBYL**”. Show how she/he can use a brute-force attack to break the cipher!!

Ciphertext: UVACLYFZLJBYL

K = 1	→	Plaintext: tuzbkxeykiaxk
K = 2	→	Plaintext: styajwdxjhzwj
K = 3	→	Plaintext: rsxzivcwigyvi
K = 4	→	Plaintext: qrwyhubvhfxuh
K = 5	→	Plaintext: pqvxgtaugewtg
K = 6	→	Plaintext: opuwfsztfdvsf
K = 7	→	Plaintext: notverysecure

Additive/Shift cipher: Secure?? (Cryptanalysis)

- Shift Cipher is not Secure
- Brute-force cryptanalysis easily performed on the shift cipher by trying all 25 possible keys.
- Given a ciphertext string, Eve/Darth successively try the decryption process with $k = 0, 1, 2, \text{ etc.}$ until get a meaningful text.
- Additive ciphers are also subject to **statistical attacks**.

Additive/Shift ciphers: Prone to Statistical Attacks!!

Letter	Frequency	Letter	Frequency	Letter	Frequency	Letter	Frequency
E	12.7	H	6.1	W	2.3	K	0.08
T	9.1	R	6.0	F	2.2	J	0.02
A	8.2	D	4.3	G	2.0	Q	0.01
O	7.5	L	4.0	Y	2.0	X	0.01
I	7.0	C	2.8	P	1.9	Z	0.01
N	6.7	U	2.8	B	1.5		
S	6.3	M	2.4	V	1.0		

Figure 19: Frequency of occurrence of letters in an English text of 100 characters.

Digram	TH, HE, IN, ER, AN, RE, ED, ON, ES, ST, EN, AT, TO, NT, HA, ND, OU, EA, NG, AS, OR, TI, IS, ET, IT, AR, TE, SE, HI, OF
Trigram	THE, ING, AND, HER, ERE, ENT, THA, NTH, WAS, ETH, FOR, DTH

Grouping of digrams and trigrams based on their frequency in English.

Multiplicative Ciphers

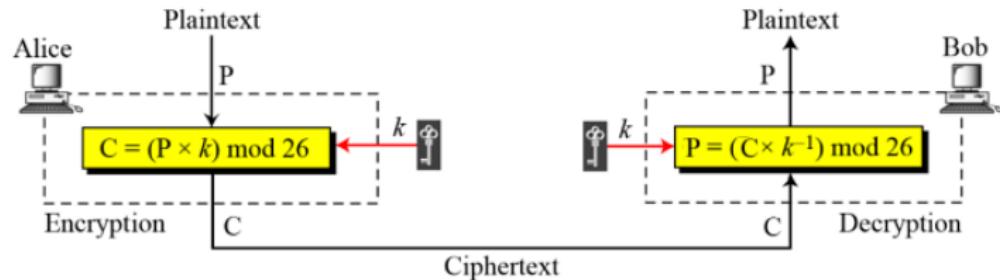


Figure 21: Multiplicative Cipher

- In a multiplicative cipher, the plaintext and ciphertext are integers in Z_{26} ; the key is an integer in Z_{26}^* .

Multiplicative Cipher: Example

- What is the key domain for any multiplicative cipher, if keys are from English alphabet set?

Multiplicative Cipher: Example

- What is the key domain for any multiplicative cipher, if keys are from English alphabet set?
- **The key needs to be in Z_{26}^* . This set has only 12 members: 1, 3, 5, 7, 9, 11, 15, 17, 19, 21, 23, 25.**

Multiplicative Cipher: Example

- What is the key domain for any multiplicative cipher, if keys are from English alphabet set?
- **The key needs to be in Z_{26}^* . This set has only 12 members: 1, 3, 5, 7, 9, 11, 15, 17, 19, 21, 23, 25.**

Plaintext: h → 07	Encryption: $(07 \times 07) \bmod 26$	ciphertext: 23 → X
Plaintext: e → 04	Encryption: $(04 \times 07) \bmod 26$	ciphertext: 02 → C
Plaintext: l → 11	Encryption: $(11 \times 07) \bmod 26$	ciphertext: 25 → Z
Plaintext: l → 11	Encryption: $(11 \times 07) \bmod 26$	ciphertext: 25 → Z
Plaintext: o → 14	Encryption: $(14 \times 07) \bmod 26$	ciphertext: 20 → U

Figure 22: Use of multiplicative cipher to encrypt the message "hello" with a key of 7.

Affine Ciphers

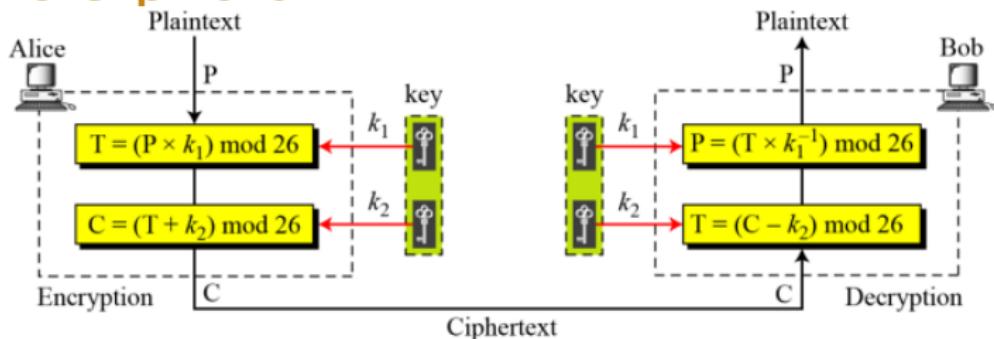


Figure 23: Affine Cipher: A combination of additive and multiplicative cipher with a pair of keys.

$$C = (P \times k_1 + k_2) \bmod 26$$

$$P = ((C - k_2) \times k_1^{-1}) \bmod 26$$

where k_1^{-1} is the multiplicative inverse of k_1 and $-k_2$ is the additive inverse of k_2

Key Domain

- The affine cipher uses a pair of keys in which the first key is from Z_{26}^* and the second is from Z_{26} .
- What is the size of key domain for Additive/Shift cipher?

Key Domain

- The affine cipher uses a pair of keys in which the first key is from Z_{26}^* and the second is from Z_{26} .
- What is the size of key domain for Additive/Shift cipher?
- What is the size of key domain for Multiplicative cipher?

Key Domain

- The affine cipher uses a pair of keys in which the first key is from Z_{26}^* and the second is from Z_{26} .
- What is the size of key domain for Additive/Shift cipher?
- What is the size of key domain for Multiplicative cipher?
- What is the size of key domain for Affine cipher?

Key Domain

- The affine cipher uses a pair of keys in which the first key is from Z_{26}^* and the second is from Z_{26} .
- What is the size of key domain for Additive/Shift cipher?
- What is the size of key domain for Multiplicative cipher?
- What is the size of key domain for Affine cipher?
(i) 25, (ii) 11, (iii) $26 \times 12 - 1 = 312 - 1 = 311$

Key Domain

- The affine cipher uses a pair of keys in which the first key is from Z_{26}^* and the second is from Z_{26} .
- What is the size of key domain for Additive/Shift cipher?
- What is the size of key domain for Multiplicative cipher?
- What is the size of key domain for Affine cipher?
 - (i) 25, (ii) 11, (iii) $26 \times 12 - 1 = 312 - 1 = 311$
- **The additive cipher is a special case of an affine cipher in which $k_1 = 1$. The multiplicative cipher is a special case of affine cipher in which $k_2 = 0$.**

Affine Cipher: Example

P: h → 07	Encryption: $(07 \times 7 + 2) \bmod 26$	C: 25 → Z
P: e → 04	Encryption: $(04 \times 7 + 2) \bmod 26$	C: 04 → E
P: l → 11	Encryption: $(11 \times 7 + 2) \bmod 26$	C: 01 → B
P: l → 11	Encryption: $(11 \times 7 + 2) \bmod 26$	C: 01 → B
P: o → 14	Encryption: $(14 \times 7 + 2) \bmod 26$	C: 22 → W

Figure 24: Use of an affine cipher to encrypt the message “hello” with the key pair (7,2).

C: Z → 25	Decryption: $((25 - 2) \times 7^{-1}) \bmod 26$	P: 07 → h
C: E → 04	Decryption: $((04 - 2) \times 7^{-1}) \bmod 26$	P: 04 → e
C: B → 01	Decryption: $((01 - 2) \times 7^{-1}) \bmod 26$	P: 11 → l
C: B → 01	Decryption: $((01 - 2) \times 7^{-1}) \bmod 26$	P: 11 → l
C: W → 22	Decryption: $((22 - 2) \times 7^{-1}) \bmod 26$	P: 14 → o

Figure 25: Use of an affine cipher to decrypt the message “ZEBBW” with the key pair (7,2) in modulus 26.

Monoalphabetic Substitution Cipher

- Additive, multiplicative, and affine ciphers have small key domains; therefore, they are very vulnerable to brute-force attack.
- **Monoalphabetic Substitution Cipher:** A better solution is to create a mapping between each plaintext character and the corresponding ciphertext character.
 - Alice and Bob can agree on a table showing the mapping for each character.

Plaintext →	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
Ciphertext →	N	O	A	T	R	B	E	C	F	U	X	D	Q	G	Y	L	K	H	V	I	J	M	P	Z	S	W

Figure 26: An example key for monoalphabetic substitution cipher.

Monoalphabetic Substitution Cipher: Examples

- We can use the key in Figure to encrypt the message

this message is easy to encrypt but hard to find the key

- The ciphertext is

ICFVQRVVNEFVRNVSIYRGAHSLIOJICNHTIYBFGTICRXRS

Monoalphabetic Substitution Cipher: Secure?? (Cryptanalysis)

- Each alphabetic character is mapped to a unique alphabetic character. **One-to-One**
- We use arbitrary monoalphabetic substitution, so the key space for monoalphabetic subsitution cipher is $26!$, or almost $4 \times 10^{26} \approx 2^{88}$ possible permutations, which is a very large number. Thus, **brute-force** seems infeasible.

Monoalphabetic Substitution Cipher: Secure?? (Cryptanalysis)

- Each alphabetic character is mapped to a unique alphabetic character. **One-to-One**
- We use arbitrary monoalphabetic substitution, so the key space for monoalphabetic subsitution cipher is $26!$, or almost $4 \times 10^{26} \approx 2^{88}$ possible permutations, which is a very large number. Thus, **brute-force** seems infeasible.
- **However, a Monoalphabetic Substitution Cipher is insecure against frequency analysis.**

Polyalphabetic Cipher

- In polyalphabetic substitution, each occurrence of a character may have a different substitute.
- The relationship between a character in the plaintext to a character in the ciphertext is **one-to-many**.

Autokey Cipher

- “autokey” implies that the subkeys are automatically created from the plaintext cipher characters during the encryption process.

$$P = P_1 P_2 P_3 \dots$$

$$C = C_1 C_2 C_3 \dots$$

$$k = (k_1, P_1, P_2, \dots)$$

$$\text{Encryption: } C_i = (P_i + k_i) \bmod 26$$

$$\text{Decryption: } P_i = (C_i - k_i) \bmod 26$$

Autokey Cipher

- “autokey” implies that the subkeys are automatically created from the plaintext cipher characters during the encryption process.

$$P = P_1 P_2 P_3 \dots$$

$$C = C_1 C_2 C_3 \dots$$

$$k = (k_1, P_1, P_2, \dots)$$

Encryption: $C_i = (P_i + k_i) \bmod 26$

Decryption: $P_i = (C_i - k_i) \bmod 26$

Plaintext:	a	t	t	a	c	k	i	s	t	o	d	a	y
P's Values:	00	19	19	00	02	10	08	18	19	14	03	00	24
Key stream:	12	00	19	19	00	02	10	08	18	19	14	03	00
C's Values:	12	19	12	19	02	12	18	00	11	7	17	03	24
Ciphertext:	M	T	M	T	C	M	S	A	L	H	R	D	Y

Figure 27: Assume that Alice and Bob agreed to use an autokey cipher with initial key value $k_1 = 12$. Now Alice wants to send Bob the message “Attack is today”. Enciphering is done character by character.

Autokey Cipher: Secure?? (Cryptanalysis)

- The autokey cipher hides the single-layer frequency statistics of the plaintext. But..
- Vulnerable to brute-force attack as the additive cipher.

Autokey Cipher: Secure?? (Cryptanalysis)

- The autokey cipher hides the single-layer frequency statistics of the plaintext. But..
- Vulnerable to brute-force attack as the additive cipher.
- The first sub-key can be from one of the 25 values.
- We need polyalphabetic ciphers that not only hide the characteristics of the language but also have large key domains.

Playfair Cipher

- Variant of polyalphabetic cipher, used by the British army during World War I.
- The secret key in this cipher is made of 25 alphabet letters arranged in a 5x5 matrix (letter I and J are considered the same when encrypting).

c	h	a	r	l
e	s	b	d	f
g	i/j	k	m	n
o	p	q	t	u
v	w	x	y	z

Figure 28: An example of a secret key in the Playfair cipher.

Playfair Cipher: Rules..

- Plaintext: "meet me at the bridge"
 - Split the sentence into digrams removing spaces, 'x' used to make even number of letters:
me et me at th eb ri dg ex

Playfair Cipher: Rules..

- Plaintext: "meet me at the bridge"
 - Split the sentence into digrams removing spaces, 'x' used to make even number of letters:
me et me at th eb ri dg ex
 - Repeating plaintext letters that are in the same pair are separated with a filler letter, such as x:
"balloon" would be treated as **ba lx lo on**

Playfair Cipher: Rules..

- Plaintext: "meet me at the bridge"
 - Split the sentence into digrams removing spaces, 'x' used to make even number of letters:
me et me at th eb ri dg ex
 - Repeating plaintext letters that are in the same pair are separated with a filler letter, such as x:
"balloon" would be treated as **ba lx lo on**
 - Two plaintext letters in the same row are each replaced by the letter to the right, with the first element of the row circularly following the last.
eb is replaced by sd
ng is replaced by gi (or gj as preferred)

Playfair Cipher: Rules..cont

- Two plaintext letters that fall in the same column are each replaced by the letter beneath, with the top element of the column circularly following the last.
dt would be replaced by my
ty would be replaced by yr
 - Otherwise, each plaintext letter in a pair is replaced by the letter that lies in its own row and the column occupied by the other plaintext letter.
me becomes gd
- Ciphertext therefore is:
“gd do gd rq pr sd hm em bv”

Playfair Cipher: Secure?? (Cryptanalysis)

- Brute force attack is difficult as the size of the key domain is 25!
- Single letter frequency is obscured.
- But digrams are preserved.
- A cryptanalyst can use a ciphertext-only attack based on the digram frequency test to find the key.

Vigenere Cipher

- Simplest polyalphabetic substitution cipher, designed by Blaise de Vigenere, a sixteenth-century french mathematician.
- Consider the set of all Caesar ciphers:
 $\{C_a, C_b, C_c, \dots, C_z\}$
- Key: e.g. **security**
- Encrypt each letter using $C_s, C_e, C_c, C_u, C_r, C_i, C_t, C_y$ in turn.
- Repeat from start after C_y .
- Decryption simply works in reverse.

Vigenere Cipher: Mathematical Representation

- Let m be a positive integer
- $P = C = K = (\mathbb{Z}_{26})^m$
- For $k = (k_1, k_2, \dots, k_m) \in K$,
 1. $e_k(x_1, x_2, \dots, x_m) = (x_1 + k_1, x_2 + k_2, \dots, x_m + k_m)$
 2. $d_k(y_1, y_2, \dots, y_m) = (y_1 - k_1, y_2 - k_2, \dots, y_m - k_m)$
- All above operations are performed in \mathbb{Z}_{26}

Vigenere Cipher: Example

- Correspondence between alphabetic characters and integer:
 $A = 0, B = 1, \dots, Y = 24, Z = 25.$
- $m = 6.$
- Keyword is “CIPHER”, this corresponds to the numerical equivalent
 $k = (2, 8, 15, 7, 4, 17)$

Vigenere Cipher: Example..contd...

- Plaintext : “thiscryptosystemisnotsecure”.
- Encryption: add modulo 26

19	7	8	18	2	17	24	15	19	14	18	24	18	19	4	12
2	8	15	7	4	17	2	8	15	7	4	17	2	8	15	7
21	15	23	25	6	8	0	23	8	21	22	15	20	1	19	19

8	18	13	14	19	18	4	2	20	17	4
4	17	2	8	15	7	4	17	2	8	15
12	9	15	22	8	25	8	19	22	25	19

- Ciphertext:
“VPXZGIAIXWPUBTTMJPWIZITWZT”.

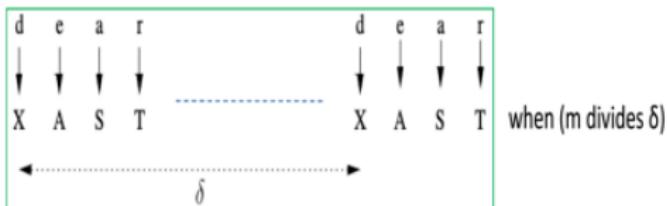
Vigenere Cipher: Secure? (Cryptanalysis)

- Frequency analysis more difficult (but not impossible)
- Attack has two steps
 1. Determine the length m of the key
 2. Determine $K = (k_1, k_2, \dots, k_m)$ by finding each k_i separately.

Determining key length: Kaisiki Test_(content)

courtesy: Chester Rebeiro

- Kasiski test by Friedrich Kasiski in 1863
- Let m be the size of the key
- **observation:** two identical plaintext segments will encrypt to the same ciphertext when they are δ apart and ($m|\delta$)



- If several such δ s are found (i.e. $\delta_1, \delta_2, \delta_3, \dots$) then
 - $m|\delta_1, m|\delta_2, m|\delta_3, \dots$
 - Thus m divides the gcd of $(\delta_1, \delta_2, \delta_3, \dots)$

Increasing Confidence of Key Length (Index of Coincidence)

- Consider a multi set of letters of size N
say $s = \{a, b, c, d, a, a, e, f, e, g, \dots\}$
- Probability of picking two 'a' characters (without replacement) is

$$\frac{n_0}{N} \times \frac{n_0 - 1}{N - 1}$$

n_0 : Number of occurrences of
'a' in S

probability the first pick is 'a'

probability the second pick is 'a'

Index of Coincidence..contd..1

- Sum of probabilities of picking two similar characters is

$$I_c = \sum_{i=0}^{25} \frac{n_i(n_i - 1)}{N(N - 1)}$$

index of coincidence

Index of coincidence.. cont..2

- Consider a random permutation of the alphabets (as in the substitution cipher)
 $s = \{a, b, c, d, a, a, e, f, e, g, \dots\} \longrightarrow S = \{X, M, D, F, X, X, Z, G, Z, J, \dots\}$
- Note that : $n_a = n_x$; thus the value of I_c remains unaltered
- Number of occurrence of an alphabet in a text depends on the language, thus each language will have a unique I_c value

English	0.0667	French	0.0778
German	0.0762	Spanish	0.0770
Italian	0.0738	Russian	0.0529

Vigenere Cipher: Cryptanalysis Example

Let us assume we have intercepted the following ciphertext:

LIOMWGEGGDVWGHHCQUCRHRWAGWIOWQLKGZETKKMEVLWPCZVGTH-
VTSGXQOVGCSVETQLTJSUMVVVEUVLXEWSLGFMVVWLGYHCUSWXQH-
KVGSHEEVFLCFDGVSUMPHKIRZDMPHHBVWWJWIXGFWLTSHGJOUEEHH-
VUCFVGOWICQLTJSUXGLW

Kasiski test for repetition of three character segments yields the results as shown in Table 3.4.

Table 3.4 Kasiski test for Example 3.19

String	First Index	Second Index	Difference
QLT	65	165	100
LTJ	66	166	100
TJS	67	167	100
JSU	68	168	100
SUM	69	117	48
VWV	72	132	60

Vigenere Cipher: Cryptanalysis Example

The greatest common divisor is thus 4, thus suggesting that the key length is a multiple of 4. We confirm this guess by the Index of Coincidence test.

We divide the ciphertext into 4 rows as shown below. We also mention the corresponding Index of Coincidence values. The high values of the IC confirms the key length reported in the Kasiski test.

1st string :

IC = 0.067677

LWGWCRAOKTEPGTQCTJVUEGVGUQGECVPRPVJGTJEUGCJG

2nd string :

IC = 0.074747

IGGGQHGWGVCTSOSQS WVWFVYSHSVFSHZHWWFSOHCQSL

3rd string:

IC = 0.070707

OFDHURWQZKLZHGVVLUVLSZWHWKHF DUKDHVIWHUHF WL UW

4th string:

IC = 0.076768

MEVHCWILEMWVVXGETMEXMLCXVELGMIMBWXLGEVVITX

Vigenere Cipher: Cryptanalysis Example

Then we perform the Mutual Index of Coincidence to obtain the actual key value. Running the test, we obtain that the key value is CODE, and the corresponding plaintext is

JULIUSCAESARUSEDACRYPTOSYSTEMINHISWARWHICHISNOWREFERR
EDTOASCAESARCIPHERITISASHIFTCIPHERWITHTHEKEYSETTOTHREEE
ACHCHARACTERINTHEPLAINTEXTISHIFTERTHREECHARACTERSOCRE
ATEACIPHERTEXT

Note that the plaintext makes sense and hence we believe the decryption is correct. We format the obtained as follows:

Julius Caesar used a cryptosystem in his wars, which is now referred to as Caesar cipher. It is an additive cipher with the key set to three. Each character in the plaintext is shifted three characters to create the ciphertext.

Hill Cipher

(Content courtesy: Chester Rebeiro)

- Encryption: $y = xK \pmod{26}$
- Decryption: $x = yK^{-1} \pmod{26}$
 - plaintext : $x \in \{0, 1, 2, 3, \dots, 25\}$
 - ciphertext : $y \in \{0, 1, 2, 3, \dots, 25\}$
 - key : K is an invertible matrix

Hill Cipher..

(Content courtesy: Chester Rebeiro)

- example

plaintext

h i f f

(7,8)(11,11)

$$K = \begin{bmatrix} 11 & 8 \\ 3 & 7 \end{bmatrix} \quad K^{-1} = \begin{bmatrix} 7 & 18 \\ 23 & 11 \end{bmatrix} \quad K \bullet K^{-1} = 1 \pmod{26}$$

$$\boxed{[7 \ 8] \times \begin{bmatrix} 11 & 8 \\ 3 & 7 \end{bmatrix} \pmod{26} = [23 \ 8]} \quad \text{encryption}$$

$$\boxed{[23 \ 8] \times \begin{bmatrix} 7 & 18 \\ 23 & 11 \end{bmatrix} \pmod{26} = [7 \ 8]} \quad \text{decryption}$$

h i f f \rightarrow (7,8)(11,11) \longrightarrow (23,8)(24,9) \rightarrow X Y J

plaintext ciphertext

Cryptanalysis of Hill Cipher (content courtesy: Chester R. and Debdeep M.)

- ciphertext only attack is difficult
- known plaintext attack

$$\begin{array}{l} (7,8)(11,11) \\ \text{known plaintext} \end{array} \times \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix} \longrightarrow \begin{array}{l} (23,8)(24,9) \\ \text{corresponding ciphertext} \end{array}$$

Form equations and solve to get the key

$$7k_{11} + 8k_{21} = 23$$

$$7k_{12} + 8k_{22} = 8$$

$$11k_{11} + 11k_{21} = 24$$

$$11k_{12} + 11k_{22} = 9$$

One-Time Pad

- One of the goals of cryptography is perfect secrecy.
- A study by **Shannon** has shown that perfect secrecy can be achieved if each plaintext symbol is encrypted with a key randomly chosen from a key domain.
- This idea is used in a cipher called **one-time pad**, invented by **Vernam**.
- The key has the same length as the plaintext and is chosen in random.
- The key is changed each time the sender sends a new message.

One-Time Pad.. contd..

- For example, an additive cipher can be easily broken because the same key is used to encrypt every character.
- However, even this simple cipher can become a perfect cipher if the key that is used to encrypt each character is chosen randomly from the key domain (00, 01, 02, ..., 25) -i.e if the first character is encrypted using the key 04, the second character is encrypted using the key 02, the third character is encrypted using the key 21; and so on.
- **ciphertext-only attack is impossible.**
- Other types of attacks are also impossible if the sender changes the key each time she/he sends a message, using another random sequence of integers.

One-Time Pad: Feasibility

- A one-time pad is a perfect cipher, but it is almost impossible to implement commercially.
- If the key must be newly generated each time, how can Alice tell Bob the new key each time she has a message to send?
- **There are some occasions when a one-time pad can be used. For example, if the president of a country needs to send a completely secret message to the president of another country, she/he can send a trusted envoy with a random key before sending the message.**

Permutation Cipher

- Ciphers we seen so far were substitution ciphers
 - Plaintext characters substituted with ciphertext characters

his
plaintext

→

X Y J
ciphertext

- Alternate technique: Permutation
 - Plaintext characters re-ordred by a random permutation

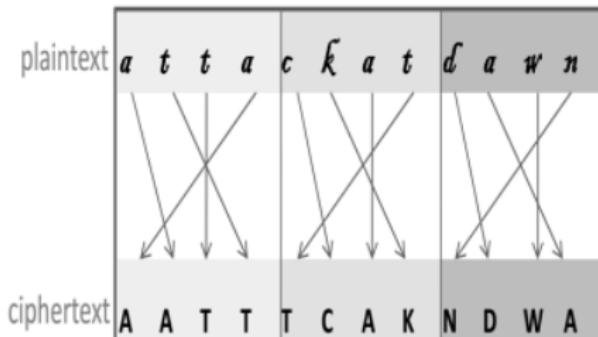
his
plaintext

→

L I H I
ciphertext

Permutation Cipher

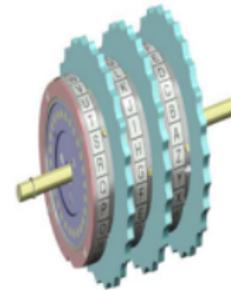
- Example plaintext: **attackatdawn**
 - key: (1,3,2,0) here is of length 4 and a permutation of (0,1,2,3)
 - It refers 0th character in plaintext goes to 1st character in ciphertext (and so on..)



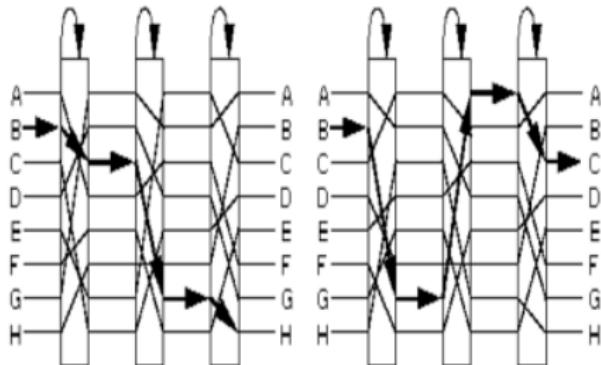
- cryptanalysis : $4!$ possibilities

Rotor Cipher Machines (German Enigma)

- Before modern ciphers, rotor machines were most common complex ciphers in use.
- Widely used in WW2.
- Used a series of rotating cylinders.
- Implemented a polyalphabetic substitution cipher of period K.



Rotor Cipher Machines (German Enigma)



- Each rotor makes a permutation
 - Adding / removing a rotor would change the ciphertext
- Additionally, the rotors rotate with a gear after a character is entered
- Broken by Alan Turing

Transposition Ciphers

- A transposition cipher reorders symbols.
- Classification:
 1. Keyless Transposition Ciphers
 2. Keyed Transposition Ciphers
 3. Combining Two Approaches

Keyless Transposition Ciphers

- Reorders the symbols.
- Simple transposition ciphers, which were used in the past, are keyless.
- Example:
 - A good example of a keyless cipher using the first method is the **rail fence cipher**. The ciphertext is created reading the pattern row by row. For example, to send the message **"Meet me at the park"** to Bob, Alice writes:

The diagram shows the letters of the message "Meet me at the park" arranged in two rows. The first row contains 'M', 'e', 't', 'e', 't', 'h', 'e', 'p', 'r', 'k'. The second row contains 'e', 'e', 'a', 't', 'h', 'e', 'a', 'r'. Arrows indicate the reading order: starting from the top left, moving down to the bottom, then back up to the top right, and so on, forming a zigzag pattern across the two rows.

- She then creates the ciphertext **"MEMATEAKETETHPR"**.

Keyless Transposition Cipher: Example 2

- Alice and Bob can agree on the number of columns and use the second method.
Alice writes the same plaintext, row by row, in a table of four columns.

1	2	3	4	5	6
M	E	E	T	M	E
A	F	T	E	R	P
A	R	T	Y		

4	2	1	6	3	5
T	E	M	E	E	M
E	F	A	P	T	R
Y	R	A		T	

Plain Text: MEET ME AFTER PARTY

Key Used: 421635

Cipher Text: TEMEEMEFAPTRYRAT

Keyless Transposition Cipher: Example 3

Alice and Bob can agree on the number of columns and use the second method. Alice writes the same plaintext, row by row, in a table of four columns.

m	e	e	t
m	e	a	t
t	h	e	p
a	r	k	

She then creates the ciphertext “MMTAEEHREAEKTP”.

Keyless Transposition Cipher: Example 3: Security? (Cryptanalysis)

The cipher in Example is actually a transposition cipher. The following shows the permutation of each character in the plaintext into the ciphertext based on the positions.

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
01	05	09	13	02	06	10	13	03	07	11	15	04	08	12

The second character in the plaintext has moved to the fifth position in the ciphertext; the third character has moved to the ninth position; and so on. Although the characters are permuted, there is a pattern in the permutation: (01, 05, 09, 13), (02, 06, 10, 13), (03, 07, 11, 15), and (08, 12). In each section, the difference between the two adjacent numbers is 4.

Keyed Transposition Ciphers

- The keyless ciphers permute the characters by using writing plaintext in one way and reading it in another way.
- The permutation is done on the whole plaintext to create the whole ciphertext.
- Another method is **to divide the plaintext into groups of predetermined size, called blocks, and then use a key to permute the characters in each block separately.**

Keyed Transposition Cipher: Example

- Alice needs to send the message "**Enemy attacks tonight**" to Bob..

e n e m y a t t a c k s k s t o n i g h t z

- The key used for encryption and decryption is a permutation key, which shows how the characters are permuted.

Encryption ↓

3	1	4	5	2
1	2	3	4	5

↑ Decryption

Keyed Transposition Cipher: Example

- Alice needs to send the message “**Enemy attacks tonight**” to Bob..

e n e m y a t t a c k s k s t o n i g h t z

- The key used for encryption and decryption is a permutation key, which shows how the characters are permuted.

Encryption ↓

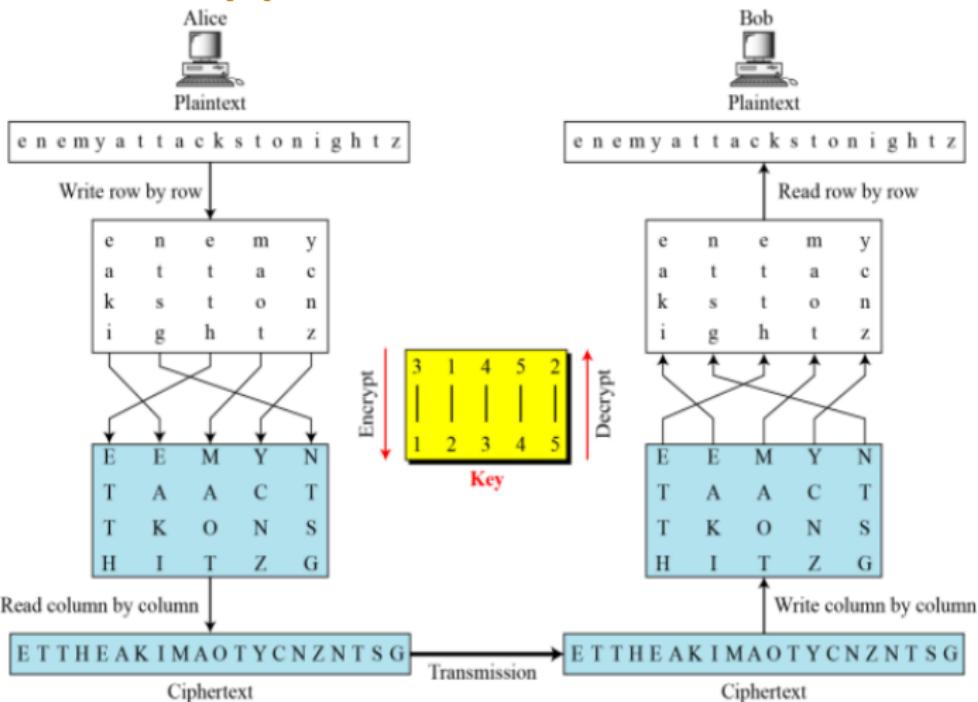
3	1	4	5	2
1	2	3	4	5

↑ Decryption

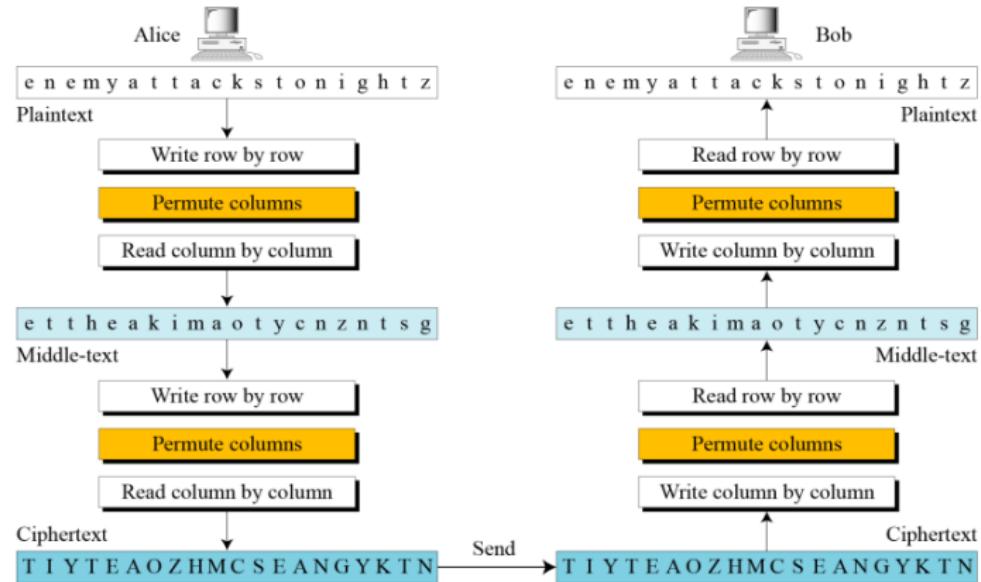
- The permutation yields

E E M Y N T A A C T T K O N S H I T Z G

Combined Approaches



Double Transposition Cipher



Bibliography: Books and Resources

- Cryptography and Network Security: Principles and Practice by William Stallings
- Cryptography and Network Security by Behrouz A Forouzan and Debdeep Mukhopadhyay
- Principles of Information Security by Michael E. Whitman and Herbert J. Mattord.
- Cisco platform, and Internet.
- Published research papers, study materials from researchers of security domain.

CO-INS:Information and Network Security

UNIT-II (Part-I)

Course Instructors:

Soma Saha

Veerendra Srivastava

Soma Saha (PhD)

Department of Computer Engineering
SGSITS Indore, India

March 8, 2021

UNIT-II (Part-I): Learning Objectives

Upon completion of this unit, you should be able to

- LO1 Define the terms and concepts of symmetric-key ciphers
- LO2 Discuss the two broad categories of traditional symmetric-key ciphers with focus on different cipher cryptanalysis
- LO3 Show the idea behind stream ciphers and block ciphers

Cryptography: Symmetric-Key Cipher



Figure 1: Symmetric-Key Encipherment.

- $c = E_s(p, k)$
- $p = D_s(c, k)$
- D_s = Decryption function (symmetric)
- c = ciphertext
- p = plaintext
- k = secret key
- E_s = Encryption function (symmetric)

Symmetric-Key Encipherment: Message exchange Prove

- We can prove that the plaintext created by Bob is the same as the one originated by Alice. We assume that Bob creates p_1 ; we prove that $p_1 = p$:

Symmetric-Key Encipherment: Message exchange Prove

- We can prove that the plaintext created by Bob is the same as the one originated by Alice. We assume that Bob creates p_1 ; we prove that $p_1 = p$:
- Alice : $c = E_s(p, k)$
- Bob: $p_1 = D_s(c, k) = D_s(E_s(p, k), k) = p$

Symmetric-Key Encipherment: Message exchange Prove

- We can prove that the plaintext created by Bob is the same as the one originated by Alice. We assume that Bob creates p_1 ; we prove that $p_1 = p$:
 - Alice : $c = E_s(p, k)$
 - Bob: $p_1 = D_s(c, k) = D_s(E_s(p, k), k) = p$
- **Kerckhoff's Principle:** A cryptosystem should be secure even if everything about the system, except the key, is public knowledge.
- Kerckhoff's principle was reformulated (or possibly independently formulated) by American mathematician **Claude Shannon** as "the enemy knows the system", i.e., "one ought to design systems under the assumption that the enemy will immediately gain full familiarity with them". In that form, it is called **Shannon's maxim**.(Source: wikipedia)

Symmetric-Key Encipherment: How to compute #keys?

- If there are m people in a group who need to communicate with each other, how many keys are needed?

Symmetric-Key Encipherment: How to compute #keys?

- If there are m people in a group who need to communicate with each other, how many keys are needed?

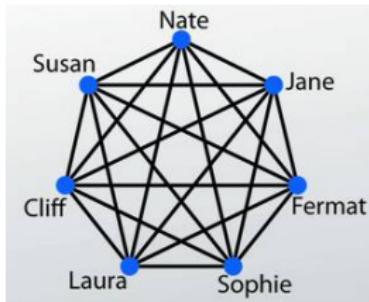


Figure 2: Example scenario.

Cryptography-Cryptanalysis-Cryptology

- Cryptography - Science and art of creating secret codes.
- Cryptanalysis - Science and art of breaking those codes.
- The study of cryptanalysis helps us create better secret codes.

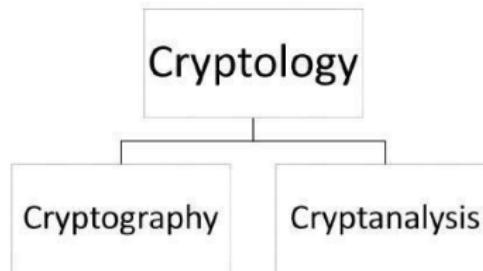


Figure 3: Cryptography-Cryptanalysis-Cryptology.

Cryptanalysis Attacks

- Classification based on encryption techniques:
 1. Ciphertext-only attack
 2. Known-plaintext attack
 3. Chosen-Plaintext attack
 4. Chosen-Ciphertext attack
 5. Chosen-text attack

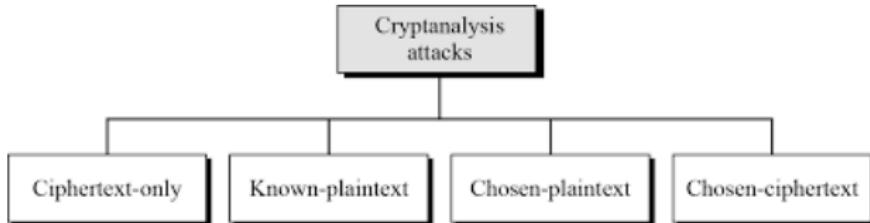


Figure 4: Classification of cryptanalysis attacks based on encryption techniques.

Ciphertext-Only Attack

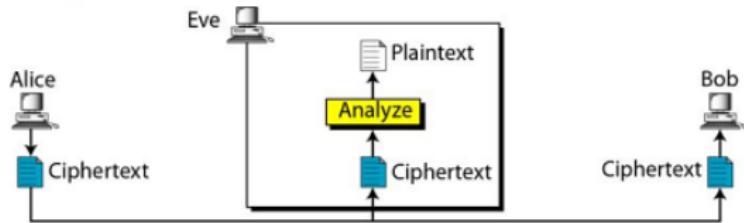


Figure 5: Ciphertext-only attack.

- Adversary, Eve/Darth has access to only some ciphertext.
- **Assumption:** Eve/Darth knows the algorithm and can intercept the ciphertext.

Ciphertext-Only Attack: various methods

- **Brute-Force Attack/Exhaustive-key-search Attack:** Eve/Darth tries to use all possible keys.
- **Assumption:**
 - Eve/Darth knows the algorithm.
 - Eve/Darth knows the key domain (the list of all possible keys).
- **Application:** Using the intercepted cipher, Eve/Darth decrypts the ciphertext with every possible key until the plaintext makes sense.
- **Prevention:** The number of possible keys must be very large.

Ciphertext-Only Attack: various methods..cont..1

- **Statistical Attack:** The cryptanalyst can benefit from some inherent characteristics of the plaintext language to launch a **statistical attack**.
- **Example:**
 - The letter E is the most frequently used letter in English text.
 - The cryptanalyst finds the mostly-used character in the ciphertext and assumes that the corresponding plaintext character is E.
 - After finding a few pairs, the analyst can find the key and use it to decrypt the message.
- **Prevention:** The cipher should hide the characteristics of the language.

Ciphertext-Only Attack: various methods..cont..2

- **Pattern Attack:** Some ciphers may hide characteristics of the language, but may create some patterns in the ciphertext. A cryptanalyst may use a **pattern attack** to break the cipher.
- **Prevention:** Important to use ciphers that make the ciphertext as random as possible.

Known-Plaintext Attack

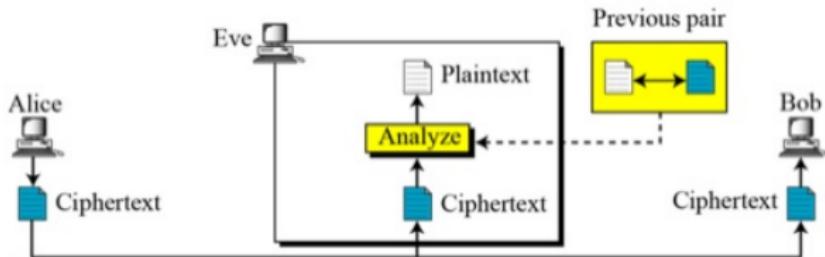


Figure 6: Known-Plaintext Attack.

- **Known-Plaintext Attack:** Eve/Darth has access to some plaintext/ciphertext pairs in addition to the intercepted ciphertext that she/he wants to break.
 - The plaintext/ciphertext pairs have been collected earlier.
 - This type of attacks are less likely to happen, because...

Chosen-Plaintext Attack

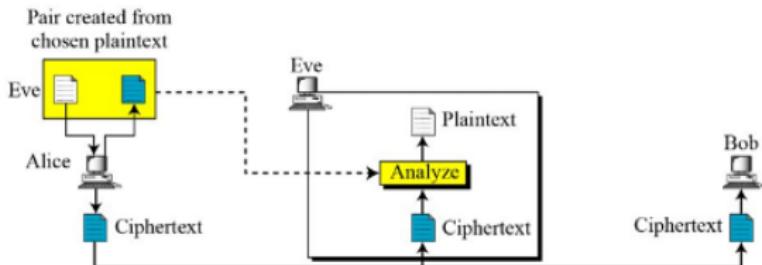


Figure 7: Chosen-Plaintext Attack.

- **Chosen-Plaintext Attack:** This attack is similar to the known-plaintext attack, but the plaintext/ciphertext pairs have been chosen by the attacker herself.
 - Example: If Eve/Darth has access to Alice's computer, she/he can choose some plaintext and intercept the captured ciphertext.
 - She/he does not have the key because the key is normally **embedded in the software** used by the sender.
 - Easy to implement, but less likely to happen, because..

Chosen-Ciphertext Attack

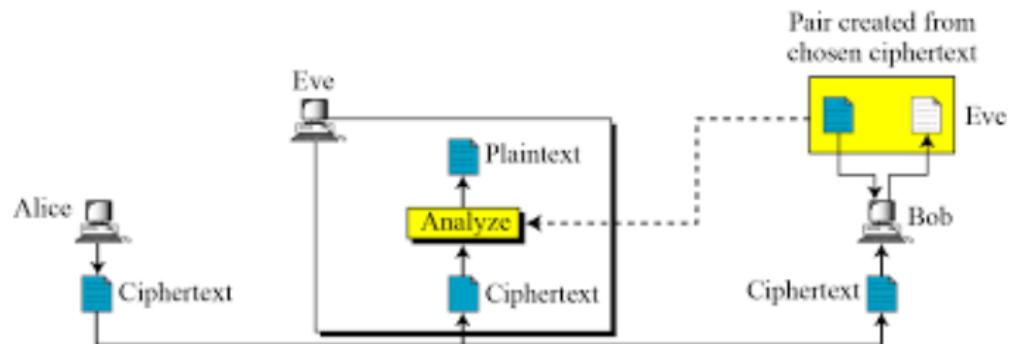


Figure 8: Chosen-Ciphertext Attack.

- **Chosen-ciphertext attack** is similar to the chosen-plaintext attack, except that Eve/Darth chooses some ciphertext and decrypts it to form a ciphertext/plaintext pair.
- This can happen if Eve/Darth has access to Bob's computer.

Hypothetical bad symmetric encryption algorithm: XOR

(Content courtesy: Tyler Bletsch, Duke Univ.)

- A lot of encryption algorithms rely on properties of XOR
 - Can think of $A \wedge B$ as "Flip a bit in A if corresponding bit in B is 1"
 - If you XOR by same thing twice, you get the data back
 - XORing by a random bit string yields NO info about original data
 - Each bit has a 50% chance of having been flipped
- Could consider XOR itself to be a symmetric encryption algorithm (but it seems dreadful at it!) - can be illustrative to explore
- Simple XOR encryption algorithm:
 - $E(p, k) = p \wedge k$ (keep repeating k as often as needed to cover p)
 - $D(c, k) = c \wedge k$ (same algorithm both ways!)

A	B	$A \wedge B$
0	0	0
0	1	1
1	0	1
1	1	0

```
>>> a=501  
>>> b=199  
>>> a ^= b  
>>> print a  
306  
>>> a ^= b  
>>> print a  
501
```

XOR “encryption” demo

(Content courtesy: Tyler Bletsch, Duke Univ.)

Plaintext: 'Hello'

Key : 'key'

H e l l o
Plaintext : 01001000 01100101 01101100 01101100 01101111
k e y Key repeats> k e

Key : 01101011 01100101 01111001 01101011 01100101

Ciphertext: 
^ XOR result

Ciphertext: 00100011 00000000 00010101 00000111 00001010

Key : 01101011 01100101 01111001 01101011 01100101

Decrypted : 
^ XOR result

Attacking XOR

(Content courtesy: Tyler Bletsch, Duke Univ.)

Figure 9: Known-Plaintext Attack:

```
Given plaintext : 01001000 01100101 01101100 01101100 01101111  
Given ciphertext : 00100011 00000000 00010101 00000111 00001010  
XOR result      : 01101011 01100101 01111001 01101011 01100101  
                  ^^ it's the key!!!
```

Figure 10: Chosen-Plaintext Attack:

```
Chosen plaintext : 00000000 00000000 00000000 00000000 00000000  
Given ciphertext : 01101011 01100101 01111001 01101011 01100101  
XOR result      : 01101011 01100101 01111001 01101011 01100101  
                  ^^ it's the key!!!
```

Attacking XOR..contd..

(Content courtesy: Tyler Bletsch, Duke Univ.)

Figure 11: Ciphertext-Only Attack:

Ciphertext: 00100011 00000000 00010101 00000111 00001010

- "I assume the plaintext had ASCII text with lowercase letters, and in all such letters bit 6 is 1, but none of the ciphertext has bit 6 set, so I bet the key is most/all lower case letters"
- "The second byte is all zeroes, which means the second byte of the key and plaintext are equal"
- etc...

**** Conclusion: XOR is a dreadful encryption algorithm**

Categories of Traditional Ciphers

- * Traditional Symmetric-Key Ciphers

- **Substitution Ciphers:** We replace one symbol in the ciphertext with another symbol.
- **Transposition Ciphers:** We reorder the position of symbols in the plaintext.

KEY	
Plaintext	ABCDEFGHIJKLMNOPQRSTUVWXYZ
Ciphertext	PQOWIEURYTLAKSJDHFGMZNXBCV
ENCRYPTION	
Plaintext	T H E M O N E Y I S I N T H E B A G
Ciphertext	M R I K J S I C Y G Y S M R I Q P U

Figure 12: Substitution Cipher.

1	2	3	4	5	6
M	E	E	T	M	E
A	F	T	E	R	P
A	R	T	Y		
4	2	1	6	3	5
T	E	M	E	E	M
E	F	A	P	T	R
Y	R	A			T

Plain Text: MEET ME AFTER PARTY

Key Used: 421635

Cipher Text: TEMEEMEFAPTRYRAT

Figure 13: Transposition Cipher.

Substitution Ciphers: Types

- **Substitution Cipher**

1. **Mono-alphabetic** : It only uses one alphabet to substitute.

- Additive/Shift/Caeser
- Multiplicative
- Affine
- Monoalphabetic Substitution Cipher

2. **Poly-alphabetic**: It may use two or more alphabets to substitute.

- Auto-Key Cipher
- Vigenere Cipher
- Playfair cipher
- Hill Cipher
- One Time Pad
- Rotor Cipher

Additive Cipher/Shift Cipher

- Each letter in the plaintext is replaced by a letter some fixed number of positions down the alphabet.
- For example, with a left shift of 3, D would be replaced by A, E would become B, and so on.
- **Caesar Cipher** is a particular case (for $k = 3$).
 - *Julius Caesar*, who used it in his private correspondence; he used fixed #3 for shifting.

Additive Cipher/Shift Cipher

- Each letter in the plaintext is replaced by a letter some fixed number of positions down the alphabet.
- For example, with a left shift of 3, D would be replaced by A, E would become B, and so on.
- **Caesar Cipher** is a particular case (for $k = 3$).
 - *Julius Caesar*, who used it in his private correspondence; he used fixed #3 for shifting.
- **Mathematically,**
 - * $Z_{26} = \{0, 1, 2, \dots, 24, 25\}$
 - * $P = C = K = Z_{26}$
 - * For $k \in K$,
 $E_k(x) = (x + k) \bmod 26$ for $x \in P$
 $D_k(y) = (y - k) \bmod 26$ for $y \in C$

Additive/Shift Cipher: Example

- The plaintext is ordinary English text.
- Correlation between alphabetic characters and integer:
 $A = 0, B = 1, \dots, Y = 24, Z = 25.$

Plaintext →	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
Ciphertext →	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Value →	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

Additive/Shift Cipher: Working Principle

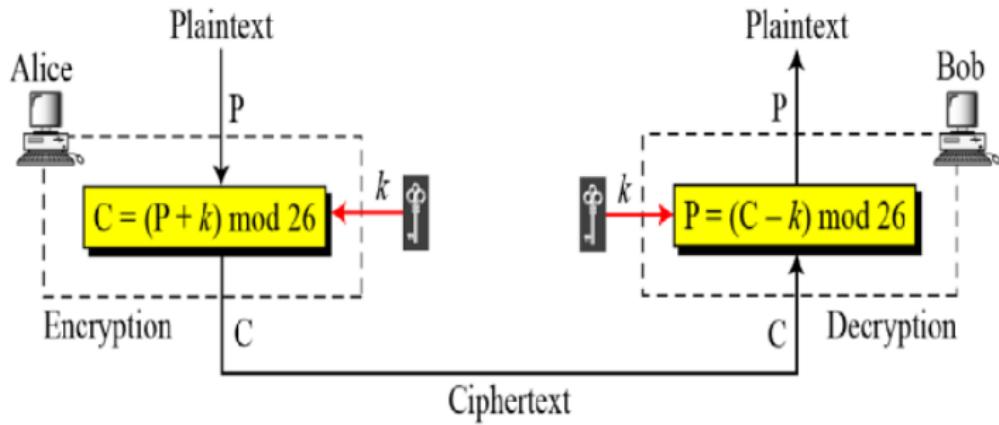


Figure 14: Additive/Shift Cipher (**Note: When the cipher is additive, the plaintext, ciphertext, and key are integers in Z_{26}**).

Additive/Shift Cipher: Encryption

- key $k = 15$
- Plaintext is “hello”
- Corresponding sequence of integers:
07 04 11 11 14
- we add 15 (key) to each value (reducing modulo 26):
22 19 00 00 03
- Convert the sequence of integers to alphabetic characters:
W T A A D

Plaintext: h → 07

Encryption: $(07 + 15) \text{ mod } 26$

Ciphertext: 22 → W

Plaintext: e → 04

Encryption: $(04 + 15) \text{ mod } 26$

Ciphertext: 19 → T

Plaintext: l → 11

Encryption: $(11 + 15) \text{ mod } 26$

Ciphertext: 00 → A

Plaintext: l → 11

Encryption: $(11 + 15) \text{ mod } 26$

Ciphertext: 00 → A

Plaintext: o → 14

Encryption: $(14 + 15) \text{ mod } 26$

Ciphertext: 03 → D

Additive/Shift Cipher: Decryption

- ciphertext : "WTAAD"
- convert the ciphertext to sequence of integers:
22 19 00 00 03
- subtract 15 from each value (reducing modulo 26):
07 04 11 11 14
- convert the sequence of integers to alphabetic characters: "hello"

Ciphertext: W → 22	Decryption: $(22 - 15) \text{ mod } 26$	Plaintext: 07 → h
Ciphertext: T → 19	Decryption: $(19 - 15) \text{ mod } 26$	Plaintext: 04 → e
Ciphertext: A → 00	Decryption: $(00 - 15) \text{ mod } 26$	Plaintext: 11 → l
Ciphertext: A → 00	Decryption: $(00 - 15) \text{ mod } 26$	Plaintext: 11 → l
Ciphertext: D → 03	Decryption: $(03 - 15) \text{ mod } 26$	Plaintext: 14 → o

Shift/Cyclic Decryption.

Caesar Cipher

- **Caesar Cipher** is the earliest known (and the simplest).
- It involves replacing each letter of the alphabet with the letter standing three places further down. This is then wrapped around on itself when the end is reached.
- For example:

with K=3

	0	1	2	3	4	5	6	7	8	9	10	11	12
plaintext	a	b	c	d	e	f	g	h	i	j	k	l	m
ciphertext	D	E	F	G	H	I	J	K	L	M	N	O	P
	13	14	15	16	17	18	19	20	21	22	23	24	25
plaintext	n	o	p	q	r	s	t	u	v	w	x	y	z
ciphertext	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

attackatdawn → DWWDFNDWFDZQ

Additive/ Shift Cipher: Example

- Eve/Darth has intercepted the ciphertext "**UVACLYFZLJBYL**". Show how she/he can use a brute-force attack to break the cipher!!

Additive/ Shift Cipher: Example

- Eve/Darth has intercepted the ciphertext “**UVACLYFZLJBYL**”. Show how she/he can use a brute-force attack to break the cipher!!

Ciphertext: UVACLYFZLJBYL

K = 1	→	Plaintext: tuzbkxeykiaxk
K = 2	→	Plaintext: styajwdxjhzwj
K = 3	→	Plaintext: rsxzivcwigyvi
K = 4	→	Plaintext: qrwyhubvhfxuh
K = 5	→	Plaintext: pqvxgtaugewtg
K = 6	→	Plaintext: opuwfsztfdvsf
K = 7	→	Plaintext: notverysecure

Additive/Shift cipher: Secure?? (Cryptanalysis)

- Shift Cipher is not Secure
- Brute-force cryptanalysis easily performed on the shift cipher by trying all 25 possible keys.
- Given a ciphertext string, Eve/Darth successively try the decryption process with $k = 0, 1, 2, \text{ etc.}$ until get a meaningful text.
- Additive ciphers are also subject to **statistical attacks**.

Additive/Shift ciphers: Prone to Statistical Attacks!!

Letter	Frequency	Letter	Frequency	Letter	Frequency	Letter	Frequency
E	12.7	H	6.1	W	2.3	K	0.08
T	9.1	R	6.0	F	2.2	J	0.02
A	8.2	D	4.3	G	2.0	Q	0.01
O	7.5	L	4.0	Y	2.0	X	0.01
I	7.0	C	2.8	P	1.9	Z	0.01
N	6.7	U	2.8	B	1.5		
S	6.3	M	2.4	V	1.0		

Figure 19: Frequency of occurrence of letters in an English text of 100 characters.

Digram	TH, HE, IN, ER, AN, RE, ED, ON, ES, ST, EN, AT, TO, NT, HA, ND, OU, EA, NG, AS, OR, TI, IS, ET, IT, AR, TE, SE, HI, OF
Trigram	THE, ING, AND, HER, ERE, ENT, THA, NTH, WAS, ETH, FOR, DTH

Grouping of digrams and trigrams based on their frequency in English.

Multiplicative Ciphers

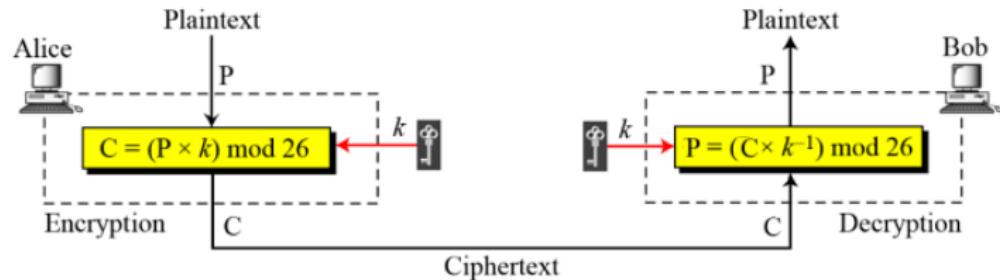


Figure 21: Multiplicative Cipher

- In a multiplicative cipher, the plaintext and ciphertext are integers in Z_{26} ; the key is an integer in Z_{26}^* .

Multiplicative Cipher: Example

- What is the key domain for any multiplicative cipher, if keys are from English alphabet set?

Multiplicative Cipher: Example

- What is the key domain for any multiplicative cipher, if keys are from English alphabet set?
- **The key needs to be in Z_{26}^* . This set has only 12 members: 1, 3, 5, 7, 9, 11, 15, 17, 19, 21, 23, 25.**

Multiplicative Cipher: Example

- What is the key domain for any multiplicative cipher, if keys are from English alphabet set?
- **The key needs to be in Z_{26}^* . This set has only 12 members: 1, 3, 5, 7, 9, 11, 15, 17, 19, 21, 23, 25.**

Plaintext: h → 07	Encryption: $(07 \times 07) \bmod 26$	ciphertext: 23 → X
Plaintext: e → 04	Encryption: $(04 \times 07) \bmod 26$	ciphertext: 02 → C
Plaintext: l → 11	Encryption: $(11 \times 07) \bmod 26$	ciphertext: 25 → Z
Plaintext: l → 11	Encryption: $(11 \times 07) \bmod 26$	ciphertext: 25 → Z
Plaintext: o → 14	Encryption: $(14 \times 07) \bmod 26$	ciphertext: 20 → U

Figure 22: Use of multiplicative cipher to encrypt the message "hello" with a key of 7.

Affine Ciphers

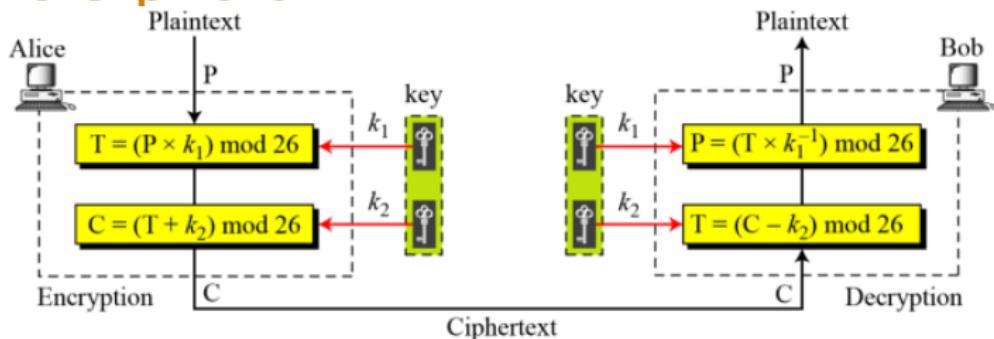


Figure 23: Affine Cipher: A combination of additive and multiplicative cipher with a pair of keys.

$$C = (P \times k_1 + k_2) \bmod 26$$

$$P = ((C - k_2) \times k_1^{-1}) \bmod 26$$

where k_1^{-1} is the multiplicative inverse of k_1 and $-k_2$ is the additive inverse of k_2

Key Domain

- The affine cipher uses a pair of keys in which the first key is from Z_{26}^* and the second is from Z_{26} .
- What is the size of key domain for Additive/Shift cipher?

Key Domain

- The affine cipher uses a pair of keys in which the first key is from Z_{26}^* and the second is from Z_{26} .
- What is the size of key domain for Additive/Shift cipher?
- What is the size of key domain for Multiplicative cipher?

Key Domain

- The affine cipher uses a pair of keys in which the first key is from Z_{26}^* and the second is from Z_{26} .
- What is the size of key domain for Additive/Shift cipher?
- What is the size of key domain for Multiplicative cipher?
- What is the size of key domain for Affine cipher?

Key Domain

- The affine cipher uses a pair of keys in which the first key is from Z_{26}^* and the second is from Z_{26} .
- What is the size of key domain for Additive/Shift cipher?
- What is the size of key domain for Multiplicative cipher?
- What is the size of key domain for Affine cipher?
(i) 25, (ii) 11, (iii) $26 \times 12 - 1 = 312 - 1 = 311$

Key Domain

- The affine cipher uses a pair of keys in which the first key is from Z_{26}^* and the second is from Z_{26} .
- What is the size of key domain for Additive/Shift cipher?
- What is the size of key domain for Multiplicative cipher?
- What is the size of key domain for Affine cipher?
 - (i) 25, (ii) 11, (iii) $26 \times 12 - 1 = 312 - 1 = 311$
- **The additive cipher is a special case of an affine cipher in which $k_1 = 1$. The multiplicative cipher is a special case of affine cipher in which $k_2 = 0$.**

Affine Cipher: Example

P: h → 07	Encryption: $(07 \times 7 + 2) \bmod 26$	C: 25 → Z
P: e → 04	Encryption: $(04 \times 7 + 2) \bmod 26$	C: 04 → E
P: l → 11	Encryption: $(11 \times 7 + 2) \bmod 26$	C: 01 → B
P: l → 11	Encryption: $(11 \times 7 + 2) \bmod 26$	C: 01 → B
P: o → 14	Encryption: $(14 \times 7 + 2) \bmod 26$	C: 22 → W

Figure 24: Use of an affine cipher to encrypt the message “hello” with the key pair (7,2).

C: Z → 25	Decryption: $((25 - 2) \times 7^{-1}) \bmod 26$	P: 07 → h
C: E → 04	Decryption: $((04 - 2) \times 7^{-1}) \bmod 26$	P: 04 → e
C: B → 01	Decryption: $((01 - 2) \times 7^{-1}) \bmod 26$	P: 11 → l
C: B → 01	Decryption: $((01 - 2) \times 7^{-1}) \bmod 26$	P: 11 → l
C: W → 22	Decryption: $((22 - 2) \times 7^{-1}) \bmod 26$	P: 14 → o

Figure 25: Use of an affine cipher to decrypt the message “ZEBBW” with the key pair (7,2) in modulus 26.

Monoalphabetic Substitution Cipher

- Additive, multiplicative, and affine ciphers have small key domains; therefore, they are very vulnerable to brute-force attack.
- **Monoalphabetic Substitution Cipher:** A better solution is to create a mapping between each plaintext character and the corresponding ciphertext character.
 - Alice and Bob can agree on a table showing the mapping for each character.

Plaintext →	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
Ciphertext →	N	O	A	T	R	B	E	C	F	U	X	D	Q	G	Y	L	K	H	V	I	J	M	P	Z	S	W

Figure 26: An example key for monoalphabetic substitution cipher.

Monoalphabetic Substitution Cipher: Examples

- We can use the key in Figure to encrypt the message

this message is easy to encrypt but hard to find the key

- The ciphertext is

ICFVQRVVNEFVRNVSIYRGAHSLIOJICNHTIYBFGTICRXRS

Monoalphabetic Substitution Cipher: Secure?? (Cryptanalysis)

- Each alphabetic character is mapped to a unique alphabetic character. **One-to-One**
- We use arbitrary monoalphabetic substitution, so the key space for monoalphabetic subsitution cipher is $26!$, or almost $4 \times 10^{26} \approx 2^{88}$ possible permutations, which is a very large number. Thus, **brute-force** seems infeasible.

Monoalphabetic Substitution Cipher: Secure?? (Cryptanalysis)

- Each alphabetic character is mapped to a unique alphabetic character. **One-to-One**
- We use arbitrary monoalphabetic substitution, so the key space for monoalphabetic subsitution cipher is $26!$, or almost $4 \times 10^{26} \approx 2^{88}$ possible permutations, which is a very large number. Thus, **brute-force** seems infeasible.
- **However, a Monoalphabetic Substitution Cipher is insecure against frequency analysis.**

Polyalphabetic Cipher

- In polyalphabetic substitution, each occurrence of a character may have a different substitute.
- The relationship between a character in the plaintext to a character in the ciphertext is **one-to-many**.

Autokey Cipher

- “autokey” implies that the subkeys are automatically created from the plaintext cipher characters during the encryption process.

$$P = P_1 P_2 P_3 \dots$$

$$C = C_1 C_2 C_3 \dots$$

$$k = (k_1, P_1, P_2, \dots)$$

$$\text{Encryption: } C_i = (P_i + k_i) \bmod 26$$

$$\text{Decryption: } P_i = (C_i - k_i) \bmod 26$$

Autokey Cipher

- “autokey” implies that the subkeys are automatically created from the plaintext cipher characters during the encryption process.

$$P = P_1 P_2 P_3 \dots$$

$$C = C_1 C_2 C_3 \dots$$

$$k = (k_1, P_1, P_2, \dots)$$

Encryption: $C_i = (P_i + k_i) \bmod 26$

Decryption: $P_i = (C_i - k_i) \bmod 26$

Plaintext:	a	t	t	a	c	k	i	s	t	o	d	a	y
P's Values:	00	19	19	00	02	10	08	18	19	14	03	00	24
Key stream:	12	00	19	19	00	02	10	08	18	19	14	03	00
C's Values:	12	19	12	19	02	12	18	00	11	7	17	03	24
Ciphertext:	M	T	M	T	C	M	S	A	L	H	R	D	Y

Figure 27: Assume that Alice and Bob agreed to use an autokey cipher with initial key value $k_1 = 12$. Now Alice wants to send Bob the message “Attack is today”. Enciphering is done character by character.

Autokey Cipher: Secure?? (Cryptanalysis)

- The autokey cipher hides the single-layer frequency statistics of the plaintext. But..
- Vulnerable to brute-force attack as the additive cipher.

Autokey Cipher: Secure?? (Cryptanalysis)

- The autokey cipher hides the single-layer frequency statistics of the plaintext. But..
- Vulnerable to brute-force attack as the additive cipher.
- The first sub-key can be from one of the 25 values.
- We need polyalphabetic ciphers that not only hide the characteristics of the language but also have large key domains.

Playfair Cipher

- Variant of polyalphabetic cipher, used by the British army during World War I.
- The secret key in this cipher is made of 25 alphabet letters arranged in a 5x5 matrix (letter I and J are considered the same when encrypting).

c	h	a	r	l
e	s	b	d	f
g	i/j	k	m	n
o	p	q	t	u
v	w	x	y	z

Figure 28: An example of a secret key in the Playfair cipher.

Playfair Cipher: Rules..

- Plaintext: "meet me at the bridge"
 - Split the sentence into digrams removing spaces, 'x' used to make even number of letters:
me et me at th eb ri dg ex

Playfair Cipher: Rules..

- Plaintext: "meet me at the bridge"
 - Split the sentence into digrams removing spaces, 'x' used to make even number of letters:
me et me at th eb ri dg ex
 - Repeating plaintext letters that are in the same pair are separated with a filler letter, such as x:
"balloon" would be treated as **ba lx lo on**

Playfair Cipher: Rules..

- Plaintext: "meet me at the bridge"
 - Split the sentence into digrams removing spaces, 'x' used to make even number of letters:
me et me at th eb ri dg ex
 - Repeating plaintext letters that are in the same pair are separated with a filler letter, such as x:
"balloon" would be treated as **ba lx lo on**
 - Two plaintext letters in the same row are each replaced by the letter to the right, with the first element of the row circularly following the last.
eb is replaced by sd
ng is replaced by gi (or gj as preferred)

Playfair Cipher: Rules..cont

- Two plaintext letters that fall in the same column are each replaced by the letter beneath, with the top element of the column circularly following the last.
dt would be replaced by my
ty would be replaced by yr
 - Otherwise, each plaintext letter in a pair is replaced by the letter that lies in its own row and the column occupied by the other plaintext letter.
me becomes gd
- Ciphertext therefore is:
“gd do gd rq pr sd hm em bv”

Playfair Cipher: Secure?? (Cryptanalysis)

- Brute force attack is difficult as the size of the key domain is 25!
- Single letter frequency is obscured.
- But digrams are preserved.
- A cryptanalyst can use a ciphertext-only attack based on the digram frequency test to find the key.

Vigenere Cipher

- Simplest polyalphabetic substitution cipher, designed by Blaise de Vigenere, a sixteenth-century french mathematician.
- Consider the set of all Caesar ciphers:
 $\{C_a, C_b, C_c, \dots, C_z\}$
- Key: e.g. **security**
- Encrypt each letter using $C_s, C_e, C_c, C_u, C_r, C_i, C_t, C_y$ in turn.
- Repeat from start after C_y .
- Decryption simply works in reverse.

Vigenere Cipher: Mathematical Representation

- Let m be a positive integer
- $P = C = K = (\mathbb{Z}_{26})^m$
- For $k = (k_1, k_2, \dots, k_m) \in K$,
 1. $e_k(x_1, x_2, \dots, x_m) = (x_1 + k_1, x_2 + k_2, \dots, x_m + k_m)$
 2. $d_k(y_1, y_2, \dots, y_m) = (y_1 - k_1, y_2 - k_2, \dots, y_m - k_m)$
- All above operations are performed in \mathbb{Z}_{26}

Vigenere Cipher: Example

- Correspondence between alphabetic characters and integer:
 $A = 0, B = 1, \dots, Y = 24, Z = 25.$
- $m = 6.$
- Keyword is “CIPHER”, this corresponds to the numerical equivalent
 $k = (2, 8, 15, 7, 4, 17)$

Vigenere Cipher: Example..contd...

- Plaintext : “thiscryptosystemisnotsecure”.
- Encryption: add modulo 26

19	7	8	18	2	17	24	15	19	14	18	24	18	19	4	12
2	8	15	7	4	17	2	8	15	7	4	17	2	8	15	7
21	15	23	25	6	8	0	23	8	21	22	15	20	1	19	19

8	18	13	14	19	18	4	2	20	17	4
4	17	2	8	15	7	4	17	2	8	15
12	9	15	22	8	25	8	19	22	25	19

- Ciphertext:
“VPXZGIAIXWPUBTTMJPWIZITWZT”.

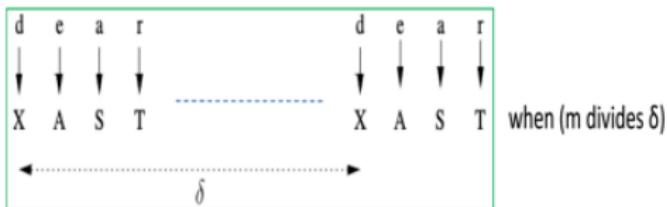
Vigenere Cipher: Secure? (Cryptanalysis)

- Frequency analysis more difficult (but not impossible)
- Attack has two steps
 1. Determine the length m of the key
 2. Determine $K = (k_1, k_2, \dots, k_m)$ by finding each k_i separately.

Determining key length: Kaisiki Test_(content)

courtesy: Chester Rebeiro

- Kasiski test by Friedrich Kasiski in 1863
- Let m be the size of the key
- **observation:** two identical plaintext segments will encrypt to the same ciphertext when they are δ apart and ($m|\delta$)



- If several such δ s are found (i.e. $\delta_1, \delta_2, \delta_3, \dots$) then
 - $m|\delta_1, m|\delta_2, m|\delta_3, \dots$
 - Thus m divides the gcd of $(\delta_1, \delta_2, \delta_3, \dots)$

Increasing Confidence of Key Length (Index of Coincidence)

- Consider a multi set of letters of size N
say $s = \{a, b, c, d, a, a, e, f, e, g, \dots\}$
- Probability of picking two 'a' characters (without replacement) is

$$\frac{n_0}{N} \times \frac{n_0 - 1}{N - 1}$$

n_0 : Number of occurrences of
'a' in S

probability the first pick is 'a'

probability the second pick is 'a'

Index of Coincidence..contd..1

- Sum of probabilities of picking two similar characters is

$$I_c = \sum_{i=0}^{25} \frac{n_i(n_i - 1)}{N(N - 1)}$$

index of coincidence

Index of coincidence.. cont..2

- Consider a random permutation of the alphabets (as in the substitution cipher)
 $s = \{a, b, c, d, a, a, e, f, e, g, \dots\} \longrightarrow S = \{X, M, D, F, X, X, Z, G, Z, J, \dots\}$
- Note that : $n_a = n_x$; thus the value of I_c remains unaltered
- Number of occurrence of an alphabet in a text depends on the language, thus each language will have a unique I_c value

English	0.0667	French	0.0778
German	0.0762	Spanish	0.0770
Italian	0.0738	Russian	0.0529

Vigenere Cipher: Cryptanalysis Example

Let us assume we have intercepted the following ciphertext:

LIOMWGEGGDVWGHHCQUCRHRWAGWIOWQLKGZETKKMEVLWPCZVGTH-
VTSGXQOVGCSVETQLTJSUMVVVEUVLXEWSLGFMVVWLGYHCUSWXQH-
KVGSHEEVFLCFDGVSUMPHKIRZDMPHHBVWWJWIXGFWLTSHGJOUEEHH-
VUCFVGOWICQLTJSUXGLW

Kasiski test for repetition of three character segments yields the results as shown in Table 3.4.

Table 3.4 Kasiski test for Example 3.19

String	First Index	Second Index	Difference
QLT	65	165	100
LTJ	66	166	100
TJS	67	167	100
JSU	68	168	100
SUM	69	117	48
VWV	72	132	60

Vigenere Cipher: Cryptanalysis Example

The greatest common divisor is thus 4, thus suggesting that the key length is a multiple of 4. We confirm this guess by the Index of Coincidence test.

We divide the ciphertext into 4 rows as shown below. We also mention the corresponding Index of Coincidence values. The high values of the IC confirms the key length reported in the Kasiski test.

1st string :

IC = 0.067677

LWGWCRAOKTEPGTQCTJVUEGVGUQGECVPRPVJGTJEUGCJG

2nd string :

IC = 0.074747

IGGGQHGWGVCTSOSQS WVWFVYSHSVFSHZHWWFSOHCQSL

3rd string:

IC = 0.070707

OFDHURWQZKLZHGVVLUVLSZWHWKHF DUKDHVIWHUHF WL UW

4th string:

IC = 0.076768

MEVHCWILEMWVVXGETMEXMLCXVELGMIMBWXLGEVVITX

Vigenere Cipher: Cryptanalysis Example

Then we perform the Mutual Index of Coincidence to obtain the actual key value. Running the test, we obtain that the key value is CODE, and the corresponding plaintext is

JULIUSCAESARUSEDACRYPTOSYSTEMINHISWARWHICHISNOWREFERR
EDTOASCAESARCIPHERITISASHIFTCIPHERWITHTHEKEYSETTOTHREEE
ACHCHARACTERINTHEPLAINTEXTISHIFTERTHREECHARACTERSOCRE
ATEACIPHERTEXT

Note that the plaintext makes sense and hence we believe the decryption is correct. We format the obtained as follows:

Julius Caesar used a cryptosystem in his wars, which is now referred to as Caesar cipher. It is an additive cipher with the key set to three. Each character in the plaintext is shifted three characters to create the ciphertext.

Hill Cipher

(Content courtesy: Chester Rebeiro)

- Encryption: $y = xK \pmod{26}$
- Decryption: $x = yK^{-1} \pmod{26}$
 - plaintext : $x \in \{0, 1, 2, 3, \dots, 25\}$
 - ciphertext : $y \in \{0, 1, 2, 3, \dots, 25\}$
 - key : K is an invertible matrix

Hill Cipher..

(Content courtesy: Chester Rebeiro)

- example

plaintext

h i f f

(7,8)(11,11)

$$K = \begin{bmatrix} 11 & 8 \\ 3 & 7 \end{bmatrix} \quad K^{-1} = \begin{bmatrix} 7 & 18 \\ 23 & 11 \end{bmatrix} \quad K \bullet K^{-1} = 1 \pmod{26}$$

$$\boxed{[7 \ 8] \times \begin{bmatrix} 11 & 8 \\ 3 & 7 \end{bmatrix} \pmod{26} = [23 \ 8]} \quad \text{encryption}$$

$$\boxed{[23 \ 8] \times \begin{bmatrix} 7 & 18 \\ 23 & 11 \end{bmatrix} \pmod{26} = [7 \ 8]} \quad \text{decryption}$$

h i f f \rightarrow (7,8)(11,11) \longrightarrow (23,8)(24,9) \rightarrow X Y J

plaintext ciphertext

Cryptanalysis of Hill Cipher (content courtesy: Chester R. and Debdeep M.)

- ciphertext only attack is difficult
- known plaintext attack

$$\begin{array}{l} (7,8)(11,11) \\ \text{known plaintext} \end{array} \times \begin{bmatrix} k_{11} & k_{12} \\ k_{21} & k_{22} \end{bmatrix} \longrightarrow \begin{array}{l} (23,8)(24,9) \\ \text{corresponding ciphertext} \end{array}$$

Form equations and solve to get the key

$$7k_{11} + 8k_{21} = 23$$

$$7k_{12} + 8k_{22} = 8$$

$$11k_{11} + 11k_{21} = 24$$

$$11k_{12} + 11k_{22} = 9$$

One-Time Pad

- One of the goals of cryptography is perfect secrecy.
- A study by **Shannon** has shown that perfect secrecy can be achieved if each plaintext symbol is encrypted with a key randomly chosen from a key domain.
- This idea is used in a cipher called **one-time pad**, invented by **Vernam**.
- The key has the same length as the plaintext and is chosen in random.
- The key is changed each time the sender sends a new message.

One-Time Pad.. contd..

- For example, an additive cipher can be easily broken because the same key is used to encrypt every character.
- However, even this simple cipher can become a perfect cipher if the key that is used to encrypt each character is chosen randomly from the key domain (00, 01, 02, ..., 25) -i.e if the first character is encrypted using the key 04, the second character is encrypted using the key 02, the third character is encrypted using the key 21; and so on.
- **ciphertext-only attack is impossible.**
- Other types of attacks are also impossible if the sender changes the key each time she/he sends a message, using another random sequence of integers.

One-Time Pad: Feasibility

- A one-time pad is a perfect cipher, but it is almost impossible to implement commercially.
- If the key must be newly generated each time, how can Alice tell Bob the new key each time she has a message to send?
- **There are some occasions when a one-time pad can be used. For example, if the president of a country needs to send a completely secret message to the president of another country, she/he can send a trusted envoy with a random key before sending the message.**

Permutation Cipher

- Ciphers we seen so far were substitution ciphers
 - Plaintext characters substituted with ciphertext characters

his
plaintext

→

X Y J
ciphertext

- Alternate technique: Permutation
 - Plaintext characters re-ordred by a random permutation

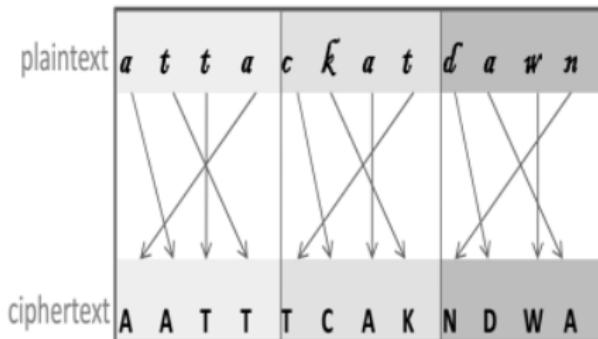
his
plaintext

→

L I H I
ciphertext

Permutation Cipher

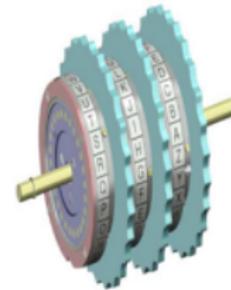
- Example plaintext: **attackatdawn**
 - key: (1,3,2,0) here is of length 4 and a permutation of (0,1,2,3)
 - It refers 0th character in plaintext goes to 1st character in ciphertext (and so on..)



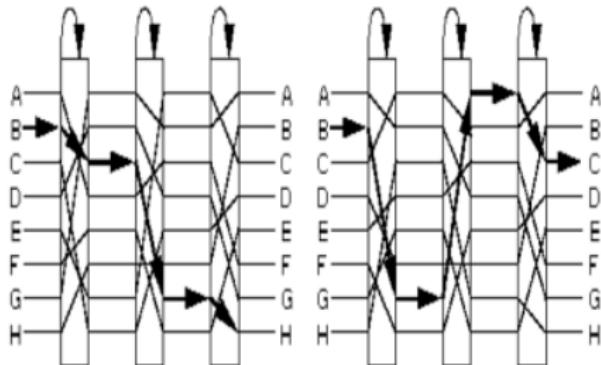
- cryptanalysis : $4!$ possibilities

Rotor Cipher Machines (German Enigma)

- Before modern ciphers, rotor machines were most common complex ciphers in use.
- Widely used in WW2.
- Used a series of rotating cylinders.
- Implemented a polyalphabetic substitution cipher of period K.



Rotor Cipher Machines (German Enigma)



- Each rotor makes a permutation
 - Adding / removing a rotor would change the ciphertext
- Additionally, the rotors rotates with a gear after a character is entered
- Broken by Alan Turing

Transposition Ciphers

- A transposition cipher reorders symbols.
- Classification:
 1. Keyless Transposition Ciphers
 2. Keyed Transposition Ciphers
 3. Combining Two Approaches

Keyless Transposition Ciphers

- Reorders the symbols.
- Simple transposition ciphers, which were used in the past, are keyless.
- Example:
 - A good example of a keyless cipher using the first method is the **rail fence cipher**. The ciphertext is created reading the pattern row by row. For example, to send the message **"Meet me at the park"** to Bob, Alice writes:

The diagram shows the letters of the message "Meet me at the park" arranged in two rows. The first row contains 'M', 'e', 't', 'e', 't', 'h', 'e', 'p', 'r', 'k'. The second row contains 'e', 'e', 'a', 't', 'h', 'e', 'a', 'r'. Arrows indicate the reading order: starting from the top left, moving down to the bottom, then back up to the top right, and so on, forming a zigzag pattern across the two rows.

- She then creates the ciphertext **"MEMATEAKETETHPR"**.

Keyless Transposition Cipher: Example 2

- Alice and Bob can agree on the number of columns and use the second method.
Alice writes the same plaintext, row by row, in a table of four columns.

1	2	3	4	5	6
M	E	E	T	M	E
A	F	T	E	R	P
A	R	T	Y		

4	2	1	6	3	5
T	E	M	E	E	M
E	F	A	P	T	R
Y	R	A		T	

Plain Text: MEET ME AFTER PARTY

Key Used: 421635

Cipher Text: TEMEEMEFAPTRYRAT

Keyless Transposition Cipher: Example 3

Alice and Bob can agree on the number of columns and use the second method. Alice writes the same plaintext, row by row, in a table of four columns.

m	e	e	t
m	e	a	t
t	h	e	p
a	r	k	

She then creates the ciphertext “MMTAEEHREAEKTP”.

Keyless Transposition Cipher: Example 3: Security? (Cryptanalysis)

The cipher in Example is actually a transposition cipher. The following shows the permutation of each character in the plaintext into the ciphertext based on the positions.

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
01	05	09	13	02	06	10	13	03	07	11	15	04	08	12

The second character in the plaintext has moved to the fifth position in the ciphertext; the third character has moved to the ninth position; and so on. Although the characters are permuted, there is a pattern in the permutation: (01, 05, 09, 13), (02, 06, 10, 13), (03, 07, 11, 15), and (08, 12). In each section, the difference between the two adjacent numbers is 4.

Keyed Transposition Ciphers

- The keyless ciphers permute the characters by using writing plaintext in one way and reading it in another way.
- The permutation is done on the whole plaintext to create the whole ciphertext.
- Another method is **to divide the plaintext into groups of predetermined size, called blocks, and then use a key to permute the characters in each block separately.**

Keyed Transposition Cipher: Example

- Alice needs to send the message "**Enemy attacks tonight**" to Bob..

e n e m y a t t a c k s k s t o n i g h t z

- The key used for encryption and decryption is a permutation key, which shows how the characters are permuted.

Encryption ↓

3	1	4	5	2
1	2	3	4	5

↑ Decryption

Keyed Transposition Cipher: Example

- Alice needs to send the message “**Enemy attacks tonight**” to Bob..

e n e m y a t t a c k s k s t o n i g h t z

- The key used for encryption and decryption is a permutation key, which shows how the characters are permuted.

Encryption ↓

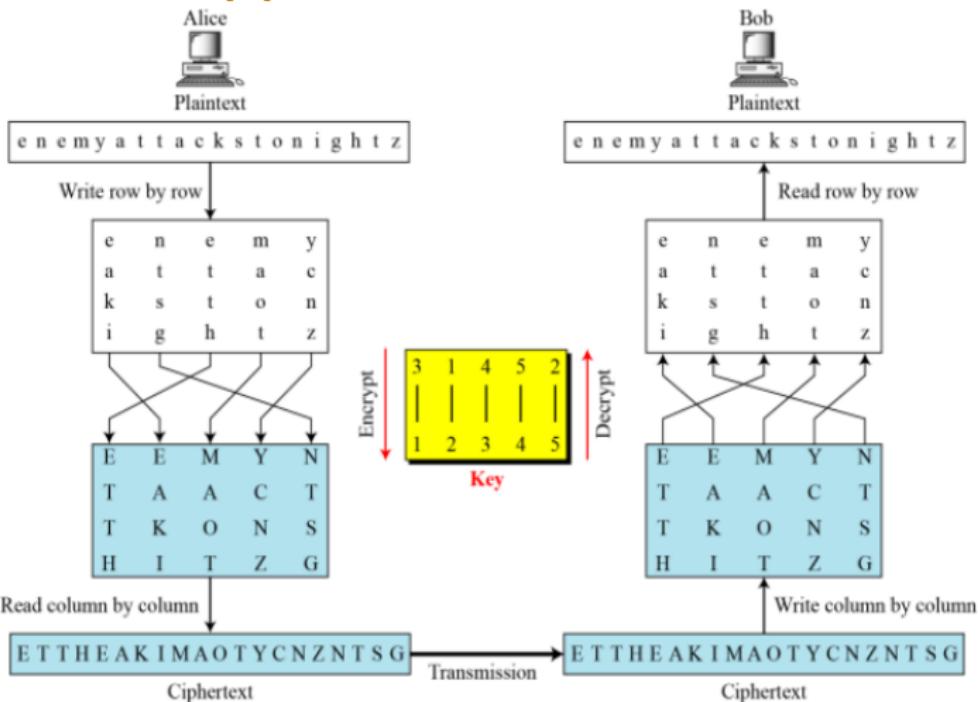
3	1	4	5	2
1	2	3	4	5

↑ Decryption

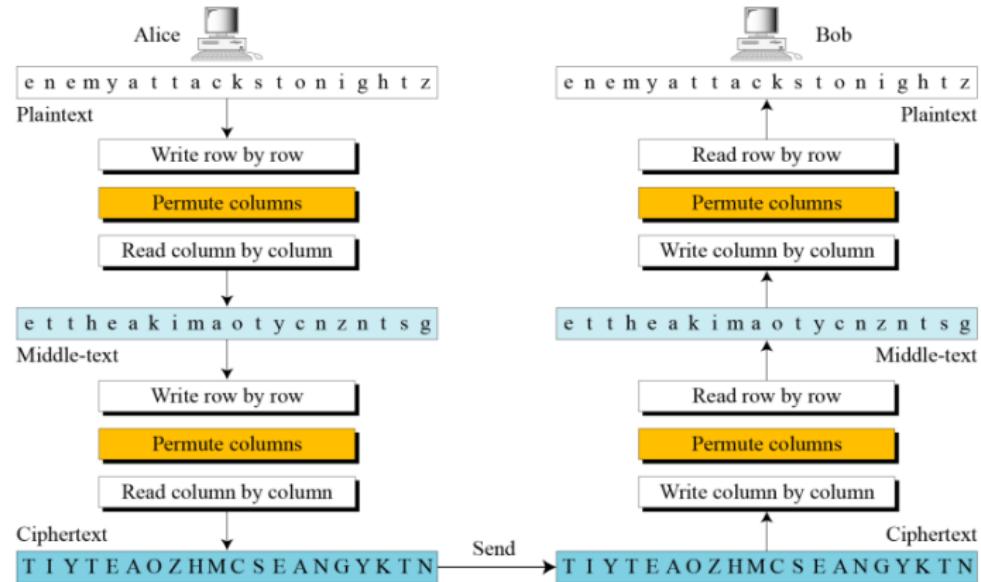
- The permutation yields

E E M Y N T A A C T T K O N S H I T Z G

Combined Approaches



Double Transposition Cipher



Bibliography: Books and Resources

- Cryptography and Network Security: Principles and Practice by William Stallings
- Cryptography and Network Security by Behrouz A Forouzan and Debdeep Mukhopadhyay
- Principles of Information Security by Michael E. Whitman and Herbert J. Mattord.
- Cisco platform, and Internet.
- Published research papers, study materials from researchers of security domain.

SGSITS Indore, Department of Computer Engineering
Information and Network Security
MID SEMESTER-I-Part-II Question Paper

Instructors: Dr. Soma Saha, Mr. Veerendra Srivastava

Duration: 40mins.

March 8, 2021

1 Problem set– Part 2. All questions are compulsory.

1. Let $C = P = \mathbb{Z}_{26}$, and let the **encryption** key for the Hill Cipher be

$$K \begin{bmatrix} 9 & 4 \\ 1 & 3 \end{bmatrix}$$

Find the **plaintext** which encrypts to the ciphertext **#name** when using the Hill cipher and the key K. (Hint: First find the decryption key K^{-1} .) Clearly mention each step. (Marks: 5)

Note: **#name** signifies the first four letters of student's name in captial letters. For example,

1. If student's name is "**Prabhu Deva**", the data will be **#name** = "**PRAV**".
 2. If student name is "**Ivy Sharma**", the data will be **#name** = "**IVYS**", S is taken from surname in sequence.
 3. If two students in the class have same first name like, "**Madhuri Dixit**" and "**Madhuri Padukone**", the data will be **#name** = "**DIXI**", and the data will be **#name** = "**PADU**", respectively.
2. (Marks: 2+3)
- a. Find the number of multiplicative inverses and the values in the set \mathbb{Z}_n^* , where **n** represents the value **#last_digit**+11. **#last_digit** represents the last digit of student's enrollment number.
 - b. For a message transfer, encrypt the message "**life begins where fear ends**" using Vigenère Cipher where the key is student's 12 digit **unique "enrollment_number"**. Find the ciphertext. Clearly mention each step.
3. (Marks: 5)
- a. What is the difference between IP security transport mode and tunnel mode?
 - b. Draw the packet formats for the ESP and AH protocol for both transport mode and tunnel mode.
4. (Marks: 5)

- a. What are the services provided by PGP?
- b. If PGP or any specific protocol is NOT used then what are the other alternatives to secure the GMAIL emailing services?

SGSITS Indore, Department of Computer Engineering
Information and Network Security
MID SEMESTER-II Question Paper

Dr. Soma Saha and Mr. Veerendra Shrivastava

April 5, 2021

1 Problem set– Part 2

1. (a) Answer the following question: (Marks: 2)

You receive the following email from the Help Desk:

Dear Sgsits Gsuite Email User,

Beginning next week, we will be deleting all inactive email accounts in order to create space for more users. You are required to send the following information in order to continue using your email account. If we do not receive this information from you by the end of the week, your email account will be closed.

*Name (first and last):

*Email Login:

*Password:

*Date of birth:

*Alternate email:

Please contact the Sgsits Web Services Team with any questions.

Thank you for your immediate attention.

What should you do?

- (b) Let x be an element of Z_n and y denote its multiplicative-inverse-mod- n . (Marks: 3)
 - When does y exist?
 - Give the equation that x and y satisfy.
 2. (a) Evaluate the following (Marks: 3)
 - $5^{45x} \bmod 10$
 - $8^{86x} \bmod 10$
- NOTE: x represents the $\#letters_in_your_name + surname \bmod 10$. For example, if your name is “MUNNA CIRCUIT”, the value of x will be $12 \bmod 10 = 2$, where $\#letters_in_your_name + surname = 12$.
- (b) What are the main differences between symmetric-key and asymmetric-key cryptography? Which one is mostly used, and why? (Marks: 2)
 3. Draw the SSL Protocol stack and explain each of the protocol of SSL protocol stack. (Marks: 5)
 4. Differentiate between MIME and S/MIME Protocol. (Marks: 5)

21.	Define information security.
Ans.	According to the SANS Institute “Information security refers to the processes and methodologies that are designed and implemented to protect print, electronic, or any other form of confidential, private and sensitive information or data from unauthorized access, use, misuse, disclosure, destruction, modification or disruption.”
22.	Differentiate between Information Security, Cyber Security and Network Security.
Ans.	Information Security refers to the processes and methodologies that are designed and implemented to protect print, electronic, or any other form of confidential, private and sensitive information or data from unauthorized access, use, misuse, disclosure, destruction, modification or disruption. Cyber Security, a subset of information security, is the practice of defending organization’s networks, computers and data from unauthorized digital access, attack or damage by implementing various processes, technologies and practices. Network Security is the process of taking physical and software preventative measures to protect the underlying networking infrastructure from unauthorized access, misuse, malfunction, modification, destruction, or improper disclosure, thereby creating a secure platform for computers, users and programs to perform their permitted critical functions within a secure environment.
23.	Define the following terms: a) Vulnerability b) Threat c) Exploit d) Risk
Ans.	a) Vulnerability: Vulnerability is a cyber-security term that refers to a flaw in a system that can leave it open to attack. Vulnerability may also refer to any type of weakness in a computer system itself, in a set of procedures, or in anything that leaves information security exposed to a threat. b) Threat: A threat refers to a new or newly discovered incident with the potential to do harm to a system or your overall organization. There are three main types of threats – natural threats (e.g., floods or a tornado), unintentional threats (such as an employee mistakenly accessing the wrong information) and intentional threats. c) Exploit: The term exploit is commonly used to describe a software program that has been developed to attack an asset by taking advantage of vulnerability. The objective of many exploits is to gain control over an asset. d) Risk: Risk refers to the potential for loss or damage when a threat exploits vulnerability. Examples of risk include financial losses as a result of business disruption, loss of privacy, reputational damage, and legal implications and can even include loss of life.
24.	What do you mean by active attacks? What are the different types of active attacks?
Ans.	Active attacks involve some modification of the data stream or the creation of a false stream. The active attacks can be subdivided into four categories: 1. Masquerade 2. Replay 3. Modification of messages 4. Denial of service
25.	What do you mean by passive attacks? What are the different types of passive attacks?
Ans.	A Passive Attack attempts to learn or make use of information from the system but does not affect system resources. Passive attacks are in the nature of eavesdropping on, or monitoring of, transmissions. The goal of the opponent is to obtain information that is being transmitted. There are two types of passive attacks: 1. Release of message contents 2. Traffic analysis.
26.	Differentiate between DOS and DDoS attack.
Ans.	A Denial-of-Service (DoS) is a type of attack where the attackers (hackers) attempt to prevent legitimate users from accessing the service. A DoS attack can be done in several ways. The basic types of DoS attack include: 1. Flooding the network to prevent legitimate network traffic. 2. Disrupting the connections between two machines, thus preventing access to a service 3. Preventing a particular individual from accessing a service. 4. Disrupting the state of information, such as resetting of TCP sessions A Distributed denial of service (DDoS) attacks represents the next step in the evolution of DoS attacks as a way of disrupting the Internet. Cyber criminals began using DDoS attacks around 2000. The attacks use large numbers of compromised computers, as well as other electronic devices — such as webcams and smart televisions that make up the ever-increasing Internet of Things — to force the shutdown of the targeted website, server or network. In contrast, a DoS attack generally uses a single computer and a single IP address to attack its target, making it easier to defend against.
27.	What do you mean by Zero Day Attack? Explain.
Ans.	The term “zero-day” refers to newly discovered software vulnerability. Because the developer has just learned of the flaw, it also means an official patch or update to fix the issue hasn’t been released. So, “zero-day” refers to the fact that the developers have “zero days” to fix the problem that has just been exposed — and perhaps already exploited by hackers. Once the vulnerability becomes publicly known, the vendor has to work quickly to fix the issue to protect its users. But the software vendor may fail to release a patch before hackers manage to exploit the security hole. That’s known as a zero-day attack.
28.	Define the security goals.
Ans.	There are three main security goals:

	<p>Confidentiality: Confidentiality means keeping the secrets secret. Information has confidentiality when it is protected from disclosure or exposure to unauthorized individuals or systems. Confidentiality ensures that only those with the rights and privileges to access information are able to do so.</p> <p>Integrity: It can be achieved by - Identification, Authentication and Authorization. Integrity can apply to a stream of messages, a single message, or selected fields within a message. A connection-oriented integrity service, one that deals with a stream of messages, assures that messages are received as sent, with no duplication, insertion, modification, reordering, or replays. A connection-less integrity service, one that deals with individual messages without regard to any larger context, generally provides protection against message modification only.</p> <p>Availability: Availability enables authorized users—persons or computer systems—to access information without interference or obstruction and to receive it in the required format.</p>
29.	What are the different components of Information System?
Ans.	An information system (IS) is much more than computer hardware; it is the entire set of software, hardware, data, people, procedures, and networks that make possible the use of information resources in the organization. These six critical components enable information to be input, processed, output, and stored. Each of these IS components has its own strengths and weaknesses, as well as its own characteristics and uses. Each component of the information system also has its own security requirements.
30.	Differentiate between authentication and authorization.
Ans.	In authentication process, the identity of users is checked for providing the access to the system. While in authorization process, person's or user's authorities are checked for accessing the resources. In authentication process, users or persons are verified. While in authorization process, users or persons are validated.
31.	What is Risk? What did you understand from Risk Management?
Ans.	<p>Risk: Risk refers to the potential for loss or damage when a threat exploits vulnerability. Examples of risk include financial losses as a result of business disruption, loss of privacy, reputational damage, and legal implications and can even include loss of life. Risk management involves three major undertakings: risk identification, risk assessment, and risk control.</p> <p>Risk identification is the examination and documentation of the security posture of an organization's information technology and the risks it faces.</p> <p>Risk assessment is the determination of the extent to which the organization's information assets are exposed or at risk.</p> <p>Risk control is the application of controls to reduce the risks to an organization's data and information systems.</p>
32.	What are the different types of security policies?
Ans.	There are four types of security policies: <ol style="list-style-type: none"> 1. General security policies 2. Program security policies 3. Issue-specific policies 4. Systems-specific policies
33.	What do you mean by Cyber Law? Explain.
Ans.	The virtual world of internet is known as cyberspace and the laws governing this area are known as Cyber laws and all the netizens of this space come under the ambit of these laws as it carries a kind of universal jurisdiction. Cyber law can also be described as that branch of law that deals with legal issues related to use of inter-networked information technology. In short, cyber law is the law governing computers and the internet.
34.	What do you mean Man In The Middle Attack? Explain.
Ans.	A man-in-the-middle attack is a type of cyberattack where a malicious actor inserts him/herself into a conversation between two parties, impersonates both parties and gains access to information that the two parties were trying to send to each other. A man-in-the-middle attack allows a malicious actor to intercept, send and receive data meant for someone else, or not meant to be sent at all, without either outside party knowing until it is too late.
35.	Define Nonrepudiation?
Ans.	Nonrepudiation prevents either sender or receiver from denying a transmitted message. Thus, when a message is sent, the receiver can prove that the alleged sender in fact sent the message. Similarly, when a message is received, the sender can prove that the alleged receiver in fact received the message.
36.	What is XSS attack? How it can be performed?
Ans.	<p>Cross-site scripting (XSS) is a code injection attack that allows an attacker to execute malicious JavaScript in another user's browser. The attacker does not directly target his victim. Instead, he exploits vulnerability in a website that the victim visits, in order to get the website to deliver the malicious JavaScript for him. To the victim's browser, the malicious JavaScript appears to be a legitimate part of the website, and the website has thus acted as an unintentional accomplice to the attacker. There are three types of XSS attacks:</p> <p>Stored XSS – Stored XSS also known as persistent XSS, occurs when user input is stored on the target server such as database/message forum/comment field etc. Then the victim is able to retrieve the stored data from the web application.</p> <p>Reflected XSS – Reflected XSS also known as non-persistent XSS, occurs when user input is immediately returned by a web application in an error message/search result or the input provided by the user as part of the</p>

	<p>request and without permanently storing the user provided data.</p> <p>DOM Based XSS – DOM Based XSS is a form of XSS when the source of the data is in the DOM, the sink is also in the DOM, and the data flow never leaves the browser.</p> <p>Cross Site Scripting attack means sending and injecting malicious code or script. Malicious code is usually written with client-side programming languages such as Javascript, HTML, VBScript, Flash, etc. However, Javascript and HTML are mostly used to perform this attack.</p> <p>This attack can be performed in different ways. Depending upon the type of XSS attack, the malicious script may be reflected on the victim's browser or stored in the database and executed every time, when the user calls the appropriate function. The main reason for this attack is inappropriate user's input validation, where malicious input can get into the output. A malicious user can enter a script, which will be injected into the website's code. Then the browser is not able to know if the executed code is malicious or not. Therefore malicious script is being executed on the victim's browser or any faked form is being displayed for the users. There are several forms in which XSS attack can occur.</p> <p>Main forms of Cross Site Scripting are as follows:</p> <ul style="list-style-type: none"> a) Cross Site Scripting can occur on the malicious script executed at the client side. b) Fake page or form displayed to the user (where the victim types credentials or clicks a malicious link). c) On the websites with displayed advertisements. d) Malicious emails sent to the victim. <p>This attack occurs when the malicious user finds the vulnerable parts of the website and sends it as appropriate malicious input. Malicious script is being injected into the code and then sent as the output to the final user.</p>
37.	What is SQL injection attack? How it can be performed?
Ans.	<p>An SQL query is a request for some action to be performed on a database, most commonly on a web page that asks for a username or password. But since most websites don't monitor inputs other than usernames and passwords, a hacker can use the input boxes to send their own requests – that is, inject SQL into the database. This way, hackers can create, read, update, alter or delete data stored in the back-end database, usually to access sensitive information such as social security numbers and credit card data as well as other financial information. SQL injection usually occurs when you ask a user for input, like their username/userid, and instead of a name/id, the user gives you an SQL statement that you will unknowingly run on your database.</p> <p>Consider the following example which creates a SELECT statement by adding a variable (txtUserId) to a select string. The variable is fetched from user input (getRequestParam):</p> <p>Example:</p> <pre>txtUserId = getRequestParam("UserId"); txtSQL = "SELECT * FROM Users WHERE UserId = " + txtUserId;</pre> <p>The original purpose of the code was to create an SQL statement to select a user, with a given user id. If there is nothing to prevent a user from entering "wrong" input, the user can enter some "smart" input like this:</p> <p>UserId: 105 OR 1=1</p> <p>Then, the SQL statement will look like this: SELECT * FROM Users WHERE UserId = 105 OR 1=1;</p> <p>The SQL above is valid and will return ALL rows from the "Users" table, since OR 1=1 is always TRUE.</p>
38.	What do you mean by Risk Assessment? Explain.
Ans.	<p>After identifying the information asset and threat & vulnerabilities, we evaluate the relative risk for each of the vulnerabilities. This process is called Risk Assessment. Risk assessment assigns a risk rating or score to each information asset. While this number does not mean anything in absolute terms, it is useful in gauging the relative risk to each vulnerable information asset and facilitates the development of comparative ratings later in the risk control process. Likelihood is the probability that a specific vulnerability will be the object of a successful attack. In risk assessment, we assign a numeric value to likelihood. The National Institute of Standards and Technology (NIST) recommends in Special Publication 800-30 assigning a number between 0.1 (low) and 1.0 (high).</p>
39.	What do you mean by Risk control? What are the different strategies to control the risk?
Ans.	<p>Once the project team for information security development has created the ranked vulnerability worksheet, the team must choose one of five basic strategies to control each of the risks that result from these vulnerabilities. The five strategies are: defend, transfer, mitigate, accept, and terminate.</p> <ol style="list-style-type: none"> 1. Defend: The defend control strategy attempts to prevent the exploitation of the vulnerability. This is the preferred approach and is accomplished by means of countering threats, removing vulnerabilities from assets, limiting access to assets, and adding protective safeguards. 2. Transfer: The transfer control strategy attempts to shift risk to other assets, other processes, or other organizations. This can be accomplished by rethinking how services are offered, revising deployment models, outsourcing to other organizations, purchasing insurance, or implementing service contracts with providers. 3. Mitigate: The mitigate control strategy attempts to reduce the impact caused by the exploitation of

	<p>vulnerability through planning and preparation. Mitigation begins with the early detection that an attack is in progress and a quick, efficient, and effective response.</p> <p>4. Accept: The accept control strategy is the choice to do nothing to protect vulnerability and to accept the outcome of its exploitation. This may or may not be a conscious business decision.</p> <p>5. Terminate: The terminate control strategy directs the organization to avoid those business activities that introduce uncontrollable risks.</p>
40.	<p><u>Explain the legal and ethical issues in computer security?</u></p> <p>Ans. Law: The law may be understood as the systematic set of universally accepted rules and regulation created by an appropriate authority such as government, which may be regional, national, international, etc.</p> <p>Ethics: Also described as moral philosophy, is a system of moral principles which is concerned with what is good for individuals and society.</p> <p>Nobody will be punished when they violate ethics; but whoever violates laws is going to receive punishment carried out by relevant authorities. Besides, an action can be illegal, but morally right.</p> <p>For example, in ancient China, some people rob properties from rich people, and give it to poor people, and it is considered to be morally right but be illegal.</p> <p>Similarly, an action that is legal can be morally wrong. For instance, some people spend thousands of dollars on their pets while some poor people on the street cannot have enough food. Ethics emphasizes more on positive aspects while laws are more concerned with negative actions.</p> <p>There are different types of laws:</p> <p>Civil law, Criminal law, Private law, Public law</p> <p>There are four types of security policies:</p> <ol style="list-style-type: none"> 1. General security policies 2. Program security policies, 3. Issue-specific policies 4. Systems-specific policies.
41.	<p><u>What is replay attack? How it works? Explain.</u></p> <p>Ans. A replay attack occurs when a cybercriminal eavesdrops on a secure network communication, intercepts it, and then fraudulently delays or resends it to misdirect the receiver into doing what the hacker wants.</p> <p>One of the best techniques to avert replay attacks is by using strong digital signatures with timestamps.</p> <p>A one-time password for each request also helps in preventing replay attacks and is frequently used in banking operations.</p> <p>Consider this real-world example of an attack. A staff member at a company asks for a financial transfer by sending an encrypted message to the company's financial administrator. An attacker eavesdrops on this message, captures it, and is now in a position to resend it.</p> <p>Because it's an authentic message that has simply been resent, the message is already correctly encrypted and looks legitimate to the financial administrator.</p> <p>In this scenario, the financial administrator is likely to respond to this new request unless he or she has a good reason to be suspicious. That response could include sending a large sum of money to the attacker's bank account.</p>
42.	<p><u>What do you mean by confidentiality in security goals? How it can be achieved?</u></p> <p>Ans. Confidentiality means keeping the secrets secret. Information has confidentiality when it is protected from disclosure or exposure to unauthorized individuals or systems. Confidentiality ensures that only those with the rights and privileges to access information are able to do so. When unauthorized individuals or systems can view information, confidentiality is breached.</p> <p>Confidentiality is the protection of transmitted data from passive attacks. With respect to the content of a data transmission, several levels of protection can be identified. The other aspect of confidentiality is the protection of traffic flow from analysis. This requires that an attacker not be able to observe the source and destination, frequency, length, or other characteristics of the traffic on a communication facility. The value of confidentiality of information is especially high when it is personal information about employees, customers, or patients. The confidentiality can be achieved by using the various cryptographic algorithms.</p>
43.	<p><u>What do you mean by integrity in security goals? How it can be achieved?</u></p> <p>Ans. Integrity can apply to a stream of messages, a single message, or selected fields within a message. A connection-oriented integrity service, one that deals with a stream of messages, assures that messages are received as sent, with no duplication, insertion, modification, reordering, or replays. A connection-less integrity service, one that deals with individual messages without regard to any larger context, generally provides protection against message modification only.</p> <p>The integrity of information is threatened when the information is exposed to corruption, damage, destruction, or other disruption of its authentic state. Corruption can occur while information is being stored or transmitted. Many computer viruses and worms are designed with the explicit purpose of corrupting data.</p> <p>Another key method of assuring information integrity is file hashing, in which a file is read by a special algorithm that uses the value of the bits in the file to compute a single large number called a hash value. The hash value for any combination of bits is unique. If a computer system performs the same hashing algorithm on a file and obtains a different number than the recorded hash value for that file, the file has been compromised and the integrity of the information is lost. Noise in the transmission media, for instance, can also cause data to</p>

	<p>lose its integrity.</p> <p>Transmitting data on a circuit with a low voltage level can alter and corrupt the data. Redundancy bits and check bits can compensate for internal and external threats to the integrity of information.</p> <p>During each transmission, algorithms, hash values, and the error-correcting codes ensure the integrity of the information. Data whose integrity has been compromised is retransmitted.</p> <p>It can be achieved by: Identification, Authentication, Authorization</p>
44.	<p>Describe about the Cyber Laws.</p> <p>Ans. The virtual world of internet is known as cyberspace and the laws governing this area are known as Cyber laws and all the netizens of this space come under the ambit of these laws as it carries a kind of universal jurisdiction. Cyber law can also be described as that branch of law that deals with legal issues related to use of inter-networked information technology.</p> <p>In short, cyber law is the law governing computers and the internet. Cyber law is important because it touches almost all aspects of transactions and activities on and involving the internet, World Wide Web and cyberspace. Every action and reaction in cyberspace has some legal and cyber legal perspectives.</p> <p>Cyber law encompasses laws relating to:</p> <ol style="list-style-type: none"> 1. Cyber crimes 2. Electronic and digital signatures 3. Intellectual property 4. Data protection and privacy
45.	<p>If an organization must evaluate the following three information assets for risk management, which vulnerability should be evaluated first for additional safety controls? Which should be evaluated last?</p> <p>a) Switch L47 connects a network to the Internet. It has two vulnerabilities: it is susceptible to hardware failure at a likelihood of 0.2, and it is subject to an SNMP buffer overflow attack at a likelihood of 0.1. This switch has an impact rating of 90 and has no current controls in place. You are 75% certain of the assumptions and data.</p> <p>b) Server WebSrv6 hosts a company Web site and performs e-commerce transactions. It has a Web server version that can be attacked by sending it invalid Unicode values. The likelihood of that attack is estimated at 0.1. The server has been assigned an impact value of 100, and a control has been implanted that reduces the impact of vulnerability by 75%. You are 80% certain of the assumptions and data.</p> <p>c) Operators use an MGMT45 control console to monitor operations in the server room. It has no passwords and is susceptible to unlogged misuse by the operators. Estimates show the likelihood of the misuse is 0.1. There are no controls in place on this asset; it has an impact rating of 5. You are 90% certain of the assumptions and data.</p>
Ans.	<p>First, we will calculate the risk of vulnerability by using the formula:</p> $Rr = (Lv \times I)(1 - Rc + U)$ <p>Switch L47 vulnerability 1 = $(0.2 \times 90)(1 - 0 + 0.25)$ = 22.5</p> <p>Switch L47 vulnerability 2 = $(0.1 \times 90)(1 - 0 + 0.25)$ = 11.25</p> <p>Server WebSrv6 vulnerability 3 = $(0.1 \times 100)(1 - 0.75 + 0.2)$ = 4.5</p> <p>MGMT45 control console vulnerability 4 = $(0.1 \times 5)(1 - 0 + 0.1)$ = 0.55</p> <p>Therefore, the vulnerability of Switch L47 would need to evaluated first because it has the highest risk rate (22.5) and the MGMT45 control console would be evaluated last because it has the lowest risk rate (0.55).</p>
46.	<p>Consider the information stored on your personal computer. For each of the terms listed, find an example and document it: threat, threat agent, vulnerability, exposure, risk, attack, and exploit.</p>
Ans.	<p>a. Threat: Theft of Media</p> <p>b. Threat Agent: Hacker</p> <p>c. Vulnerability: Unprotected system port</p> <p>d. Exposure: Using a website monitored by malicious hackers, reveals a vulnerability – i.e. Unprotected system port</p> <p>e. Risk: Low level risk – The probability that theft of media will occur is low.</p> <p>f. Attack: Hacker is made aware of system vulnerability (unprotected system port) by monitoring the website mediamadness.com. The hacker then navigates to and enters the exposed port; the hacker continues to steal media files from the user's computer. This results in the user experiencing a loss.</p> <p>g. Exploit: Hacker uses software tools to gain access to the unprotected system port; gaining access to the user's computer.</p>
47.	<p>What are the general guidelines for secure coding in an application as per Ministry of Electronics and Information Technology, Government of India?</p>
Ans.	<p>1. All your web-applications should be security Audited initially (for Web-application/mobile apps)</p> <ul style="list-style-type: none"> • In every two years • Or whenever new module/page is added or modified or functionality is changed

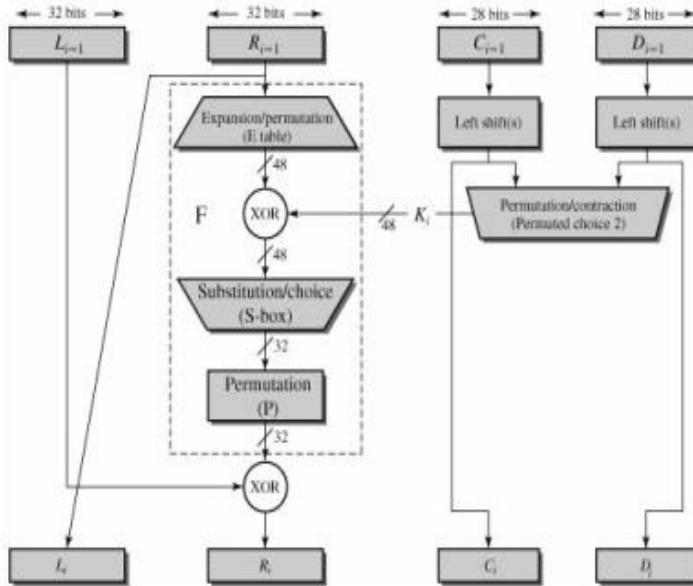
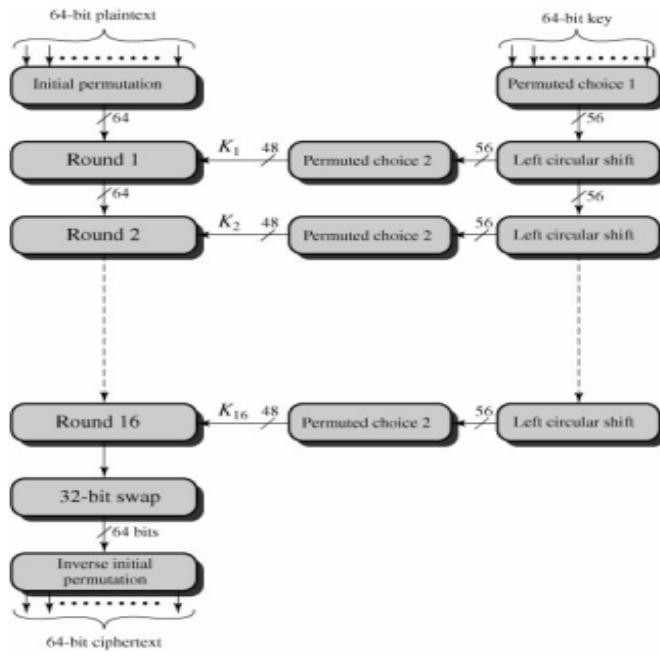
	<p>2. In all web-applications/mobile-apps incorporate security requirements at the design and development phases.</p> <p>3. Ensure that web-applications are deployed on hardened servers/infrastructures.</p> <p>4. All components on server should be hardened and latest stable (non-vulnerable) versions should be upgraded.</p> <p>5. All server environment/infrastructure should be configured for least privileged access, at all layers.</p> <p>6. Effectively monitor system for any changes or intrusion.</p> <p>7. Configure system logs on server [e.g.: Web-Access logs, Application Logs, Security Logs etc.]</p> <p>8. Incorporate proper security advisories across all layers of infrastructure and servers.</p> <p>9. Ensure proper backups of system/server/devices content/logs on a segregated server (preferable on disconnected server or storage devices)</p> <p>10. Whenever any suspicious/intrusion incident is detected :</p> <ul style="list-style-type: none"> • Block the site for public access • Report incident to Incident handling agency • DO NOT CHANGE ARTIFACTS
48.	What do you mean phishing attack? How it can be performed by attacker? Explain.
Ans.	<p>"Phishing" refers to an attempt to steal sensitive information, typically in the form of usernames, passwords, credit card numbers, bank account information or other important data in order to utilize or sell the stolen information. By masquerading as a reputable source with an enticing request, an attacker lures in the victim in order to trick them, similarly to how a fisherman uses bait to catch a fish.</p> <pre> graph LR Attacker[Attacker] -- "1" --> Email[Email] Email --> Victim[Victim] Victim -- "2" --> Phishing[Phishing Website] Phishing -- "3" --> Credentials[Attacker collects victim's credentials] Credentials --> Legitimate[Legitimate Website] Legitimate -- "4" --> Attacker </pre>
49.	What are the possible results of an attack on a computer network?
Ans.	<p>There are some of the possible results of an attack on a computer network:</p> <ol style="list-style-type: none"> 1. Loss or corruption of sensitive data that is essential for a company's survival and success 2. Diminished reputation and trust among customers 3. The decline in value with shareholders 4. Reduced brand value 5. Reduction in profits
50.	What are the types of password attacks? What can a systems administrator do to protect against them?
Ans.	<p>Following are the password attacks:</p> <ol style="list-style-type: none"> 1. Brute force attack: In a brute force attack, a hacker uses a computer program to login to a user's account with all possible password combinations. Moreover, brute force accounts don't start at random; instead, they start with the easiest-to-guess passwords. 2. Dictionary Attack: Conversely, a dictionary attack allows hackers to employ a program which cycles through common words. A brute force attack goes letter by letter, whereas a dictionary attack only tries possibilities most likely to succeed. Also, dictionary attacks rely on a few key factors of users' psychology. For example, users tend to pick short passwords and base their passwords off common words. So a dictionary attack starts with those words and variations (adding numbers at the end, replacing letters with numbers, etc.). 3. Keylogger Attack: Keylogger attacks install a program on users' endpoints to track all of a users' keystrokes. So as the user types in their usernames and passwords, the hackers record them for use later. This technically falls under the category of malware or a digital virus, so it must first infect the users' endpoints (often through a phishing download). 4. Traffic interception: In this attack, the cybercriminal uses software such as packet sniffers to monitor network traffic and capture passwords as they're passed. Similar to eavesdropping or tapping a phone line, the software monitors and captures critical information. Obviously, if that information—such as passwords—is unencrypted, the task is easier. But even encrypted information may be decryptable, depending on the strength of the encryption method used. <p>Strong passwords are usually the first defence against password attacks. The latest NIST guidelines recommend easy to remember/hard to guess passwords. A good mix of upper and lowercase characters, numbers, and special characters can help. Even better, avoid use of common words and common phrases. Definitely avoid site-specific words (including the name of the app you're logging into in the password, for instance). NIST also recommends checking passwords against a dictionary of known poor passwords. Employee education is also important. One of the best defences against social engineering tactics is teaching users the techniques hackers use and how to recognize them.</p>
51.	The study of cryptography and cryptanalysis together are called _____.

Ans.	cryptology
52.	Which of the following algorithms are block ciphers: a) DES b) AES c) RSA d) All of these
Ans.	d) All of these
53.	In public key cryptography, both sender and receiver share a common secret key. (True/False)
Ans.	False
54.	The number of round functions in DES algorithm are _____.
Ans.	16
55.	The key length in 3DES algorithm is _____.
Ans.	192 bits
56.	AES algorithm uses _____ rounds and _____ subkeys.
Ans.	10, 44
57.	The MD5 function is a cryptographic algorithm that takes an input of arbitrary length and produces a message digest that is _____ bits long.
Ans.	128
58.	What is the block size in SHA-512 algorithm?
Ans.	1024 bit
59.	Which of the following operation is NOT performed by AES: a) Left Circular Shift b) Mix Columns c) Substitute Bytes d) Add Round Key
Ans.	a) Left Circular Shift
60.	In MAC, two communicating parties, say X and Y, share the same secret key?(True/False)
Ans.	True
61.	Which of the following parameter is available in X.509 certificate format: a) Issuer Name b) Validity Period c) Public Key Info d) All of these
Ans.	d) All of these
62.	In RSA the _____ key of a receiver is used to encrypt messages.
Ans.	public
63.	What are the numbers of possible keys for a key of length 128 bits?
Ans.	2^{128}
64.	Which among the following is an advantage of modern cryptography? a) Analyzed by best minds b) Low cost in implementation c) Can work over images, not just text d) All of the above
Ans.	d) All of the above
65.	When A wants to communicate with B using symmetric key algorithms, how many keys are needed?
Ans.	1
66.	What is the table size (in bits) to implement the S-Box? a) 8 b) 2^8 c) 2^{11} d) 2^{12}
Ans.	c) 2^{11}
67.	Which among the following is true of 3DES (compared to DES)? a) Less secure b) slower c) Both a and b d) None of these
Ans.	b) slower
68.	In AES-128, a set contains how many words? a) 4 b) 8 c) 12 d) 16
Ans.	a) 4
69.	If the message size is 1000 bits, for a given hash value, how many messages in the message space map to this hash value? Note that the hash size is 160 bits. a) 2^{1000} b) 2^{840} c) 2^{160} d) 840
Ans.	b) 2^{840}
70.	How many number of 64bit buffer registers are used in SHA-512 algorithm.
Ans.	8
71.	Define cryptography? Write the application of cryptography.
Ans.	Cryptography, which comes from the Greek words kryptos, meaning "hidden," and graphein, meaning "to write," is the process of making and using codes to secure the transmission of information. There are lots of benefits of cryptography in the modern world and a few of them are: 1. Chip based payment cards 2. Computer and other passwords 3. E-commerce 4. Defence communications 5. Digital Currencies 6. Designing protocols 7. Data authenticity
72.	Differentiate between the public key and secret key cryptography.
Ans.	In secret key cryptography, both sender and receiver share a common secret key; the same secret key is used for both encryption and decryption. This form of cryptography is also known as symmetric key cryptography. In public key cryptography algorithms, two distinct keys forming a key pair are used. The encryption key or public key and the decryption key or private key. The public key of a user is used to encrypt messages to that

	user. It is intended to be known to the outside world. The corresponding private key, however should not be revealed to anyone. It is the private key of the recipient that is used to decrypt the message. This form of cryptography is also known as Asymmetric Key Cryptography.	
73.	Define the following terms: a) Encryption Decryption	b.) Decryption
Ans.	a) Encryption: The process of converting from plaintext to ciphertext is known as enciphering or encryption. b) Decryption: The process of restoring the plaintext from the ciphertext is known deciphering or decryption.	
74.	Define the following terms: a) Plain text b) Cipher text	
Ans.	An original message is known as the plaintext, while the coded message is called the ciphertext.	
75.	What is substitution cipher? Write the name of different substitution cipher methods?	
Ans.	A substitution technique is one in which the letters of plaintext are replaced by other letters or by numbers or symbols. If the plaintext is viewed as a sequence of bits, then substitution involves replacing plaintext bit patterns with ciphertext bit patterns. Substitution cipher methods are Caesar cipher, Polyalphabetic cipher, Hill Cipher, One time pad.	
76.	What is transposition cipher? Write the name of different transposition cipher methods?	
Ans.	A very different kind of mapping is achieved by performing some sort of permutation on the plaintext letters. This technique is referred to as a transposition cipher. The simplest such cipher is the rail fence technique, in which the plaintext is written down as a sequence of diagonals and then read off as a sequence of rows.	
77.	Consider a Hill Cipher using a block size of 2(m=2). Let K = 3 7 15 12 The plaintext is [H I]. Calculate the corresponding cipher text?	
Ans.	The numeric equivalent of the given plaintext block P is = [7 8] Encryption C = (P*K) mod 26 $\begin{matrix} P * K = & 7 & 8 & * & 3 & 7 \\ & & & & 15 & 12 \end{matrix}$ $\begin{matrix} C = & [141 & 145] & \text{mod} & 26 \\ & = & [11 & 15] \end{matrix}$ The corresponding cipher text C = [L P]	
78.	Define message authentication? What are the different types of message authentication?	
Ans.	Message authentication is a procedure to verify that received messages come from the alleged source and have not been altered. Types of message authentication: 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. Hash function: A function that maps a message of any length into a fixed-length hash value, which serves as the authenticator.	
79.	What is digital signature? Write the types of digital signature?	
Ans.	Message authentication protects two parties who exchange messages from any third party. However, it does not protect the two parties against each other. Several forms of dispute between the two are possible. In situations where there is not complete trust between sender and receiver, something more than authentication is needed. The most attractive solution to this problem is the digital signature. The digital signature is analogous to the handwritten signature. It must have the following properties: a. It must verify the author and the date and time of the signature. b. It must authenticate the contents at the time of the signature. c. It must be verifiable by third parties, to resolve disputes. There are two approaches to digital signature: i. RSA Approach ii. DSS Approach	
80.	What do you mean by X.509 standard? Explain.	
Ans.	X.509 defines a framework for the provision of authentication services by the X.500 directory to its users. The directory may serve as a repository of public-key certificates. Each certificate contains the public key of a user and is signed with the private key of a trusted certification authority. In addition, X.509 defines alternative authentication protocols based on the use of public-key certificates. X.509 is an important standard because the certificate structure and authentication protocols defined in X.509 are used in a variety of contexts. For example, the X.509 certificate format is used in S/MIME, IP Security, and SSL/TLS and SET. The heart of the X.509 scheme is the public-key certificate associated with each user. These user certificates are assumed to be created by some trusted certification authority (CA) and placed in the directory by the CA or by the user. The directory server itself is not responsible for the creation of public keys or for the certification function; it merely provides an easily accessible location for users to obtain certificates.	
81.	Explain the frame format of X.509 standard?	
Ans.	The X.509 standard contains the following fields: Version: Differentiates among successive versions of the certificate format; the default is version 1. If the Issuer Unique Identifier or Subject Unique Identifier are present, the value must be version 2. If one or more extensions are present, the version must be version 3.	

	<p>Serial number: An integer value, unique within the issuing CA, that is unambiguously associated with this certificate.</p> <p>Signature algorithm identifier: The algorithm used to sign the certificate, together with any associated parameters. Because this information is repeated in the Signature field at the end of the certificate, this field has little, if any, utility.</p> <p>Issuer name: X.500 name of the CA that created and signed this certificate.</p> <p>Period of validity: Consists of two dates: the first and last on which the certificate is valid.</p> <p>Subject name: The name of the user to whom this certificate refers. That is, this certificate certifies the public key of the subject who holds the corresponding private key.</p> <p>Subject's public-key information: The public key of the subject, plus an identifier of the algorithm for which this key is to be used, together with any associated parameters.</p> <p>Issuer unique identifier: An optional bit string field used to identify uniquely the issuing CA in the event the X.500 name has been reused for different entities.</p> <p>Subject unique identifier: An optional bit string field used to identify uniquely the subject in the event the X.500 name has been reused for different entities.</p> <p>Extensions: A set of one or more extension fields. Extensions were added in version 3.</p> <p>Signature: Covers all of the other fields of the certificate; it contains the hash code of the other fields, encrypted with the CA's private key. This field includes the signature algorithm identifier.</p>
82.	Differentiate between SHA-1 and SHA-2.
Ans.	The Secure Hash Algorithm (SHA) was developed by the National Institute of Standards and Technology (NIST) and published as a federal information processing standard (FIPS 180) in 1993; a revised version was issued as FIPS 180-1 in 1995 and is generally referred to as SHA-1. SHA-1 produces a hash value of 160 bits. In 2002, NIST produced a revised version of the standard, FIPS 180-2, that defined three new versions of SHA, with hash value lengths of 256, 384, and 512 bits, known as SHA-256, SHA-384, and SHA-512. These new versions have the same underlying structure and use the same types of modular arithmetic and logical binary operations as SHA-1.
83.	What is Public Key Infrastructure (PKI)? Explain.
Ans.	<p>Public-key Infrastructure (PKI) is an integrated system of software, encryption methodologies, protocols, legal agreements, and third-party services that enables users to communicate securely. PKI systems are based on public-key cryptosystems and include digital certificates and certificate authorities (CAs). A typical PKI solution protects the transmission and reception of secure information by integrating the following components:</p> <p>A certificate authority (CA), which issues, manages, authenticates, signs, and revokes users' digital certificates, which typically contain the user name, public key, and other identifying information.</p> <p>A registration authority (RA), which operates under the trusted collaboration of the certificate authority and can handle day-to-day certification functions, such as verifying registration information, generating end-user keys, revoking certificates, and validating user certificates.</p>
84.	Differentiate between DES and Triple DES.
Ans.	<p>The most widely used encryption scheme is based on the Data Encryption Standard (DES) adopted in 1977 by the National Bureau of Standards, now the National Institute of Standards and Technology (NIST). The algorithm itself is referred to as the Data Encryption Algorithm (DEA). For DES, data are encrypted in 64-bit blocks using a 64-bit key. The algorithm transforms 64-bit input in a series of steps into a 64-bit output. The same steps, with the same key, are used to reverse the encryption. DES works on bits, or binary numbers--the 0s and 1s common to digital computers. DES is a block cipher--meaning it operates on plaintext blocks of a given size (64-bits) and returns ciphertext blocks of the same size. Thus DES results in a permutation among the 2⁶⁴ possible arrangements of 64 bits, each of which may be either 0 or 1. Each block of 64 bits is divided into two blocks of 32 bits each.</p> <p>3DES was created to provide a level of security far beyond that of standard DES. 3DES uses three 64-bit keys for an overall key length of 192 bits. 3DES encryption is the same as that of standard DES, repeated three times. 3DES can be employed using two or three keys and a combination of encryption or decryption for additional security. The most common implementations involve encrypting and/or decrypting with two or three different keys. 3DES employs forty-eight rounds in its encryption computation, generating ciphers that are approximately 256 times stronger than standard DES ciphers but require only three times longer to process.</p>
85.	What do you mean by Kerberos? Explain.

Ans.	<p>Kerberos is an authentication service developed as part of Project Athena at MIT. The problem that Kerberos addresses is this:</p> <p>Assume an open distributed environment in which users at workstations wish to access services on servers distributed throughout the network. We would like for servers to be able to restrict access to authorized users and to be able to authenticate requests for service. In this environment, a workstation cannot be trusted to identify its users correctly to network services. In particular, the following three threats exist:</p> <ul style="list-style-type: none"> i. A user may gain access to a particular workstation and pretend to be another user operating from that workstation. ii. A user may alter the network address of a workstation so that the requests sent from the altered workstation appear to come from the impersonated workstation. iii. A user may eavesdrop on exchanges and use a replay attack to gain entrance to a server or to disrupt operations. <p>In any of these cases, an unauthorized user may be able to gain access to services and data that he or she is not authorized to access. Rather than building in elaborate authentication protocols at each server, Kerberos provides a centralized authentication server whose function is to authenticate users to servers and servers to users.</p>
86.	How Caesar cipher works? Explain the encryption and decryption process of Caesar cipher?
Ans.	<p>The earliest known use of a substitution cipher, and the simplest, was by Julius Caesar. The Caesar cipher involves replacing each letter of the alphabet with the letter standing three places further down the alphabet.</p> <p>For example:</p> <pre style="margin-left: 40px;">plain: meet me after the toga party cipher: PHHW PH DIWHU WKH WRJD SDUWB</pre> <p>A shift may be of any amount, so that the general Caesar algorithm is</p> $C = E(k, p) = (p + k) \text{ mod } 26$ <p>where k takes on a value in the range 1 to 25.</p> <p>The decryption algorithm is simply</p> $p = D(k, C) = (C - k) \text{ mod } 26$ <p>This type of substitution is based on a monoalphabetic substitution, because it only uses one alphabet. More advanced substitution ciphers use two or more alphabets, and are referred to as polyalphabetic substitutions.</p>
87.	Differentiate between monoalphabetic and polyalphabetic cipher with suitable examples.
Ans.	<p>The Caesar cipher substitution is based on a monoalphabetic substitution, because it only uses one alphabet. More advanced substitution ciphers use two or more alphabets, and are referred to as polyalphabetic substitutions.</p> <p>In polyalphabetic cipher, the ciphertext corresponding to a particular character in the plain text is not fixed. An advanced type of substitution cipher that uses a simple polyalphabetic code is the Vigenère cipher. The cipher is implemented using the Vigenère square (or table), which is made up of twenty-six distinct cipher alphabets.</p>
88.	What do you mean by one time pad encryption process? Why it is unbreakable? Explain.
Ans.	<p>Mauborgne suggested using a random key that is as long as the message, so that the key need not be repeated. In addition, the key is to be used to encrypt and decrypt a single message, and then is discarded. Each new message requires a new key of the same length as the new message. Such a scheme, known as a one-time pad, is unbreakable. It produces random output that bears no statistical relationship to the plaintext. Because the ciphertext contains no information whatsoever about the plaintext, there is simply no way to break the code. The security of the one-time pad is entirely due to the randomness of the key. If the stream of characters that constitute the key is truly random, then the stream of characters that constitute the ciphertext will be truly random. Thus, there are no patterns or regularities that a cryptanalyst can use to attack the ciphertext.</p>
89.	Explain the DES algorithm encryption process with block diagram.
Ans.	<p>The most widely used encryption scheme is based on the Data Encryption Standard (DES) adopted in 1977 by the National Bureau of Standards, now the National Institute of Standards and Technology (NIST). The algorithm itself is referred to as the Data Encryption Algorithm (DEA). For DES, data are encrypted in 64-bit blocks using a 64-bit key. The algorithm transforms 64-bit input in a series of steps into a 64-bit output. The same steps, with the same key, are used to reverse the encryption. DES works on bits, or binary numbers--the 0s and 1s common to digital computers. DES is a block cipher--meaning it operates on plaintext blocks of a given size (64-bits) and returns ciphertext blocks of the same size. Thus DES results in a permutation among the 2^{64} possible arrangements of 64 bits, each of which may be either 0 or 1. Each block of 64 bits is divided into two blocks of 32 bits each.</p>



90. How 3DES is more powerful than DES? Explain.

Ans. 3DES was created to provide a level of security far beyond that of standard DES. 3DES uses three 64-bit keys for an overall key length of 192 bits. 3DES encryption is the same as that of standard DES, repeated three times. 3DES can be employed using two or three keys and a combination of encryption or decryption for additional security. The most common implementations involve encrypting and/or decrypting with two or three different keys. 3DES employs forty-eight rounds in its encryption computation, generating ciphers that are approximately 256 times stronger than standard DES ciphers but require only three times longer to process.

91. What do you mean Advanced Encryption Standard algorithm? What are the different operations performed under AES algorithm?

Ans. The Advanced Encryption Standard (AES) was published by NIST (National Institute of Standards and Technology) in 2001. AES is a symmetric block cipher that is intended to replace DES as the approved standard for a wide range of applications. AES is a block cipher intended to replace DES for commercial applications. It uses a 128-bit block size and a key size of 128, 192, or 256 bits.

- No. of rounds: 10
- No. of subkeys: 44
- Each sub key size: 32bit/1word/4 bytes
- Each round uses 4 sub-keys.
- Pre-round calculation uses 4 sub-keys.
- Size of one word is 32bits or 4bytes.

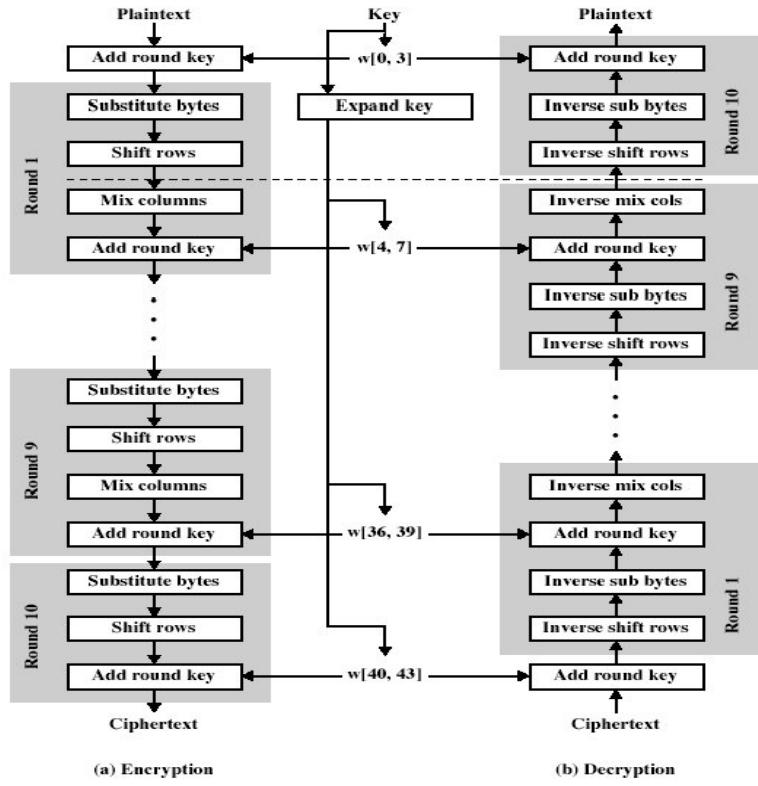
Four different stages are used, one of permutation and three of substitution:

1. Substitute bytes: Uses an S-box to perform a byte-by-byte substitution of the block
2. ShiftRows: A simple permutation

3. MixColumns: A substitution that makes use of arithmetic over GF(28)
 4. AddRoundKey: A simple bitwise XOR of the current block with a portion of the expanded key

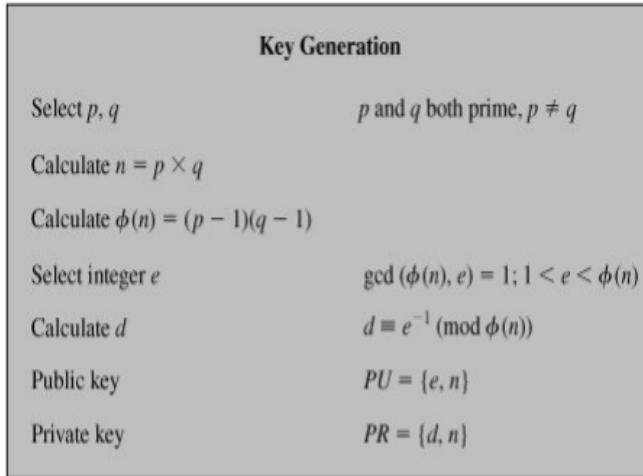
92. Explain the encryption and decryption process of AES algorithm with block diagram.

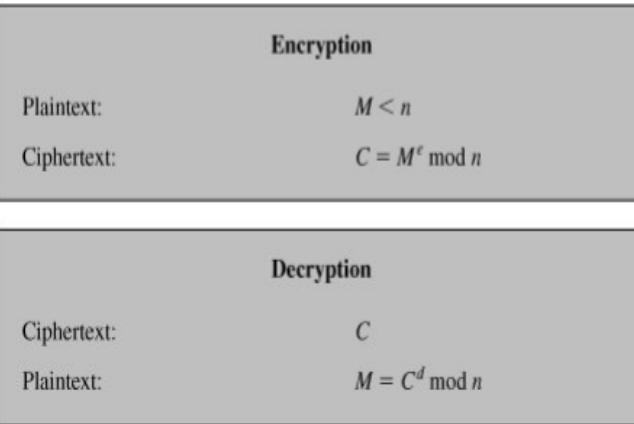
Ans.



93. Explain the RSA algorithm with suitable example.

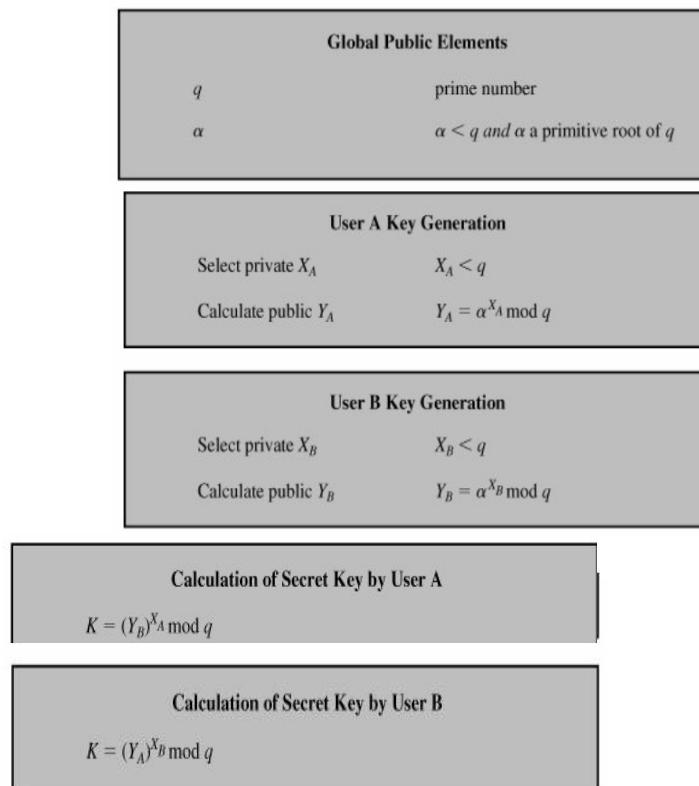
Ans. Developed in 1977 by Ron Rivest, Adi Shamir, and Len Adleman at MIT and first published in 1978. The Rivest-Shamir-Adleman (RSA) scheme has since that time reigned supreme as the most widely accepted and implemented general-purpose approach to public-key encryption. The RSA scheme is a block cipher in which the plaintext and ciphertext are integers between 0 and $n-1$ for some n .





94. What is the purpose of Diffie Hellman key exchange algorithm? Write the process to generate the key using this algorithm. Explain.

Ans. The purpose of the algorithm is to enable two users to securely exchange a key that can then be used for subsequent encryption of messages. The algorithm itself is limited to the exchange of secret values



95. What is a hash function? How hash function is different from message authentication code? Explain.

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

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. The hash code is a function of all the bits of the message and provides an error-detection capability. A change to any bit or bits in the message results in a change to the hash code.

96. Explain the processing of single 512bit block using MD5 algorithm.

Ans. The MD5 function is a cryptographic algorithm that takes an input of arbitrary length and produces a message digest that is 128 bits long. MD5 was designed by well-known cryptographer Ronald Rivest in 1991.

The MD5 algorithm first divides the input in blocks of 512bits each.

Step 1: Append padding bits, starting from 1 then put 0's.(i.e. 10000.....)

Step 2: The remaining bits are filled up with 64 bits representing the length of the original message.

Step 3: Initialize the buffer. Each of size 32bits. Four buffers are required(A,B,C,D).

Step 4: Process each 512bits block.

Step 5: Output i.e. message digest stored in buffer.

The processing of a message block consists of four similar stages, termed rounds; each round is composed of 16 similar operations based on a non-linear function F, modular addition, and left rotation. There are four possible functions; a different one is used in each round:

$$\begin{aligned} F(B, C, D) &= (B \wedge C) \vee (\neg B \wedge D) \\ G(B, C, D) &= (B \wedge D) \vee (C \wedge \neg D) \\ H(B, C, D) &= B \oplus C \oplus D \\ I(B, C, D) &= C \oplus (B \vee \neg D) \end{aligned}$$

97. Explain the working of SHA-512 algorithm.

Ans. The algorithm takes as input a message with a maximum length of less than 2128 bits and produces as output a 512-bit message digest. The input is processed in 1024-bit blocks.

Step 1: Append padding bits. The message is padded so that its length is congruent to 896 modulo 1024 . Padding is always added, even if the message is already of the desired length. Thus, the number of padding bits is in the range of 1 to 1024. The padding consists of a single 1-bit followed by the necessary number of 0-bits.

Step 2: Append length. A block of 128 bits is appended to the message. This block is treated as an unsigned 128-bit integer (most significant byte first) and contains the length of the original message (before the padding). The outcome of the first two steps yields a message that is an integer multiple of 1024 bits in length. The expanded message is represented as the sequence of 1024-bit blocks M₁, M₂, ..., M_N, so that the total length of the expanded message is N x 1024 bits.

Step 3: Initialize hash buffer. A 512-bit buffer is used to hold intermediate and final results of the hash function. The buffer can be represented as eight 64-bit registers (a, b, c, d, e, f, g, h). These registers are initialized to the following 64-bit integers (hexadecimal values):

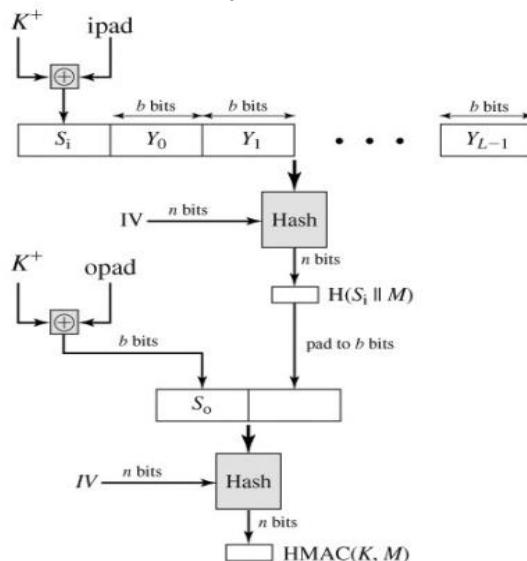
a = 6A09E667F3BCC908
 b = BB67AE8584CAA73B
 c = 3C6EF372FE94F82B
 d = A54FF53A5F1D36F1
 e = 510E527FADE682D1
 f = 9B05688C2B3E6C1F
 g = 1F83D9ABFB41BD6B
 h = 5BE0CDI9137E2179

Step 4: Process message in 1024-bit (128-word) blocks. The heart of the algorithm is a module that consists of 80 rounds.

Step 5: Output. After all N 1024-bit blocks have been processed, the output from the Nth stage is the 512-bit message digest.

98. What is HMAC? Explain the HMAC structure with block diagram.

Ans. A hash function such as SHA 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, and is used in other Internet protocols, such as SSL. HMAC has also been issued as a NIST standard. HMAC consists of twin benefits of Hashing and MAC, and thus is more secure than any other authentication codes.



99. Explain the signing and verifying process of Digital Signature Algorithm.

Ans. Message authentication protects two parties who exchange messages from any third party. However, it does not protect the two parties against each other. Several forms of dispute between the two are possible. In situations where there is not complete trust between sender and receiver, something more than authentication is needed. The most attractive solution to this problem is the digital signature. The digital signature is analogous to the

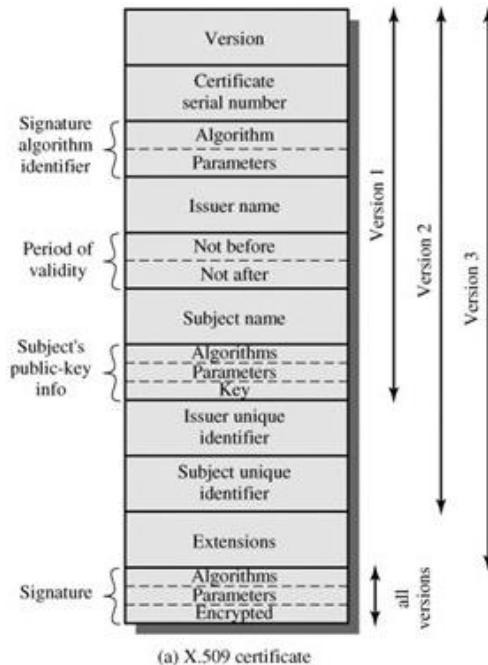
handwritten signature. It must have the following properties:

- It must verify the author and the date and time of the signature.
- It must authenticate the contents at the time of the signature.
- It must be verifiable by third parties, to resolve disputes.

Global Public-Key Components	
P	prime number where $2^L - 1 < p < 2^L$ for $512 \leq L \leq 1024$ and L a multiple of 64; i.e., bit length of between 512 and 1024 bits in increments of 64 bits
q	prime divisor of $(p - 1)$, where $2^{159} < q < 2^{160}$; i.e., bit length of 160 bits
g	$= h^{(p-1)/q} \pmod{p}$, where h is any integer with $1 < h < (p - 1)$ such that $h^{(p-1)/q} \pmod{p} > 1$
User's Private Key	
x	random or pseudorandom integer with $0 < x < q$
User's Public Key	
y	$= g^x \pmod{p}$
User's Per-Message Secret Number	
k	random or pseudorandom integer with $0 < k < q$
Signing	
r	$= (g^k \pmod{p}) \pmod{q}$
s	$= [k^{-1} (H(M) + xr)] \pmod{q}$
Signature = (r, s)	
Verifying	
w	$= (s')^{-1} \pmod{q}$
u1	$=[H(M')w] \pmod{q}$
u2	$=(r')w \pmod{q}$
v	$=[(g^{u_1} y^{u_2}) \pmod{p}] \pmod{q}$
TEST: $v = r'$	
M	= message to be signed
H(M)	= hash of M using SHA-1
M', r', s'	= received versions of M, r, s

100. Explain the X.509 certificate format.

Ans.



a) Version: Differentiates among successive versions of the certificate format; the default is version 1. If the Issuer Unique Identifier or Subject Unique Identifier are present, the value must be version 2. if one or more

	<p>extensions are present, the version must be version 3.</p> <p>b) Serial number: An integer value, unique within the issuing CA, that is unambiguously associated with this certificate.</p> <p>c) Signature algorithm identifier: The algorithm used to sign the certificate, together with any associated parameters. Because this information is repeated in the Signature field at the end of the certificate, this field has little, if any, utility.</p> <p>d) Issuer name: X.500 name of the CA that created and signed this certificate.</p> <p>e) Period of validity: Consists of two dates: the first and last on which the certificate is valid.</p> <p>f) Subject name: The name of the user to whom this certificate refers. That is, this certificate certifies the public key of the subject who holds the corresponding private key.</p> <p>g) Subject's public-key information: The public key of the subject, plus an identifier of the algorithm for which this key is to be used, together with any associated parameters.</p> <p>h) Issuer unique identifier: An optional bit string field used to identify uniquely the issuing CA in the event the X.500 name has been reused for different entities.</p> <p>i) Subject unique identifier: An optional bit string field used to identify uniquely the subject in the event the X.500 name has been reused for different entities.</p> <p>j) Extensions: A set of one or more extension fields. Extensions were added in version 3.</p> <p>k) Signature: Covers all of the other fields of the certificate; it contains the hash code of the other fields, encrypted with the CA's private key. This field includes the signature algorithm identifier.</p>
101.	The _____ is actually an IETF version of _____ a)TLS;TSS b) SSL;TLS c) TLS;SSL d) SSL;SLT
Ans.	c) TLS;SSL
102.	The _____ protocol provides security at transport layer.
Ans.	SSL and TLS
103.	SSL provides _____. a) message integrity b) confidentiality c) compression d) all of the above
Ans.	d) all of the above
104.	Protocol for the email security is _____
Ans.	PGP
105.	A _____ protocol is a collection of protocols designed by IETF to provide security for a packet at the network layer. a) IPSec b) SSL c) PGP d) None of the above
Ans.	a) IPSec
106.	The _____ mode is normally used when we need host-to-host protection of data. a) transport b) tunnel c) Both A and B d) Neither A and nor B
Ans	c) Both A and B
107.	In tunnel mode, IPSec protects the _____ a) Entire IP packet b) IP header c) IP payload d) IP trailer
Ans.	a) Entire IP packet
108.	IPSec is designed to provide security at the _____ a) Transport layer b) Network layer c) Application layer d) Session layer
Ans.	b) Network layer
109.	Which component is included in IP security? a) Authentication Header (AH) b) Encapsulating Security Payload (ESP) c) Internet key Exchange (IKE) d) All of the mentioned
Ans.	d) All of the mentioned
110.	WPA2 is used for security in _____.
Ans.	Wi-Fi
111.	PGP encrypts data by using a block cipher called _____ a) International data encryption algorithm b) Private data encryption algorithm c) Internet data encryption algorithm d) Local data encryption algorithm

Ans.	a) International data encryption algorithm
112.	Network layer firewall has two sub-categories as _____.
Ans.	State full firewall and stateless firewall
113.	A proxy firewall filters at _____. a) Physical layer b) Data link layer c) Network layer d) Application layer
Ans.	d) Application layer
114.	A stateful firewall maintains a _____ which is a list of active connections. a) Routing table b) Bridging table c) State table d) Connection table
Ans.	a) Routing table
115.	Which of the following is not a software firewall? a) Windows Firewall b) Outpost Firewall Pro c) Endian Firewall d) Linksys Firewall
Ans.	d) Linksys Firewall
116.	A firewall protects which of the following attacks? a) Phishing b) Dumpster diving c) Denial of Service (DoS) d) Shoulder surfing
Ans.	c) Denial of Service (DoS)
117.	Packet filtering firewalls are deployed on _____. a) routers b) switches c) hubs d) repeaters
Ans.	a) routers
118.	The _____ protocol is an open encryption and security specification designed to protect credit card transactions on the Internet.
Ans.	SET
119.	The RFC _____ Specification of key management capabilities.
Ans.	2408
120.	Both PGP and S/MIME make use of an encoding technique referred to as _____ conversion
Ans.	radix-64
121.	Which security protocols are predominantly used in Web-based electronic commerce?
Ans.	Many Web-based technologies make use of the S-HTTP, SET, SSL, SSH-2, and IPSec protocols.
122.	Which security protocols are used to protect e-mail?
Ans.	E-mail security is most often provided using the S/MIME, PEM, and PGP protocols.
123.	IPSec can be used in two modes. What are they?
Ans.	IPSec is provisioned using the transport and tunnel modes.
124.	What are the five principal services provided by PGP?
Ans.	The following are the services offered by PGP: 1. Authentication 2. Confidentiality 3. Compression 4. Email Compatibility 5. Segmentation
125.	What is the utility of a detached signature?
Ans.	A detached signature is useful in several contexts. A user may wish to maintain a separate signature log of all messages sent or received. A detached signature of an executable program can detect subsequent virus infection. Finally, detached signatures can be used when more than one party must sign a document, such as a legal contract. Each person's signature is independent and therefore is applied only to the document. Otherwise, signatures would have to be nested, with the second signer signing both the document and the first signature, and so on.
126.	Why does PGP generate a signature before applying compression?
Ans.	a) It is preferable to sign an uncompressed message so that one can store only the uncompressed message together with the signature for future verification. If one signed a compressed document, then it would be necessary either to store a compressed version of the message for later verification or to recompress the message when verification is required. b) Even if one were willing to generate dynamically a recompressed message for verification, PGP's compression algorithm presents a difficulty. The algorithm is not deterministic; various implementations of the algorithm achieve different tradeoffs in running speed versus compression ratio and, as a result, produce different compressed forms. However, these different compression algorithms are interoperable because any version of the algorithm can correctly decompress the output of any other version. Applying the hash function and signature after compression would constrain all PGP implementations to the same version of the compression algorithm.
127.	What is R64 conversion?
Ans.	R64 converts a raw 8-bit binary stream to a stream of printable ASCII characters. Each group of three octets of binary data is mapped into four ASCII characters.

128.	Why is the segmentation and reassembly function in PGP needed?
Ans.	E-mail facilities often are restricted to a maximum message length.
129.	What is S/MIME?
Ans.	Secure Multipurpose Internet Mail Extensions (S/MIME) builds on the encoding format of the Multipurpose Internet Mail Extensions (MIME) protocol and uses digital signatures based on public key cryptosystems to secure e-mail. S/MIME provides the following functions: Enveloped data, Signed data, Clear-signed data, Signed and enveloped data.
130.	Give examples of applications of IPSec.
Ans.	Secure branch office connection over the internet, Secure remote access over the internet, Establishing extranet and intranet connectivity with partners, Establishing electronic commerce security
131.	What parameters identify an SA and what parameters characterize the nature of a particular SA?
Ans.	Security Associations (SA) are identified by the following three parameters: 1. Security Parameter Index 2. IP Destination Address 3. Security Protocol Identifier The following parameters characterize the nature of a particular SA: 1. Secret Key 2. Encapsulation Mode
132.	What is the difference between transport mode and tunnel mode?
Ans.	Authentication Headers and Encapsulation Security Payloads support two modes of use: transport mode and tunnel mode. Transport mode provides protection, primarily, for upper-layer protocols whereas tunnel mode provides security for the entire IP Packet being transmitted.
133.	What is the difference between an SSL connection and an SSL session?
Ans.	A SSL connection is a transport that provides a suitable type of service. In the case of SSL, such a connection is peer-to-peer and the connections are transient. Furthermore, each connection is associated with one SSL session, which is defined as the association between a client and a server. A session is created by the Handshake Protocol, and it defines a set of cryptographic security parameters which can be shared among multiple connections.
134.	What services are provided by the SSL Record Protocol?
Ans.	The SSL Record Protocol provides two services for SSL connections: 1. Confidentiality: The Handshake Protocol defines a shared secret key that is used for conventional encryption of SSL payloads. 2. Message Integrity: The Handshake Protocol also defines a shared secret key that is used to form a message authentication code (MAC). In SSL Record Protocol application data is divided into fragments. The fragment is compressed and then encrypted MAC (Message Authentication Code) generated by algorithms like SHA (Secure Hash Protocol) and MD5 (Message Digest) is appended. After that encryption of the data is done and in last SSL header is appended to the data.
135.	Write the name of SET participants?
Ans.	SET includes the following participants: <ul style="list-style-type: none">• Cardholder – customer• Issuer – customer financial institution• Merchant• Acquirer – Merchant financial• Certificate authority – Authority which follows certain standards and issues certificates (like X.509V3) to all other participants.
136.	What is the difference between digital signatures and digital certificates?
Ans.	A digital signature is a mathematical technique used to validate the authenticity and integrity of a message, software or digital document. Key Generation Algorithms: Digital signature are electronic signatures, which assures that the message was sent by a particular sender. While performing digital transactions authenticity and integrity should be assured, otherwise the data can be altered or someone can also act as if he was the sender and expect a reply. Signing Algorithms: To create a digital signature, signing algorithms like email programs create a one-way hash of the electronic data which is to be signed. The signing algorithm then encrypts the hash value using the private key (signature key). This encrypted hash along with other information like the hashing algorithm is the digital signature. This digital signature is appended with the data and sent to the verifier. The reason for encrypting the hash instead of the entire message or document is that a hash function converts any arbitrary input into a much shorter fixed length value. This saves time as now instead of signing a long message a shorter hash value has to be signed and moreover hashing is much faster than signing. Signature Verification Algorithms: Verifier receives Digital Signature along with the data. It then uses Verification algorithm to process on the digital signature and the public key (verification key) and generates some value. It also applies the same hash function on the received data and generates a hash value. Then the hash value and the output of the verification algorithm are compared. If they both are equal, then the digital

	<p>signature is valid else it is invalid.</p> <p>Digital certificate is issued by a trusted third party which proves sender's identity to the receiver and receiver's identity to the sender.</p> <p>A digital certificate is a certificate issued by a Certificate Authority (CA) to verify the identity of the certificate holder. The CA issues an encrypted digital certificate containing the applicant's public key and a variety of other identification information. Digital certificate is used to attach public key with a particular individual or an entity.</p> <p>Digital certificate contains:-</p> <ul style="list-style-type: none"> • Name of certificate holder • Serial number which is used to uniquely identify a certificate, the individual or the entity identified by the certificate • Expiration dates • Copy of certificate holder's public key.(used for decrypting messages and digital signatures) • Digital Signature of the certificate issuing authority • Digital certificate is also sent with the digital signature and the message
137.	<p>How to protect the electronic transaction over Internet? Explain.</p> <p>Ans. Secure Electronic Transaction or SET is a system which ensures security and integrity of electronic transactions done using credit cards in a scenario. SET is not some system that enables payment but it is a security protocol applied on those payments. It uses different encryption and hashing techniques to secure payments over internet done through credit cards. SET protocol was supported in development by major organizations like Visa, Mastercard, Microsoft which provided its Secure Transaction Technology (STT) and NetScape which provided technology of Secure Socket Layer (SSL). SET protocol restricts revealing of credit card details to merchants thus keeping hackers and thieves at bay. SET protocol includes Certification Authorities for making use of standard Digital Certificates like X.509 Certificate.</p> <p>SET functionalities :</p> <p>Provide Authentication:</p> <ul style="list-style-type: none"> • Merchant Authentication – To prevent theft, SET allows customers to check previous relationships between merchant and financial institution. Standard X.509V3 certificates are used for this verification. • Customer / Cardholder Authentication – SET checks if use of credit card is done by an authorized user or not using X.509V3 certificates. <p>Provide Message Confidentiality: Confidentiality refers to preventing unintended people from reading the message being transferred. SET implements confidentiality by using encryption techniques. Traditionally DES is used for encryption purpose.</p> <p>Provide Message Integrity: SET doesn't allow message modification with the help of signatures. Messages are protected against unauthorized modification using RSA digital signatures with SHA-1 and some using HMAC with SHA-1.</p>
138.	<p>How to secure the wireless networks? Explain.</p> <p>Ans. There are some ways to secure the wireless networks:</p> <p>a) Change default passwords: Most network devices, including wireless access points, are pre-configured with default administrator passwords to simplify setup. These default passwords are easily available to obtain online, and so provide only marginal protection. Changing default passwords makes it harder for attackers to access a device. Use and periodic changing of complex passwords is your first line of defense in protecting your device.</p> <p>b) Restrict access: Only allow authorized users to access your network. Each piece of hardware connected to a network has a media access control (MAC) address. You can restrict access to your network by filtering these MAC addresses. Consult your user documentation for specific information about enabling these features. You can also utilize the "guest" account, which is a widely used feature on many wireless routers. This feature allows you to grant wireless access to guests on a separate wireless channel with a separate password, while maintaining the privacy of your primary credentials.</p> <p>c) Encrypt the data on your network: Encrypting your wireless data prevents anyone who might be able to access your network from viewing it. There are several encryption protocols available to provide this protection. Wi-Fi Protected Access (WPA), WPA2, and WPA3 encrypt information being transmitted between wireless routers and wireless devices. WPA3 is currently the strongest encryption. WPA and WPA2 are still available; however, it is advisable to use equipment that specifically supports WPA3, as using the other protocols could leave your network open to exploitation.</p> <p>d) Protect your Service Set Identifier (SSID): To prevent outsiders from easily accessing your network, avoid publicizing your SSID. All Wi-Fi routers allow users to protect their device's SSID, which makes it more difficult for attackers to find a network. At the very least, change your SSID to something unique. Leaving it as the manufacturer's default could allow a potential attacker to identify the type of router and possibly exploit any known vulnerabilities.</p> <p>e) Install a firewall: Consider installing a firewall directly on your wireless devices (a host-based firewall), as well as on your home network (a router- or modem-based firewall). Attackers who can directly tap into your wireless network may be able to circumvent your network firewall—a host-based firewall will add a layer of protection to the data on your computer.</p>

	<p>f) Maintain antivirus software: Install antivirus software and keep your virus definitions up to date. Many antivirus programs also have additional features that detect or protect against spyware and adware.</p> <p>g) Use file sharing with caution: File sharing between devices should be disabled when not needed. You should always choose to only allow file sharing over home or work networks, never on public networks. You may want to consider creating a dedicated directory for file sharing and restrict access to all other directories. In addition, you should password protect anything you share. Never open an entire hard drive for file sharing.</p> <p>h) Keep your access point software patched and up to date. The manufacturer of your wireless access point will periodically release updates to and patches for a device's software and firmware. Be sure to check the manufacturer's website regularly for any updates or patches for your device.</p> <p>i) Check your internet provider's or router manufacturer's wireless security options: Your internet service provider and router manufacturer may provide information or resources to assist in securing your wireless network. Check the customer support area of their websites for specific suggestions or instructions.</p> <p>j) Connect using a Virtual Private Network (VPN): Many companies and organizations have a VPN. VPNs encrypt connections at the sending and receiving ends and keep out traffic that is not properly encrypted. If a VPN is available to you, make sure you log onto it any time you need to use a public wireless access point.</p>
--	---

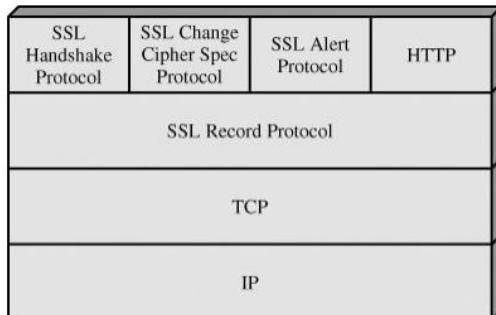
139. What do you mean by SSL protocol? How it works?

Ans. Netscape developed the Secure Sockets Layer (SSL) protocol to use public key encryption to secure a channel over the Internet, thus enabling secure communications. Most popular browsers, including Internet Explorer, use SSL. In addition to providing data encryption, integrity, and server authentication, SSL can, when properly configured, provide client authentication.

The SSL protocol works as follows: during a normal client/server HTTP session, the client requests access to a portion of the Web site that requires secure communications and the server sends a message to the client indicating that a secure connection must be established. The client sends its public key and security parameters. This handshaking phase is complete when the server finds a public key match and sends a digital certificate to the client in order to authenticate itself. Once the client verifies that the certificate is valid and trustworthy, the SSL session is established. Until the client or the server terminates the session, any amount of data can be transmitted securely.

140. Explain the SSL protocol stacks with diagram?

Ans. SSL is designed to make use of TCP to provide a reliable end-to-end secure service. SSL is not a single protocol but rather two layers of protocols.



The SSL Record Protocol provides basic security services to various higher-layer protocols. In particular, the Hypertext Transfer Protocol (HTTP), which provides the transfer service for Web client/server interaction, can operate on top of SSL. Three higher-layer protocols are defined as part of SSL: the Handshake Protocol, The Change Cipher Spec Protocol, and the Alert Protocol. These SSL-specific protocols are used in the management of SSL exchanges.

141. What do you mean by Secured HTTP? Explain.

Ans. Secure HTTP (S-HTTP) is an extended version of Hypertext Transfer Protocol that provides for the encryption of individual messages transmitted via the Internet between a client and server. S-HTTP is the application of SSL over HTTP, which allows the encryption of all information passing between two computers through a protected and secure virtual connection. Unlike SSL, in which a secure channel is established for the duration of a session, S-HTTP is designed for sending individual messages over the Internet and therefore a session for each individual exchange of data must be established. To establish a session, the client and server must have compatible cryptosystems and agree on the configuration. The S-HTTP client then must send the server its public key so that the server can generate a session key. The session key from the server is then encrypted with the client's public key and returned to the client. The client decrypts the key using its private key, and the client and server now possess identical session keys, which they can use to encrypt the messages sent between them. S-HTTP can provide confidentiality, authentication, and data integrity through a variety of trust models and cryptographic algorithms. In addition, this protocol is designed for easy integration with existing HTTP applications and for implementation in conjunction with HTTP.

142. Why is the segmentation and reassembly function in PGP needed?

Ans. E-mail facilities often are restricted to a maximum message length. For example, many of the facilities accessible through the Internet impose a maximum length of 50,000 octets. Any message longer than that must be broken up into smaller segments, each of which is mailed separately. To accommodate this restriction, PGP automatically subdivides a message that is too large into segments that are small enough to send via e-mail. The

	segmentation is done after all of the other processing, including the radix-64 conversion. Thus, the session key component and signature component appear only once, at the beginning of the first segment. Reassembly at the receiving end is required before verifying signature or decryption.
143.	<p>What do you mean by PGP? What are the functions or services provided by PGP? Explain.</p> <p>Ans. PGP is a remarkable phenomenon. Largely the effort of a single person, Phil Zimmermann, PGP provides a confidentiality and authentication service that can be used for electronic mail and file storage applications. The actual operation of PGP, as opposed to the management of keys, consists of five services: authentication, confidentiality, compression, e-mail compatibility, and segmentation.</p> <ul style="list-style-type: none"> • Digital signature: A hash code of a message is created using SHA-1. This message digest is encrypted using DSS or RSA with the sender's private key and included with the message. • Message encryption: A message is encrypted using CAST-128 or IDEA or 3DES with a one-time session key generated by the sender. The session key is encrypted using Diffie Hellman or RSA with the recipient's public key and included with the message. • Compression: A message may be compressed, for storage or transmission, using ZIP. • Email compatibility: To provide transparency for email applications, an encrypted message may be converted to an ASCII string using radix 64 conversions. • Segmentation: To accommodate maximum message size limitations, PGP performs segmentation and reassembly.
144.	Explain the parameters that define an SSL session state?
Ans.	<ol style="list-style-type: none"> 1. Session identifier: This is an arbitrary byte sequence by the server to identify an active or resumable session state. 2. Peer certificate: This is an X509.v3 certificate of the peer. This element of the state may be null. 3. Compression method: This is the algorithm used to compress data prior to the encryption. 4. Cipher spec: This specifies the bulk data encryption (null, AES, etc.) and a hash algorithm (MD5 or SHA-1, etc.) used for MAC calculation. This also defines cryptographic attributes such as the hash size. 5. Master secret: This is a 48-byte secret shared between the client and the server. 6. Is resumable: This is a flag indicating whether the session can be used to initiate new connections.
145.	What are the limitations of existing mail transfer protocol SMTP?
Ans.	<p>MIME is an extension to the RFC 822 framework that is intended to address some of the problems and limitations of the use of SMTP (Simple Mail Transfer Protocol) or some other mail transfer protocol and RFC 822 for electronic mail. Following limitations of the SMTP/822 scheme:</p> <ol style="list-style-type: none"> 1. SMTP cannot transmit executable files or other binary objects. A number of schemes are in use for converting binary files into a text form that can be used by SMTP mail systems, including the popular UNIX UUencode/UUdecode scheme. However, none of these is a standard or even a defacto standard. 2. SMTP cannot transmit text data that includes national language characters because these are represented by 8-bit codes with values of 128 decimal or higher, and SMTP is limited to 7-bit ASCII. 3. SMTP servers may reject mail message over a certain size. 4. SMTP gateways that translate between ASCII and the character code EBCDIC do not use a consistent set of mappings, resulting in translation problems. 5. SMTP gateways to X.400 electronic mail networks cannot handle non-textual data included in X.400 messages. 6. Some SMTP implementations do not adhere completely to the SMTP standards defined in RFC 821. <p>Common problems include:</p> <ul style="list-style-type: none"> • Deletion, addition, or reordering of carriage return and linefeed • Truncating or wrapping lines longer than 76 characters • Removal of trailing white space (tab and space characters) • Padding of lines in a message to the same length • Conversion of tab characters into multiple space characters <p>MIME is intended to resolve these problems in a manner that is compatible with existing RFC 822 implementations.</p>
146.	What do you mean by MIME? Explain the five header field defined in MIME?
Ans.	<p>MIME is an extension to the RFC 822 framework that is intended to address some of the problems and limitations of the use of SMTP (Simple Mail Transfer Protocol) or some other mail transfer protocol and RFC 822 for electronic mail.</p> <p>The MIME specification includes the following elements:</p> <ol style="list-style-type: none"> 1. Five new message header fields are defined, which may be included in an RFC 822 header. These fields provide information about the body of the message. 2. A number of content formats are defined, thus standardizing representations that support multimedia electronic mail. 3. Transfer encodings are defined that enable the conversion of any content format into a form that is protected from alteration by the mail system. <p>The five header fields defined in MIME are as follows:</p> <ul style="list-style-type: none"> • MIME-Version: Must have the parameter value 1.0. This field indicates that the message conforms to

	<p>RFCs 2045 and 2046.</p> <ul style="list-style-type: none"> • Content-Type: Describes the data contained in the body with sufficient detail that the receiving user agent can pick an appropriate agent or mechanism to represent the data to the user or otherwise deal with the data in an appropriate manner. • Content-Transfer-Encoding: Indicates the type of transformation that has been used to represent the body of the message in a way that is acceptable for mail transport. • Content-ID: Used to identify MIME entities uniquely in multiple contexts. • Content-Description: A text description of the object with the body; this is useful when the object is not readable (e.g., audio data)
147.	Explain the parameters that define an SSL session connection?
Ans.	<ol style="list-style-type: none"> 1. Server and client random: These are byte sequences that are chosen by the server and client for each connection. 2. Server write MAC secret: This is the secret key used in MAC operations on data sent by the server. 3. Client write MAC secret: This is the secret key used in MAC operations on data sent by the client. 4. Server write key: This is the secret encryption key for data encrypted by the server and decrypted by the client. 5. Client write key: This is the symmetric encryption key for data encrypted by the client and decrypted by the server. 6. Initialization vectors: When a block cipher in CBC mode is used, an initialization vector (IV) is maintained for each key. This field is first initialized by the SSL Handshake Protocol. Thereafter, the final ciphertext block from each record is preserved for use as the IV with the following record. 7. Sequence numbers: Each party maintains separate sequence numbers for transmitted and received messages for each connection. When a party sends or receives a change cipher spec message, the appropriate sequence number is set to zero. Sequence numbers may not exceed $2^{64}-1$.
148.	What do you mean by WEP? What are the shortcomings of WEP protocol?
Ans.	<p>WEP was an early attempt to provide security with the 802.11 network protocol. It is now considered too cryptographically weak to provide any meaningful protection from eavesdropping, but for a time it did provide some measure of security for low-sensitivity networks. WEP uses the RC4 cipher stream to encrypt each packet using a 64-bit key. This key is created using a 24-bit initialization vector and a 40-bit key value. The packets are formed using an XOR function to use the RC4 key value stream to encrypt the data packet. A 4-byte integrity check value (ICV) is calculated for each packet and then appended. According to many experts, WEP is too weak for use in most network settings because:</p> <p>Key management is not effective since most networks use a single shared secret key value for each node. Synchronizing key changes is a tedious process, and no key management is defined in the protocol, so keys are seldom changed.</p> <p>The initialization vector (IV) is too small, resulting in the recycling of IVs. An attacker can reverse engineer the RC4 cipher stream and decrypt subsequent packets, or can forge future packets. In 2007, this was accomplished in less than one minute.</p> <p>In summary, an intruder who collects enough data can threaten a WEP network in just a few minutes by decrypting or altering the data being transmitted, or by forging the WEP key to gain unauthorized access to the network. WEP also lacks a means of validating user credentials to ensure that only those who should be on the network are allowed to access it.</p>
149.	What do you mean by TLS? How TLS is different from SSL? Explain.
Ans.	<p>TLS is an IETF standardization initiative whose goal is to produce an Internet standard version of SSL. TLS is defined as a Proposed Internet Standard in RFC 2246. RFC 2246 is very similar to SSLv3.</p> <ol style="list-style-type: none"> 1. Version Number: The TLS Record Format is the same as that of the SSL Record Format, and the fields in the header have the same meanings. The one difference is in version values. For the current version of TLS, the Major Version is 3 and the Minor Version is 1. 2. Message Authentication Code: There are two differences between the SSLv3 and TLS MAC schemes: the actual algorithm and the scope of the MAC calculation. TLS makes use of the HMAC algorithm defined in RFC 2104. SSLv3 uses the same algorithm, except that the padding bytes are concatenated with the secret key rather than being XORED with the secret key padded to the block length. The level of security should be about the same in both cases. For TLS, the MAC calculation encompasses the fields indicated in the following expression: <pre>HMAC_hash(MAC_write_secret, seq_num TLSCompressed.type TLSCompressed.version TLSCompressed.length TLSCompressed.fragment)</pre> <p>The MAC calculation covers all of the fields covered by the SSLv3 calculation, plus the field TLSCompressed.version, which is the version of the protocol being employed.</p> <ol style="list-style-type: none"> 3. Pseudorandom Function: TLS makes use of a pseudorandom function referred to as PRF to expand secrets into blocks of data for purposes of key generation or validation. The objective is to make use of a relatively small shared secret value but to generate longer blocks of data in a way that is secure from the kinds of attacks made on hash functions and MACs. 4. Alert Codes: TLS supports all of the alert codes defined in SSLv3 with the exception of no_certificate.

	<p>5. Cipher Suites: There are several small differences between the cipher suites available under SSLv3 and under TLS:</p> <p>6. Key Exchange: TLS supports all of the key exchange techniques of SSLv3 with the exception of Fortezza.</p> <p>7. Symmetric Encryption Algorithms: TLS includes all of the symmetric encryption algorithms found in SSLv3, with the exception of Fortezza.</p> <p>8. Padding: In SSL, the padding added prior to encryption of user data is the minimum amount required so that the total size of the data to be encrypted is a multiple of the cipher's block length. In TLS, the padding can be any amount that results in a total that is a multiple of the cipher's block length, up to a maximum of 255 bytes.</p>
150.	<p>How do viruses avoid basic pattern match of antivirus?</p> <ul style="list-style-type: none"> a) They are encrypted b) They act with special permissions c) They modify themselves d) None of the mentioned
Ans.	c) They modify themselves
151.	<p>How does an antivirus of today identify viruses?</p> <ul style="list-style-type: none"> a) Previously known patterns b) It can detect unknown patterns c) It can take high priority to increase scanning speed d) None of the mentioned
Ans.	a) Previously known patterns
152.	<p>What is known as a sandbox?</p> <ul style="list-style-type: none"> a) It is a program which can be molded to do the desired task b) It is a program that is controlled or emulated section of OS c) It is a special mode of antivirus d) None of the mentioned
Ans.	b) It is a program that is controlled or emulated section of OS
153.	<p>What are the different ways to intrude?</p> <ul style="list-style-type: none"> a) Buffer overflows b) Unexpected combinations and unhandled input c) Race conditions d) All of the above
Ans.	d) All of the above
154.	<p>What are the major components of the intrusion detection system?</p> <ul style="list-style-type: none"> a) Analysis Engine b) Event provider c) Alert Database d) All of the above
Ans.	d) All of the above
155.	<p>What are the characteristics of anomaly based IDS?</p> <ul style="list-style-type: none"> a) It models the normal usage of network as a noise characterization b) It doesn't detect novel attacks c) Anything distinct from the noise is not assumed to be intrusion activity d) It detects based on signature
Ans.	a) It models the normal usage of network as a noise characterization
156.	<p>Who unleashed famous worm attack in 1988 which effected UNIX systems and caused losses in millions?</p> <ul style="list-style-type: none"> a) Robert Morris b) Bob Milano c) Mark zuckerberg d) Bill Gates
Ans.	a) Robert Morris
157.	<p>Which is not a port scan type?</p> <ul style="list-style-type: none"> a) TCP scanning b) SYN scanning c) UDP scanning d) SYSTEM Scanning
Ans.	d) SYSTEM Scanning
158.	<p>From the following, which is not a common file permission?</p> <ul style="list-style-type: none"> a) Write b) Execute c) Stop d) Read
Ans.	c) Stop
159.	<p>What does Light Directory Access Protocol (LDAP) doesn't store?</p> <ul style="list-style-type: none"> a) Users b) Address

	c) Passwords d) Security Keys
Ans.	b) Address
160.	Provide the security components of OS Security. a) User accounts Security b) BIOS Security c) Anti-virus Security d) All of these
Ans.	d) All of these
161.	Figure out the issues in Operating System Security. a) Authentication b) Malwares c) Software vulnerabilities d) All of these
Ans.	d) All of these
162.	An intrusion occurs when an attacker attempts to gain entry into or disrupt the normal operations of an information system.(True/False)
Ans.	True
163.	A network-based IDPS protects the server or host's information assets.(True/False)
Ans.	False
164.	A _____ IDPS resides on a computer or appliance connected to a segment of an organization's network and monitors network traffic on that network segment, looking for indications of ongoing or successful attacks.
Ans.	network-based
165.	A _____ IDPS resides on a particular computer or server, known as the host, and monitors activity only on that system.
Ans.	host-based
166.	During its lifetime, a typical virus goes through how much number of phases?
Ans.	4
167.	A _____ virus infects a master boot record or boot record and spreads when a system is booted from the disk containing the virus.
Ans.	Boot sector
168.	A GFI LANguard is a _____ scanner tools.
Ans.	vulnerability
169.	A packet sniffer is a network tool that collects copies of packets from the network and analyses them. (True/False)
Ans.	True
170.	List and briefly define three classes of intruders.
Ans.	1. Masquerader: An individual who is not authorized to use the computer and who penetrates a system's access controls to exploit a legitimate user's account. 2. Misfeasor: A legitimate user who accesses data, programs, or resources for which such access is not authorized, or who is authorized for such access but misuses his or her privileges. 3. Clandestine user: An individual who seizes supervisory control of the system and uses this control to evade auditing and access controls or to suppress audit collection.
171.	What are two common techniques used to protect a password file?
Ans.	1. One-way encryption: The system stores only an encrypted form of the user's password. When the user presents a password, the system encrypts that password and compares it with the stored value. In practice, the system usually performs a one-way transformation (not reversible) in which the password is used to generate a key for the encryption function and in which a fixed-length output is produced. 2. Access control: Access to the password file is limited to one or a very few accounts
172.	What are three benefits that can be provided by an intrusion detection system?
Ans.	1. If an intrusion is detected quickly enough, the intruder can be identified and ejected from the system before any damage is done or any data are compromised. Even if the detection is not sufficiently timely to preempt the intruder, the sooner that the intrusion is detected, the less the amount of damage and the more quickly that recovery can be achieved. 2. An effective intrusion detection system can serve as a deterrent, so acting to prevent intrusions. 3. Intrusion detection enables the collection of information about intrusion techniques that can be used to strengthen the intrusion prevention facility.
173.	What is a monitoring (or SPAN) port? What is it used for?
Ans.	A switched-port analysis port is a data port on a switched device that replicates all designated traffic from the switch device so that the traffic can be captured, stored or analyzed for IDS or other purposes.
174.	What is a honeypot? How is it different from a honeynet?
Ans.	A honeypot is a security mechanism that creates a virtual trap to lure attackers. An intentionally compromised computer system allows attackers to exploit vulnerabilities so we can study them to improve our security policies. We can apply a honeypot to any computing resource from software and networks to file servers and

	<p>routers. Honeypots are a type of deception technology that allows us to understand attacker behavior patterns. Security teams can use honeypots to investigate cybersecurity breaches to collect intel on how cybercriminals operate. They also reduce the risk of false positives, when compared to traditional cybersecurity measures, because they are unlikely to attract legitimate activity.</p> <p>A honeynet is a decoy network that contains one or more honeypots. It looks like a real network and contains multiple systems but is hosted on one or only a few servers, each representing one environment. For example, a Windows honeypot machine, a Mac honeypot machine and a Linux honeypot machine.</p>
175.	What is network footprinting? How are they related?
Ans.	An activity where the information about the organization along with their network activities and assets are being gathered in called as network footprinting. It is a process where an organized research and investigations is made on internet address possessed by a targeted organization. Footprinting is a process where information that is available in public are gathered about a particular organization. The attackers generally collect information such as IP address of an organization to make organized attack. Footprinting is performed with the help of public internet data sources such as organizations webpage which can contain information about the internal systems.
176.	What is a vulnerability scanner? How is it used to improve security?
Ans.	Vulnerability scanners are automated tools that allow organizations to check if their networks, systems and applications have security weaknesses that could expose them to attacks. Vulnerability scanning is a common practice across enterprise networks and is often mandated by industry standards and government regulations to improve the organization's security posture.
177.	What is the difference between active and passive vulnerability scanners?
Ans.	<p>Active Scanners: Active scanners send transmissions to the network's nodes, examining the responses they receive to evaluate whether a specific node represents a weak point within the network. A network administrator can also use an active scanner to simulate an attack on the network, uncovering weaknesses a potential hacker would spot, or examine a node following an attack to determine how a hacker breached security. Active scanners can take action to autonomously resolve security issues, such as blocking a potentially dangerous IP address.</p> <p>Passive Scanners: Passive scanners identify the active operating systems, applications and ports throughout a network, monitoring activity to determine the network's vulnerabilities. However, while passive scanners can provide information about weaknesses, they can't take action to resolve security problems. These scanners can check the current software and patch versions on networked devices, indicating which devices are using software that presents a potential gateway for hackers or trojan attacks, and reference this information against public databases containing lists of current patches. A network administrator can set passive scanners to run continuously or to operate at specified intervals.</p>
178.	Define packet sniffing.
Ans.	<p>When any data has to be transmitted over the computer network, it is broken down into smaller units at the sender's node called data packets and reassembled at receiver's node in original format. It is the smallest unit of communication over a computer network. It is also called a block, a segment, a datagram or a cell. The act of capturing data packet across the computer network is called packet sniffing.</p> <p>Packet sniffing is done by using tools called packet sniffer. It can be either filtered or unfiltered. Filtered is used when only specific data packets have to be captured and Unfiltered is used when all the packets have to be captured. Wireshark, SmartSniff are examples of packet sniffing tools.</p>
179.	Define open and close port.
Ans.	All communication that happens over the internet is exchanged via ports. Every IP address contains two kinds of ports, TCP and UDP, and there can be up to 65,535 of each for any given IP address. Services that connect to the internet (like web browsers, email clients, and file transfer services) use specific ports to receive information. In security parlance, the term open port is used to mean a TCP or UDP port number that is configured to accept packets. In contrast, a port which rejects connections or ignores all packets directed at it is called a closed port.
180.	Define the following terms: a) False negative b) False Positive c) Evasion d) Tuning
Ans.	<p>a. False negative: The failure of an IDPS to react to an actual attack event. This is the most grievous failure, since the purpose of an IDPS is to detect and respond to attacks.</p> <p>b. False positive: An alert or alarm that occurs in the absence of an actual attack. A false positive can sometimes be produced when an IDPS mistakes normal system activity for an attack. False positives tend to make users insensitive to alarms and thus reduce their reactivity to actual intrusion events.</p> <p>c. Evasion: The process by which attackers change the format and/or timing of their activities to avoid being detected by the IDPS.</p> <p>d. Tuning: The process of adjusting an IDPS to maximize its efficiency in detecting true positives, while minimizing both false positives and false negatives.</p>
181.	What are the different types of IDPS?
Ans.	<p>There are two types of IDPS:</p> <ol style="list-style-type: none"> 1. Network-based IDPS (NIDPS): A network-based IDPS is focused on protecting network information assets. 2. Host-based IDPS (HIDPS): A host-based IDPS protects the server or host's information assets. It monitors both network connection activity and current information states on host servers.

182.	Differentiate between virus and worm.
Ans.	A virus is a piece of software that can "infect" other programs by modifying them; the modification includes a copy of the virus program, which can then go on to infect other programs. A worm is a program that propagates copies of itself to other computers.
183.	What is firewall? What are the different types of firewall?
Ans.	A firewall forms a barrier through which the traffic going in each direction must pass. A firewall security policy dictates which traffic is authorized to pass in each direction. A firewall may be designed to operate as a filter at the level of IP packets, or may operate at a higher protocol layer. There are three types of firewall: 1. Packet Filtering Firewall 2. Application Level Gateway 3. Circuit Level Gateway
184.	What do you mean by port scanning? Explain.
Ans.	Port scanning utilities, or port scanners, are tools used by both attackers and defenders to identify (or fingerprint) the computers that are active on a network, as well as the ports and services active on those computers, the functions and roles the machines are fulfilling, and other useful information. A port is a network channel or connection point in a data communications system. There are 65,536 port numbers in use for TCP and another 65,536 port numbers for UDP. Services using the TCP/IP protocol can run on any port; however, services with reserved ports generally run on ports 1–1023. Port 0 is not used. Ports greater than 1023 are typically referred to as ephemeral ports and may be randomly allocated to server and client processes. An open port can be used by an attacker to send commands to a computer, potentially gain access to a server, and possibly exert control over a networking device. The general rule of thumb is to remove from service or secure any port not absolutely necessary to conducting business. For example, if a business doesn't host Web services, there is no need for port 80 to be available on its servers.
185.	How does a network-based IDPS differ from a host-based IDPS? Explain.
Ans.	<p>Network-Based IDPS: A network-based IDPS (NIDPS) resides on a computer or appliance connected to a segment of an organization's network and monitors network traffic on that network segment, looking for indications of ongoing or successful attacks. When the NIDPS identifies activity that it is programmed to recognize as an attack, it responds by sending notifications to administrators. When examining incoming packets, an NIDPS looks for patterns within network traffic such as large collections of related items of a certain type—which could indicate that a denial-of-service attack is underway—or the exchange of a series of related packets in a certain pattern—which could indicate that a port scan is in progress. An NIDPS can detect many more types of attacks than a host-based IDPS, but it requires a much more complex configuration and maintenance program. A NIDPS is installed at a specific place in the network (such as on the inside of an edge router) from where it is possible to monitor the traffic going into and out of a particular network segment.</p> <p>Host-Based IDPS: While a network-based IDPS resides on a network segment and monitors activities across that segment, a host-based IDPS (HIDPS) resides on a particular computer or server, known as the host, and monitors activity only on that system. HIDPSs are also known as system integrity verifiers because they benchmark and monitor the status of key system files and detect when an intruder creates, modifies, or deletes monitored files.</p> <p>An HIDPS has an advantage over an NIDPS in that it can access encrypted information traveling over the network and use it to make decisions about potential or actual attacks. Also, since the HIDPS works on only one computer system, all the traffic it examines traverses that system. An HIDPS is also capable of monitoring system configuration databases, such as windows registries, in addition to stored configuration files like .ini, .cfg, and .dat files. Most HIDPSs work on the principle of configuration or change management, which means that they record the sizes, locations, and other attributes of system files.</p>
186.	How does a signature-based IDPS differ from a behavior-based IDPS? Explain.
Ans.	A signature-based system looks for patterns of behavior that match a library of known behaviors. A behavior-based system watches for activities that suggest an alert-level activity is occurring based on sequences of actions or the timing between otherwise unrelated events.
187.	What kind of data and information can be found using a packet sniffer?
Ans.	A packet sniffer (sometimes called a network protocol analyzer) is a network tool that collects copies of packets from the network and analyzes them. It can provide a network administrator with valuable information for diagnosing and resolving networking issues. In the wrong hands, however, a sniffer can be used to eavesdrop on network traffic. An excellent free, client-based network protocol analyzer is Wireshark, formerly known as Ethereal. Wireshark allows the administrator to examine data from both live network traffic and captured traffic. Wireshark has several features, including a language filter and TCP session reconstruction utility.
188.	What is a virus? Explain the different phases of virus?
Ans.	A virus is a piece of software that can "infect" other programs by modifying them; the modification includes a copy of the virus program, which can then go on to infect other programs. A virus can do anything that other programs do. The only difference is that it attaches itself to another program and executes secretly when the host program is run. Once a virus is executing, it can perform any function, such as erasing files and programs. Parasitic virus: The traditional and still most common form of virus. A parasitic virus attaches itself to executable files and replicates, when the infected program is executed, by finding other executable files to infect. 1. Memory-resident virus: Lodges in main memory as part of a resident system program. From that point on, the virus infects every program that executes.

	<p>2. Boot sector virus: Infects a master boot record or boot record and spreads when a system is booted from the disk containing the virus.</p> <p>3. Stealth virus: A form of virus explicitly designed to hide itself from detection by antivirus software.</p> <p>4. Polymorphic virus: A virus that mutates with every infection, making detection by the "signature" of the virus impossible.</p> <p>5. Metamorphic virus: As with a polymorphic virus, a metamorphic virus mutates with every infection. The difference is that a metamorphic virus rewrites itself completely at each iteration, increasing the difficulty of detection. Metamorphic viruses may change their behavior as well as their appearance.</p>
189.	Differentiate between application level gateway and circuit level gateway firewall.
Ans.	<p>An application-level gateway, also called a proxy server, acts as a relay of application-level traffic. The user contacts the gateway using a TCP/IP application, such as Telnet or FTP, and the gateway asks the user for the name of the remote host to be accessed. When the user responds and provides a valid user ID and authentication information, the gateway contacts the application on the remote host and relays TCP segments containing the application data between the two endpoints. If the gateway does not implement the proxy code for a specific application, the service is not supported and cannot be forwarded across the firewall. Application-level gateways tend to be more secure than packet filters. Rather than trying to deal with the numerous possible combinations that are to be allowed and forbidden at the TCP and IP level, the application-level gateway need only scrutinize a few allowable applications. In addition, it is easy to log and audit all incoming traffic at the application level. A prime disadvantage of this type of gateway is the additional processing overhead on each connection.</p> <p>A circuit-level gateway can be a stand-alone system or it can be a specialized function performed by an application-level gateway for certain applications. A circuit-level gateway does not permit an end-to-end TCP connection; rather, the gateway sets up two TCP connections, one between itself and a TCP user on an inner host and one between itself and a TCP user on an outside host. Once the two connections are established, the gateway typically relays TCP segments from one connection to the other without examining the contents. The security function consists of determining which connections will be allowed.</p>
190.	What is packet filtering firewall? Explain
Ans.	<p>A packet-filtering router applies a set of rules to each incoming and outgoing IP packet and then forwards or discards the packet. The router is typically configured to filter packets going in both directions (from and to the internal network). Filtering rules are based on information contained in a network packet.</p>
191.	What do you mean by Memory and Address Protection?
Ans.	Memory protection includes protection for the memory that the OS itself uses as well as the memory of user processes. Major challenge in multi-programming system is to prevent one program from affecting the data and programs in the memory space of other users.
192.	What is the difference between rule-based anomaly detection and rule-based penetration identification?
Ans.	<p>With rule-based anomaly detection, historical audit records are analyzed to identify usage patterns and to generate automatically rules that describe those patterns. Rules may represent past behavior patterns of users, programs, privileges, time slots, terminals, and so on. Current behavior is then observed, and each transaction is matched against the set of rules to determine if it conforms to any historically observed pattern of behavior.</p> <p>Rule-based penetration identification uses rules for identifying known penetrations or penetrations that would exploit known weaknesses. Rules can also be defined that identify suspicious behavior, even when the behavior is within the bounds of established patterns of usage. Typically, the rules used in these systems are specific to the machine and operating system. Also, such rules are generated by "experts" rather than by means of an automated analysis of audit records.</p>
193.	What is a salt in the context of UNIX password management?
Ans.	The salt is combined with the password at the input to the one-way encryption routine.
194.	What is the difference between statistical anomaly detection and rule-based intrusion detection?
Ans.	<p>Statistical anomaly detection involves the collection of data relating to the behavior of legitimate users over a period of time. Then statistical tests are applied to observed behavior to determine with a high level of confidence whether that behavior is not legitimate user behavior.</p> <p>Rule-Based Detection involves an attempt to define a set of rules that can be used to decide that a given behavior is that of an intruder.</p>
195.	What is an Intrusion Detection System? Explain.
Ans.	An intrusion occurs when an attacker attempts to gain entry into or disrupt the normal operations of an information system, almost always with the intent to do harm. Intrusion detection consists of procedures and systems that identify system intrusions. An IDS works like a burglar alarm in that it detects a violation (some system activity analogous to an opened or broken window) and activates an alarm. This alarm can be audible and/or visual (producing noise and lights, respectively), or it can be silent (an e-mail message or pager alert).

	With almost all IDSSs, system administrators can choose the configuration of the various alerts and the alarm levels associated with each type of alert.
196.	What is the difference between stateful firewall and stateless firewall?
Ans.	<p>A stateful firewall is a firewall that monitors the full state of active network connections. This means that stateful firewalls are constantly analyzing the complete context of traffic and data packets, seeking entry to a network rather than discrete traffic and data packets in isolation. Once a certain kind of traffic has been approved by a stateful firewall, it is added to a state table and can travel more freely into the protected network. Traffic and data packets that don't successfully complete the required handshake will be blocked. By taking multiple factors into consideration before adding a type of connection to an approved list, such as TCP stages, stateful firewalls are able to observe traffic streams in their entirety. However, this method of protection does come with a few vulnerabilities. For example, stateful firewalls can fall prey to DDoS attacks due to the intense compute resources and unique software-network relationship necessary to verify connections.</p> <p>Stateless firewalls are designed to protect networks based on static information such as source and destination. Whereas stateful firewalls filter packets based on the full context of a given network connection, stateless firewalls filter packets based on the individual packets themselves. To do so, stateless firewalls use packet filtering rules that specify certain match conditions. If match conditions are met, stateless firewall filters will then use a set of preapproved actions to guide packets into the network. If match conditions are not met, unidentified or malicious packets will be blocked. Because stateless firewalls do not take as much into account as stateful firewalls, they're generally considered to be less rigorous. For example, stateless firewalls can't consider the overall pattern of incoming packets, which could be useful when it comes to blocking larger attacks happening beyond the individual packet level.</p>
197.	What are the different ways for memory and address protection of Operating System? Explain.
Ans.	<p>Memory protection includes protection for the memory that the OS itself uses as well as the memory of user processes. Major challenge in multi-programming system is to prevent one program from affecting the data and programs in the memory space of other users.</p> <p>The various methods for memory and address protection are:</p> <ul style="list-style-type: none"> i. Fence: A fence or fence address is simplest form of memory protection which can be used only for single user operating system. A fence is a particular address that users and their processes cannot cross. Only the OS can operate on one side of the fence and users are restricted to the other side. A fence could be static, in which case there is a fixed fence address. Alternatively, a dynamic fence can be used, which can be implemented using a fence register to specify the current fence address. ii. Base and Bounds registers: This type of protection can be used in multi-user environment where one user's program needs to be protected from the other. Each user has a base register which is the lower address and a Bound register which is the upper address limit. The base and bounds register approach implicitly assumes that the user or process space is contiguous in memory. The OS must determine what protection to apply to a specific memory location. In some cases it might be sufficient to apply the same protection to all of a user's memory. The disadvantage is that the registers confine access to consecutive range of addresses. iii. Tagging: This specifies the protection for each individual address. In this method of protection every word of machine memory has one or more extra bits to identify the access rights to that word. Only privileged instructions can set these access bits. While this is as fine-grained protection as possible, it introduces significant overhead. The overhead can be reduced by tagging sections of the address space instead of each individual address. Another drawback to tagging is compatibility, since tagging schemes are not in common use. iv. Segmentation: This method divides the memory into logical units such as individual procedures or the data in one array. Once they are divided, appropriate access control can be enforced on each segment. A benefit of segmentation is that any segment can be placed in any memory location provided the location is large enough to hold it. The OS must keep track of the locations of all segments, which is accomplished using <segment,offset> pairs, where the named segment specifies the segment, and the offset is the starting address of the specified segment. With segmentation, all address references must go through the OS, so the OS can, in this respect, achieve complete mediation. Depending on the access control applied to particular segments, users can share access to some segments or users can be restricted to specific segments. v. Paging: Paging discards the disadvantage of segmentation. In paging all segments are of a fixed size called as pages and the memory divided is known as page frames. In paging a particular page can be accessed using a pair of the form <page, offset=""> where page is the page number and offset is location within a page. The advantages of paging over segmentation include no fragmentation, improved efficiency, and the fact that there are no variable sizes to worry about. The disadvantages are that there is, in general, no logical unity to pages, which makes it more difficult to determine the proper access control to apply to a given page.
198.	What do you mean by Intrusion Prevention Systems? Explain. Why IDPS is required?
Ans.	An intrusion occurs when an attacker attempts to gain entry into or disrupt the normal operations of an information system, almost always with the intent to do harm. Intrusion prevention consists of activities that deter an intrusion. Some important intrusion prevention activities are writing and implementing good enterprise information security policy, planning and executing effective information security programs, installing and testing technology-based information security countermeasures (such as firewalls and intrusion

	<p>detection systems), and conducting and measuring the effectiveness of employee training and awareness activities.</p> <p>According to the NIST documentation on industry best practices, there are several compelling reasons to acquire and use an IDPS:</p> <ol style="list-style-type: none"> 1. To prevent problem behaviors by increasing the perceived risk of discovery and punishment for those who would attack or otherwise abuse the system. 2. To detect attacks and other security violations that are not prevented by other security measures. 3. To detect and deal with the preambles to attacks (commonly experienced as network probes and other "doorknob rattling" activities). 4. To document the existing threat to an organization. 5. To act as quality control for security design and administration, especially in large and complex enterprises. 6. To provide useful information about intrusions that do take place, allowing improved diagnosis, recovery, and correction of causative factors.
199.	What are the different types of viruses? Explain,
Ans.	<p>Parasitic virus: The traditional and still most common form of virus. A parasitic virus attaches itself to executable files and replicates, when the infected program is executed, by finding other executable files to infect.</p> <p>Memory-resident virus: Lodges in main memory as part of a resident system program. From that point on, the virus infects every program that executes.</p> <p>Boot sector virus: Infects a master boot record or boot record and spreads when a system is booted from the disk containing the virus.</p> <p>Stealth virus: A form of virus explicitly designed to hide itself from detection by antivirus software.</p> <p>Polymorphic virus: A virus that mutates with every infection, making detection by the "signature" of the virus impossible.</p> <p>Metamorphic virus: As with a polymorphic virus, a metamorphic virus mutates with every infection. The difference is that a metamorphic virus rewrites itself completely at each iteration, increasing the difficulty of detection. Metamorphic viruses may change their behavior as well as their appearance.</p>
200.	The EISP is based on and directly supports the _____ of the organization and sets the strategic direction, scope, and tone for all security efforts. a) Vision b) Mission c) Direction d) All of these
Ans.	d) All of these
201.	Which of the following thing/s comes under the issue specific security policy: a) Use of phone b) Use of photocopy equipment c) Use of pen drive d) All of these
Ans.	d) All of these
202.	Information leakage is one of the threats of computer system specifically distributed systems where sensitive information can easily be revealed to unauthorized users that results to lack of integrity. (True/False)
Ans.	False
203.	Authorization is a prerequisite for authentication.
Ans.	False
204.	Lack of access control policy is a _____. a) Bug b) Threat c) Vulnerability d) Attack
Ans.	c) Vulnerability
205.	Security features that control that can access resources in the OS. a) Authentication b) Identification c) Validation d) Access control
Ans.	d) Access control
206.	A _____ is a plan or course of action that conveys instructions from an organization's senior management to those who make decisions, take actions, and perform other duties.
Ans.	policy
207.	Information security safeguards provide three levels of controls; what are they?
Ans.	managerial, operational, and technical
208.	A buffer against outside attacks is frequently referred to as a _____.
Ans.	demilitarized zone (DMZ)
209.	Which direction access cannot happen using DMZ zone by default? a) Company computer to DMZ b) Internet to DMZ

	c) Internet to company computer d) Company computer to internet
Ans.	c) Internet to company computer
210.	The ACL that consists of a list related to an object that states all the subjects that can be allowed to access the object, as well as the rights to the object. (True/False)
Ans.	True
211.	Access control is the method by which systems determine whether and how to admit a user into a trusted area of the organization.(True/False)
Ans.	True
212.	Nondiscretionary access controls (MACs) use data classification schemes; they give users and data owners limited control over access to information resources. (True/False)
Ans.	False
213.	A variation of Mandatory access controls is called_____, in which users are assigned a matrix of authorizations for particular areas of access.
Ans.	lattice-based access control
214.	Discretionary controls are a strictly-enforced version of MACs that are managed by a central authority in the organization and can be based on an individual's role.(True/False)
Ans.	False
215.	An information security policy provides rules for the protection of the information assets of the organization.(True/False)
Ans.	True
216.	Issue specific security policies often function as standards or procedures to be used when configuring or maintaining systems.(True/False)
Ans.	False
217.	A _____ might describe the configuration and operation of a network firewall.
Ans.	System Specific Security Policy
218.	Indicate the security standard that specifies a management system to bring information security under management control. a) ISO 27002 b) ISO 27001 c) ISO 3200 d) ISO 3201
Ans.	b) ISO 27001
219.	Which of the following threat may occur in the distributed system? a) Denial of Service b)Information Leakage c) Unauthorized Access d)All of these
Ans.	d) All of these
220.	What do you mean by access control?
Ans.	Access control is the method by which systems determine whether and how to admit a user into a trusted area of the organization—that is, information systems, restricted areas such as computer rooms, and the entire physical location. Access control is achieved by means of a combination of policies, programs, and technologies.
221.	What are the different types of access control?
Ans.	Access controls can be of three types: 1. Mandatory Access Control 2. Nondiscretionary Access Control 3. Discretionary Access Control
222.	What are the different methods for logical access control?
Ans.	Logical access control is done via access control lists (ACLs), group policies, passwords, and account restrictions.
223.	What did you understand from information security policies?
Ans.	A policy is a plan or course of action that conveys instructions from an organization's senior management to those who make decisions, take actions, and perform other duties. Policies are organizational laws in that they dictate acceptable and unacceptable behavior within the organization. An information security policy provides rules for the protection of the information assets of the organization.
224.	What are the different types of information security policies?
Ans.	There are three types of security policy, according to the National Institute of Standards and Technology's: 1. Enterprise information security policies 2. Issue-specific security policies 3. Systems-specific security policies
225.	Write the name of different threats to the distributed system?
Ans.	There are different threats when distributed system is concerned, as any networked computer system can face it. It is important to implement countermeasures for all expected threats for the purpose of the system to remain constant and cost effective. Those threats can be distinguished depending on their interaction as follows below: 1. Denial of service

	2. Information leakage 3. Unauthorized access
226.	What are the major design issues in building secure distributed system? Ans. The major design issues in building secure distributed systems are: 1. Focus of control 2. Layering of security mechanism Focus of control: There are three approaches that can be followed to protect a distributed application: a. Protection against invalid operations on secure data b. Protection against unauthorized invocations c. Protection against unauthorized users Layering of security mechanism: One of the important aspect of designing secure systems is to decide which level the security mechanism should be placed. Security mechanism is normally placed in middleware in a distributed system.
227.	What do you mean by database security? Explain. Ans. All systems have ASSETS and security is about protecting assets. The first thing, then, is to know your assets and their value. The second thing to know is what THREATs are putting your assets at risk. These include things such as power failure and employee fraud. Note that threats are partly hypothetical, always changing and always imperfectly known. Security activity is directed at protecting the system from perceived threats.
228.	Write the names of the different threats to database security? Ans. • Unauthorised modification • Unauthorised disclosure • Loss of availability • Commercial sensitivity • Personal privacy and data protection • Computer misuse
229.	What do you mean by database security models? Ans. A security model establishes the external criteria for the examination of security issues in general, and provides the context for database considerations, including implementation and operation. There are two database security models: 1. Authentication 2. Authorization
230.	What is a distributed denial-of-service attack? Explain. Ans. A distributed denial of-service (DDoS) is an attack in which a coordinated stream of requests is launched against a target from many locations at the same time. Most DDoS attacks are preceded by a preparation phase in which many systems, perhaps thousands, are compromised. The compromised machines are turned into zombies, machines that are directed remotely (usually by a transmitted command) by the attacker to participate in the attack. DDoS attacks are the most difficult to defend against, and there are presently no controls that any single organization can apply.
231.	Write the name of modules exist in the ORACLE DBSAT? Ans. The DBSAT tool contains three modules: 1. Collector 2. Reporter 3. Discoverer Collector and Reporter work together to discover any risk areas and will produce reports regarding those risk areas – the Database Security Assessment report. The Discoverer is a stand-alone module used to locate and report on sensible data – and will produce the Database Sensitive Data Assessment report.
232.	What is the ISO 27000 series of standards? Ans. The ISO 27000 family of information security management standards is a series of mutually supporting information security standards that can be combined to provide a globally recognised framework for best-practice information security management.
233.	What are the differences between a policy, a standard, and a practice? Ans. Policy - Written instructions that describe proper behavior. Standard - Detailed statement of what must be done to comply with policy. Practice - Examples of actions that would comply with policy.
234.	Where can a security administrator find information on established security frameworks? Ans. ISO/IEC 27002 gives guidelines for organizational information security standards and information security management practices including the selection, implementation and management of controls taking into consideration the organization's information security risk environment(s). It is designed to be used by organizations that intend to: • select controls within the process of implementing an Information Security Management System based on ISO/IEC 27001; • implement commonly accepted information security controls; • develop their own information security management guidelines.

235.	Write about the worst-case assumptions and design guidelines for secure Distributed Systems.
Ans.	<p>Interfaces are exposed: Distributed systems are composed of processes that offer services or share information. Their communication interfaces are necessarily open (to allow new clients to access them) - an attacker can send a message to any interface.</p> <p>Networks are insecure: For example, message sources can be falsified - messages can be made to look as though they came from Alice when they were actually sent by Mallory. Host addresses can be 'spoofed' - Mallory can connect to the network with the same address as Alice and receive copies of messages intended for her.</p> <p>Limit the lifetime and scope of each secret: When a secret key is first generated we can be confident that it has not been compromised. The longer we use it and the more widely it is known, the greater the risk. The use of secrets such as passwords and shared secret keys should be time-limited, and sharing should be restricted.</p> <p>Algorithms and program code are available to attackers: The bigger and the more widely distributed a secret is, the greater the risk of its disclosure. Secret encryption algorithms are totally inadequate for today's large-scale network environments. Best practice is to publish the algorithms used for encryption and authentication, relying only on the secrecy of cryptographic keys. This helps to ensure that the algorithms are strong by throwing them open to scrutiny by third parties.</p> <p>Attackers may have access to large resources: The cost of computing power is rapidly decreasing. We should assume that attackers will have access to the largest and most powerful computers projected in the lifetime of a system, then add a few orders of magnitude to allow for unexpected developments.</p> <p>Minimize the trusted base: The portions of a system that are responsible for the implementation of its security, and all the hardware and software components upon which they rely, have to be trusted - this is often referred to as the trusted computing base. Any defect or programming error in this trusted base can produce security weaknesses, so we should aim to minimize its size. For example, application programs should not be trusted to protect data from their users.</p>
236.	Compare Nondiscretionary Access Control and Discretionary Access Control.
Ans.	<p>Nondiscretionary controls are a strictly-enforced version of MACs that are managed by a central authority in the organization and can be based on an individual's role—role-based controls—or a specified set of tasks (subject- or object-based)—task-based controls.</p> <p>Role-based controls are tied to the role a user performs in an organization, and task-based controls are tied to a particular assignment or responsibility. The role and task controls make it easier to maintain the controls and restrictions associated with a particular role or task, especially if the individual performing the role or task changes often. Instead of constantly assigning and revoking the privileges of individuals who come and go, the administrator simply assigns the associated access rights to the role or task, and then whenever individuals are associated with that role or task, they automatically receive the corresponding access. When their turns are over, they are removed from the role or task and the access is revoked.</p> <p>Discretionary access controls (DACs) are implemented at the discretion or option of the data user. The ability to share resources in a peer-to-peer configuration allows users to control and possibly provide access to information or resources at their disposal. The users can allow general, unrestricted access, or they can allow specific individuals or sets of individuals to access these resources. For example, a user has a hard drive containing information to be shared with office co-workers. This user can elect to allow access to specific individuals by providing access, by name, in the share control function.</p>
237.	Write about the enterprise information security policies?
Ans.	<p>An enterprise information security policy (EISP) is also known as a general security policy, organizational security policy, IT security policy, or information security policy. The EISP is based on and directly supports the mission, vision, and direction of the organization and sets the strategic direction, scope, and tone for all security efforts. The EISP is an executive level document, usually drafted by or in cooperation with the chief information officer of the organization. This policy is usually two to ten pages long and shapes the philosophy of security in the IT environment. The EISP usually needs to be modified only when there is a change in the strategic direction of the organization. The EISP guides the development, implementation, and management of the security program. It sets out the requirements that must be met by the information security blueprint or framework. It defines the purpose, scope, constraints, and applicability of the security program. It also assigns responsibilities for the various areas of security, including systems administration, maintenance of the information security policies, and the practices and responsibilities of the users. Finally, it addresses legal compliance.</p> <p>According to the National Institute of Standards and Technology (NIST), the EISP typically addresses compliance in the following two areas:</p> <ol style="list-style-type: none"> 1. General compliance to ensure meeting the requirements to establish a program and the responsibilities assigned therein to various organizational components. 2. The use of specified penalties and disciplinary action.
238.	Write about the issues specific security policies?
Ans.	As an organization executes various technologies and processes to support routine operations, it must instruct employees on the proper use of these technologies and processes. In general, the issue-specific security policy,

	<p>or ISSP,</p> <ul style="list-style-type: none"> • addresses specific areas of technology as listed below, • requires frequent updates, and • contains a statement on the organization's position on a specific issue. <p>An ISSP may cover the following topics, among others:</p> <ul style="list-style-type: none"> • E-mail • Use of the Internet • Specific minimum configurations of computers to defend against worms and viruses • Prohibitions against hacking or testing organization security controls • Home use of company-owned computer equipment • Use of personal equipment on company networks • Use of telecommunications technologies (fax and phone) • Use of photocopy equipment
239.	Write about the system specific security policies?
Ans.	<p>System Specific Security Policies (SysSPs) often function as standards or procedures to be used when configuring or maintaining systems. For example, a SysSP might describe the configuration and operation of a network firewall. This document could include a statement of managerial intent; guidance to network engineers on the selection, configuration, and operation of firewalls; and an access control list that defines levels of access for each authorized user. SysSPs can be separated into two general groups, managerial guidance and technical specifications, or they can be combined into a single policy document.</p>
240.	What is ORACLE DBSAT? What does DBSAT check? Explain.
Ans.	<p>The Oracle Database Security Assessment Tool is a stand-alone command line tool that accelerates the assessment and regulatory compliance process by collecting relevant types of configuration information from the database and evaluating the current security state to provide recommendations on how to mitigate the identified risks. DBSAT enables customers to quickly find:</p> <ul style="list-style-type: none"> • Security configuration issues, and how to remediate them • Users and their entitlements • Location, type, and quantity of sensitive data <p>DBSAT analyses information on the database and listener configuration to identify configuration settings that may unnecessarily introduce risk. DBSAT goes beyond simple configuration checking, examining user accounts, privilege and role grants, authorization control, separation of duties, fine-grained access control, data encryption and key management, auditing policies, and OS file permissions. DBSAT applies rules to quickly assess the current security status of a database and produce findings in all the areas above. For each finding, DBSAT recommends remediation activities that follow best practices to reduce or mitigate risk. DBSAT also scans the database for sensitive data using customizable regular expression patterns, and reports on the amount and type of sensitive data found. Besides providing the ability to search for sensitive data on English based data dictionaries (column names and comments) it also includes support for additional major European languages such as Dutch, French, Italian, German, Portuguese and Spanish. This provides organizations with a deeper insight on how much sensitive data they have and where it resides, enabling them to then protect their databases through appropriate access controls, auditing, masking, and encryption.</p>
241.	Write a short note on Cluster Computing Security?
Ans.	<p>Clustering security is one of the most important factors that needs to be considered during clustering. Connectivity, especially when they are implemented through the internet, is susceptible to any type of attack. The attacks to clustering of nodes could come in different forms – it could be as simple as a virus wherein its sole purpose is to destroy files or could be a very powerful spyware that can easily hijack the controls of nodes for malicious purposes.</p> <p>It only takes a single security flaw to destroy the entire clustering configuration. Whenever a network opens up a connection to its administrator, it automatically opens itself to different forms of attacks. This is also possible for users who try to access the nodes and stores data.</p> <p>In gist, there is always a possibility of attack whenever an interaction happens with the client and the server. This is practically the “security nightmare” in clustering since interaction will always happen which means the nodes are always susceptible to different attacks.</p> <p>For domain based security for clusters, administrator could easily control the clusters since security is based online. It does not even matter where the administrator implements security and troubleshooting as long as there is a strong connection between the clusters and the administrator.</p> <p>However, domain based security could be easily hacked. Anything that is done online could be monitored and used against the administrator’s will. Local security on the other hand boasts of optimal security by localizing administrator credentials for access.</p> <p>Most of the network tools for clustering today are using this form of security measure. But this type of security protocol is not easy. It requires a lot of resources especially when they are configured for the first time. Access to local nodes will also be challenging especially when the administrator tries to access them through online connectivity.</p>
242.	Compare managerial, operational, and technical levels of controls.
Ans.	<p>Managerial Control: Managerial controls are security processes that are designed by strategic planners and implemented by the security administration of the organization. Management controls set the direction and</p>

	<p>scope of the security process and provide detailed instructions for its conduct, as well as addressing the design and implementation of the security planning process and security program management. They also address risk management and security control reviews, describe the necessity and scope of legal compliance, and set guidelines for the maintenance of the entire security life cycle.</p> <p>Operational Controls: Operational controls are management and lower-level planning functions that deal with the operational functionality of security in the organization, such as disaster recovery and incident response planning. Operational controls address personnel security, physical security, and the protection of production inputs and outputs. In addition, operational controls guide the development of education, training, and awareness programs for users, administrators, and management. Finally, they address hardware and software systems maintenance and the integrity of data.</p> <p>Technical Controls: Technical controls are the tactical and technical implementations of security in the organization. While operational controls address specific operational issues, such as developing and integrating controls into the business functions, technical controls are the components put in place to protect an organization's information assets. They include logical access controls, such as identification, authentication, authorization, accountability (including audit trails), cryptography, and the classification of assets and users.</p>
243.	<p>What do you mean by Access Control Lists (ACL)? Explain.</p> <p>Ans. Access control lists (ACLs) consist of the user access lists, matrices, and capability tables that govern the rights and privileges of users. ACLs can control access to file storage systems, software components, or network communications devices. A capabilities table specifies which subjects and objects users or groups can access; in some systems, capabilities tables are called user profiles or user policies. These specifications frequently take the form of complex matrices, rather than simple lists or tables. The access control matrix includes a combination of tables and lists, such that organizational assets are listed along the column headers, while users are listed along the row headers. The resulting matrix contains ACLs in columns for a particular device or asset, and capability tables in rows for a particular user. ACLs can restrict access for a particular user, computer, time, duration—even a particular file. This specificity provides powerful control to the administrator.</p> <p>In general, ACLs regulate the following:</p> <ul style="list-style-type: none"> • Who can use the system • What authorized users can access • When authorized users can access the system • Where authorized users can access the system from <p>The who of ACL access may be determined by a person's identity or by a person's membership in a group. Restricting what authorized users are permitted to access—whether by type (printers, files, communication devices, or applications), name, or location—is achieved by adjusting the resource privileges for a person or group to one of Read, Write, Create, Modify, Delete, Compare, or Copy. To control when access is allowed, some organizations implement time-of-day and/or day-of-week restrictions for some network or system resources. To control where resources can be accessed from, many network-connected assets block remote usage and also have some levels of access that are restricted to locally connected users. When these various ACL options are applied concurrently, the organization can govern how its resources can be used.</p>
244.	<p>Explain the mandatory access control?</p> <p>Ans. Mandatory access controls (MACs) use data classification schemes; they give users and data owners limited control over access to information resources. In a data classification scheme, each collection of information is rated, and each user is rated to specify the level of information that user may access. These ratings are often referred to as sensitivity levels, and they indicate the level of confidentiality the information requires. A variation of this form of access control is called lattice-based access control, in which users are assigned a matrix of authorizations for particular areas of access. The level of authorization may vary between levels, depending on the classification authorizations individuals possess for each group of information or resources. The lattice structure contains subjects and objects, and the boundaries associated with each pair are demarcated. Lattice-based control specifies the level of access each subject has to each object. With this type of control, the column of attributes associated with a particular object (such as a printer) is referred to as an access control list (ACL). The row of attributes associated with a particular subject (such as a user) is referred to as a capabilities table.</p>
245.	<p>Explain the authentication and authorization Database security models?</p> <p>Ans. A security model establishes the external criteria for the examination of security issues in general, and provides the context for database considerations, including implementation and operation.</p> <p>Authentication: The client has to establish the identity of the server and the server has to establish the identity of the client. This is done often by means of shared secrets (either a password/user-id combination, or shared biographic and/or biometric data). It can also be achieved by a system of higher authority which has previously established authentication. In client-server systems where data (not necessarily the database) is distributed, the authentication may be acceptable from a peer system. Note that authentication may be transmissible from system to system. The result, as far as the DBMS is concerned, is an authorisation identifier. Authentication does not give any privileges for particular tasks. It only establishes that the DBMS trusts that the user is who he/she claimed to be and that the user trusts that the DBMS is also the intended system. Authentication is a prerequisite for authorisation.</p>

	<p>Authorization: Authorisation relates to the permissions granted to an authorised user to carry out particular transactions, and hence to change the state of the database (write item transactions) and/or receive data from the database (read-item transactions). The result of authorisation, which needs to be on a transactional basis, is a vector: Authorisation (item, auth-id, operation). A vector is a sequence of data values at a known location in the system. How this is put into effect is down to the DBMS functionality. At a logical level, the system structure needs an authorisation server, which needs to co-operate with an auditing server. There is an issue of server-to-server security and a problem with amplification as the authorisation is transmitted from system to system. Amplification here means that the security issues become larger as a larger number of DBMS servers are involved in the transaction. Audit requirements are frequently implemented poorly. To be safe, you need to log all accesses and log all authorisation details with transaction identifiers. There is a need to audit regularly and maintain an audit trail, often for a long period.</p>
246.	Explain the distributed system security mechanisms?
Ans.	<p>a. Cryptography: The security of information transmitted from one node to another is questionable, therefore there is a need of using a proper method of transforming it into unreadable formats (secrets writing) through cryptography. The use of a single key or public key cryptographic algorithm which is suitable for protecting message content by hiding information carried by a packet during the transmission process. This can be accomplished using RSA or AES algorithms.</p> <p>b. Authentication protocol: Provides a series of communication procedures between users of the system and the server for the purpose of securing the communication process.</p> <p>c. Access control mechanism: This can be done using access control lists (ACL) that consists of a list related to an object that states all the subjects that can be allowed to access the object, as well as the rights to the object. ACL normally are implemented directly or as an approximation in recent Operating systems.</p>
247.	What do you mean by security to distributed system? What are the different threats to the distributed system? Explain.
Ans.	<p>Today, computers are not stand alone units. Several computers are being networked together to form large computer systems. Not only are computers being networked, but they are being networked into large distributed systems where each individual computer, node if you will, can make use of the applications distributed throughout the system.</p> <p>There are different threats when distributed system is concerned, as any networked computer system can face it. It is important to implement countermeasures for all expected threats for the purpose of the system to remain constant and cost effective. Those threats can be distinguished depending on their interaction as follows below:</p> <ol style="list-style-type: none"> 1. Denial of service: Involves attacks that affect the availability of information from the system to the user resulting in paralysation of the entire operation of an organization or part of activities depending on the attack. The use of resource control mechanism can help in solving the above problem by applying timing responses, sizing responses, and connection control. Also problem detection by timing latency in system can easily be done if there is a dramatic increase of latency then denial of service (DoS) can be detected as well as addressed. 2. Information leakage: is one of the threats of computer system specifically distributed systems where sensitive information can easily be revealed to unauthorized users that results to lack of confidentiality. 3. Unauthorized access: This can occur due to the reason that the physical configuration is not strong enough to protect such threats from accessing the system (distributed system). This is known as inter process communication threats. Access control policies will enable organizations to be able to specify different ways that will lead to proper management of access to resources as well as information which are the valuable assets of an organization.
248.	What are the different threats to the database security? Explain.
Ans.	<p>a. Unauthorised modification: Changing data values for reasons of sabotage, crime or ignorance which may be enabled by inadequate security mechanisms, or sharing of passwords or password guessing, for example.</p> <p>b. Unauthorised disclosure: When information that should not have been disclosed has been disclosed. A general issue of crucial importance, which can be accidental or deliberate.</p> <p>c. Loss of availability: Sometimes called denial of service. When the database is not available it incurs a loss (otherwise life is better without the system!). So any threat that gives rise to time offline, even to check whether something has occurred, is to be avoided.</p> <p>d. Commercial sensitivity: Most financial losses through fraud arise from employees. Access controls provide both protection against criminal acts and evidence of attempts (successful or otherwise) to carry out acts detrimental to the organisation, whether fraud, extraction of sensitive data or loss of availability.</p> <p>e. Personal privacy and data protection: Internationally, personal data is normally subject to legislative controls. Personal data is data about an identifiable individual. Often the individual has to be alive but the method of identification is not prescribed. So a postal code for a home may in some cases identify an individual, if only one person is living at an address with the postal code. Such data needs careful handling and control.</p>

	<p>f. Computer misuse: There is also generally legislation on the misuse of computers. Misuse includes the violation of access controls and attempts to cause damage by changing the database state or introducing worms and viruses to interfere with proper operation. These offences are often extraditable. So an unauthorised access in Hong Kong using computers in France to access databases in Germany which refer to databases in America could lead to extradition to France or Germany or the USA.</p> <p>g. Audit requirements: These are operational constraints built around the need to know who did what, who tried to do what, and where and when everything happened. They involve the detection of events (including CONNECT and GRANT transactions), providing evidence for detection, assurance as well as either defence or prosecution. There are issues related to computer-generated evidence not covered here.</p>
249.	How to protect the database from various security threats? Explain.
Ans.	<p>To protect the database system from various threats. Here are some countermeasures which are as follows:</p> <ul style="list-style-type: none"> a. Access Control: A database for an organization contains a great deal of information and usually has several users. Most of them need to access only a small part of the database. A policy defines the requirements that are to be implemented within hardware and software and those that are external to the system, including physical, personal, and procedural controls. b. Flow Control: Flow control provides the flow of information among accessible objects. Flow controls check that information contained in objects does not flow explicitly or implicitly into less protected objects. c. Encryption: An encryption algorithm should be applied to the data, using a user-specified encryption key. The output of the algorithm is the encrypted version. There is also a decryption algorithm, which takes the encrypted data and a decryption key as input and then returns the original data. d. RAID: Redundant Array of Independent Disks which protect against data loss due to disk failure. e. Authentication: Access to the database is a matter of authentication. It provides the guidelines how the database is accessed. Every access should be monitored. f. Backup: At every instant, backup should be done. In case of any disaster, Organizations can retrieve their data.

Unit V

Syllabus

Access Control Mechanisms, Security Policies: Definition, Types, various models of security; Introduction to Security in Distributed Systems, Introduction to Database security methods.

Access Control

- Access control is the method by which systems determine **whether and how to admit a user into a trusted area of the organization**—that is, information systems, restricted areas such as computer rooms, and the entire physical location.
- Access control is achieved by means of a combination of **policies, programs, and technologies**.
- Access controls can be of **three** types:
 - Mandatory Access Control
 - Nondiscretionary Access Control
 - Discretionary Access Control

Mandatory Access Control

- **Mandatory access controls (MACs)** use data classification schemes; they give **users and data owners limited control over access to information resources**.
- In a data classification scheme, each collection of information is rated, and each user is rated to specify the level of information that user may access.
- These ratings are often referred to as sensitivity levels, and they indicate the level of confidentiality the information requires.
- A variation of this form of access control is called **lattice-based access control**, in which users are assigned a **matrix of authorizations** for particular areas of access.
- The level of authorization may vary between levels, depending on the classification authorizations individuals possess for each group of information or resources.
- The lattice structure contains subjects and objects, and the boundaries associated with each pair are demarcated.
- Lattice-based control specifies the level of access each subject has to each object.
- With this type of control, the **column of attributes associated with a particular object** (such as a printer) is referred to as an **access control list (ACL)**.
- The **row of attributes associated with a particular subject** (such as a user) is referred to as a **capabilities table**.

Nondiscretionary Access Control

- **Nondiscretionary controls** are a strictly-enforced version of MACs that are managed by a central authority in the organization and can be based on an **individual's role—role-based controls**—or a specified **set of tasks** (subject- or object-based)—**task-based controls**.
- **Role-based controls** are tied to the role a user performs in an organization, and **task-based controls** are tied to a particular assignment or responsibility.
- The role and task controls make it easier to maintain the controls and restrictions associated with a particular role or task, especially if the individual performing the role or task changes often.
- Instead of constantly assigning and revoking the privileges of individuals who come and go, the **administrator simply assigns the associated access rights to the role or task, and then whenever individuals are associated with that role or task, they automatically receive the corresponding access**.
- When their turns are over, they are removed from the role or task and the access is revoked.

Discretionary Access Control

- **Discretionary access controls (DAs)** are implemented at the discretion or option of the data user.
- The ability to share resources in a peer-to-peer configuration allows users to control and possibly provide access to information or resources at their disposal.
- The users can allow general, unrestricted access, or they can allow specific individuals or sets of individuals to access these resources.
- For example, a user has a hard drive containing information to be shared with office co-workers.
- This user can elect to allow access to specific individuals by providing access, by name, in the share control function.

Contd...

- In general, all access control approaches rely on the following mechanisms:
 - Identification
 - Authentication
 - Authorization
 - Accountability

Information Security Policies

- A **policy** is a **plan or course of action** that conveys instructions from an organization's senior management to those who make decisions, take actions, and perform other duties.
- Policies are organizational laws in that they dictate acceptable and unacceptable behavior within the organization.
- The meaning of the term **security policy** depends on the context in which it is used.
- Governmental agencies view security policy in terms of national security and national policies to deal with foreign states.
- A security policy can also communicate a credit card agency's method for processing credit card numbers.
- In general, a security policy is a set of rules that protect an organization's assets.
- **An information security policy provides rules for the protection of the information assets of the organization.**

Contd...

Management must define **three types** of security policy, according to the National Institute of Standards and Technology's:

- 1. Enterprise information security policies**
- 2. Issue-specific security policies**
- 3. Systems-specific security policies**

Enterprise Information Security Policies

- An **enterprise information security policy (EISP)** is also known as a general security policy, organizational security policy, IT security policy, or information security policy.
- The **EISP is based on and directly supports the mission, vision, and direction of the organization and sets the strategic direction, scope, and tone for all security efforts.**
- The EISP is an executive level document, usually drafted by or in cooperation with the chief information officer of the organization. This policy is usually two to ten pages long and shapes the philosophy of security in the IT environment.
- The EISP usually needs to be modified only when there is a change in the strategic direction of the organization. **The EISP guides the development, implementation, and management of the security program.** It sets out the requirements that must be met by the information security blueprint or framework. It defines the purpose, scope, constraints, and applicability of the security program.
- It also assigns **responsibilities for the various areas of security, including systems administration, maintenance of the information security policies, and the practices and responsibilities of the users.** Finally, it addresses **legal compliance.**
- According to the National Institute of Standards and Technology (NIST), the EISP typically addresses compliance in the following two areas:
 1. **General compliance to ensure meeting the requirements to establish a program and the responsibilities assigned therein to various organizational components.**
 2. **The use of specified penalties and disciplinary action.**

Issue Specific Security Policies

- As an organization executes various technologies and processes to support routine operations, it must instruct employees on the proper use of these technologies and processes.
- In general, the **issue-specific security policy**, or **ISSP**,
(1) addresses specific areas of technology as listed below,
(2) requires frequent updates, and
(3) contains a statement on the organization's position on a specific issue.
- **An ISSP may cover the following topics, among others:**
 - E-mail
 - Use of the Internet
 - Specific minimum configurations of computers to defend against worms and viruses
 - Prohibitions against hacking or testing organization security controls
 - Home use of company-owned computer equipment
 - Use of personal equipment on company networks
 - Use of telecommunications technologies (fax and phone)
 - Use of photocopy equipment

Systems specific Security Policies

- While issue-specific policies are formalized as written documents readily identifiable as policy, system-specific security policies (SysSPs) sometimes have a different look.
- **SysSPs often function as standards or procedures to be used when configuring or maintaining systems.**
- **For example, a SysSP might describe the configuration and operation of a network firewall.**
- This document could include a statement of managerial intent; guidance to network engineers on the selection, configuration, and operation of firewalls; and an access control list that defines levels of access for each authorized user.
- SysSPs can be separated into two general groups, **managerial guidance and technical specifications**, or they can be combined into a single policy document.

Introduction to Security in Distributed Systems

- Today, computers are not stand alone units. Several computers are being networked together to form large computer systems.
- Not only are computers being networked, but they are being networked into large distributed systems where each individual computer, node if you will, can make use of the applications distributed throughout the system.

Threat of Distributed Systems

There are different threats when distributed system is concerned, as any networked computer system can face it. It is important to implement countermeasures for all expected threats for the purpose of the system to remain constant and cost effective. Those threats can be distinguished depending on their interaction as follows below:

- 1. Denial of service:** Involves attacks that affect the availability of information from the system to the user resulting to paralysation of the entire operation of an organization or part of activities depending on the attack. The use of resource control mechanism can help in solving the above problem by applying timing responses, sizing responses, and connection control. Also problem detection by timing latency in system can easily be done if there is a dramatic increase of latency then denial of service (DoS) can be detected as well as addressed.
- 2. Information leakage:** is one of the threats of computer system specifically distributed systems where **sensitive information can easily be revealed to unauthorized users** that results to lack of confidentiality.
- 3. Unauthorized access:** This can occur due to the reason that the physical configuration is not strong enough to protect such threats from accessing the system (distributed system). This is known as inter process communication threats. Access control policies will enable organizations to be able to specify different ways that will lead to proper management of access to resources as well as information which are the valuable assets of an organization.

Distributed System Security Mechanism

- **Cryptography:** The security of information transmitted from one node to another is questionable, therefore there is a need of using a proper method of transforming it into unreadable formats (secrets writing) through cryptography. The use of a single key or public key cryptographic algorithm which is suitable for protecting message content by hiding information carried by a packet during the transmission process. This can be accomplished using RSA or AES algorithms.
- **Authentication protocol:** Provides a series of communication procedures between users of the system and the server for the purpose of securing the communication process.
- **Access control mechanism:** This can be done using access control lists (ACL) that consists of a list related to an object that states all the subjects that can be allowed to access the object, as well as the rights to the object. ACL normally are implemented directly or as an approximation in recent Operating systems.

Introduction to Database security

- **All systems have ASSETS and security is about protecting assets.** The first thing, then, is to know your assets and their value.
- The second thing to know is what THREATs are putting your assets at risk. These include things such as power failure and employee fraud. Note that threats are partly hypothetical, always changing and always imperfectly known. Security activity is directed at protecting the system from perceived threats.

Threats to the database

- **Unauthorised modification:** **Changing data values** for reasons of sabotage, crime or ignorance which may be enabled by inadequate security mechanisms, or sharing of passwords or password guessing, for example.
- **Unauthorised disclosure:** When information that **should not have been disclosed has been disclosed**. A general issue of crucial importance, which can be accidental or deliberate
- **Loss of availability:** Sometimes called denial of service. When the database is not available it incurs a loss (otherwise life is better without the system!). So any threat that gives rise to time offline, even to check whether something has occurred, is to be avoided.
- **Commercial sensitivity:** Most **financial losses through fraud arise from employees**. Access controls provide both protection against criminal acts and evidence of attempts (successful or otherwise) to carry out acts detrimental to the organisation, whether fraud, extraction of sensitive data or loss of availability.
- **Personal privacy and data protection:** Internationally, personal data is normally subject to legislative controls. Personal data is data about an identifiable individual. Often the individual has to be alive but the method of identification is not prescribed. So a postal code for a home may in some cases identify an individual, if only one person is living at an address with the postal code. Such data needs careful handling and control.

Contd...

- **Computer misuse:** There is also generally legislation on the misuse of computers. Misuse includes the violation of access controls and attempts to cause damage by changing the database state or introducing worms and viruses to interfere with proper operation. These offences are often extraditable. So an unauthorised access in Hong Kong using computers in France to access databases in Germany which refer to databases in America could lead to extradition to France or Germany or the USA.
- **Audit requirements:** These are operational constraints built around the need to know who did what, who tried to do what, and where and when everything happened. They involve the detection of events (including CONNECT and GRANT transactions), providing evidence for detection, assurance as well as either defence or prosecution. There are issues related to computer-generated evidence not covered here.

Database Security Models

A security model establishes the external criteria for the examination of security issues in general, and provides the context for database considerations, including implementation and operation.

- Authentication
- Authorization

Authentication

- The client has to establish the identity of the server and the server has to establish the identity of the client. This is done often by means of shared secrets (either a password/user-id combination, or shared biographic and/or biometric data). It can also be achieved by a system of higher authority which has previously established authentication.
- In client-server systems where data (not necessarily the database) is distributed, the authentication may be acceptable from a peer system. Note that authentication may be transmissible from system to system.
- The result, as far as the DBMS is concerned, is an authorisation-identifier. Authentication does not give any privileges for particular tasks. It only establishes that the DBMS trusts that the user is who he/she claimed to be and that the user trusts that the DBMS is also the intended system.
- **Authentication is a prerequisite for authorisation.**

Authorization

- Authorisation relates to the permissions granted to an authorised user to carry out particular transactions, and hence to change the state of the database (write item transactions) and/or receive data from the database (read-item transactions). The result of authorisation, which needs to be on a transactional basis, is a vector: Authorisation (item, auth-id, operation).
- A vector is a sequence of data values at a known location in the system. How this is put into effect is down to the DBMS functionality. At a logical level, the system structure needs an authorisation server, which needs to co-operate with an auditing server.
- There is an issue of server-to-server security and a problem with amplification as the authorisation is transmitted from system to system.
- Amplification here means that the security issues become larger as a larger number of DBMS servers are involved in the transaction. Audit requirements are frequently implemented poorly. To be safe, you need to log all accesses and log all authorisation details with transaction identifiers.
- There is a need to audit regularly and maintain an audit trail, often for a long period.

CO-INS:Information and Network Security

UNIT-II (Part-II) Modern Symmetric-Key Ciphers

Course Instructors:

Soma Saha

Veerendra Srivastava

Soma Saha (PhD)

Department of Computer Engineering
SGSITS Indore, India

March 31, 2021

UNIT-II (Part-II): Learning Objectives

Upon completion of this unit, you should be able to

- LO1 Explain the concept of modern block ciphers and discuss their characteristics
- LO2 Discuss the components of a modern block cipher
- LO3 Relate the concept of product ciphers and distinguish between the two classes of product ciphers
- LO4 Explain modern stream ciphers and discuss two broad categories—synchronous and non-synchronous

Cryptography: Modern Symmetric-Key Ciphers

- The traditional/classical symmetric-key ciphers (that we have studied so far) are **character-oriented ciphers**.
- With the advent of the computer, we need **bit-oriented ciphers**.
- Why??

Cryptography: Modern Symmetric-Key Ciphers

- The traditional/classical symmetric-key ciphers (that we have studied so far) are **character-oriented ciphers**.
- With the advent of the computer, we need **bit-oriented ciphers**.
- Why??
 - The information to be encrypted is not just text; it can also consists of numbers, graphics, audio, and video data.

Cryptography: Modern Symmetric-Key Ciphers

- The traditional/classical symmetric-key ciphers (that we have studied so far) are **character-oriented ciphers**.
- With the advent of the computer, we need **bit-oriented ciphers**.
- Why??
 - The information to be encrypted is not just text; it can also consists of numbers, graphics, audio, and video data.
 - It is convenient to convert these types of data into stream of bits, to encrypt the stream, and then to send the encrypted stream.

Cryptography: Modern Symmetric-Key Ciphers

- The traditional/classical symmetric-key ciphers (that we have studied so far) are **character-oriented ciphers**.
- With the advent of the computer, we need **bit-oriented ciphers**.
- Why??
 - The information to be encrypted is not just text; it can also consists of numbers, graphics, audio, and video data.
 - It is convenient to convert these types of data into stream of bits, to encrypt the stream, and then to send the encrypted stream.
 - Additionally, when text is treated at the bit level, each character is replaced by 8 (or 16) bits, which means that the number of symbols becomes 8(or 16) times larger. **Mixing a larger number of symbols increases security.**

Modern Block Cipher

- A symmetric-key **modern block cipher** encrypts an n -bit block of plaintext or decrypts an n -bit block of ciphertext. The encryption or decryption algorithm uses a k -bit key.

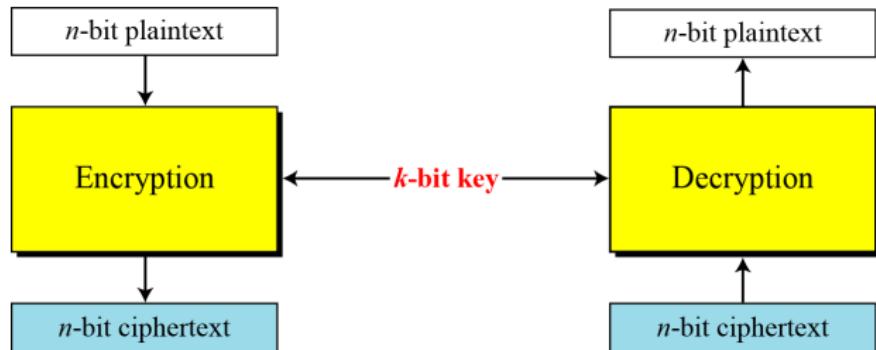


Figure 1: A modern block cipher.

Modern Block Cipher.. contd...1

- If the message has fewer than n bits, padding must be added to make it an n -bit block; if the message has more than n bits, it should be divided into n -bit blocks and appropriate padding must be added to the last block if necessary.
- **Common values for n ?**

Modern Block Cipher.. contd...1

- If the message has fewer than n bits, padding must be added to make it an n -bit block; if the message has more than n bits, it should be divided into n -bit blocks and appropriate padding must be added to the last block if necessary.
- **Common values for n ?**
 - The common values for n are 64, 128, 256, or 512 bits.

Modern Block Cipher.. contd...2

- How many padding bits must be added to a message of 100 characters if 8-bit ASCII is used for encoding and the block cipher accepts blocks of 64 bits?

Modern Block Cipher.. contd...2

- How many padding bits must be added to a message of 100 characters if 8-bit ASCII is used for encoding and the block cipher accepts blocks of 64 bits?
- Encoding 100 characters using 8-bit ASCII results in an 800-bit message. The plaintext must be divisible by 64. If $|M|$ and $|Pad|$ are the length of the message and the length of the padding,

$$|M| + |Pad| = 0 \bmod 64 \rightarrow |Pad| = -800 \bmod 64 \rightarrow 32 \bmod 64$$

Substitution or Transposition

- A modern block cipher can be designed to act as a **substitution cipher** or a **transposition cipher**.
- **Example:** If the cipher is designed as a substitution cipher, a 1-bit or a 0-bit in the plaintext can be replaced by either 0 or 1. This signifies that the ciphertext and plaintext can have a different number of 1's.

Substitution or Transposition

- A modern block cipher can be designed to act as a **substitution cipher** or a **transposition cipher**.
- **Example:** If the cipher is designed as a substitution cipher, a 1-bit or a 0-bit in the plaintext can be replaced by either 0 or 1. This signifies that the ciphertext and plaintext can have a different number of 1's.
 - a 64 bit plaintext block of 12 0's and 52 1's can be encrypted to a ciphertext of 34 0's and 30 1's.

Substitution or Transposition

- A modern block cipher can be designed to act as a **substitution cipher** or a **transposition cipher**.
- **Example:** If the cipher is designed as a substitution cipher, a 1-bit or a 0-bit in the plaintext can be replaced by either 0 or 1. This signifies that the ciphertext and plaintext can have a different number of 1's.
 - a 64 bit plaintext block of 12 0's and 52 1's can be encrypted to a ciphertext of 34 0's and 30 1's.
- If the cipher is designed as a transposition cipher, the bits are only reordered (transposed); there is same number of 1's in the plaintext and in the ciphertext.
- **Conclusion:** Modern block ciphers are designed as substitution ciphers to be resistant to exhaustive-search attack.

Modern block cipher: Substitution or Transposition:: Example

- Suppose that we have a block cipher where $n = 64$. If there are 10 1's in the ciphertext, how many trial-and-error tests does Eve need to do to recover the plaintext from the intercepted ciphertext in each of the following cases?
 - a. The cipher is designed as a substitution cipher.
 - b. The cipher is designed as a transposition cipher.
- **Solution:**

Modern block cipher: Substitution or Transposition:: Example

- Suppose that we have a block cipher where $n = 64$. If there are 10 1's in the ciphertext, how many trial-and-error tests does Eve need to do to recover the plaintext from the intercepted ciphertext in each of the following cases?
 - a. The cipher is designed as a substitution cipher.
 - b. The cipher is designed as a transposition cipher.
- **Solution:**
 - a. In the first case, Eve has no idea how many 1's are in the plaintext. Eve needs to try all possible 2^{64} 64-bit blocks to find one that makes sense.
 - If Eve could try 1 billion blocks per second, it would still take hundreds of years, on average, before she could be successful.

Modern block cipher: Substitution or Transposition:: Example

- Suppose that we have a block cipher where $n = 64$. If there are 10 1's in the ciphertext, how many trial-and-error tests does Eve need to do to recover the plaintext from the intercepted ciphertext in each of the following cases?
 - The cipher is designed as a substitution cipher.
 - The cipher is designed as a transposition cipher.
- Solution:**
 - In the first case, Eve has no idea how many 1's are in the plaintext. Eve needs to try all possible 2^{64} 64-bit blocks to find one that makes sense.
 - If Eve could try 1 billion blocks per second, it would still take hundreds of years, on average, before she could be successful.
 - In the second case, Eve knows that there are exactly 10 1's in the plaintext. Eve can launch an exhaustive-search attack using only those 64-bit blocks that have exactly 10 1's.

$$\binom{64}{10} = \frac{64!}{(10!)(54!)} = 151,473,214,816$$

(Less than 3 min...)

Block Ciphers as Permutation Groups

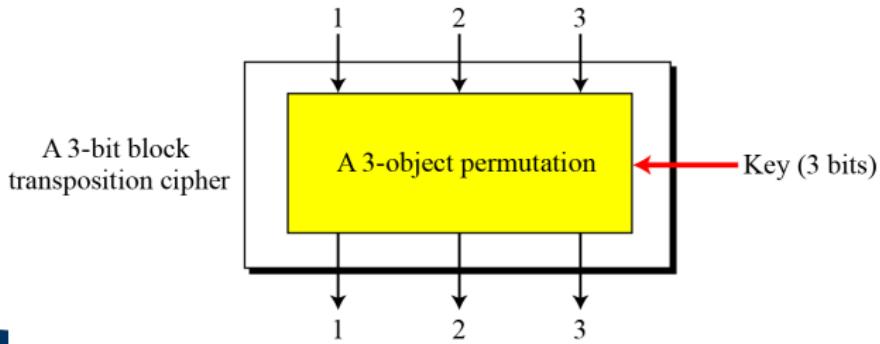
- **Full-size key ciphers:** the key is long enough to choose every possible mapping from input to output. In practice, the key is smaller (partial-key), only some mappings from the input to output are possible.
 - Full-size key ciphers are not used in practice, only partial-key ciphers are used.

Block Ciphers as Permutation Groups

- **Full-size key ciphers:** the key is long enough to choose every possible mapping from input to output. In practice, the key is smaller (partial-key), only some mappings from the input to output are possible.
 - Full-size key ciphers are not used in practice, only partial-key ciphers are used.
- **Full-Size Key Transposition Block Ciphers:** In a full-size key transposition cipher We need to have $n!$ possible keys, so the key should have $\lceil \log_2 n! \rceil$ bits.
 - Only transposes bits without changing their values.
 - So, it can be modeled as an n -object permutation with a set of $n!$ permutation tables in which the key defines which table is used by Alice and Bob.

Full-Size Key Transposition Block Ciphers: Example

- Show the model and the set of permutation tables for a 3-bit block transposition cipher where the block size is 3 bits.
- The set of permutation tables has $3! = 6$ elements, as shown below:



$\{[1\ 2\ 3], [1\ 3\ 2], [2\ 1\ 3], [2\ 3\ 1], [3\ 1\ 2], [3\ 2\ 1]\}$

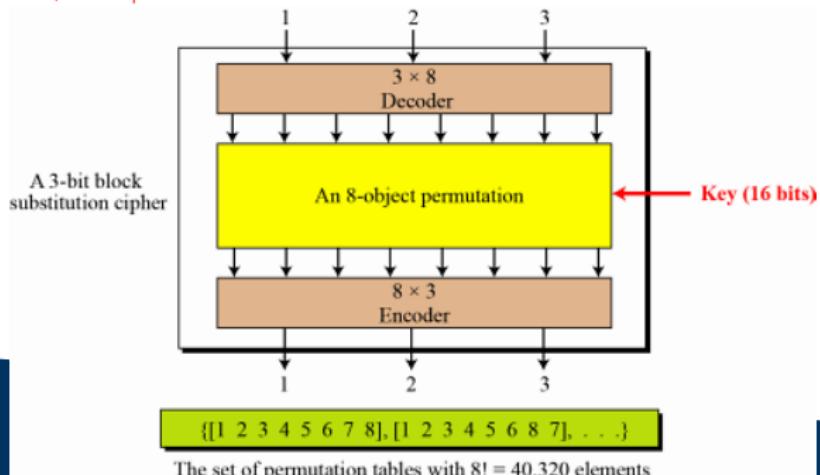
The set of permutation tables with $3! = 6$ elements

Full-Size Key Substitution Block Ciphers

- **Full-Size Key Substitution Block Ciphers:** A full-size key substitution cipher does not transpose bits; it substitutes bits. We can model the substitution cipher as a permutation if we can decode the input and encode the output.
 -
 - We can model the substitution cipher as a permutation if we can decode the input and encode the output.
 - **Decoding** means transforming an n -bit integer into a 2^n -bit string with only a single 1 and $2^n - 1$ 0's.
 - The position of the single 1 is the value of the integer, in which the positions range from 0 to $2^n - 1$.
 - **Encoding** is the reverse process. As the new input and output have always a single 1, the cipher can be modeled as a permutation of 2^n objects.

Full-Size Key Substitution Block Ciphers: Example

- Example: Show the model and the set of permutation tables for a 3-bit block substitution cipher.
- The figure shows the model and the set of permutation tables. The key is much longer, $\lceil \log_2 40,320 \rceil = 16$ bits.



NOTE:

- A full-size key n-bit transposition cipher or a substitution block cipher can be modeled as a permutation, but their key sizes are different:
 - Transposition: the key is $\lceil \log_2 n! \rceil$ bits long.
 - Substitution: the key is $\lceil \log_2(2^n)! \rceil$ bits long.

Partial-Size Key Cipher

- Two or more cascaded permutations can be always replaced with a single permutation. Hence it is useless to have more than one stage of full-size key ciphers, because the effect is the same as having a single stage.
- Modern block ciphers normally are keyed substitution ciphers in which the key allows only partial mappings from the possible inputs to the possible outputs.

Partial-Size Key Cipher

- Two or more cascaded permutations can be always replaced with a single permutation. Hence it is useless to have more than one stage of full-size key ciphers, because the effect is the same as having a single stage.
- Modern block ciphers normally are keyed substitution ciphers in which the key allows only partial mappings from the possible inputs to the possible outputs.
- **For example**, a common substitution cipher is DES (Data Encryption Standard) which uses a 64-bit block cipher. If the designer of DES had used a full-size key, the key would have been $\log_2(2^{64})! = 2^{70}$ bits. The key size for DES is only 56 bits which is only a very small fraction of the full-size key. This means that DES uses only 2^{56} mappings out of approximately $2^{2^{70}}$ possible mappings.

Components of a Modern Block Cipher

- In cryptography, **confusion** and **diffusion** are two properties of the operation of a secure cipher identified by **Claude Shannon** in his 1945 classified report: "A Mathematical Theory of Cryptography".

Diffusion

- **Diffusion:** Diffusion means that if we change a single bit of the plaintext, then (statistically) half of the bits in the ciphertext should change, and similarly, if we change one bit of the ciphertext, then approximately one half of the plaintext bits should change. Since a bit can have only two states, when they are all re-evaluated and changed from one seemingly random position to another, half of the bits will have changed state.
 - **The idea of diffusion is to hide the relationship between the ciphertext and the plain text.**
 - This will make it hard for an attacker who tries to find out the plain text and it increases the redundancy of plain text by spreading it across the rows and columns; it is achieved through transposition of algorithm and it is used by block ciphers only.

Confusion

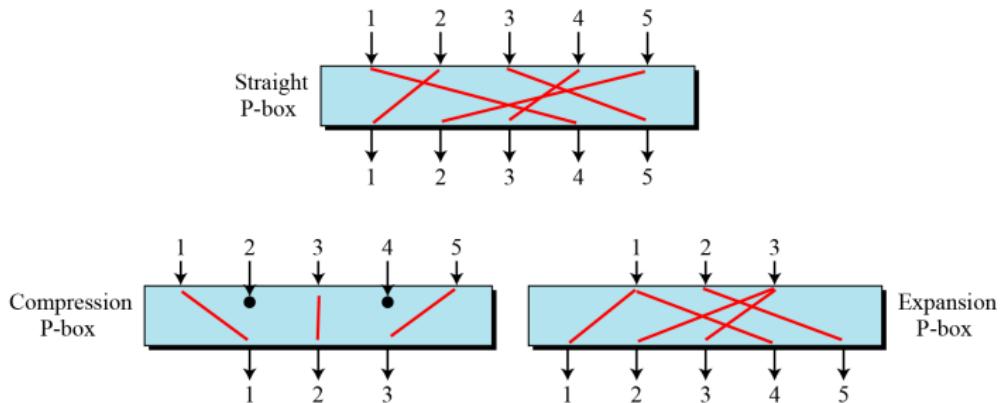
- Confusion means that each binary digit (bit) of the ciphertext should depend on several parts of the key, obscuring the connections between the two.
- **The property of confusion hides the relationship between the ciphertext and the key.**
- This property makes it difficult to find the key from the ciphertext and if a single bit in a key is changed, the calculation of the values of most or all of the bits in the ciphertext will be affected.
- Confusion increases the ambiguity of ciphertext and it is used by both block and stream ciphers.

Components of a Modern Block Cipher

- To provide required properties of a modern block cipher, such as **diffusion** and **confusion**, a modern block cipher is made of a combination of
 - transposition units for diffusion (called **D-boxes** or **P-boxes** for permutation),
 - substitution units (called **S-boxes**),
 - and some other units.

P-boxes or D-boxes

- * A P-box (permutation box) or D-box (diffusion box) parallels the traditional transposition cipher for characters. It transposes bits.



Straight P-boxes (or D-boxes)

- **straight P-boxes (or D-boxes)** : all 6 possible mappings of a 3×3 D-box.

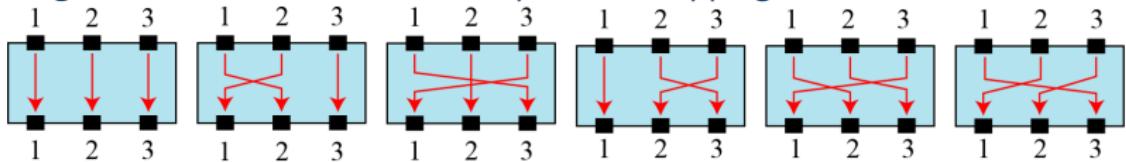


Figure 2: Although a P-box can use a key to define one of the $n!$ mappings, P boxes are normally keyless, which means the mapping is predetermined. .

Straight P-boxes (or D-boxes)

- **straight P-boxes (or D-boxes)** : all 6 possible mappings of a 3×3 D-box.

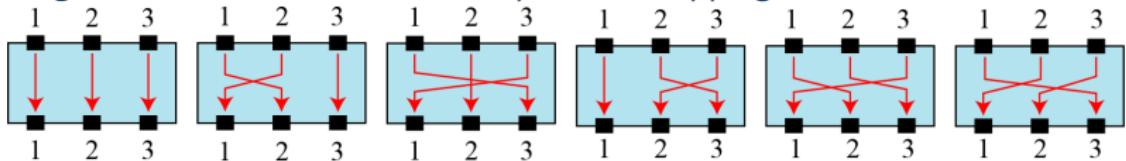


Figure 2: Although a P-box can use a key to define one of the $n!$ mappings, P boxes are normally keyless, which means the mapping is predetermined. .

58	50	42	34	26	18	10	02	60	52	44	36	28	20	12	04
62	54	46	38	30	22	14	06	64	56	48	40	32	24	16	08
57	49	41	33	25	17	09	01	59	51	43	35	27	19	11	03
61	53	45	37	29	21	13	05	63	55	47	39	31	23	15	07

Example

- Design an 8×8 permutation table for a straight P-box that moves the two middle bits (bits 4 and 5) in the input word to the two ends (bits 1 and 8) in the output words. Relative positions of other bits should not be changed.

Example

- Design an 8×8 permutation table for a straight P-box that moves the two middle bits (bits 4 and 5) in the input word to the two ends (bits 1 and 8) in the output words. Relative positions of other bits should not be changed.
- **Solution:** We need a straight P-box with the table [4 1 2 3 6 7 8 5]. The relative positions of input bits 1, 2, 3, 6, 7, and 8 have not been changed, but the first output takes the fourth input and the eighth output takes the fifth input.

Compression P-Boxes (or D-boxes)

- A compression P-box is a P-box with n inputs and m outputs where m < n.

01	02	03	21	22	26	27	28	29	13	14	17
18	19	20	04	05	06	10	11	12	30	31	32

Figure 4: Example of a 32×24 permutation table.

Expansion P-Boxes (or D-boxes)

- An expansion P-box is a P-box with n inputs and m outputs where $m > n$

01	09	10	11	12	01	02	03	03	04	05	06	07	08	09	12
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

Figure 5: Example of a 12×16 permutation table.

P-Boxes/D-Boxes: Invertibility

- A straight P-box is invertible, but compression and expansion P-boxes are not.

P-Boxes/D-Boxes: Invertibility

- A straight P-box is invertible, but compression and expansion P-boxes are not.
- How can you invert a permutation table to be represented as a one-dimensional table?

1. Original table

6	3	4	5	2	1
---	---	---	---	---	---

2. Add indices

6	3	4	5	2	1
1	2	3	4	5	6

3. Swap contents
and indices

1	2	3	4	5	6
6	3	4	5	2	1

4. Sort based
on indices

6	5	2	3	4	1
1	2	3	4	5	6

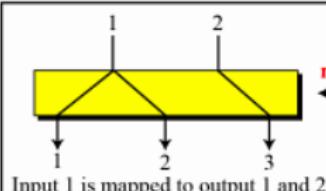
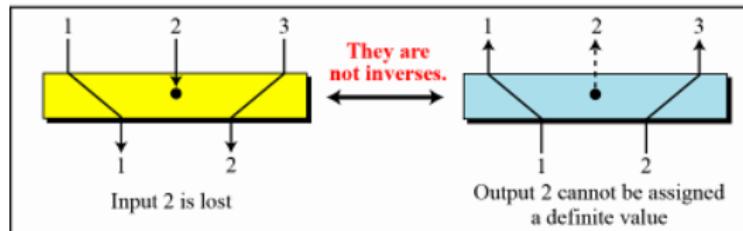
6	5	2	3	4	1
---	---	---	---	---	---

5. Inverted table

Figure 6: Inverting a permutation table.

Compression and expansion P-boxes are non-invertible

Compression P-box



Expansion P-box

S-boxes

- An S-box (substitution box) can be thought of as a miniature substitution cipher.
- **An S-box is an $m \times n$ substitution unit, where m and n are not necessarily the same.**
- For example: The following table defines the input/output relationship for an S-box of size 3×2 . The leftmost bit of the input defines the row; the two rightmost bits of the input define the column. The two output bits are values on the cross section of the selected row and column.

Leftmost bit

Rightmost bits

	00	01	10	11
0	00	10	01	11
1	10	00	11	01

Output bits

S-Boxes: Invertibility

- An S-box may or may not be invertible. In an invertible S-box, the number of input bits should be the same as the number of output bits.

S-Boxes: Invertibility.. contd

- The following shows an example of an invertible S-box. For example, if the input to the left box is 001, the output is 101. The input 101 in the right table creates the output 001, which shows that the two tables are inverses of each other.

3 bits

↓

		00	01	10	11
		011	101	111	100
0	000	010	001	110	
1					

Table used for
encryption

3 bits

↓

3 bits

↓

		00	01	10	11
		100	110	101	000
0	011	001	111	010	
1					

Table used for
decryption

3 bits

↓

XOR (Exclusive-Or)

- An important component in most block ciphers is the exclusive-or operation.

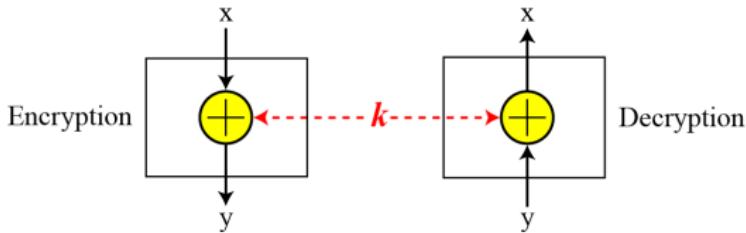


Figure 10: Invertibility of the exclusive-or operation.

XOR.. Contd..

- An important component in most block ciphers is the exclusive-or operation.
- **The five properties of the exclusive-or operation makes this operation a very interesting component for use in a block cipher: closure, associativity, commutativity, existence of identity, and existence of inverse.**

$$X \text{ EXOR } 0 = X$$

$$X \text{ EXOR } 1 = \bar{X}$$

$$X \text{ EXOR } \bar{X} = 1$$

$$X \text{ EXOR } X = 0$$

Exclusive OR ...Contd

- The inverse of a component in a cipher makes sense if the component represents a unary operation (one input and one output).

Exclusive OR ...Contd

- The inverse of a component in a cipher makes sense if the component represents a unary operation (one input and one output).
- For example, a keyless P-box or a keyless S-box can be made invertible because they have one input and one output.
- An exclusive-or operation is a binary operation. The inverse of an exclusive-or operation can make sense only if one of the inputs is fixed (is the same in encryption and decryption).

Exclusive OR ...Contd

- The inverse of a component in a cipher makes sense if the component represents a unary operation (one input and one output).
- For example, a keyless P-box or a keyless S-box can be made invertible because they have one input and one output.
- An exclusive-or operation is a binary operation. The inverse of an exclusive-or operation can make sense only if one of the inputs is fixed (is the same in encryption and decryption).
- For example, if one of the inputs is the key, which normally is the same in encryption and decryption, then an exclusive-or operation is self-invertible, as shown in Last Figure. 10.

Circular Shift

- Another component used in some modern block ciphers is the **circular shift** operation.

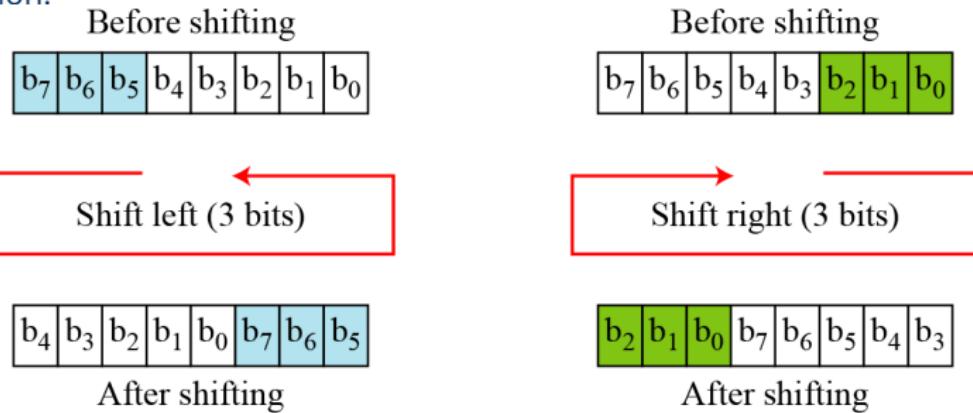


Figure 11: Circular shifting an 8-bit word to the left or right.

Swap

- The **swap** operation is a special case of the circular shift operation where $k = n/2$.

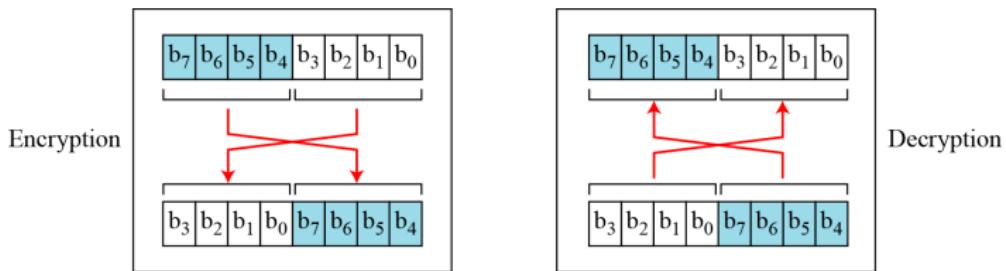


Figure 12: Swap operation on an 8-bit word.

Split and Combine

- Two other operations found in some block ciphers are split and combine.

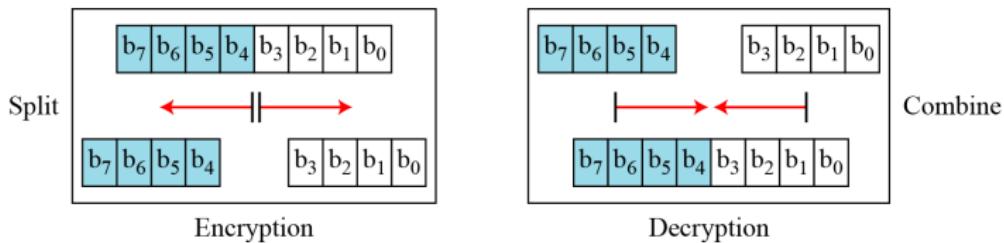


Figure 13: Split and combine operations on an 8-bit word.

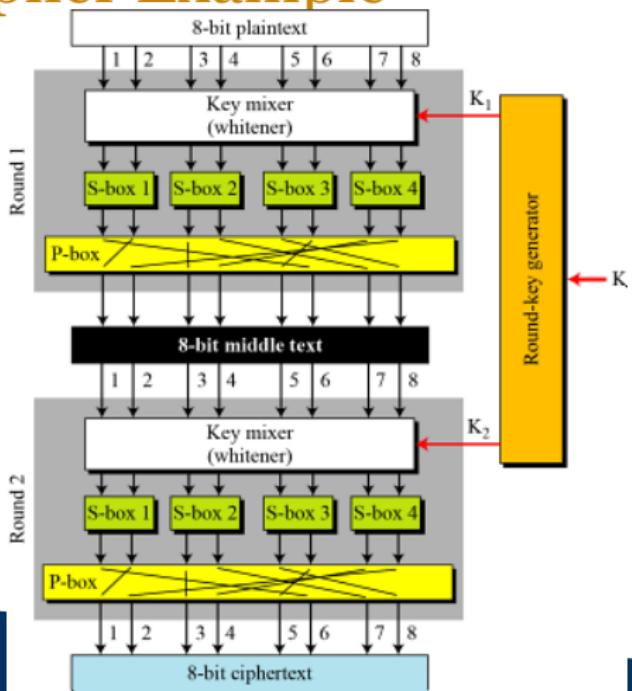
Product Cipher

- Shannon introduced the concept of **product cipher**.
- A **product cipher** is a complex cipher combining substitution, permutation, and other components discussed in previous sections.

Diffusion, confusion, and Rounds

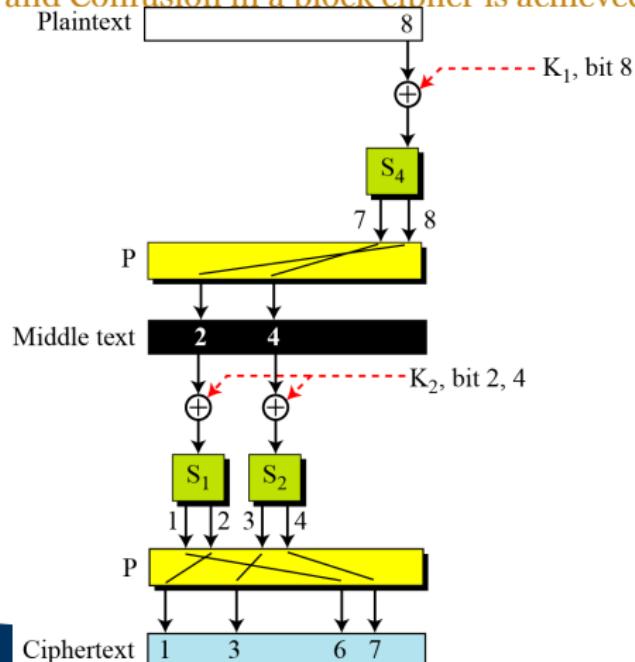
- **Diffusion:** The idea to hide the relationship between the ciphertext and the plaintext.
- **Confusion:** The idea to hide the relationship between the ciphertext and the key.
- **Rounds:** Diffusion and confusion can be achieved using iterated product ciphers where each iteration is a combination of S-boxes, P-boxes/D-boxes, and other components.

Product cipher Example



A product cipher made of two rounds

How does Diffusion and Confusion in a block cipher is achieved?



Classification of Product Ciphers

- Modern block ciphers are all product ciphers, but they are divided into two classes.
 - **Feistel ciphers**
 - **Non-Feistel ciphers**

Classification of Product Ciphers

- Modern block ciphers are all product ciphers, but they are divided into two classes.
- **Feistel ciphers**
 - Has been used for decades.
 - Can have three types of components : **self-invertible**, **invertible**, and **non-invertible**.
 - Example: **DES**
- **Non-Feistel ciphers:**
 - Uses only **invertible components**.
 - A component in the encryption cipher has the corresponding component in the decryption cipher.
 - Example: **AES**

Feistel Ciphers: First Thought

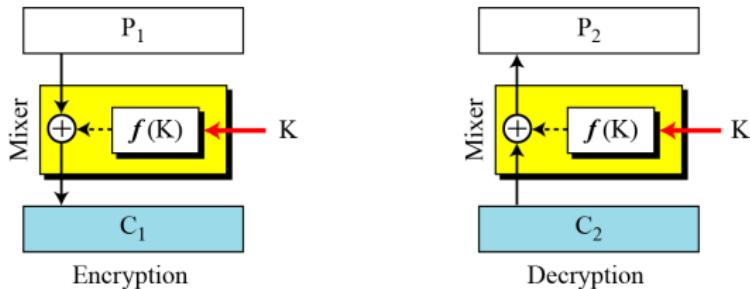


Figure 16: The first thought in Feistel cipher design: **f(K) is a non-invertible function.**

- **Encryption:** $C_1 = P_1 \oplus f(K)$
- **Decryption:**

$$P_2 = C_2 \oplus f(K) = C_1 \oplus f(K) = P_1 \oplus f(K) \oplus f(K) = P_1 \oplus (00\dots 0) = P_1$$

Feistel Ciphers: First Thought

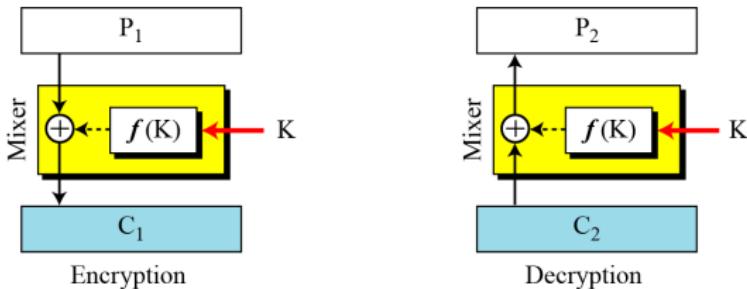


Figure 16: The first thought in Feistel cipher design: **f(K) is a non-invertible function.**

- **Encryption:** $C_1 = P_1 \oplus f(K)$
- **Decryption:**
$$P_2 = C_2 \oplus f(K) = C_1 \oplus f(K) = P_1 \oplus f(K) \oplus f(K) = P_1 \oplus (00\dots 0) = P_1$$
- **The mixer(combination of function and ex-or operation) in the Feistel design is self-invertible.**

Example

- This is a trivial example. The plaintext and ciphertext are each 4 bits long and the key is 3 bits long. Assume that the function takes the first and third bits of the key, interprets these two bits as a decimal number, squares the number, and interprets the result as a 4-bit binary pattern. Show the results of encryption and decryption if the original plaintext is 0111 and the key is 101.

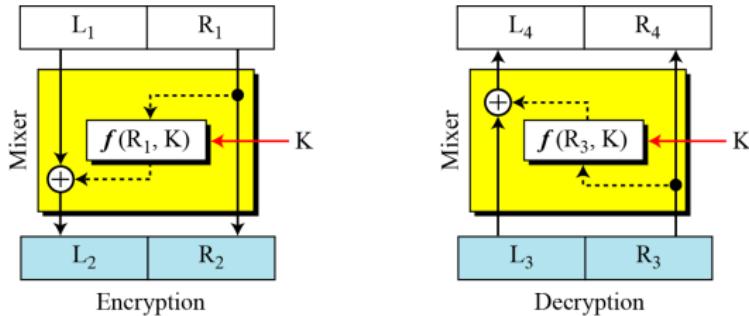
Example

- This is a trivial example. The plaintext and ciphertext are each 4 bits long and the key is 3 bits long. Assume that the function takes the first and third bits of the key, interprets these two bits as a decimal number, squares the number, and interprets the result as a 4-bit binary pattern. Show the results of encryption and decryption if the original plaintext is 0111 and the key is 101.
- Solution:**
 - The function extracts the first and second bits to get 11 in binary or 3 in decimal. The result of squaring is 9, which is 1001 in binary.

Encryption: $C = P \oplus f(K) = 0111 \oplus 1001 = 1110$

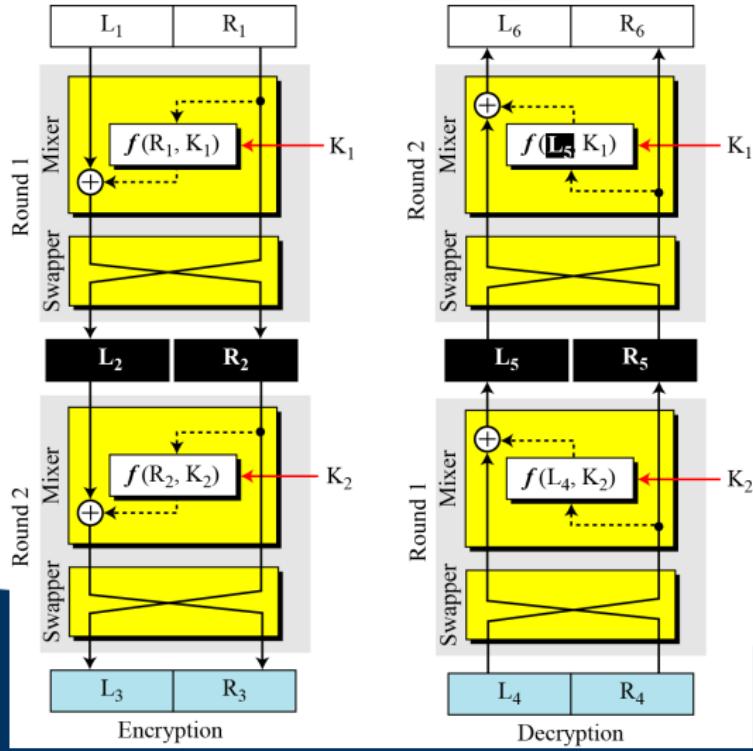
Decryption: $P = C \oplus f(K) = 1110 \oplus 1001 = 0111$

Improvement of the previous Feistel design



- $R_4 = R_3 = R_2 = R_1$
- $L_4 = L_3 \oplus f(R_3, K) = L_2 \oplus f(R_2, K) = L_1 \oplus f(R_1, K) \oplus f(R_1, K) = L_1$
- **Flaw in this design: Right half of the plaintext never changes.**

Final design of a Feistel cipher with two rounds



Feistel Cipher design.. contd..

- Let us see if $L_6 = L_1$ and $R_6 = R_1$, assuming that $L_4 = L_3$ and $R_4 = R_3$ (no change in ciphertext during transmission).
- We first prove the equality for the middle text.

$$\begin{aligned}L_5 &= R_4 \oplus f(L_4, K_2) = R_3 \oplus f(R_2, K_2) = L_2 \oplus f(R_2, K_2) \oplus f(R_2, K_2) = L_2 \\R_5 &= L_4 = L_3 = R_2\end{aligned}$$

- Then it is easy to prove that the equality holds for two plaintext blocks.
$$\begin{aligned}L_6 &= R_5 \oplus f(L_5, K_1) = R_2 \oplus f(L_2, K_1) = L_1 \oplus f(R_1, K_1) \oplus f(R_1, K_1) = L_1 \\R_6 &= L_5 = L_2 = R_1\end{aligned}$$

Feistel Ciphers

- Blowfish.
- Camellia.
- CAST-128.
- DES.
- FEAL.
- GOST 28147-89.
- ICE.

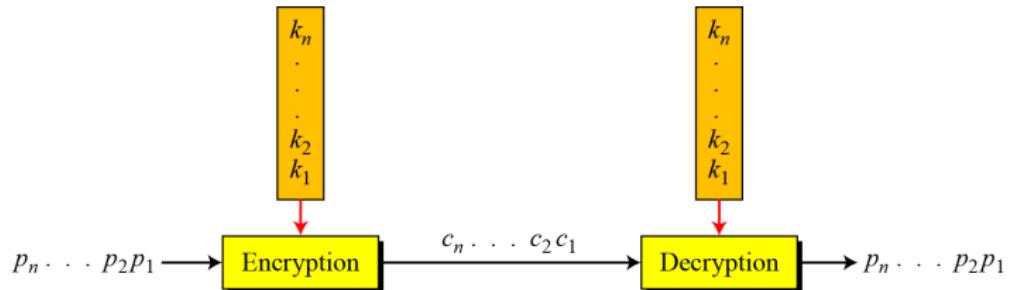
Modern Block Cipher: Secure??

- Attacks on traditional ciphers can also be used on modern block ciphers, but today's block ciphers resist most of the attacks discussed in classes/unit2_Cryptography_part1 slides.
- **Linear cryptanalysis and Differential cryptanalysis are the two most widely used attacks on block ciphers.**
- Eli Biham and Adi Shamir introduced the idea of **differential cryptanalysis**. This is a chosen-plaintext attack.
 - Differential cryptanalysis is based on a nonuniform differential distribution table of the S-boxes in a block cipher.
- **Linear cryptanalysis** was presented by Mitsuru Matsui in 1993. The analysis uses known plaintext attacks.

Modern Stream Ciphers

- In a modern stream cipher, encryption and decryption are done r bits at a time. We have a plaintext bit stream $P = p_n \dots p_2 p_1$, a ciphertext bit stream $C = c_n \dots c_2 c_1$, and a key bit stream $K = k_n \dots k_2 k_1$, in which p_i , c_i , and k_i are r -bit words.
- Encryption is $c_i = E(k_i, p_i)$, and
- Decryption is $p_i = D(k_i, c_i)$.
- **Classification:**
 1. Synchronous Stream Ciphers
 2. Nonsynchronous Stream Ciphers

Stream Cipher



- In a modern stream cipher, each r-bit word in the plaintext stream is enciphered using an r-bit word in the key stream to create the corresponding r-bit word in the ciphertext stream.

Synchronous Stream Ciphers

- In a synchronous stream cipher the key is independent of the plaintext or ciphertext.

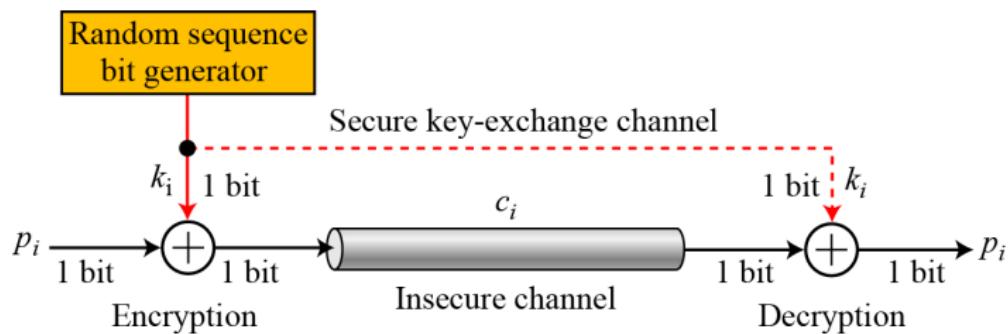


Figure 17: **One-time pad**: one-time pad invented and patented by Gilbert Vernam..

Example

- What is the pattern in the ciphertext of a one-time pad cipher in each of the following cases?
 - a. The plaintext is made of n 0's.
 - b. The plaintext is made of n 1's.
 - c. The plaintext is made of alternating 0's and 1's.
 - d. The plaintext is a random string of bits.

Example

- What is the pattern in the ciphertext of a one-time pad cipher in each of the following cases?
 - a. The plaintext is made of n 0's.
 - b. The plaintext is made of n 1's.
 - c. The plaintext is made of alternating 0's and 1's.
 - d. The plaintext is a random string of bits.
- **solution**
 - a. Because $0 \oplus k_i = k_i$, the ciphertext stream is the same as the key stream. If the key stream is random, the ciphertext is also random. The patterns in the plaintext are not preserved in the ciphertext.

Example..contd...

- Because $1 \oplus k_i = \bar{k}_i$ where \bar{k}_i is the complement of k_i , the ciphertext stream is the complement of the key stream. If the key stream is random, the ciphertext is also random. Again the patterns in the plaintext are not preserved in the ciphertext.
- In this case, each bit in the ciphertext stream is either the same as the corresponding bit in the key stream or the complement of it. Therefore, the result is also a random string if the key stream is random.
- In this case, the ciphertext is definitely random because the exclusive-or of two random bits results in a random bit.

Feedback Shift Register

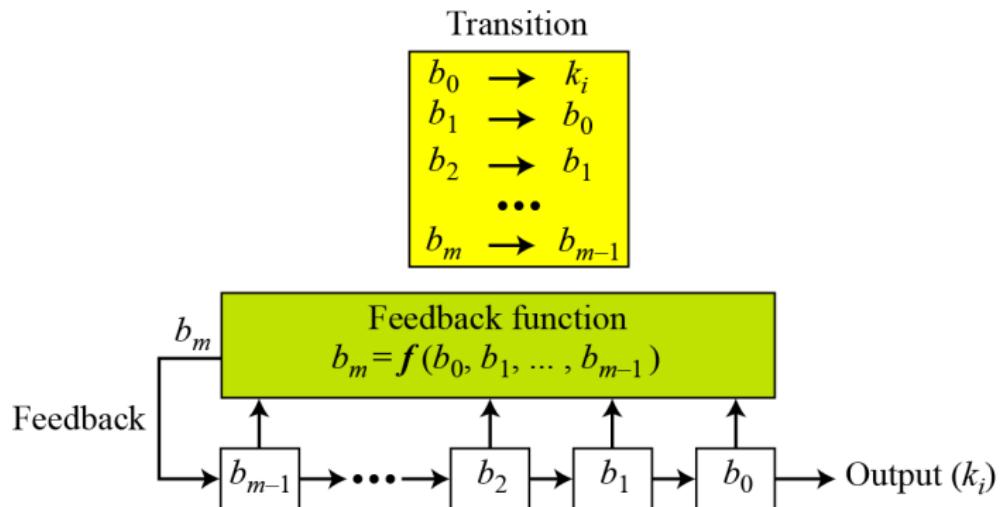
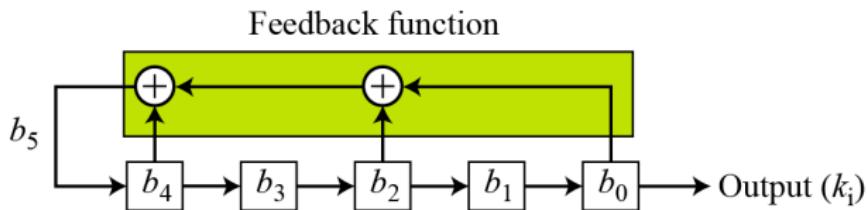


Figure 18: Feedback shift Register.

Example..1

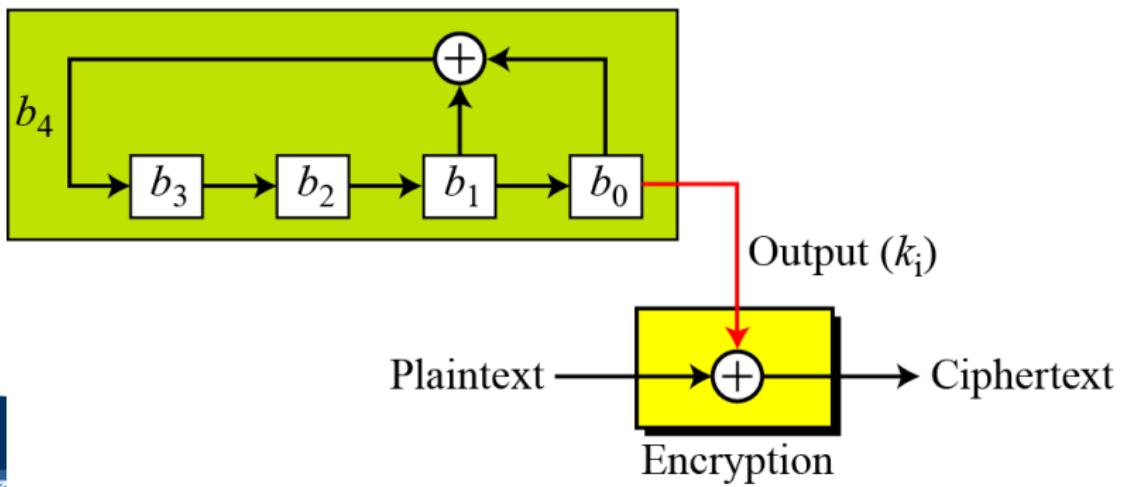
- Create a linear feedback shift register with 5 cells in which $b_5 = b_4 \oplus b_2 \oplus b_0$.
- If $c_i = 0$, b_i has no role in calculation of b_m . This means that b_i is not connected to the feedback function. If $c_i = 1$, b_i is involved in calculation of b_m . In this example, c_1 and c_3 are 0's, which means that we have only three connections. following figure shows the design.



Example..2

- Create a linear feedback shift register with 4 cells in which $b_4 = b_1 \oplus b_0$. Show the value of output for 20 transitions (shifts) if the seed is $(0001)_2$.

Key stream generator



Example2: Cell values and key sequence..

States	b_4	b_3	b_2	b_1	b_0	k_i
Initial	1	0	0	0	1	
1	0	1	0	0	0	1
2	0	0	1	0	0	0
3	1	0	0	1	0	0
4	1	1	0	0	1	0
5	0	1	1	0	0	1
6	1	0	1	1	0	0
7	0	1	0	1	1	0
8	1	0	1	0	1	1
9	1	1	0	1	0	1
10	1	1	1	0	1	0

Example2: Cell values and key sequence... cont..2

11	1	1	1	1	0	1
12	0	1	1	1	1	0
13	0	0	1	1	1	1
14	0	0	0	1	1	1
15	1	0	0	0	1	1
16	0	1	0	0	0	1
17	0	0	1	0	0	0
18	1	0	0	1	0	0
19	1	1	0	0	1	0
20	1	1	1	0	0	1

Nonsynchronous Stream Cipher

- In a nonsynchronous stream cipher, each key in the key stream depends on previous plaintext or ciphertext.

Data Encryption Standard (DES)

- The most widely used cipher in civilian applications.
- Developed by IBM; Evolved from Lucifer.
- Accepted as an US NBS standard in 1977, and later as an international standard, the National Institute of Standards and Technology (NIST).
- A block cipher with **N = 64 bit blocks**.
- **56-bit keys** (eight bytes, in each byte seven bits are used; the eighth bit can be used as a parity bit).
- Exhaustive search requires 2^{56} encryption steps (2^{55} on average).
- Iterates a round-function 16 times in **16 rounds**. The round-function mixes the data with the key.
- Each round, the key information entered to the round function is called a subkey. The subkeys K_1, \dots, K_{16} are computed by **a key scheduling algorithm**.

DES Overview

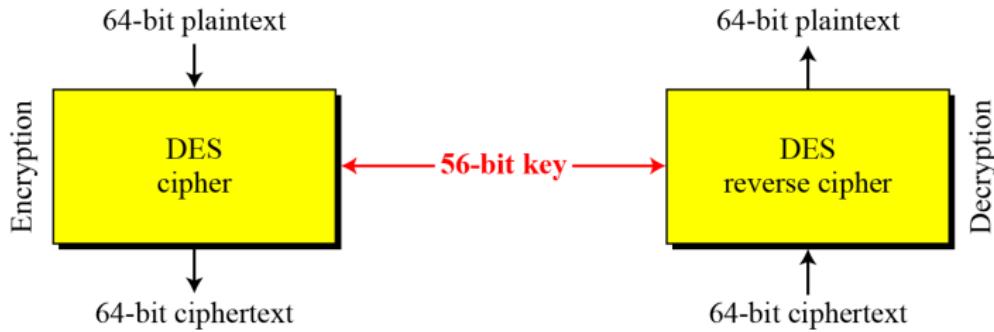
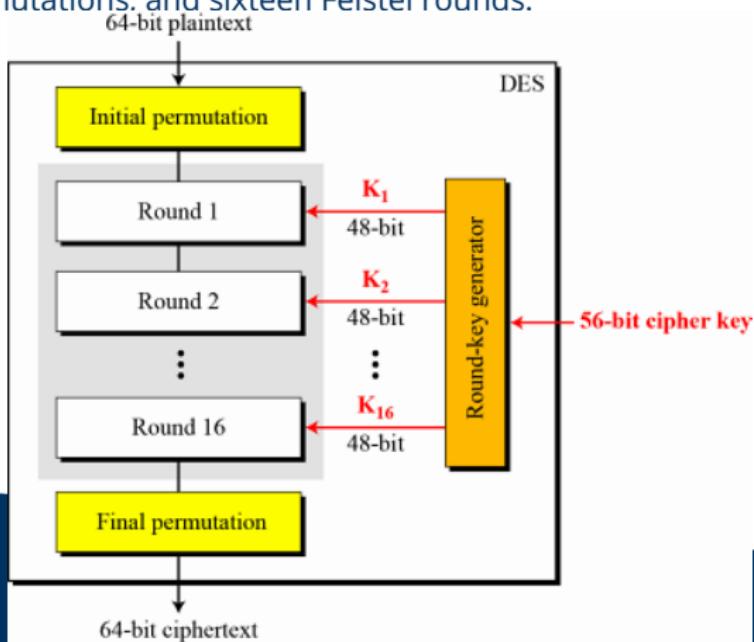


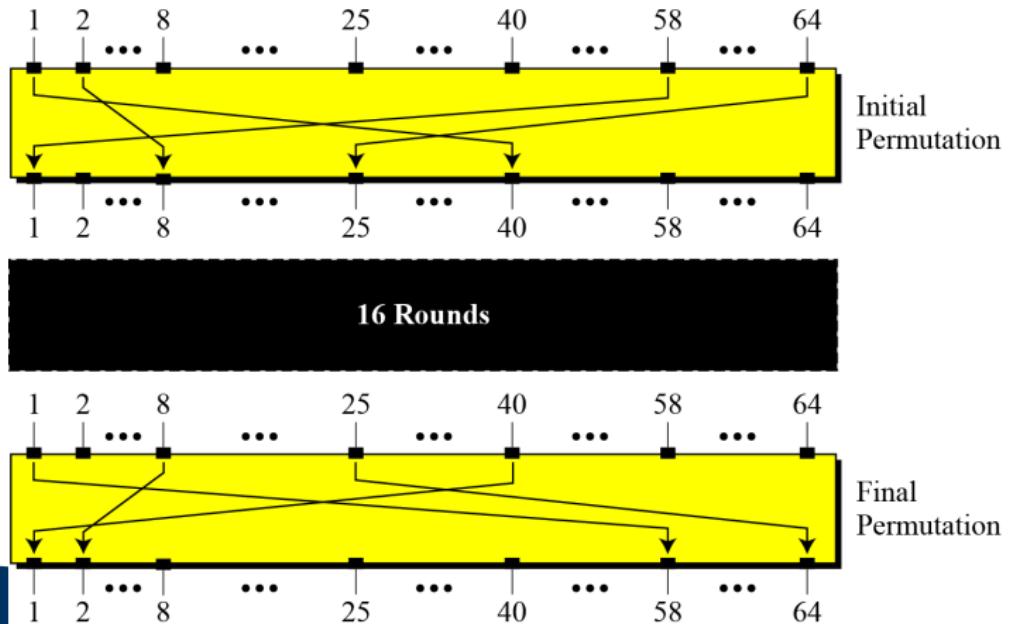
Figure 19: Encryption and decryption with DES.

General Structure of DES

- The encryption process is made of two permutations (P-boxes), which we call initial and final permutations, and sixteen Feistel rounds.



Initial and final permutation in DES



Initial and final permutation tables

<i>Initial Permutation</i>	<i>Final Permutation</i>
58 50 42 34 26 18 10 02	40 08 48 16 56 24 64 32
60 52 44 36 28 20 12 04	39 07 47 15 55 23 63 31
62 54 46 38 30 22 14 06	38 06 46 14 54 22 62 30
64 56 48 40 32 24 16 08	37 05 45 13 53 21 61 29
57 49 41 33 25 17 09 01	36 04 44 12 52 20 60 28
59 51 43 35 27 19 11 03	35 03 43 11 51 19 59 27
61 53 45 37 29 21 13 05	34 02 42 10 50 18 58 26
63 55 47 39 31 23 15 07	33 01 41 09 49 17 57 25

Figure 22: Initial and final permutation tables in DES.

Example1

- Find the output of the final permutation box when the input is given in hexadecimal as:

0x0000 0080 0000 0002

- Solution:**

Only bit 25 and bit 63 are 1s; the other bits are 0s. In the final permutation, bit 25 becomes bit 64 and bit 63 becomes bit 15. The result is

0x0002 0000 0000 0001

Example2

- Prove that the initial and final permutations are the inverse of each other by finding the output of the initial permutation if the input is

0x0002 0000 0000 0001

- Solution:** The input has only two 1s; the output must also have only two 1s. Using Table 6.1, we can find the output related to these two bits. Bit 15 in the input becomes bit 63 in the output. Bit 64 in the input becomes bit 25 in the output. So the output has only two 1s, bit 25 and bit 63. The result in hexadecimal is

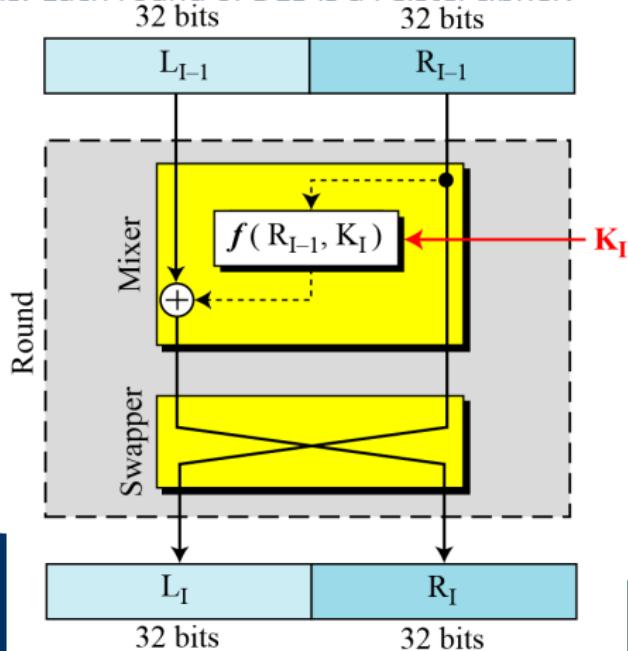
0x0000 0080 0000 0002

Cryptographic significance of initial and final permutations

- The initial and final permutations are straight P-boxes that are inverses of each other. They have no cryptography significance in DES.

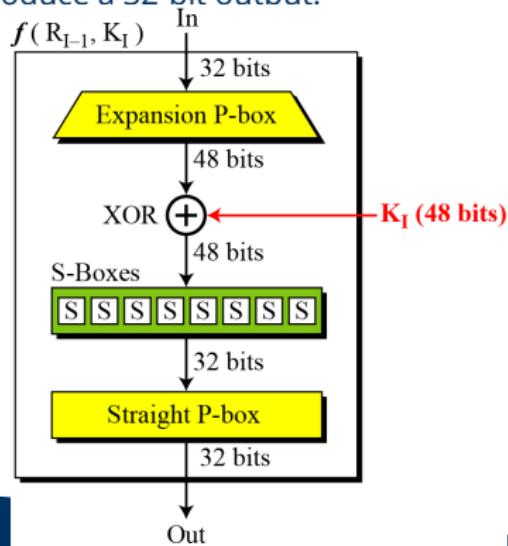
Rounds

- DES uses 16 rounds. Each round of DES is a Feistel cipher.



DES function

- The heart of DES is the DES function. The DES function applies a 48-bit key to the rightmost 32 bits to produce a 32-bit output.



Expansion P-Box in DES

- Since R_{i-1} is a 32-bit input and K_i is a 48-bit key, we first need to expand R_{i-1} to 48 bits.

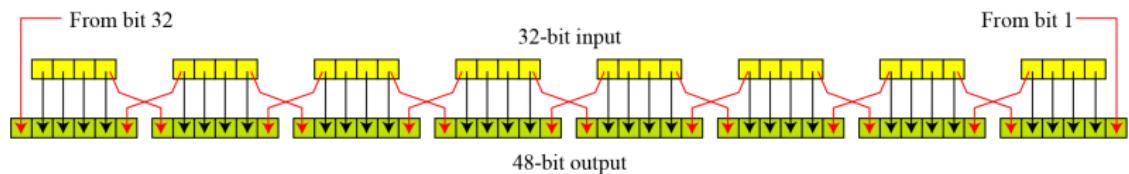


Figure 25: Expansion P-box

Contd...

- Although the relationship between the input and output can be defined mathematically, DES uses Table 6.2 to define this P-box.

32	01	02	03	04	05
04	05	06	07	08	09
08	09	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	31	31	32	01

Figure 26: Expansion P-box Table

Whitener (XOR) in DES

- After the expansion permutation, DES uses the XOR operation on the expanded right section and the round key. Note that both the right section and the key are 48-bits in length. Also note that the round key is used only in this operation.

S-Boxes in DES

- The S-boxes do the real mixing (confusion). DES uses 8 S-boxes, each with a 6-bit input and a 4-bit output.

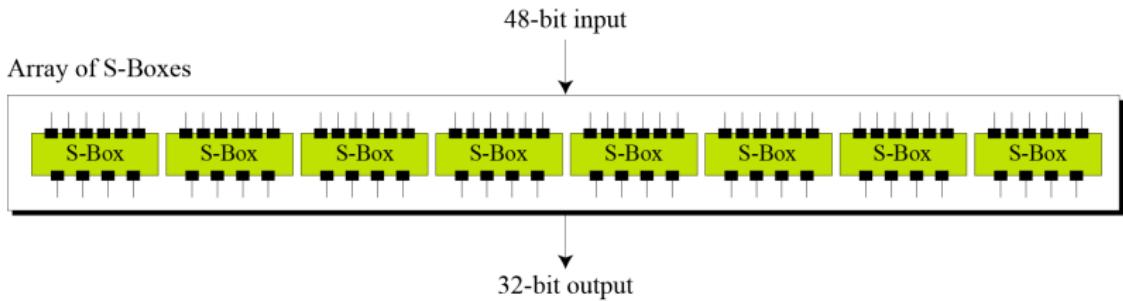
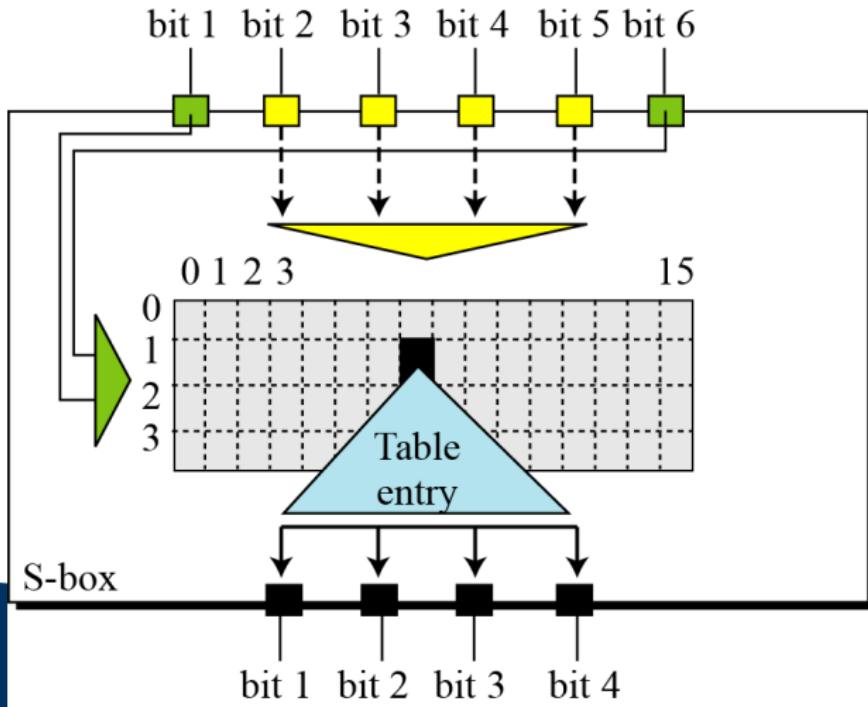


Figure 27: S-boxes

S-box rule for DES



permutation for S-box 1

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	04	13	01	02	15	11	08	03	10	06	12	05	09	00	07
1	00	15	07	04	14	02	13	10	03	06	12	11	09	05	03	08
2	04	01	14	08	13	06	02	11	15	12	09	07	03	10	05	00
3	15	12	08	02	04	09	01	07	05	11	03	14	10	00	06	13

Figure 29: permutation for S-box 1, rest can be checked from textbook.

Example 1

- When the S-box 1 (in book Table 6.3) is referred and the input to S-box 1 is 100011.
What is the output?
- Solution:**
If we write the first and the sixth bits together, we get 11 in binary, which is 3 in decimal. The remaining bits are 0001 in binary, which is 1 in decimal. We look for the value in row 3, column 1, in Table 6.3 (S-box 1). The result is 12 in decimal, which in binary is 1100. So the input 100011 yields the output 1100.

Example 2

- When the S-box 8 (in book Table 6.10) is referred and the input to S-box 8 is 000000. What is the output?
- Solution:**
If we write the first and the sixth bits together, we get 00 in binary, which is 0 in decimal. The remaining bits are 0000 in binary, which is 0 in decimal. We look for the value in row 0, column 0, in Table 6.10 (S-box 8). The result is 13 in decimal, which is 1101 in binary. So the input 000000 yields the output 1101.

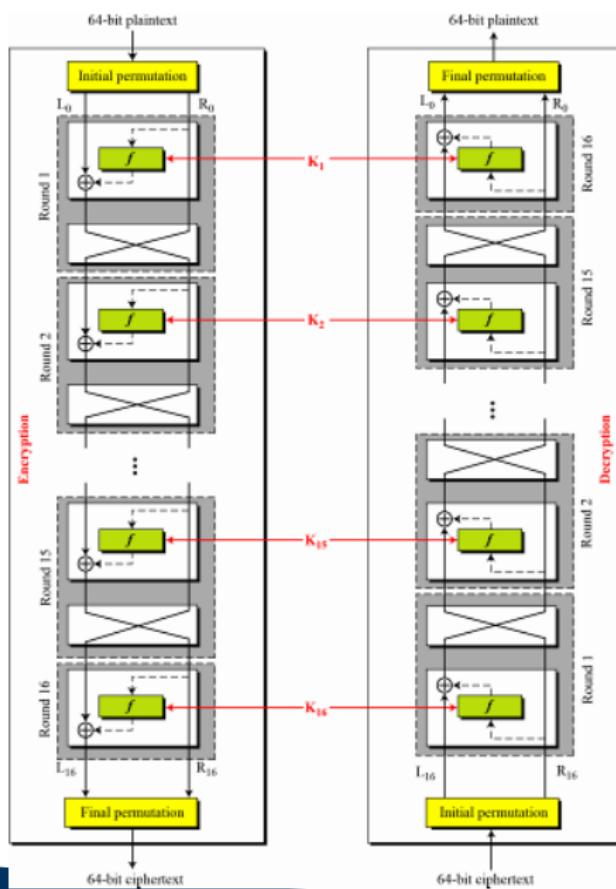
Straight Permutation table in DES function

16	07	20	21	29	12	28	17
01	15	23	26	05	18	31	10
02	08	24	14	32	27	03	09
19	13	30	06	22	11	04	25

Figure 30: Straight Permutation table

Cipher and Reverse Cipher

- Using mixers and swappers, we can create the cipher and reverse cipher, each having 16 rounds.
- **First approach:** To achieve this goal, one approach is to make the last round (round 16) different from the others; it has only a mixer and no swapper.
 - *In the first approach, there is no swapper in the last round.*



DES cipher and reverse cipher for the first approach.

Pseudocode for DES cipher

```
Cipher (plainBlock[64], RoundKeys[16, 48], cipherBlock[64])
{
    permute (64, 64, plainBlock, inBlock, InitialPermutationTable)
    split (64, 32, inBlock, leftBlock, rightBlock)
    for (round = 1 to 16)
    {
        mixer (leftBlock, rightBlock, RoundKeys[round])
        if (round!=16) swapper (leftBlock, rightBlock)
    }
    combine (32, 64, leftBlock, rightBlock, outBlock)
    permute (64, 64, outBlock, cipherBlock, FinalPermutationTable)
}
```

Pseudocode for DES cipher.. Contd...1

```
mixer (leftBlock[48], rightBlock[48], RoundKey[48])
{
    copy (32, rightBlock, T1)
    function (T1, RoundKey, T2)
        exclusiveOr (32, leftBlock, T2, T3)
        copy (32, T3, rightBlock)
}

swapper (leftBlock[32], righthBlock[32])
{
    copy (32, leftBlock, T)
    copy (32, rightBlock, leftBlock)
    copy (32, T, rightBlock)
}
```

Pseudocode for DES cipher.. Contd...2

```
substitute (inBlock[32], outBlock[48], SubstitutionTables[8, 4, 16])
{
    for (i = 1 to 8)
    {
        row ← 2 × inBlock[i × 6 + 1] + inBlock [i × 6 + 6]
        col ← 8 × inBlock[i × 6 + 2] + 4 × inBlock[i × 6 + 3] +
              2 × inBlock[i × 6 + 4] + inBlock[i × 6 + 5]

        value = SubstitutionTables [i][row][col]

        outBlock[[i × 4 + 1] ← value / 8;           value ← value mod 8
        outBlock[[i × 4 + 2] ← value / 4;           value ← value mod 4
        outBlock[[i × 4 + 3] ← value / 2;           value ← value mod 2
        outBlock[[i × 4 + 4] ← value
    }
}
```

Alternative approach

- We can make all 16 rounds the same by including one swapper to the 16th round and add an extra swapper after that (two swappers cancel the effect of each other).

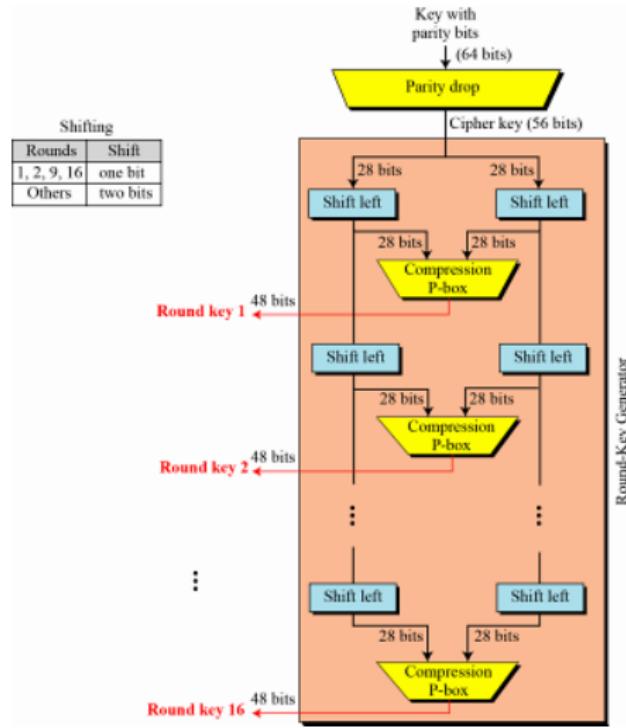


Figure 32: Key Generation: The round-key generator creates sixteen 48-bit keys out of a 56-bit cipher key.

Parity-bit Drop Table

- The preprocess before key expansion is a compression transposition step that we call **parity-bit drop**.
- It drops the parity bits (bits 8, 16, 24, 32,..., 64) from 64 -bit key and permutes the rest of the bits according to the following table.

57	49	41	33	25	17	09	01
58	50	42	34	26	18	10	02
59	51	43	35	27	19	11	03
60	52	44	36	63	55	47	39
31	23	15	07	62	54	46	38
30	22	14	06	61	53	45	37
29	21	13	05	28	20	12	04

Shift-Left

- After the straight permutation, the key is divided into two 28-bit parts. Each part is shifted left (circular shift) one or two bits.
- In rounds 1, 2, 9, and 16, shifting is one bit; in the other rounds, it is two bits.
- The two parts are then combined to form 56-bit part.

Round	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Bit shifts	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Figure 33: Number of shifts for each round.

Key-compression in Key Generation in DES

- The compression D-box or P-box changes the 58 bits to 48 bits, which are used as a key for a round.

14	17	11	24	01	05	03	28
15	06	21	10	23	19	12	04
26	08	16	07	27	20	13	02
41	52	31	37	47	55	30	40
51	45	33	48	44	49	39	56
34	53	46	42	50	36	29	32

Figure 34: **Key-compression table**

Algorithm for round-key generation..part1

```
Key_Generator (keyWithParities[64], RoundKeys[16, 48], ShiftTable[16])
{
    permute (64, 56, keyWithParities, cipherKey, ParityDropTable)
    split (56, 28, cipherKey, leftKey, rightKey)
    for (round = 1 to 16)
    {
        shiftLeft (leftKey, ShiftTable[round])
        shiftLeft (rightKey, ShiftTable[round])
        combine (28, 56, leftKey, rightKey, preRoundKey)
        permute (56, 48, preRoundKey, RoundKeys[round], KeyCompressionTable)
    }
}
```

Algorithm for round-key generation..part2

```
shiftLeft (block[28], numOfShifts)
{
    for (i = 1 to numOfShifts)
    {
        T ← block[1]
        for (j = 2 to 28)
        {
            block [j-1] ← block [j]
        }
        block[28] ← T
    }
}
```

Example1

- We choose a random plaintext block and a random key, and determine what the ciphertext block would be (all in hexadecimal):

Plaintext: 123456ABCD132536

Key: AABB09182736CCDD

CipherText: C0B7A8D05F3A829C

Plaintext: 123456ABCD132536

After initial permutation: 14A7D67818CA18AD

After splitting: $L_0 = 14A7D678$ $R_0 = 18CA18AD$

Round	Left	Right	Round Key
Round 1	18CA18AD	5A78E394	194CD072DE8C
Round 2	5A78E394	4A1210F6	4568581ABCCE
Round 3	4A1210F6	B8089591	06EDA4ACF5B5
Round 4	B8089591	236779C2	DA2D032B6EE3

Example1..Trace of Data.. continued..

<i>Round 5</i>	236779C2	A15A4B87	69A629FEC913
<i>Round 6</i>	A15A4B87	2E8F9C65	C1948E87475E
<i>Round 7</i>	2E8F9C65	A9FC20A3	708AD2DDB3C0
<i>Round 8</i>	A9FC20A3	308BEE97	34F822F0C66D
<i>Round 9</i>	308BEE97	10AF9D37	84BB4473DCCC
<i>Round 10</i>	10AF9D37	6CA6CB20	02765708B5BF
<i>Round 11</i>	6CA6CB20	FF3C485F	6D5560AF7CA5
<i>Round 12</i>	FF3C485F	22A5963B	C2C1E96A4BF3
<i>Round 13</i>	22A5963B	387CCDAA	99C31397C91F
<i>Round 14</i>	387CCDAA	BD2DD2AB	251B8BC717D0
<i>Round 15</i>	BD2DD2AB	CF26B472	3330C5D9A36D
<i>Round 16</i>	19BA9212	CF26B472	181C5D75C66D
<i>After combination:</i> 19BA9212CF26B472			
<i>Ciphertext:</i> C0B7A8D05F3A829C		<i>(after final permutation)</i>	

Decryption/Deciphering at Receiver's end

- Let us see how Bob, at the destination, can decipher the ciphertext received from Alice using the same key. The following Table shows some interesting points.

Ciphertext: C0B7A8D05F3A829C			
After initial permutation: 19BA9212CF26B472			
After splitting: $L_0=19BA9212$ $R_0=CF26B472$			
Round	Left	Right	Round Key
Round 1	CF26B472	BD2DD2AB	181C5D75C66D
Round 2	BD2DD2AB	387CCDAA	3330C5D9A36D
...
Round 15	5A78E394	18CA18AD	4568581ABCCE
Round 16	14A7D678	18CA18AD	194CD072DE8C
After combination: 14A7D67818CA18AD			
Plaintext: 123456ABCD132536		(after final permutation)	

Bibliography: Books and Resources

- Cryptography and Network Security: Principles and Practice by William Stallings
- Cryptography and Network Security by Behrouz A Forouzan and Debdeep Mukhopadhyay
- Principles of Information Security by Michael E. Whitman and Herbert J. Mattord.
- Cisco platform, and Internet.
- Published research papers, study materials from researchers of security domain.

CO-INS:Information and Network Security

UNIT-II (Part-II) Modern Symmetric-Key Ciphers

Course Instructors:

Soma Saha

Veerendra Srivastava

Soma Saha (PhD)

Department of Computer Engineering
SGSITS Indore, India

March 31, 2021

UNIT-II (Part-II): Learning Objectives

Upon completion of this unit, you should be able to

- LO1 Explain the concept of modern block ciphers and discuss their characteristics
- LO2 Discuss the components of a modern block cipher
- LO3 Relate the concept of product ciphers and distinguish between the two classes of product ciphers
- LO4 Explain modern stream ciphers and discuss two broad categories—synchronous and non-synchronous

Cryptography: Modern Symmetric-Key Ciphers

- The traditional/classical symmetric-key ciphers (that we have studied so far) are **character-oriented ciphers**.
- With the advent of the computer, we need **bit-oriented ciphers**.
- Why??

Cryptography: Modern Symmetric-Key Ciphers

- The traditional/classical symmetric-key ciphers (that we have studied so far) are **character-oriented ciphers**.
- With the advent of the computer, we need **bit-oriented ciphers**.
- Why??
 - The information to be encrypted is not just text; it can also consists of numbers, graphics, audio, and video data.

Cryptography: Modern Symmetric-Key Ciphers

- The traditional/classical symmetric-key ciphers (that we have studied so far) are **character-oriented ciphers**.
- With the advent of the computer, we need **bit-oriented ciphers**.
- Why??
 - The information to be encrypted is not just text; it can also consists of numbers, graphics, audio, and video data.
 - It is convenient to convert these types of data into stream of bits, to encrypt the stream, and then to send the encrypted stream.

Cryptography: Modern Symmetric-Key Ciphers

- The traditional/classical symmetric-key ciphers (that we have studied so far) are **character-oriented ciphers**.
- With the advent of the computer, we need **bit-oriented ciphers**.
- Why??
 - The information to be encrypted is not just text; it can also consists of numbers, graphics, audio, and video data.
 - It is convenient to convert these types of data into stream of bits, to encrypt the stream, and then to send the encrypted stream.
 - Additionally, when text is treated at the bit level, each character is replaced by 8 (or 16) bits, which means that the number of symbols becomes 8(or 16) times larger. **Mixing a larger number of symbols increases security.**

Modern Block Cipher

- A symmetric-key **modern block cipher** encrypts an n -bit block of plaintext or decrypts an n -bit block of ciphertext. The encryption or decryption algorithm uses a k -bit key.

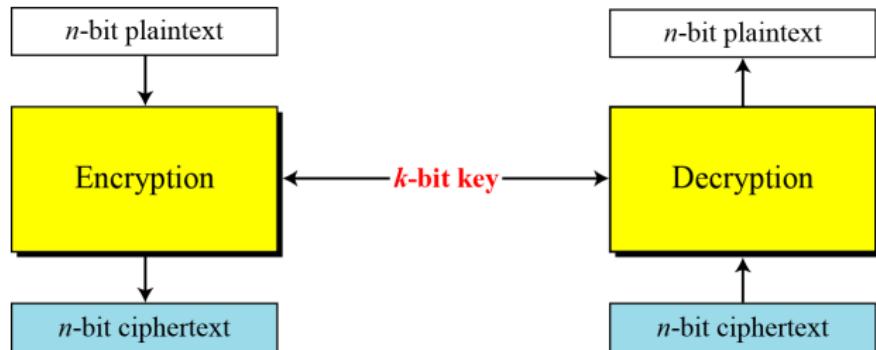


Figure 1: A modern block cipher.

Modern Block Cipher.. contd...1

- If the message has fewer than n bits, padding must be added to make it an n -bit block; if the message has more than n bits, it should be divided into n -bit blocks and appropriate padding must be added to the last block if necessary.
- **Common values for n ?**

Modern Block Cipher.. contd...1

- If the message has fewer than n bits, padding must be added to make it an n -bit block; if the message has more than n bits, it should be divided into n -bit blocks and appropriate padding must be added to the last block if necessary.
- **Common values for n ?**
 - The common values for n are 64, 128, 256, or 512 bits.

Modern Block Cipher.. contd...2

- How many padding bits must be added to a message of 100 characters if 8-bit ASCII is used for encoding and the block cipher accepts blocks of 64 bits?

Modern Block Cipher.. contd...2

- How many padding bits must be added to a message of 100 characters if 8-bit ASCII is used for encoding and the block cipher accepts blocks of 64 bits?
- Encoding 100 characters using 8-bit ASCII results in an 800-bit message. The plaintext must be divisible by 64. If $|M|$ and $|Pad|$ are the length of the message and the length of the padding,

$$|M| + |Pad| = 0 \bmod 64 \rightarrow |Pad| = -800 \bmod 64 \rightarrow 32 \bmod 64$$

Substitution or Transposition

- A modern block cipher can be designed to act as a **substitution cipher** or a **transposition cipher**.
- **Example:** If the cipher is designed as a substitution cipher, a 1-bit or a 0-bit in the plaintext can be replaced by either 0 or 1. This signifies that the ciphertext and plaintext can have a different number of 1's.

Substitution or Transposition

- A modern block cipher can be designed to act as a **substitution cipher** or a **transposition cipher**.
- **Example:** If the cipher is designed as a substitution cipher, a 1-bit or a 0-bit in the plaintext can be replaced by either 0 or 1. This signifies that the ciphertext and plaintext can have a different number of 1's.
 - a 64 bit plaintext block of 12 0's and 52 1's can be encrypted to a ciphertext of 34 0's and 30 1's.

Substitution or Transposition

- A modern block cipher can be designed to act as a **substitution cipher** or a **transposition cipher**.
- **Example:** If the cipher is designed as a substitution cipher, a 1-bit or a 0-bit in the plaintext can be replaced by either 0 or 1. This signifies that the ciphertext and plaintext can have a different number of 1's.
 - a 64 bit plaintext block of 12 0's and 52 1's can be encrypted to a ciphertext of 34 0's and 30 1's.
- If the cipher is designed as a transposition cipher, the bits are only reordered (transposed); there is same number of 1's in the plaintext and in the ciphertext.
- **Conclusion:** Modern block ciphers are designed as substitution ciphers to be resistant to exhaustive-search attack.

Modern block cipher: Substitution or Transposition:: Example

- Suppose that we have a block cipher where $n = 64$. If there are 10 1's in the ciphertext, how many trial-and-error tests does Eve need to do to recover the plaintext from the intercepted ciphertext in each of the following cases?
 - a. The cipher is designed as a substitution cipher.
 - b. The cipher is designed as a transposition cipher.
- **Solution:**

Modern block cipher: Substitution or Transposition:: Example

- Suppose that we have a block cipher where $n = 64$. If there are 10 1's in the ciphertext, how many trial-and-error tests does Eve need to do to recover the plaintext from the intercepted ciphertext in each of the following cases?
 - a. The cipher is designed as a substitution cipher.
 - b. The cipher is designed as a transposition cipher.
- **Solution:**
 - a. In the first case, Eve has no idea how many 1's are in the plaintext. Eve needs to try all possible 2^{64} 64-bit blocks to find one that makes sense.
 - If Eve could try 1 billion blocks per second, it would still take hundreds of years, on average, before she could be successful.

Modern block cipher: Substitution or Transposition:: Example

- Suppose that we have a block cipher where $n = 64$. If there are 10 1's in the ciphertext, how many trial-and-error tests does Eve need to do to recover the plaintext from the intercepted ciphertext in each of the following cases?
 - The cipher is designed as a substitution cipher.
 - The cipher is designed as a transposition cipher.
- Solution:**
 - In the first case, Eve has no idea how many 1's are in the plaintext. Eve needs to try all possible 2^{64} 64-bit blocks to find one that makes sense.
 - If Eve could try 1 billion blocks per second, it would still take hundreds of years, on average, before she could be successful.
 - In the second case, Eve knows that there are exactly 10 1's in the plaintext. Eve can launch an exhaustive-search attack using only those 64-bit blocks that have exactly 10 1's.

$$\binom{64}{10} = \frac{64!}{(10!)(54!)} = 151,473,214,816$$

(Less than 3 min...)

Block Ciphers as Permutation Groups

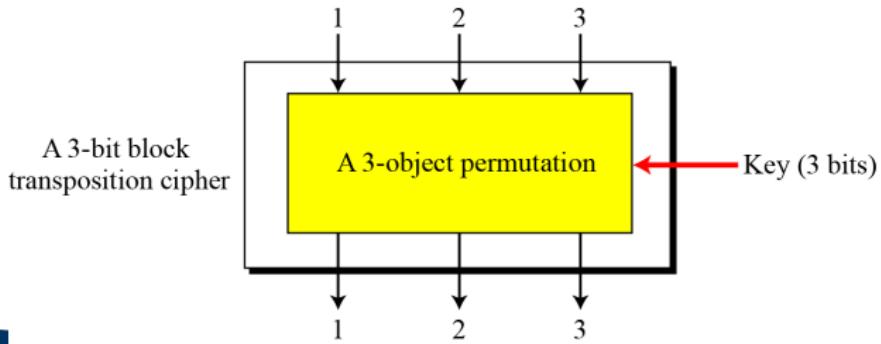
- **Full-size key ciphers:** the key is long enough to choose every possible mapping from input to output. In practice, the key is smaller (partial-key), only some mappings from the input to output are possible.
 - Full-size key ciphers are not used in practice, only partial-key ciphers are used.

Block Ciphers as Permutation Groups

- **Full-size key ciphers:** the key is long enough to choose every possible mapping from input to output. In practice, the key is smaller (partial-key), only some mappings from the input to output are possible.
 - Full-size key ciphers are not used in practice, only partial-key ciphers are used.
- **Full-Size Key Transposition Block Ciphers:** In a full-size key transposition cipher We need to have $n!$ possible keys, so the key should have $\lceil \log_2 n! \rceil$ bits.
 - Only transposes bits without changing their values.
 - So, it can be modeled as an n -object permutation with a set of $n!$ permutation tables in which the key defines which table is used by Alice and Bob.

Full-Size Key Transposition Block Ciphers: Example

- Show the model and the set of permutation tables for a 3-bit block transposition cipher where the block size is 3 bits.
- The set of permutation tables has $3! = 6$ elements, as shown below:



$\{[1\ 2\ 3], [1\ 3\ 2], [2\ 1\ 3], [2\ 3\ 1], [3\ 1\ 2], [3\ 2\ 1]\}$

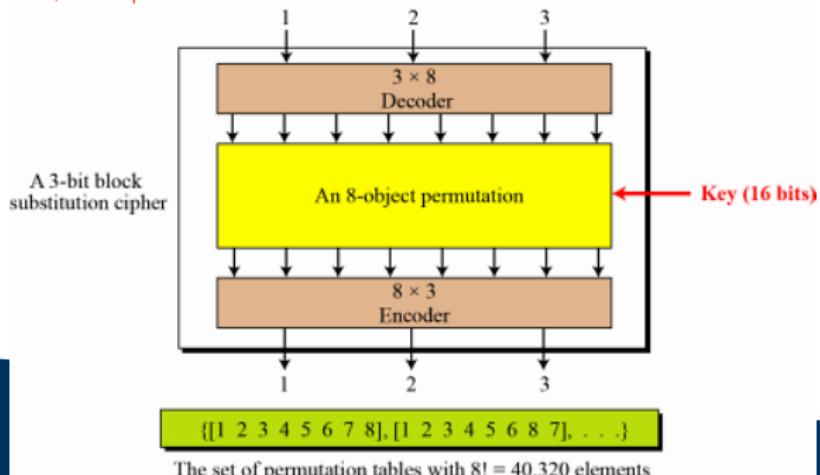
The set of permutation tables with $3! = 6$ elements

Full-Size Key Substitution Block Ciphers

- **Full-Size Key Substitution Block Ciphers:** A full-size key substitution cipher does not transpose bits; it substitutes bits. We can model the substitution cipher as a permutation if we can decode the input and encode the output.
 -
 - We can model the substitution cipher as a permutation if we can decode the input and encode the output.
 - **Decoding** means transforming an n -bit integer into a 2^n -bit string with only a single 1 and $2^n - 1$ 0's.
 - The position of the single 1 is the value of the integer, in which the positions range from 0 to $2^n - 1$.
 - **Encoding** is the reverse process. As the new input and output have always a single 1, the cipher can be modeled as a permutation of 2^n objects.

Full-Size Key Substitution Block Ciphers: Example

- Example: Show the model and the set of permutation tables for a 3-bit block substitution cipher.
- The figure shows the model and the set of permutation tables. The key is much longer, $\lceil \log_2 40,320 \rceil = 16$ bits.



NOTE:

- A full-size key n-bit transposition cipher or a substitution block cipher can be modeled as a permutation, but their key sizes are different:
 - Transposition: the key is $\lceil \log_2 n! \rceil$ bits long.
 - Substitution: the key is $\lceil \log_2(2^n)! \rceil$ bits long.

Partial-Size Key Cipher

- Two or more cascaded permutations can be always replaced with a single permutation. Hence it is useless to have more than one stage of full-size key ciphers, because the effect is the same as having a single stage.
- Modern block ciphers normally are keyed substitution ciphers in which the key allows only partial mappings from the possible inputs to the possible outputs.

Partial-Size Key Cipher

- Two or more cascaded permutations can be always replaced with a single permutation. Hence it is useless to have more than one stage of full-size key ciphers, because the effect is the same as having a single stage.
- Modern block ciphers normally are keyed substitution ciphers in which the key allows only partial mappings from the possible inputs to the possible outputs.
- **For example**, a common substitution cipher is DES (Data Encryption Standard) which uses a 64-bit block cipher. If the designer of DES had used a full-size key, the key would have been $\log_2(2^{64})! = 2^{70}$ bits. The key size for DES is only 56 bits which is only a very small fraction of the full-size key. This means that DES uses only 2^{56} mappings out of approximately $2^{2^{70}}$ possible mappings.

Components of a Modern Block Cipher

- In cryptography, **confusion** and **diffusion** are two properties of the operation of a secure cipher identified by **Claude Shannon** in his 1945 classified report: "A Mathematical Theory of Cryptography".

Diffusion

- **Diffusion:** Diffusion means that if we change a single bit of the plaintext, then (statistically) half of the bits in the ciphertext should change, and similarly, if we change one bit of the ciphertext, then approximately one half of the plaintext bits should change. Since a bit can have only two states, when they are all re-evaluated and changed from one seemingly random position to another, half of the bits will have changed state.
 - **The idea of diffusion is to hide the relationship between the ciphertext and the plain text.**
 - This will make it hard for an attacker who tries to find out the plain text and it increases the redundancy of plain text by spreading it across the rows and columns; it is achieved through transposition of algorithm and it is used by block ciphers only.

Confusion

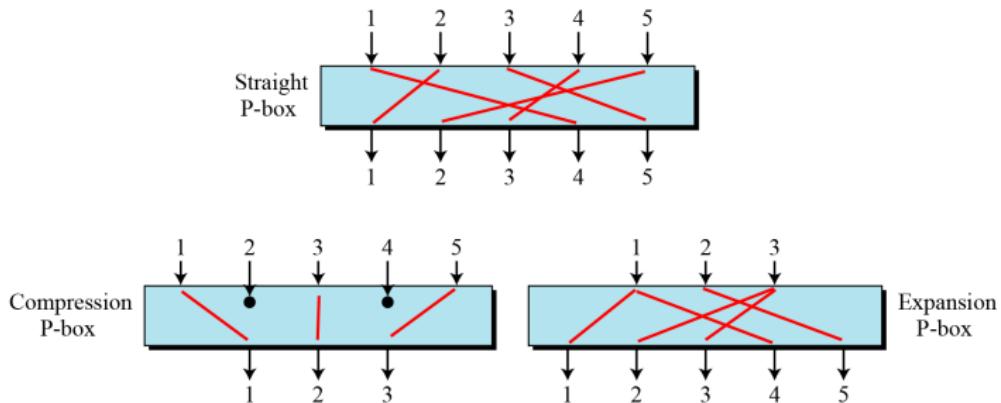
- Confusion means that each binary digit (bit) of the ciphertext should depend on several parts of the key, obscuring the connections between the two.
- **The property of confusion hides the relationship between the ciphertext and the key.**
- This property makes it difficult to find the key from the ciphertext and if a single bit in a key is changed, the calculation of the values of most or all of the bits in the ciphertext will be affected.
- Confusion increases the ambiguity of ciphertext and it is used by both block and stream ciphers.

Components of a Modern Block Cipher

- To provide required properties of a modern block cipher, such as **diffusion** and **confusion**, a modern block cipher is made of a combination of
 - transposition units for diffusion (called **D-boxes** or **P-boxes** for permutation),
 - substitution units (called **S-boxes**),
 - and some other units.

P-boxes or D-boxes

- * A P-box (permutation box) or D-box (diffusion box) parallels the traditional transposition cipher for characters. It transposes bits.



Straight P-boxes (or D-boxes)

- **straight P-boxes (or D-boxes)** : all 6 possible mappings of a 3×3 D-box.

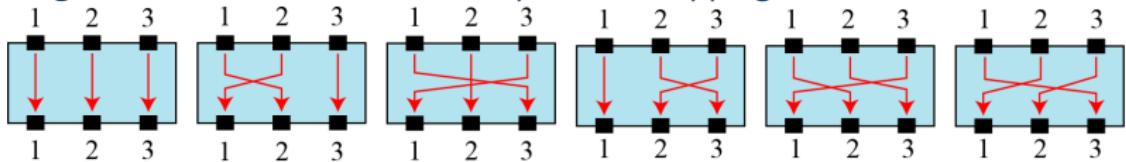


Figure 2: Although a P-box can use a key to define one of the $n!$ mappings, P boxes are normally keyless, which means the mapping is predetermined. .

Straight P-boxes (or D-boxes)

- **straight P-boxes (or D-boxes)** : all 6 possible mappings of a 3×3 D-box.

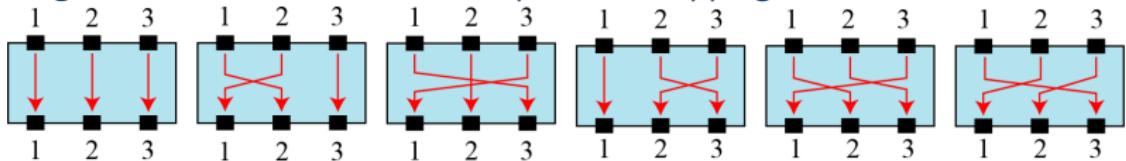


Figure 2: Although a P-box can use a key to define one of the $n!$ mappings, P boxes are normally keyless, which means the mapping is predetermined. .

58	50	42	34	26	18	10	02	60	52	44	36	28	20	12	04
62	54	46	38	30	22	14	06	64	56	48	40	32	24	16	08
57	49	41	33	25	17	09	01	59	51	43	35	27	19	11	03
61	53	45	37	29	21	13	05	63	55	47	39	31	23	15	07

Example

- Design an 8×8 permutation table for a straight P-box that moves the two middle bits (bits 4 and 5) in the input word to the two ends (bits 1 and 8) in the output words. Relative positions of other bits should not be changed.

Example

- Design an 8×8 permutation table for a straight P-box that moves the two middle bits (bits 4 and 5) in the input word to the two ends (bits 1 and 8) in the output words. Relative positions of other bits should not be changed.
- **Solution:** We need a straight P-box with the table [4 1 2 3 6 7 8 5]. The relative positions of input bits 1, 2, 3, 6, 7, and 8 have not been changed, but the first output takes the fourth input and the eighth output takes the fifth input.

Compression P-Boxes (or D-boxes)

- A compression P-box is a P-box with n inputs and m outputs where m < n.

01	02	03	21	22	26	27	28	29	13	14	17
18	19	20	04	05	06	10	11	12	30	31	32

Figure 4: Example of a 32×24 permutation table.

Expansion P-Boxes (or D-boxes)

- An expansion P-box is a P-box with n inputs and m outputs where $m > n$

01	09	10	11	12	01	02	03	03	04	05	06	07	08	09	12
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

Figure 5: Example of a 12×16 permutation table.

P-Boxes/D-Boxes: Invertibility

- A straight P-box is invertible, but compression and expansion P-boxes are not.

P-Boxes/D-Boxes: Invertibility

- A straight P-box is invertible, but compression and expansion P-boxes are not.
- How can you invert a permutation table to be represented as a one-dimensional table?

1. Original table

6	3	4	5	2	1
---	---	---	---	---	---

2. Add indices

6	3	4	5	2	1
1	2	3	4	5	6

3. Swap contents
and indices

1	2	3	4	5	6
6	3	4	5	2	1

4. Sort based
on indices

6	5	2	3	4	1
1	2	3	4	5	6

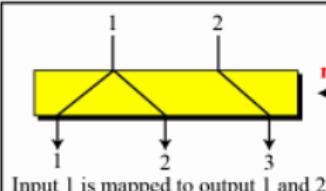
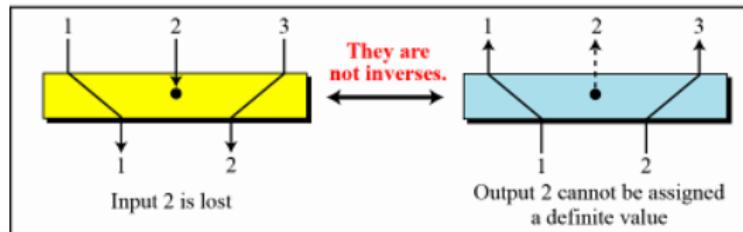
6	5	2	3	4	1
---	---	---	---	---	---

5. Inverted table

Figure 6: Inverting a permutation table.

Compression and expansion P-boxes are non-invertible

Compression P-box



Expansion P-box

S-boxes

- An S-box (substitution box) can be thought of as a miniature substitution cipher.
- **An S-box is an $m \times n$ substitution unit, where m and n are not necessarily the same.**
- For example: The following table defines the input/output relationship for an S-box of size 3×2 . The leftmost bit of the input defines the row; the two rightmost bits of the input define the column. The two output bits are values on the cross section of the selected row and column.

Leftmost bit

Rightmost bits

	00	01	10	11
0	00	10	01	11
1	10	00	11	01

Output bits

S-Boxes: Invertibility

- An S-box may or may not be invertible. In an invertible S-box, the number of input bits should be the same as the number of output bits.

S-Boxes: Invertibility.. contd

- The following shows an example of an invertible S-box. For example, if the input to the left box is 001, the output is 101. The input 101 in the right table creates the output 001, which shows that the two tables are inverses of each other.

3 bits

↓

		00	01	10	11
0	011	101	111	100	
	000	010	001	110	

Table used for
encryption

3 bits

3 bits

↓

		00	01	10	11
0	100	110	101	000	
	011	001	111	010	

Table used for
decryption

3 bits

XOR (Exclusive-Or)

- An important component in most block ciphers is the exclusive-or operation.

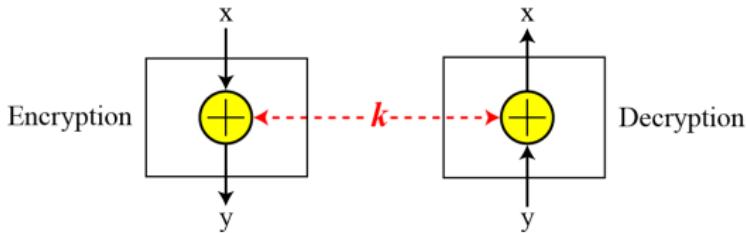


Figure 10: Invertibility of the exclusive-or operation.

XOR.. Contd..

- An important component in most block ciphers is the exclusive-or operation.
- **The five properties of the exclusive-or operation makes this operation a very interesting component for use in a block cipher: closure, associativity, commutativity, existence of identity, and existence of inverse.**

$$X \text{ EXOR } 0 = X$$

$$X \text{ EXOR } 1 = \bar{X}$$

$$X \text{ EXOR } \bar{X} = 1$$

$$X \text{ EXOR } X = 0$$

Exclusive OR ...Contd

- The inverse of a component in a cipher makes sense if the component represents a unary operation (one input and one output).

Exclusive OR ...Contd

- The inverse of a component in a cipher makes sense if the component represents a unary operation (one input and one output).
- For example, a keyless P-box or a keyless S-box can be made invertible because they have one input and one output.
- An exclusive-or operation is a binary operation. The inverse of an exclusive-or operation can make sense only if one of the inputs is fixed (is the same in encryption and decryption).

Exclusive OR ...Contd

- The inverse of a component in a cipher makes sense if the component represents a unary operation (one input and one output).
- For example, a keyless P-box or a keyless S-box can be made invertible because they have one input and one output.
- An exclusive-or operation is a binary operation. The inverse of an exclusive-or operation can make sense only if one of the inputs is fixed (is the same in encryption and decryption).
- For example, if one of the inputs is the key, which normally is the same in encryption and decryption, then an exclusive-or operation is self-invertible, as shown in Last Figure. 10.

Circular Shift

- Another component used in some modern block ciphers is the **circular shift** operation.

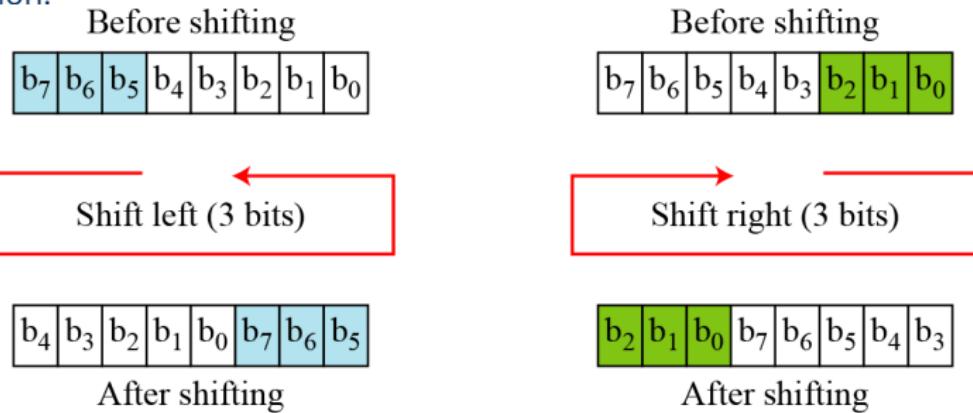


Figure 11: Circular shifting an 8-bit word to the left or right.

Swap

- The **swap** operation is a special case of the circular shift operation where $k = n/2$.

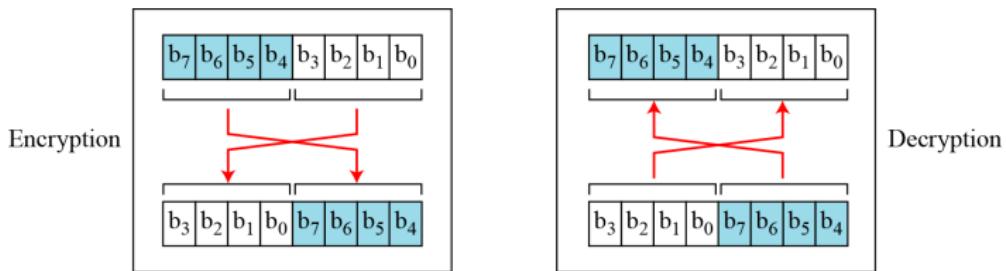


Figure 12: Swap operation on an 8-bit word.

Split and Combine

- Two other operations found in some block ciphers are split and combine.

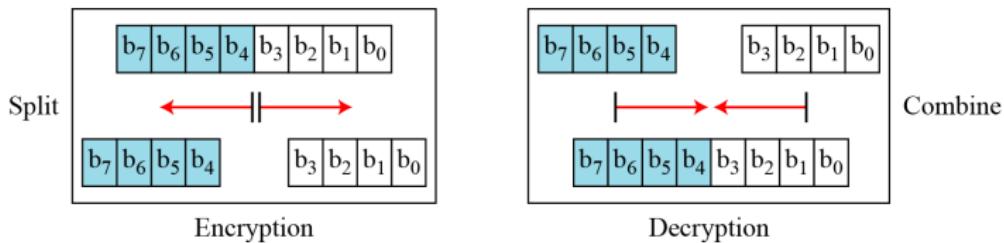


Figure 13: Split and combine operations on an 8-bit word.

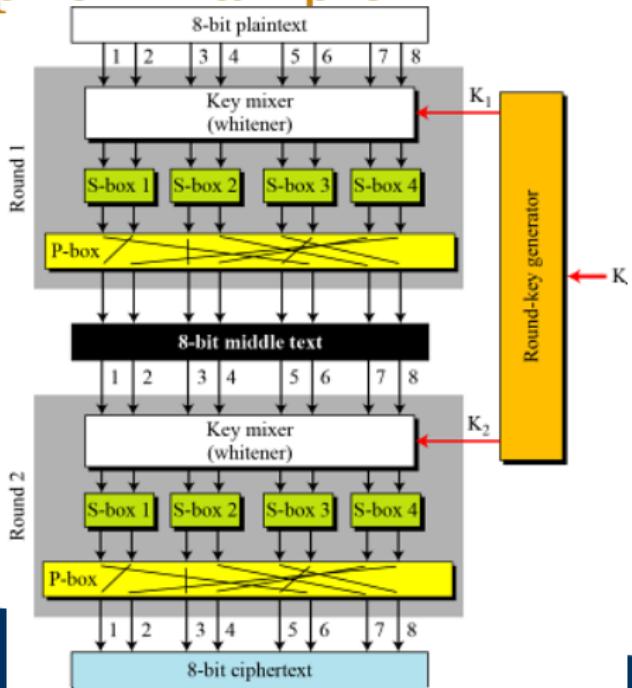
Product Cipher

- Shannon introduced the concept of **product cipher**.
- A **product cipher** is a complex cipher combining substitution, permutation, and other components discussed in previous sections.

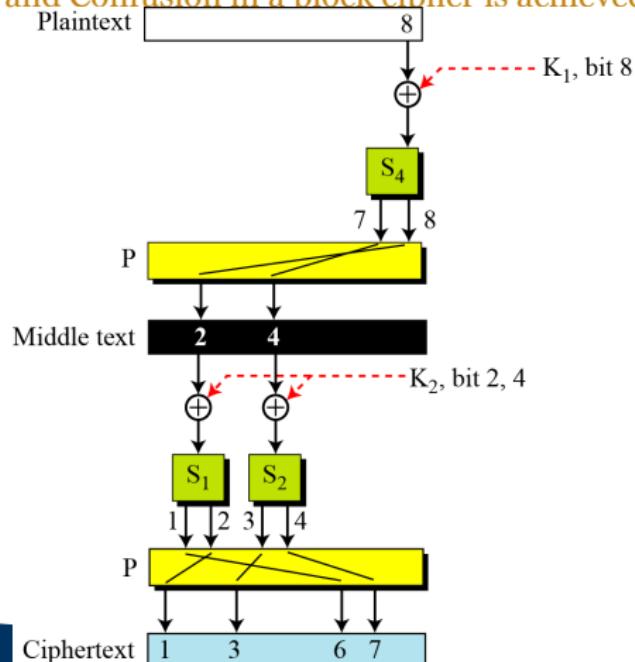
Diffusion, confusion, and Rounds

- **Diffusion:** The idea to hide the relationship between the ciphertext and the plaintext.
- **Confusion:** The idea to hide the relationship between the ciphertext and the key.
- **Rounds:** Diffusion and confusion can be achieved using iterated product ciphers where each iteration is a combination of S-boxes, P-boxes/D-boxes, and other components.

Product cipher Example



How does Diffusion and Confusion in a block cipher is achieved?



Classification of Product Ciphers

- Modern block ciphers are all product ciphers, but they are divided into two classes.
 - **Feistel ciphers**
 - **Non-Feistel ciphers**

Classification of Product Ciphers

- Modern block ciphers are all product ciphers, but they are divided into two classes.
- **Feistel ciphers**
 - Has been used for decades.
 - Can have three types of components :
self-invertible, invertible, and non-invertible.
 - Example: **DES**
- **Non-Feistel ciphers:**
 - Uses only *invertible components*.
 - A component in the encryption cipher has the corresponding component in the decryption cipher.
 - Example: **AES**

Feistel Ciphers: First Thought

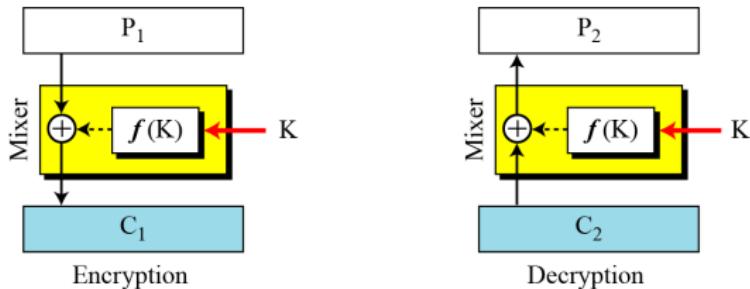


Figure 16: The first thought in Feistel cipher design: **f(K) is a non-invertible function.**

- **Encryption:** $C_1 = P_1 \oplus f(K)$
- **Decryption:**

$$P_2 = C_2 \oplus f(K) = C_1 \oplus f(K) = P_1 \oplus f(K) \oplus f(K) = P_1 \oplus (00\dots 0) = P_1$$

Feistel Ciphers: First Thought

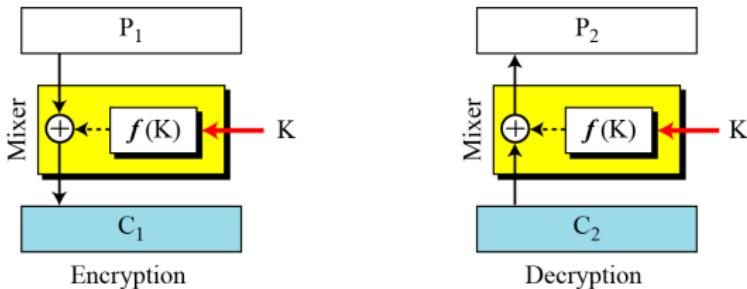


Figure 16: The first thought in Feistel cipher design: **f(K) is a non-invertible function.**

- **Encryption:** $C_1 = P_1 \oplus f(K)$
- **Decryption:**
$$P_2 = C_2 \oplus f(K) = C_1 \oplus f(K) = P_1 \oplus f(K) \oplus f(K) = P_1 \oplus (00\dots 0) = P_1$$
- **The mixer(combination of function and ex-or operation) in the Feistel design is self-invertible.**

Example

- This is a trivial example. The plaintext and ciphertext are each 4 bits long and the key is 3 bits long. Assume that the function takes the first and third bits of the key, interprets these two bits as a decimal number, squares the number, and interprets the result as a 4-bit binary pattern. Show the results of encryption and decryption if the original plaintext is 0111 and the key is 101.

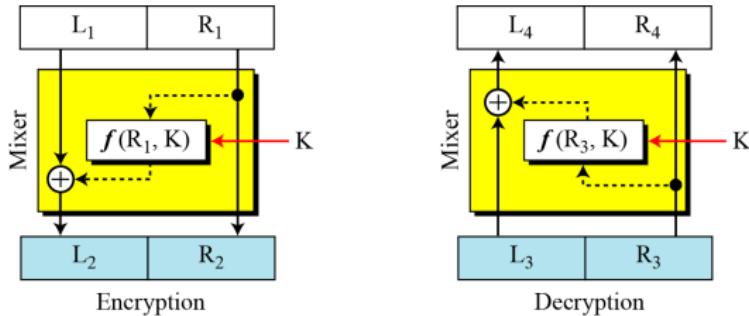
Example

- This is a trivial example. The plaintext and ciphertext are each 4 bits long and the key is 3 bits long. Assume that the function takes the first and third bits of the key, interprets these two bits as a decimal number, squares the number, and interprets the result as a 4-bit binary pattern. Show the results of encryption and decryption if the original plaintext is 0111 and the key is 101.
- Solution:**
 - The function extracts the first and second bits to get 11 in binary or 3 in decimal. The result of squaring is 9, which is 1001 in binary.

Encryption: $C = P \oplus f(K) = 0111 \oplus 1001 = 1110$

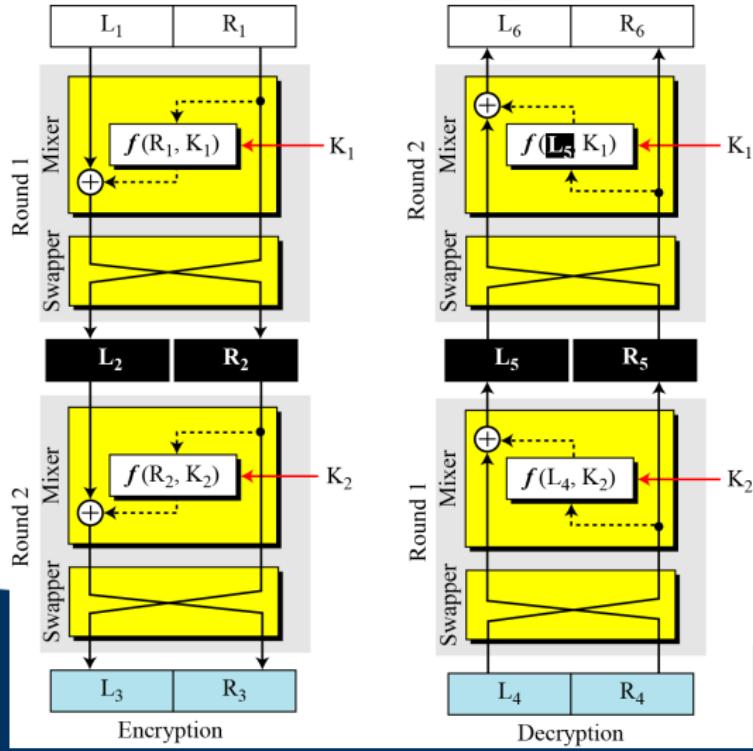
Decryption: $P = C \oplus f(K) = 1110 \oplus 1001 = 0111$

Improvement of the previous Feistel design



- $R_4 = R_3 = R_2 = R_1$
- $L_4 = L_3 \oplus f(R_3, K) = L_2 \oplus f(R_2, K) = L_1 \oplus f(R_1, K) \oplus f(R_1, K) = L_1$
- **Flaw in this design: Right half of the plaintext never changes.**

Final design of a Feistel cipher with two rounds



Feistel Cipher design.. contd..

- Let us see if $L_6 = L_1$ and $R_6 = R_1$, assuming that $L_4 = L_3$ and $R_4 = R_3$ (no change in ciphertext during transmission).
- We first prove the equality for the middle text.

$$\begin{aligned}L_5 &= R_4 \oplus f(L_4, K_2) = R_3 \oplus f(R_2, K_2) = L_2 \oplus f(R_2, K_2) \oplus f(R_2, K_2) = L_2 \\R_5 &= L_4 = L_3 = R_2\end{aligned}$$

- Then it is easy to prove that the equality holds for two plaintext blocks.
$$\begin{aligned}L_6 &= R_5 \oplus f(L_5, K_1) = R_2 \oplus f(L_2, K_1) = L_1 \oplus f(R_1, K_1) \oplus f(R_1, K_1) = L_1 \\R_6 &= L_5 = L_2 = R_1\end{aligned}$$

Feistel Ciphers

- Blowfish.
- Camellia.
- CAST-128.
- DES.
- FEAL.
- GOST 28147-89.
- ICE.

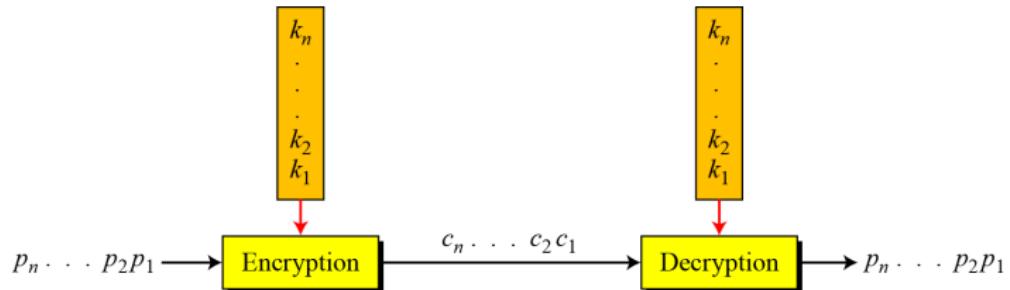
Modern Block Cipher: Secure??

- Attacks on traditional ciphers can also be used on modern block ciphers, but today's block ciphers resist most of the attacks discussed in classes/unit2_Cryptography_part1 slides.
- **Linear cryptanalysis and Differential cryptanalysis are the two most widely used attacks on block ciphers.**
- Eli Biham and Adi Shamir introduced the idea of **differential cryptanalysis**. This is a chosen-plaintext attack.
 - Differential cryptanalysis is based on a nonuniform differential distribution table of the S-boxes in a block cipher.
- **Linear cryptanalysis** was presented by Mitsuru Matsui in 1993. The analysis uses known plaintext attacks.

Modern Stream Ciphers

- In a modern stream cipher, encryption and decryption are done r bits at a time. We have a plaintext bit stream $P = p_n \dots p_2 p_1$, a ciphertext bit stream $C = c_n \dots c_2 c_1$, and a key bit stream $K = k_n \dots k_2 k_1$, in which p_i , c_i , and k_i are r -bit words.
- Encryption is $c_i = E(k_i, p_i)$, and
- Decryption is $p_i = D(k_i, c_i)$.
- **Classification:**
 1. Synchronous Stream Ciphers
 2. Nonsynchronous Stream Ciphers

Stream Cipher



- In a modern stream cipher, each r-bit word in the plaintext stream is enciphered using an r-bit word in the key stream to create the corresponding r-bit word in the ciphertext stream.

Synchronous Stream Ciphers

- In a synchronous stream cipher the key is independent of the plaintext or ciphertext.

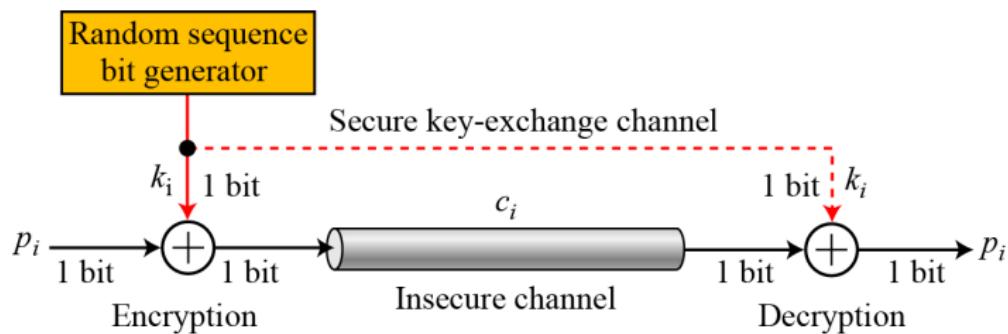


Figure 17: **One-time pad**: one-time pad invented and patented by Gilbert Vernam..

Example

- What is the pattern in the ciphertext of a one-time pad cipher in each of the following cases?
 - a. The plaintext is made of n 0's.
 - b. The plaintext is made of n 1's.
 - c. The plaintext is made of alternating 0's and 1's.
 - d. The plaintext is a random string of bits.

Example

- What is the pattern in the ciphertext of a one-time pad cipher in each of the following cases?
 - a. The plaintext is made of n 0's.
 - b. The plaintext is made of n 1's.
 - c. The plaintext is made of alternating 0's and 1's.
 - d. The plaintext is a random string of bits.
- **solution**
 - a. Because $0 \oplus k_i = k_i$, the ciphertext stream is the same as the key stream. If the key stream is random, the ciphertext is also random. The patterns in the plaintext are not preserved in the ciphertext.

Example..contd...

- Because $1 \oplus k_i = \bar{k}_i$ where \bar{k}_i is the complement of k_i , the ciphertext stream is the complement of the key stream. If the key stream is random, the ciphertext is also random. Again the patterns in the plaintext are not preserved in the ciphertext.
- In this case, each bit in the ciphertext stream is either the same as the corresponding bit in the key stream or the complement of it. Therefore, the result is also a random string if the key stream is random.
- In this case, the ciphertext is definitely random because the exclusive-or of two random bits results in a random bit.

Feedback Shift Register

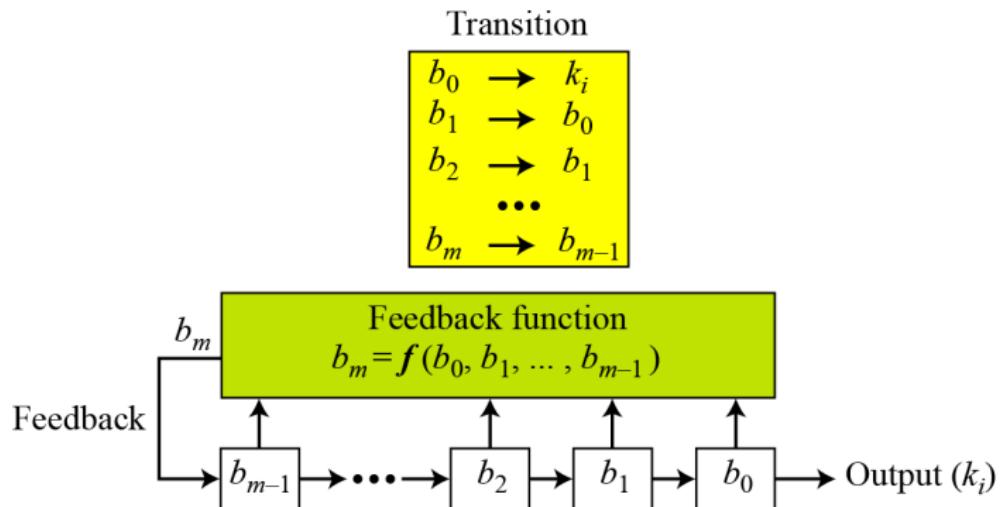
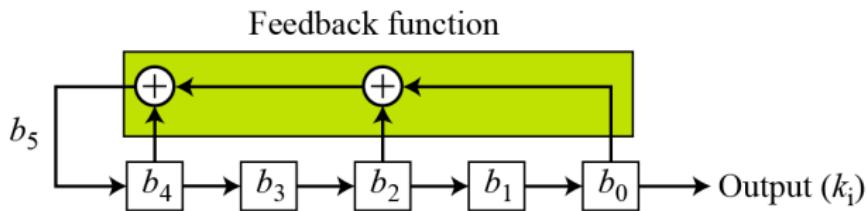


Figure 18: Feedback shift Register.

Example..1

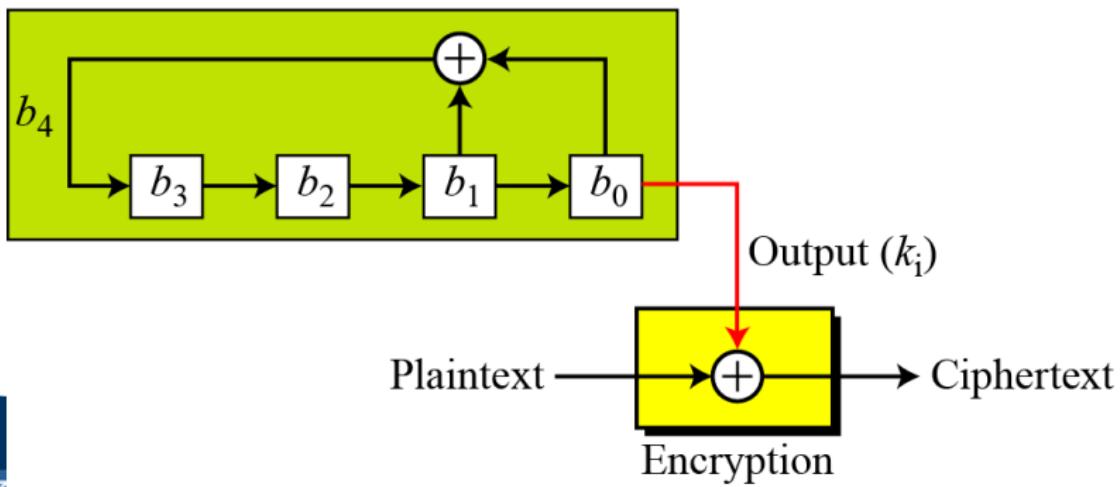
- Create a linear feedback shift register with 5 cells in which $b_5 = b_4 \oplus b_2 \oplus b_0$.
- If $c_i = 0$, b_i has no role in calculation of b_m . This means that b_i is not connected to the feedback function. If $c_i = 1$, b_i is involved in calculation of b_m . In this example, c_1 and c_3 are 0's, which means that we have only three connections. following figure shows the design.



Example..2

- Create a linear feedback shift register with 4 cells in which $b_4 = b_1 \oplus b_0$. Show the value of output for 20 transitions (shifts) if the seed is $(0001)_2$.

Key stream generator



Example2: Cell values and key sequence..

States	b_4	b_3	b_2	b_1	b_0	k_i
Initial	1	0	0	0	1	
1	0	1	0	0	0	1
2	0	0	1	0	0	0
3	1	0	0	1	0	0
4	1	1	0	0	1	0
5	0	1	1	0	0	1
6	1	0	1	1	0	0
7	0	1	0	1	1	0
8	1	0	1	0	1	1
9	1	1	0	1	0	1
10	1	1	1	0	1	0

Example2: Cell values and key sequence... cont..2

11	1	1	1	1	0	1
12	0	1	1	1	1	0
13	0	0	1	1	1	1
14	0	0	0	1	1	1
15	1	0	0	0	1	1
16	0	1	0	0	0	1
17	0	0	1	0	0	0
18	1	0	0	1	0	0
19	1	1	0	0	1	0
20	1	1	1	0	0	1

Nonsynchronous Stream Cipher

- In a nonsynchronous stream cipher, each key in the key stream depends on previous plaintext or ciphertext.

Data Encryption Standard (DES)

- The most widely used cipher in civilian applications.
- Developed by IBM; Evolved from Lucifer.
- Accepted as an US NBS standard in 1977, and later as an international standard, the National Institute of Standards and Technology (NIST).
- A block cipher with **N = 64 bit blocks**.
- **56-bit keys** (eight bytes, in each byte seven bits are used; the eighth bit can be used as a parity bit).
- Exhaustive search requires 2^{56} encryption steps (2^{55} on average).
- Iterates a round-function 16 times in **16 rounds**. The round-function mixes the data with the key.
- Each round, the key information entered to the round function is called a subkey. The subkeys K_1, \dots, K_{16} are computed by **a key scheduling algorithm**.

DES Overview

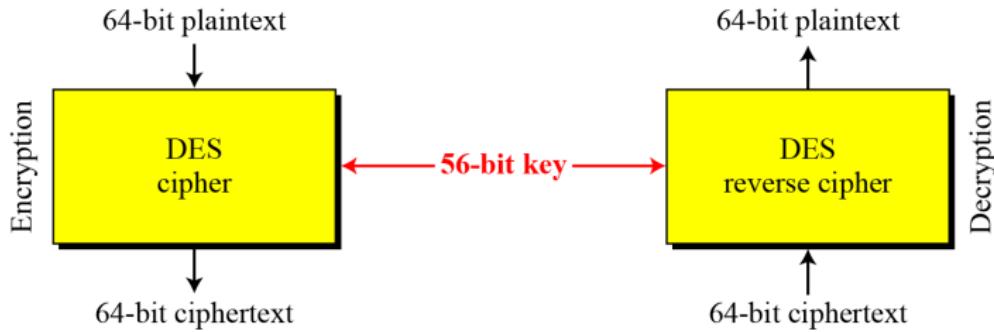
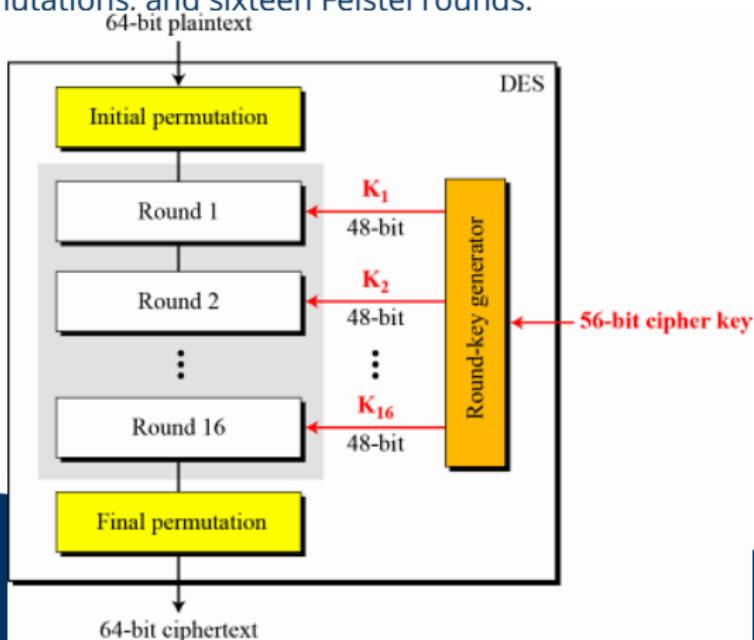


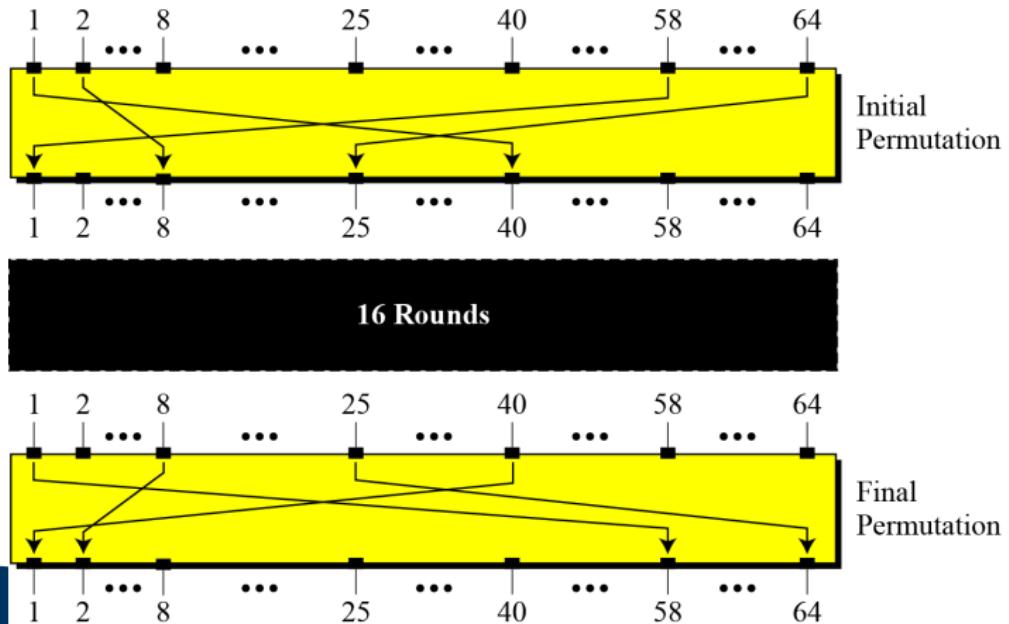
Figure 19: Encryption and decryption with DES.

General Structure of DES

- The encryption process is made of two permutations (P-boxes), which we call initial and final permutations, and sixteen Feistel rounds.



Initial and final permutation in DES



Initial and final permutation tables

<i>Initial Permutation</i>	<i>Final Permutation</i>
58 50 42 34 26 18 10 02	40 08 48 16 56 24 64 32
60 52 44 36 28 20 12 04	39 07 47 15 55 23 63 31
62 54 46 38 30 22 14 06	38 06 46 14 54 22 62 30
64 56 48 40 32 24 16 08	37 05 45 13 53 21 61 29
57 49 41 33 25 17 09 01	36 04 44 12 52 20 60 28
59 51 43 35 27 19 11 03	35 03 43 11 51 19 59 27
61 53 45 37 29 21 13 05	34 02 42 10 50 18 58 26
63 55 47 39 31 23 15 07	33 01 41 09 49 17 57 25

Figure 22: Initial and final permutation tables in DES.

Example1

- Find the output of the final permutation box when the input is given in hexadecimal as:

0x0000 0080 0000 0002

- Solution:**

Only bit 25 and bit 63 are 1s; the other bits are 0s. In the final permutation, bit 25 becomes bit 64 and bit 63 becomes bit 15. The result is

0x0002 0000 0000 0001

Example2

- Prove that the initial and final permutations are the inverse of each other by finding the output of the initial permutation if the input is

0x0002 0000 0000 0001

- Solution:** The input has only two 1s; the output must also have only two 1s. Using Table 6.1, we can find the output related to these two bits. Bit 15 in the input becomes bit 63 in the output. Bit 64 in the input becomes bit 25 in the output. So the output has only two 1s, bit 25 and bit 63. The result in hexadecimal is

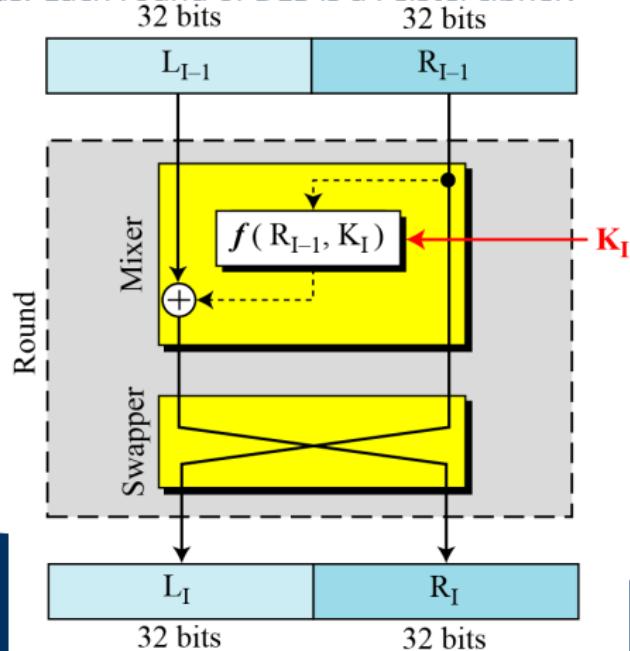
0x0000 0080 0000 0002

Cryptographic significance of initial and final permutations

- The initial and final permutations are straight P-boxes that are inverses of each other. They have no cryptography significance in DES.

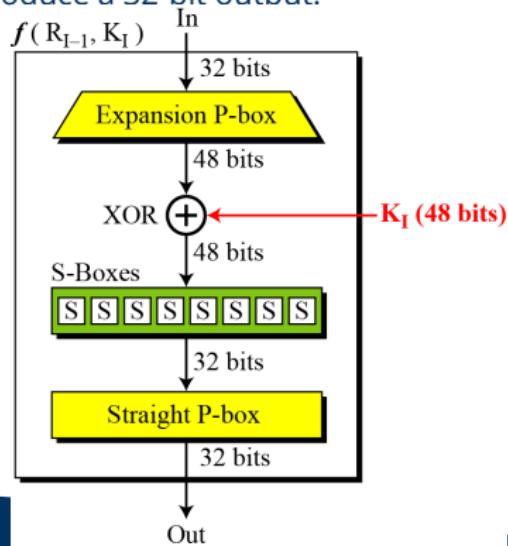
Rounds

- DES uses 16 rounds. Each round of DES is a Feistel cipher.



DES function

- The heart of DES is the DES function. The DES function applies a 48-bit key to the rightmost 32 bits to produce a 32-bit output.



Expansion P-Box in DES

- Since R_{i-1} is a 32-bit input and K_i is a 48-bit key, we first need to expand R_{i-1} to 48 bits.

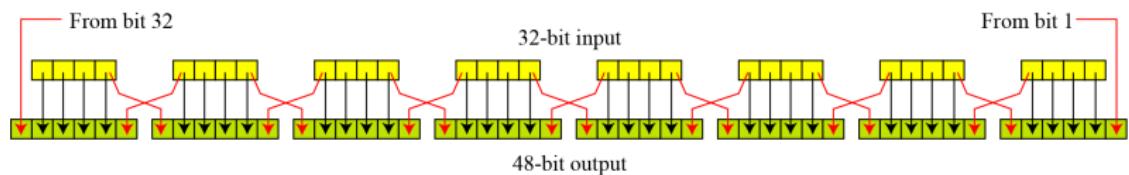


Figure 25: Expansion P-box

Contd...

- Although the relationship between the input and output can be defined mathematically, DES uses Table 6.2 to define this P-box.

32	01	02	03	04	05
04	05	06	07	08	09
08	09	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	31	31	32	01

Figure 26: Expansion P-box Table

Whitener (XOR) in DES

- After the expansion permutation, DES uses the XOR operation on the expanded right section and the round key. Note that both the right section and the key are 48-bits in length. Also note that the round key is used only in this operation.

S-Boxes in DES

- The S-boxes do the real mixing (confusion). DES uses 8 S-boxes, each with a 6-bit input and a 4-bit output.

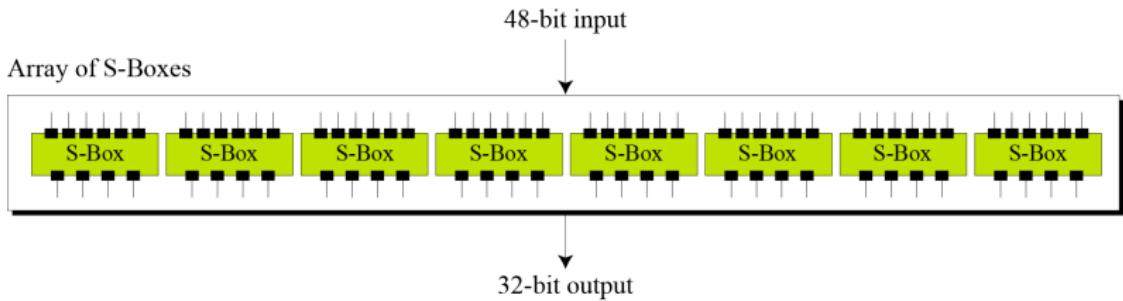
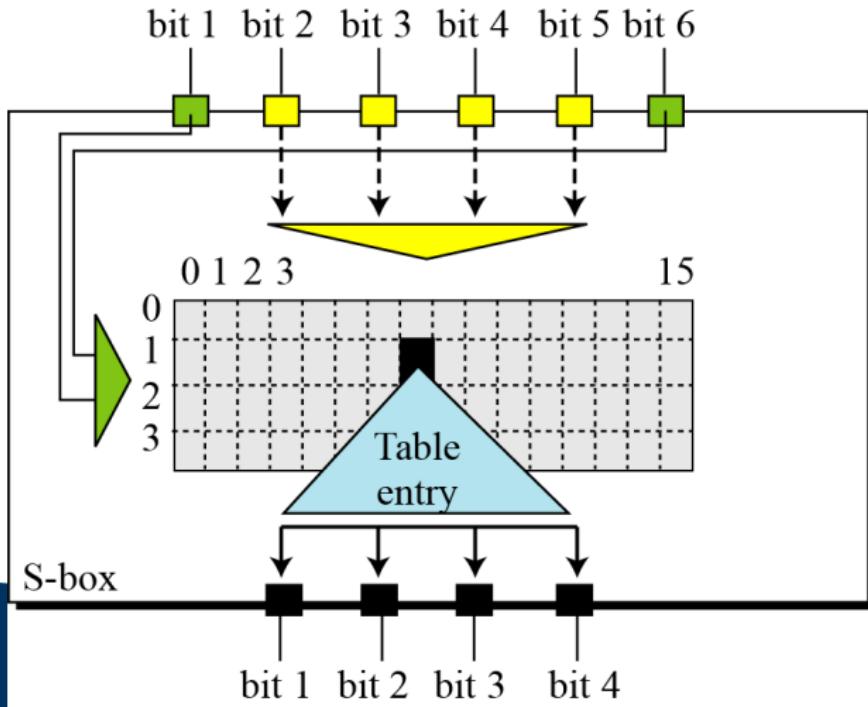


Figure 27: S-boxes

S-box rule for DES



permutation for S-box 1

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	04	13	01	02	15	11	08	03	10	06	12	05	09	00	07
1	00	15	07	04	14	02	13	10	03	06	12	11	09	05	03	08
2	04	01	14	08	13	06	02	11	15	12	09	07	03	10	05	00
3	15	12	08	02	04	09	01	07	05	11	03	14	10	00	06	13

Figure 29: permutation for S-box 1, rest can be checked from textbook.

Example 1

- When the S-box 1 (in book Table 6.3) is referred and the input to S-box 1 is 100011.
What is the output?
- Solution:**
If we write the first and the sixth bits together, we get 11 in binary, which is 3 in decimal. The remaining bits are 0001 in binary, which is 1 in decimal. We look for the value in row 3, column 1, in Table 6.3 (S-box 1). The result is 12 in decimal, which in binary is 1100. So the input 100011 yields the output 1100.

Example 2

- When the S-box 8 (in book Table 6.10) is referred and the input to S-box 8 is 000000. What is the output?
- Solution:**
If we write the first and the sixth bits together, we get 00 in binary, which is 0 in decimal. The remaining bits are 0000 in binary, which is 0 in decimal. We look for the value in row 0, column 0, in Table 6.10 (S-box 8). The result is 13 in decimal, which is 1101 in binary. So the input 000000 yields the output 1101.

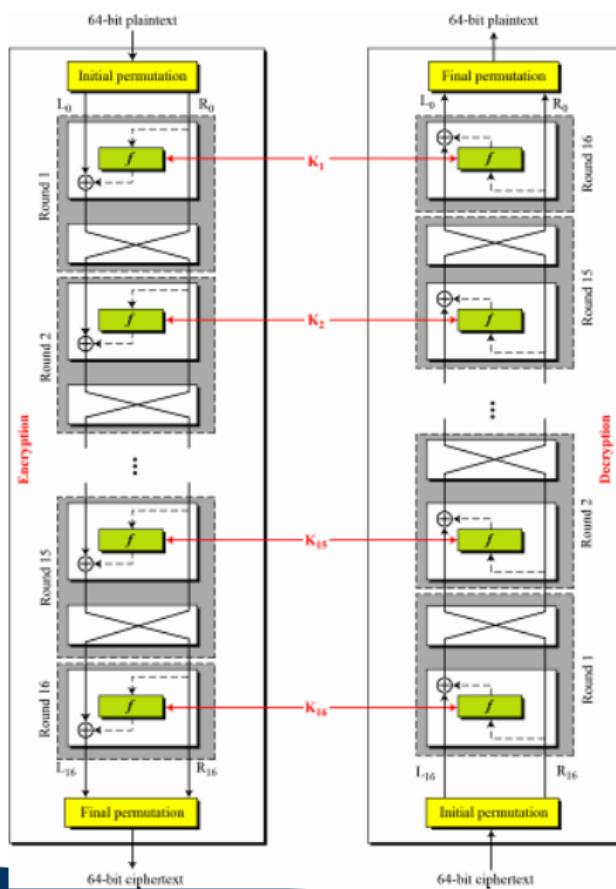
Straight Permutation table in DES function

16	07	20	21	29	12	28	17
01	15	23	26	05	18	31	10
02	08	24	14	32	27	03	09
19	13	30	06	22	11	04	25

Figure 30: Straight Permutation table

Cipher and Reverse Cipher

- Using mixers and swappers, we can create the cipher and reverse cipher, each having 16 rounds.
- **First approach:** To achieve this goal, one approach is to make the last round (round 16) different from the others; it has only a mixer and no swapper.
 - *In the first approach, there is no swapper in the last round.*



DES cipher and reverse cipher for the first approach.

Pseudocode for DES cipher

```
Cipher (plainBlock[64], RoundKeys[16, 48], cipherBlock[64])
{
    permute (64, 64, plainBlock, inBlock, InitialPermutationTable)
    split (64, 32, inBlock, leftBlock, rightBlock)
    for (round = 1 to 16)
    {
        mixer (leftBlock, rightBlock, RoundKeys[round])
        if (round!=16) swapper (leftBlock, rightBlock)
    }
    combine (32, 64, leftBlock, rightBlock, outBlock)
    permute (64, 64, outBlock, cipherBlock, FinalPermutationTable)
}
```

Pseudocode for DES cipher.. Contd...1

```
mixer (leftBlock[48], rightBlock[48], RoundKey[48])
{
    copy (32, rightBlock, T1)
    function (T1, RoundKey, T2)
        exclusiveOr (32, leftBlock, T2, T3)
        copy (32, T3, rightBlock)
}

swapper (leftBlock[32], righthBlock[32])
{
    copy (32, leftBlock, T)
    copy (32, rightBlock, leftBlock)
    copy (32, T, rightBlock)
}
```

Pseudocode for DES cipher.. Contd...2

```
substitute (inBlock[32], outBlock[48], SubstitutionTables[8, 4, 16])
{
    for (i = 1 to 8)
    {
        row ← 2 × inBlock[i × 6 + 1] + inBlock [i × 6 + 6]
        col ← 8 × inBlock[i × 6 + 2] + 4 × inBlock[i × 6 + 3] +
              2 × inBlock[i × 6 + 4] + inBlock[i × 6 + 5]

        value = SubstitutionTables [i][row][col]

        outBlock[[i × 4 + 1] ← value / 8;           value ← value mod 8
        outBlock[[i × 4 + 2] ← value / 4;           value ← value mod 4
        outBlock[[i × 4 + 3] ← value / 2;           value ← value mod 2
        outBlock[[i × 4 + 4] ← value
    }
}
```

Alternative approach

- We can make all 16 rounds the same by including one swapper to the 16th round and add an extra swapper after that (two swappers cancel the effect of each other).

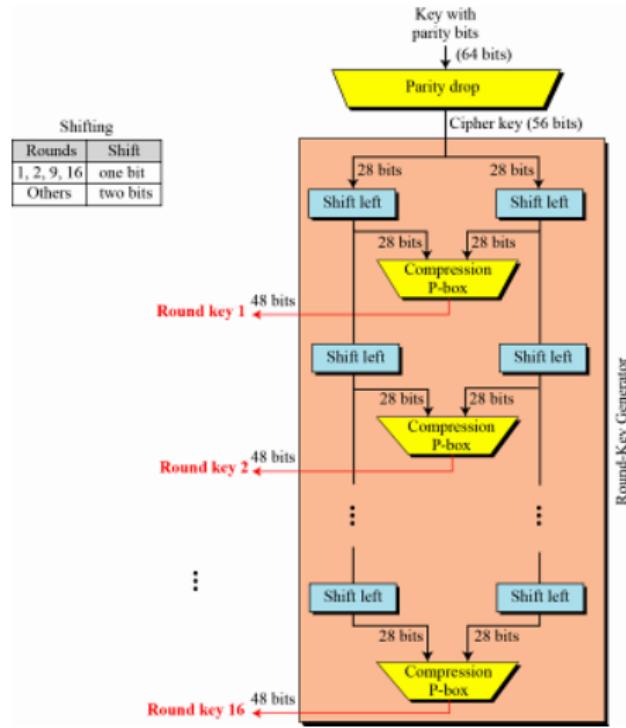


Figure 32: Key Generation: The round-key generator creates sixteen 48-bit keys out of a 56-bit cipher key.

Parity-bit Drop Table

- The preprocess before key expansion is a compression transposition step that we call **parity-bit drop**.
- It drops the parity bits (bits 8, 16, 24, 32,..., 64) from 64 -bit key and permutes the rest of the bits according to the following table.

57	49	41	33	25	17	09	01
58	50	42	34	26	18	10	02
59	51	43	35	27	19	11	03
60	52	44	36	63	55	47	39
31	23	15	07	62	54	46	38
30	22	14	06	61	53	45	37
29	21	13	05	28	20	12	04

Shift-Left

- After the straight permutation, the key is divided into two 28-bit parts. Each part is shifted left (circular shift) one or two bits.
- In rounds 1, 2, 9, and 16, shifting is one bit; in the other rounds, it is two bits.
- The two parts are then combined to form 56-bit part.

Round	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Bit shifts	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Figure 33: Number of shifts for each round.

Key-compression in Key Generation in DES

- The compression D-box or P-box changes the 58 bits to 48 bits, which are used as a key for a round.

14	17	11	24	01	05	03	28
15	06	21	10	23	19	12	04
26	08	16	07	27	20	13	02
41	52	31	37	47	55	30	40
51	45	33	48	44	49	39	56
34	53	46	42	50	36	29	32

Figure 34: **Key-compression table**

Algorithm for round-key generation..part1

```
Key_Generator (keyWithParities[64], RoundKeys[16, 48], ShiftTable[16])
{
    permute (64, 56, keyWithParities, cipherKey, ParityDropTable)
    split (56, 28, cipherKey, leftKey, rightKey)
    for (round = 1 to 16)
    {
        shiftLeft (leftKey, ShiftTable[round])
        shiftLeft (rightKey, ShiftTable[round])
        combine (28, 56, leftKey, rightKey, preRoundKey)
        permute (56, 48, preRoundKey, RoundKeys[round], KeyCompressionTable)
    }
}
```

Algorithm for round-key generation..part2

```
shiftLeft (block[28], numOfShifts)
{
    for (i = 1 to numOfShifts)
    {
        T ← block[1]
        for (j = 2 to 28)
        {
            block [j-1] ← block [j]
        }
        block[28] ← T
    }
}
```

Example1

- We choose a random plaintext block and a random key, and determine what the ciphertext block would be (all in hexadecimal):

Plaintext: 123456ABCD132536

Key: AABB09182736CCDD

CipherText: C0B7A8D05F3A829C

Plaintext: 123456ABCD132536

After initial permutation: 14A7D67818CA18AD

After splitting: $L_0 = 14A7D678$ $R_0 = 18CA18AD$

Round	Left	Right	Round Key
Round 1	18CA18AD	5A78E394	194CD072DE8C
Round 2	5A78E394	4A1210F6	4568581ABCCE
Round 3	4A1210F6	B8089591	06EDA4ACF5B5
Round 4	B8089591	236779C2	DA2D032B6EE3

Example1..Trace of Data.. continued..

<i>Round 5</i>	236779C2	A15A4B87	69A629FEC913
<i>Round 6</i>	A15A4B87	2E8F9C65	C1948E87475E
<i>Round 7</i>	2E8F9C65	A9FC20A3	708AD2DDB3C0
<i>Round 8</i>	A9FC20A3	308BEE97	34F822F0C66D
<i>Round 9</i>	308BEE97	10AF9D37	84BB4473DCCC
<i>Round 10</i>	10AF9D37	6CA6CB20	02765708B5BF
<i>Round 11</i>	6CA6CB20	FF3C485F	6D5560AF7CA5
<i>Round 12</i>	FF3C485F	22A5963B	C2C1E96A4BF3
<i>Round 13</i>	22A5963B	387CCDAA	99C31397C91F
<i>Round 14</i>	387CCDAA	BD2DD2AB	251B8BC717D0
<i>Round 15</i>	BD2DD2AB	CF26B472	3330C5D9A36D
<i>Round 16</i>	19BA9212	CF26B472	181C5D75C66D
<i>After combination:</i> 19BA9212CF26B472			
<i>Ciphertext:</i> C0B7A8D05F3A829C		<i>(after final permutation)</i>	

Decryption/Deciphering at Receiver's end

- Let us see how Bob, at the destination, can decipher the ciphertext received from Alice using the same key. The following Table shows some interesting points.

Ciphertext: C0B7A8D05F3A829C			
After initial permutation: 19BA9212CF26B472			
After splitting: $L_0=19BA9212$ $R_0=CF26B472$			
Round	Left	Right	Round Key
Round 1	CF26B472	BD2DD2AB	181C5D75C66D
Round 2	BD2DD2AB	387CCDAA	3330C5D9A36D
...
Round 15	5A78E394	18CA18AD	4568581ABCCE
Round 16	14A7D678	18CA18AD	194CD072DE8C
After combination: 14A7D67818CA18AD			
Plaintext: 123456ABCD132536		(after final permutation)	

Bibliography: Books and Resources

- Cryptography and Network Security: Principles and Practice by William Stallings
- Cryptography and Network Security by Behrouz A Forouzan and Debdeep Mukhopadhyay
- Principles of Information Security by Michael E. Whitman and Herbert J. Mattord.
- Cisco platform, and Internet.
- Published research papers, study materials from researchers of security domain.

CO-INS:Information and Network Security

UNIT-I

Course Instructors:

Soma Saha

Virendra Srivastava

Soma Saha (PhD)

Department of Computer Engineering
SGSITS Indore, India

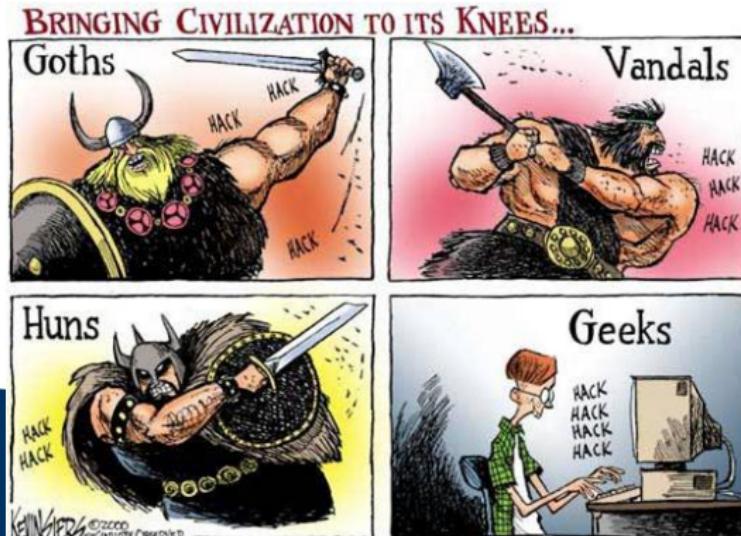
Feb 2021

Information Security

- What is Information?
- Why do we need security?
 1. Past

Information Security

- What is Information?
- Why do we need security?
 1. Past
 2. Present



Information Security in an Enterprise

"well-informed sense of assurance that the information risks and controls are in balance."—James Anderson, executive consultant at Emagined Security, Inc.

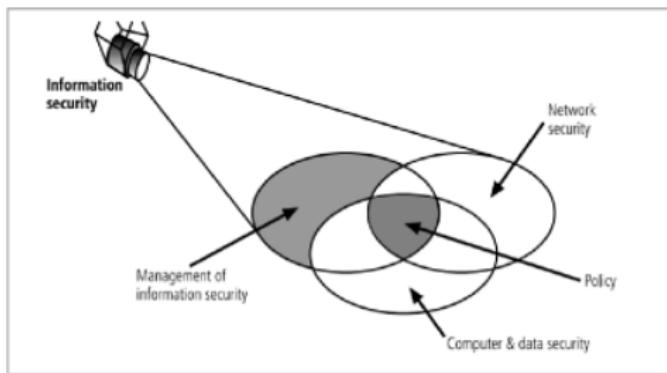


Figure 2: Components of Information Security.

Need for Security

Adversary	Goal
Student	To have fun snooping on people's e-mail
Cracker/Hacker	To test out someone's security system &/ steal data
Sales Rep.	To claim to represent all of Europe, not just Seychelles
Businessman	To discover nearest competitors strategic marketing plan
Ex-Employee	To get revenge for being fired
Accountant	To embezzle/rob money from a company
Stockbroker	To deny a promise made to a customer by e-mail
Spy	To learn an enemy's military or industrial secrets
Terrorist	To steal germ warfare secrets
Con Man	To steal credit card numbers for sale

Who is vulnerable?

- Government and defense agencies
- Financial institutions and banks
- Internet service providers
- Pharmaceutical companies
- Contractors to various government agencies
- Multinational corporations
- **ANYONE ON THE NETWORK**

Security Terminologies

- Information Security
- Network Security
- Cyber Security

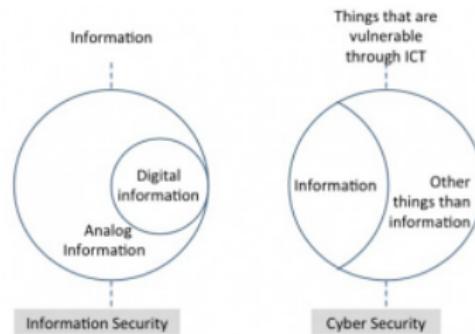


Figure 3: Information-ICT-Cyber Security.

InfoSec, ICT Security, Cyber Security

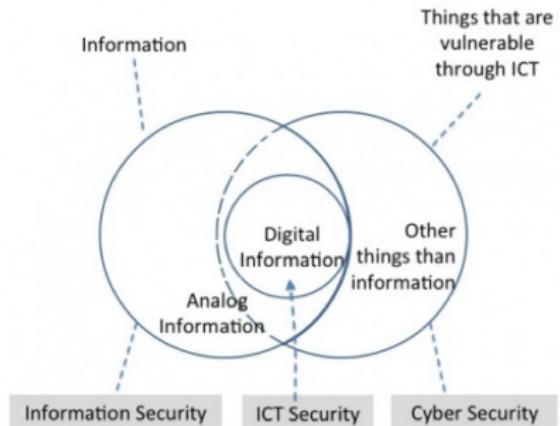


Figure 4: InfoSec-ICT Security-Cyber Security.

Threat-Vulnerability-Risk



Figure 5: Threat-Vulnerability-Risk.

UNIT-I: Learning Objectives

Upon completion of this unit, you should be able to

- LO1 Define the three types of security goals and explain attacks that threaten confidentiality, integrity, and availability
- LO2 Infer the different categories of cryptographic attacks
- LO3 Relate to the relationship between security services and their mechanisms
- LO4 Outline the different techniques needed for implementing security goals
- LO5 Threat, vulnerability, and risk estimation in an enterprise, legal and ethical issues in computer security

LO1: Security Goals



Figure 6: Security Goals.

CIA Triad

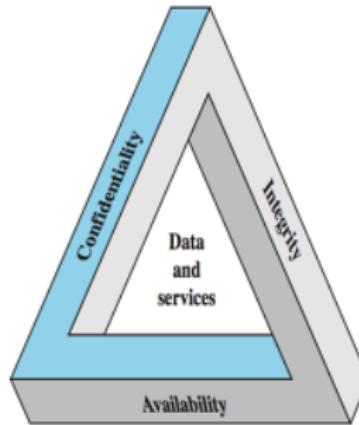


Figure 7: Security Concepts.

LO2: Cryptographic Attacks

- * What is Cryptography?

LO2: Cryptographic Attacks

- * What is Cryptography?

Example: **rwox{vj}rxw**

LO2: Cryptographic Attacks

- * What is Cryptography?

Example: **rwox{vj}rxw**

Original Message: information

1. Cryptanalytic attacks.
2. Non-Cryptanalytic attacks.

Cryptographic Attacks

- Classification based on encryption techniques:
 1. Ciphertext-only attack
 2. Known plaintext attack
 3. Chosen Plaintext attack
 4. Chosen Ciphertext attack
 5. Chosen text attack

Cryptanalytic attacks

- "Skill in the production of cryptanalysis has always been heavily on the side of the professionals, but innovation, particularly in the design of new types of cryptographic systems, has come primarily from amateurs."—Diffie and Hellman.
- **Cryptanalytic attacks tries to attack mathematical weaknesses in the algorithms.**

Cryptanalytic attacks

- Cryptanalytic attacks:
 1. Linear Cryptanalysis
 2. Differential Cryptanalysis
- **Side Channel Attacks/Implementation Attacks:**
Tries to attack the specific implementation of the cipher. (such as a smartcard system).
 - Power Analysis.
 - Timing Analysis.
 - Fault Induction.
 - TEMPEST.
 - Differential Power Analysis.

Differential Power Analysis

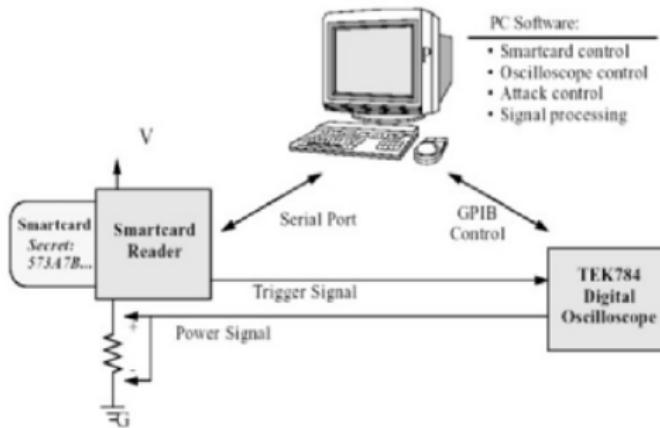
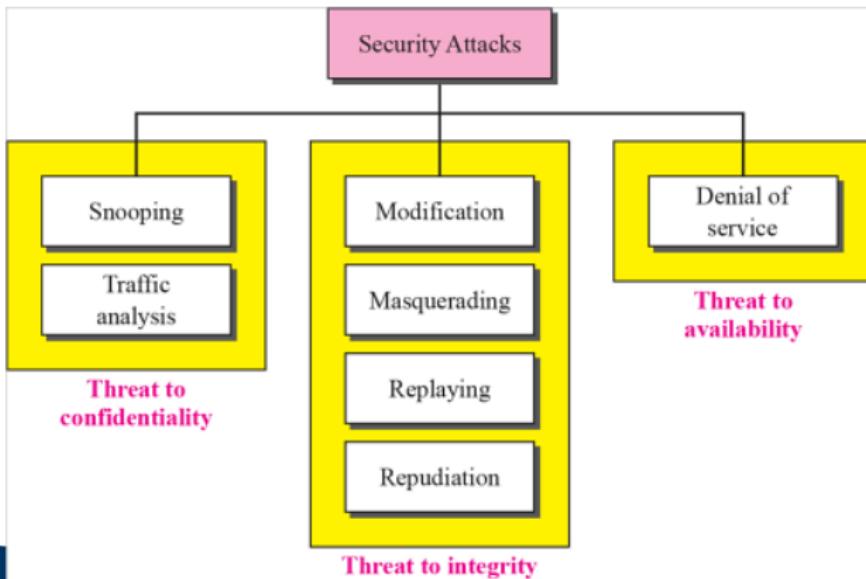


Figure 8: An example setup for a Differential Power Analysis attack on a smartcard.

Non-cryptanalytic Attacks



Threat to confidentiality: Snooping

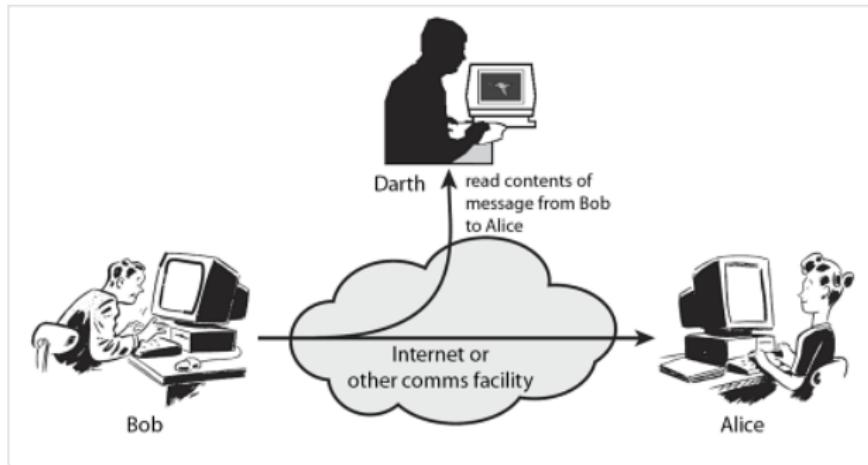


Figure 10: Snooping/Interception.

Threat to confidentiality: Traffic Analysis

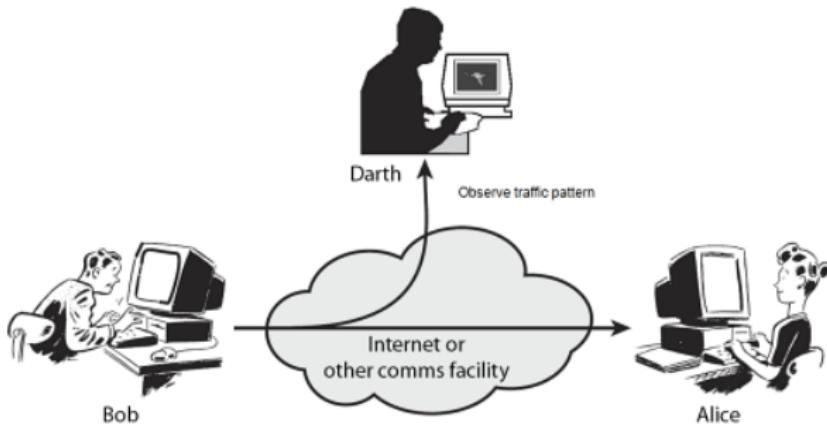


Figure 11: Traffic analysis.

Threat to Integrity: Modification

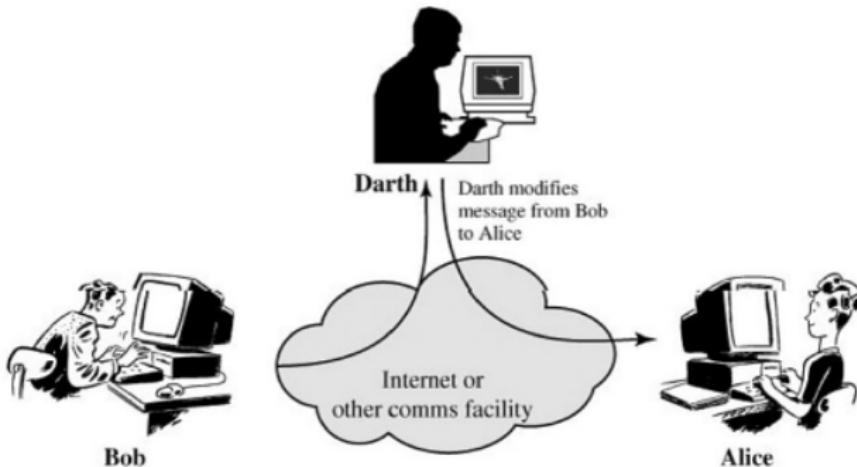


Figure 12: Modification.

Threat to Integrity: Masquerading

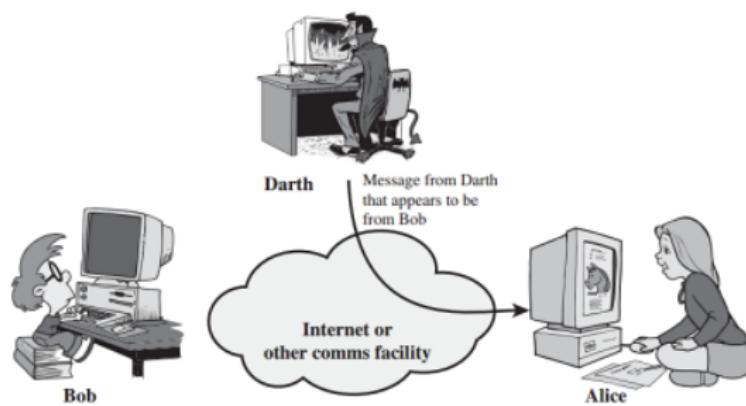


Figure 13: Masquerading.

Threat to Integrity: Replayng

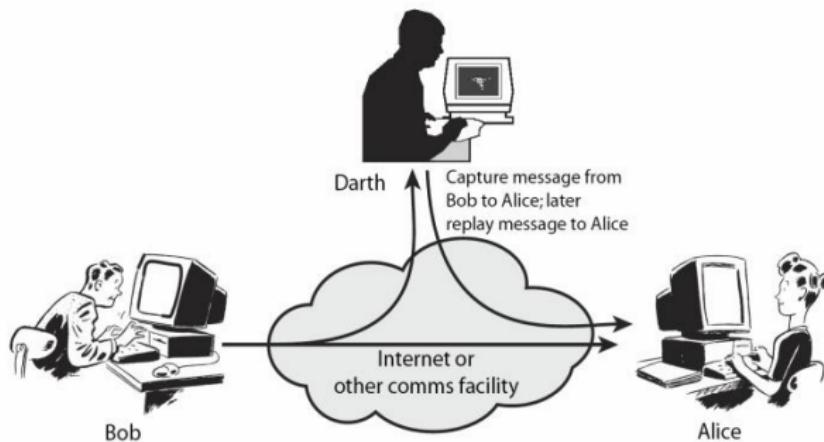


Figure 14: Replayng.

Threat to Integrity: Repudiation

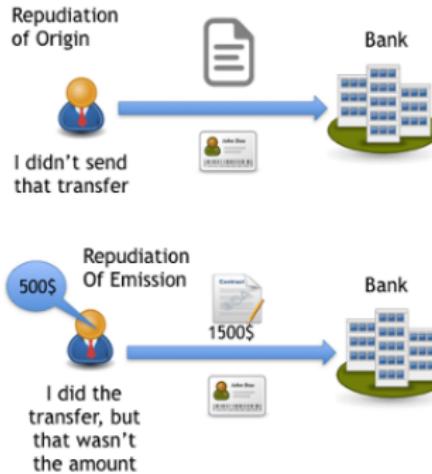
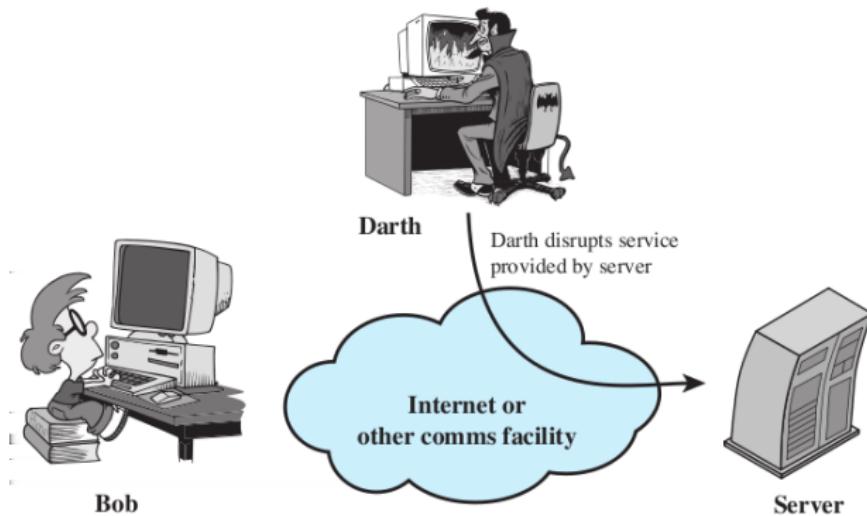


Figure 15: Different modes of repudiation.

Threat to Availability: Denial of Service



Handling Attacks

- * Passive Attacks – focus on Prevention
 - 1. Easy to stop
 - 2. Hard to detect

Handling Attacks

- * Passive Attacks – focus on Prevention

1. Easy to stop
2. Hard to detect

- * Active Attacks – focus on Detection and Recovery

1. Hard to stop
2. Easy to detect

Categorization of passive and active attacks

Attacks	Passive/Active	Threatening
Snooping, Traffic Analysis	Passive	Confidentiality
Modification, Masquerading, Replaying, Repudiation	Active	Integrity
Denial of service	Active	Availability

Table 2: Categorization of passive and active attacks.

Common Attacks

- XSS (Cross-Site Scripting)
- Cross-Site Request Forgery (CSRF)
- SQL Injection
- Man In The Middle
- DoS and DDoS
- Phishing Attack
- Zero Day Attack

XSS/Cross-site Scripting

- Security vulnerability typically found in web applications.
- Code injection attack, allows an attacker to execute malicious code (e.g. JavaScripts) into victim's web browser.
- **Attacker's Goal:** To steal the victim's credentials, such as cookies.
 1. Server-side XSS Attack
 2. Client-side XSS attack

For questing mind: <https://engineering.purdue.edu/kak/compsec/NewLectures/Lecture28.pdf>

XSS/Cross-site Scripting: Example

(slide courtesy: Veerendra Srivastava)

```
http://localhost:8080/DVWA/vulnerabilities/xss_r/?name=<h3>Please login to proceed</h3><form action=http://192.168.149.128>Username:<br><input type="username" name="username"></br>Password:<br><input type="password" name="password"></br><br><input type="submit" value="Logon"></br>
```

Figure 17: JavaScript code injected in localhost.

XSS/Cross-site Scripting: Example..(slide courtesy: Veerendra Srivastava)

The screenshot shows a web application interface. On the left, a vertical menu bar lists various security vulnerabilities: Home, Instructions, Setup / Reset DB, Brute Force, Command Injection, CSRF, File Inclusion, File Upload, Insecure CAPTCHA, SQL Injection, SQL Injection (Blind), Weak Session IDs, XSS (DOM), XSS (Reflected) (which is highlighted in green), and XSS (Stored). The main content area has a title "Vulnerability: Reflected Cross Site Scripting (XSS)". It contains a form with a text input field labeled "What's your name?" and a "Submit" button. Below the input field, the text "Hello" appears in red, followed by "Please login to proceed". A red rectangular box highlights this text and the subsequent login form fields: "Username:" and "Password:", each with its own input box, and a "Logon" button.

Vulnerability: Reflected Cross Site Scripting (XSS)

What's your name? Submit

Hello **Please login to proceed**

Username:

Password:

Logon

More Information

- [https://www.owasp.org/index.php/Cross_site_Scripting_\(XSS\)](https://www.owasp.org/index.php/Cross_site_Scripting_(XSS))
- https://www.owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet
- https://en.wikipedia.org/wiki/Cross-site_scripting
- <http://www.cgisecurity.com/xss-faq.html>
- <http://www.scriptalert1.com/>

Cross-Site Request Forgery (CSRF)

- A CSRF attack involves a victim user, a trusted site, and a malicious site.
- The victim user holds an active session with a trusted site and simultaneously visits a malicious site. The malicious site injects a HTTP request for the trusted site into the victim user session compromising its integrity.

SQL Injection

- A code injection technique, exploits the vulnerabilities in the interface between web applications and database servers.

SQL Injection

- A code injection technique, exploits the vulnerabilities in the interface between web applications and database servers.
- The vulnerability is present when user's inputs are not correctly checked within the web applications before sending to the back-end database servers.

Basic Picture: SQL Injection

(Slide courtesy: Dan Boneh)

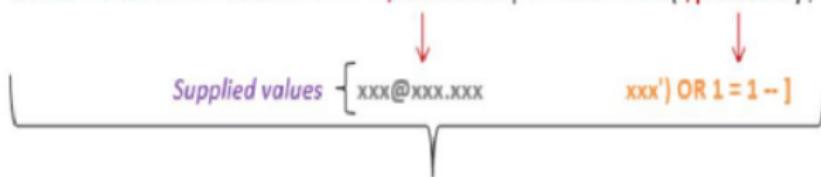


SQL Injection: Examples

Recent Past: Bloodx1.0: <https://www.exploit-db.com/exploits/47842>

SQL Injection: Examples..(slide courtesy: Veerendra Srivastava)

```
SELECT * FROM users WHERE email = '$email' AND password = md5('$password');
```



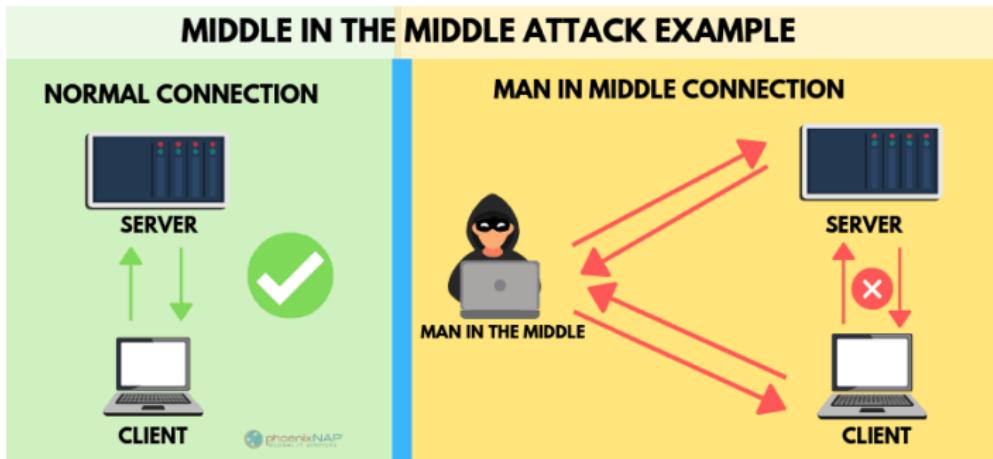
```
SELECT * FROM users WHERE email = 'xxx@xxx.xxx' AND password = md5('xxx') OR 1=1--');
```

```
SELECT * FROM users WHERE FALSE AND FALSE OR TRUE
```

```
SELECT * FROM users WHERE FALSE OR TRUE
```

```
SELECT * FROM users WHERE TRUE
```

Man-In-The-Middle (MITM) Attack/Janus Attack/Fire Brigade Attack.



Janus Attack/Fire Brigade Attack.

Requirement to execute an MITM Attack/Janus Attack/Fire Brigade Attack

- Sniffing the victim

Requirement to execute an MITM Attack/Janus Attack/Fire Brigade Attack

- Sniffing the victim
- Ensuring that the original packet does not reach the recipient
- Forwarding the modified packet

Requirement to execute an MITM Attack/Janus Attack/Fire Brigade Attack

- Sniffing the victim
- Ensuring that the original packet does not reach the recipient
- Forwarding the modified packet
- Tricking potential security systems such as SSL

DoS Attack and DDoS Attack

- DOS Attack is a malicious attempt by a single person or a group of people to cause the victim, site or node to deny service to its legitimate customers.

DoS Attack and DDoS Attack

- DOS Attack is a malicious attempt by a single person or a group of people to cause the victim, site or node to deny service to its legitimate customers.
 - * **DoS** -> when a single host attacks
 - * **DDoS** -> when multiple hosts attack simultaneously
- The goal of DoS or DDoS is usually service denial or setting up a different, second attack.

DoS Attack: Variety

- **Bandwidth Consumption:** All available bandwidth used by the attacker e.g., ICMP ECHO attack.
- **Resource Consumption:** Resources like web server, print or mail server flooded with useless requests e.g., mail bomb
- **Network Connectivity:** The attacker forces the server to stop communicating on the network e.g., SYN Flooding.
- Quick Guide for DDoS Attacks: <https://us-cert.cisa.gov/sites/default/files/publications/DDoS%20Quick%20Guide.pdf>

Phishing Attack

- Phishing is a type of **social engineering attack** often used to steal user data, including login credentials and credit card numbers.
- Occurs when an attacker pretends to be a trusted entity to dupe a victim into clicking a malicious link, that can lead to the installation of malware, freezing of the system as part of a ransomware attack, or revealing of sensitive information.
- Variety: <https://www2.deloitte.com/content/dam/Deloitte/sg/Documents/risk/sea-risk-cyber-101-part10.pdf>

Zero Day Attack

- "Zero-day" refers a **newly discovered software vulnerability**.
- Refers the fact that the developers have "**zero days**" to fix the problem which has just been exposed — and may be already exploited by hackers.
- Link: <https://www.blackhat.com/docs/eu-17/materials/eu-17-Ablon-Zero-Days-Thousands-Of-Nights-The-Life-And-Times-Of.pdf>

Zero Day Attack...



Figure 22: Hiding vulnerabilities.(Image source: Economics Times, India, Feb 14, 2019)

Prevention of Zero-Day Attack

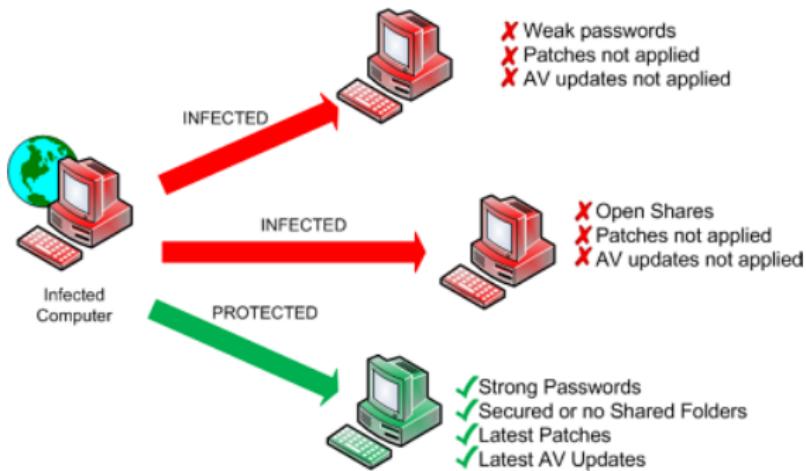


Figure 23: Zero-Day Attack preventive measures.

Security Services and Mechanism

- ITU-T (International Telecommunication Union-Telecommunication Standardization Sector) provides some security services and some mechanisms to implement those services.
- A mechanism or combination of mechanisms are used to provide a service.
- A mechanism can be used in one or more services.

Security Services and Mechanism

- ITU-T (International Telecommunication Union-Telecommunication Standardization Sector) provides some security services and some mechanisms to implement those services.
- A mechanism or combination of mechanisms are used to provide a service.
- A mechanism can be used in one or more services.
- Thus, the three aspects of security, **(i) security attacks, (ii) security mechanisms and (iii) security services** are related.

Security Services

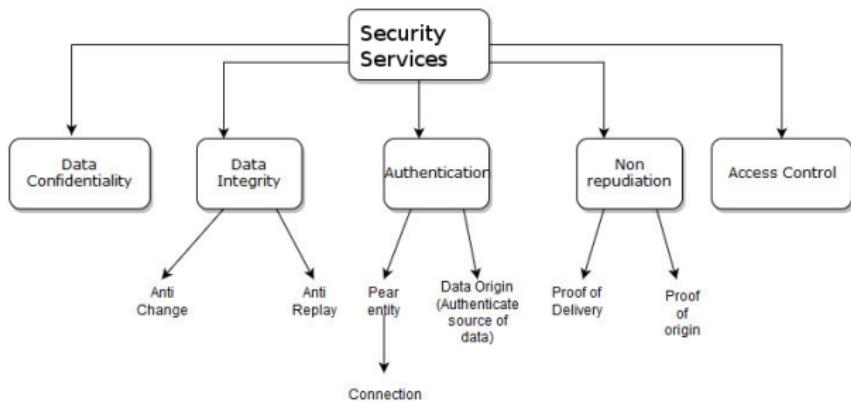


Figure 24: Security Services: ITU-T (X.800) has define five services related to the security goals and attacks (discussed..).

Security Services: Protection

- **Data Confidentiality:**

- Designed to protect data from disclosure attack.
- The service defined by X.800 is very broad and encompasses confidentiality of whole message or part of a message and also protect against traffic analysis.

- **Data Integrity:**

- Designed to protect data from modification, insertion, deletion, and replaying by an adversary.
- It may protect the whole message or part of the message.

Security Services: Protection....cont..1

- **Authentication:** This service provides the **authentication of the party** at the other end of the line.

Security Services: Protection....cont..1

- **Authentication:** This service provides the **authentication of the party** at the other end of the line.
 - **Peer entity authentication:** It provides authentication of the sender or receiver during the connection establishment in connection-oriented communication.
 - **Data origin authentication:** It authenticates the source of the data in connection-less communication.

Security Services: Protection....cont..1

- **Authentication:** This service provides the **authentication of the party** at the other end of the line.
 - **Peer entity authentication:** It provides authentication of the sender or receiver during the connection establishment in connection-oriented communication.
 - **Data origin authentication:** It authenticates the source of the data in connection-less communication.
- **Non-repudiation:** This service protects against repudiation by either the sender or the receiver of the data.

Security Services: Protection....cont..1

- **Authentication:** This service provides the **authentication of the party** at the other end of the line.
 - **Peer entity authentication:** It provides authentication of the sender or receiver during the connection establishment in connection-oriented communication.
 - **Data origin authentication:** It authenticates the source of the data in connection-less communication.
- **Non-repudiation:** This service protects against repudiation by either the sender or the receiver of the data.
 - **Proof of Origin:** The receiver of the data can later prove the identity of the sender if denied.

Security Services: Protection....cont..1

- **Authentication:** This service provides the **authentication of the party** at the other end of the line.
 - **Peer entity authentication:** It provides authentication of the sender or receiver during the connection establishment in connection-oriented communication.
 - **Data origin authentication:** It authenticates the source of the data in connection-less communication.
- **Non-repudiation:** This service protects against repudiation by either the sender or the receiver of the data.
 - **Proof of Origin:** The receiver of the data can later prove the identity of the sender if denied.
 - **Proof of Delivery:** The sender of the data can later prove that data were delivered to the intended recipient.

Security Services: Protection....cont..2

- **Access Control:**

- Provides protection against unauthorized access to data.
- Access control can involve reading, writing, modifying, executing programs, and so on.

Security Mechanisms



Figure 25: Security Mechanisms: ITU-T (X.800) recommends some security mechanisms to provide the security services (defined earlier section).

Security Mechanisms: Encipherment and Data Integrity

- **Encipherment:**

- Hiding or covering of data which provides confidentiality.
- Can also be used to complement other mechanisms to provide other services.
- Two techniques, **Cryptography** and **Steganography** are used for enciphering.

Security Mechanisms: Encipherment and Data Integrity

- **Encipherment:**

- Hiding or covering of data which provides confidentiality.
- Can also be used to complement other mechanisms to provide other services.
- Two techniques, **Cryptography** and **Steganography** are used for enciphering.

- **Data Integrity:**

- This mechanism appends to the data a short check value that has been created by a specific process from the data itself.
- Data integrity is preserved by comparing check value received to the check value generated.

Security Mechanisms: Digital Signature

- **Digital Signature:**

- A digital signature is a means by which the sender can electronically sign the data and the receiver can electronically verify the signature.
- The sender uses a process that involves showing that she owns a private key related to the public key that she has announced publicly.
- The receiver uses sender's public key to prove that the message is indeed signed by the sender who claims to have sent the message.

Security Mechanisms: Authentication Exchange and Traffic Padding

- **Authentication Exchange:**

- Two entities exchange some messages to prove their identity to each other.
- For example, one entity can prove that she knows a secret that only she is supposed to know.

Security Mechanisms: Authentication Exchange and Traffic Padding

- **Authentication Exchange:**

- Two entities exchange some messages to prove their identity to each other.
- For example, one entity can prove that she knows a secret that only she is supposed to know.

- **Traffic Padding:**

- Inserting some bogus data into the data traffic to thwart the adversary's attempt to use the traffic analysis.

Security Mechanisms: Routing Control and Notarization

- **Routing Control:**

- Selecting and continuously changing different available routes between the sender and the receiver to prevent the opponent from eavesdropping on a particular route.

Security Mechanisms: Routing Control and Notarization

- **Routing Control:**

- Selecting and continuously changing different available routes between the sender and the receiver to prevent the opponent from eavesdropping on a particular route.

- **Notarization:**

- Selecting a third trusted party to control the communication between two entities.
 - Various purposes, one is to prevent repudiation.
 - The receiver can involve a trusted party to store the sender request in order to prevent the sender from later denying that she has made such a request.

Security Mechanisms: Access control

- **Access control:**

- Uses methods to prove that a user has the access right to the data or resources owned by a system.
- Examples of proofs are passwords and PINs.

Security Mechanisms: Access control

- **Access control:**

- Uses methods to prove that a user has the access right to the data or resources owned by a system.
- Examples of proofs are passwords and PINs.

Relation between Services and Mechanisms

<i>Security Service</i>	<i>Security Mechanism</i>
Data confidentiality	Encipherment and routing control
Data integrity	Encipherment, digital signature, data integrity
Authentication	Encipherment, digital signature, authentication exchanges
Nonrepudiation	Digital signature, data integrity, and notarization
Access control	Access control mechanism

Figure 26: Relation between security services and security mechanisms.

Techniques for security goals implementation

- Security mechanisms (discussed..) are **only theoretical recipes** to implement security.
- The actual implementation of security goals needs some techniques.
- Two most important techniques:
 - Cryptography
 - Steganography

Cryptography Mechanisms

- **Past**
 - Encryption
 - Decryption
- **Today**
 - Symmetric-key Encipherment
 - Asymmetric-key Encipherment
 - Hashing

Symmetric-key Encipherment

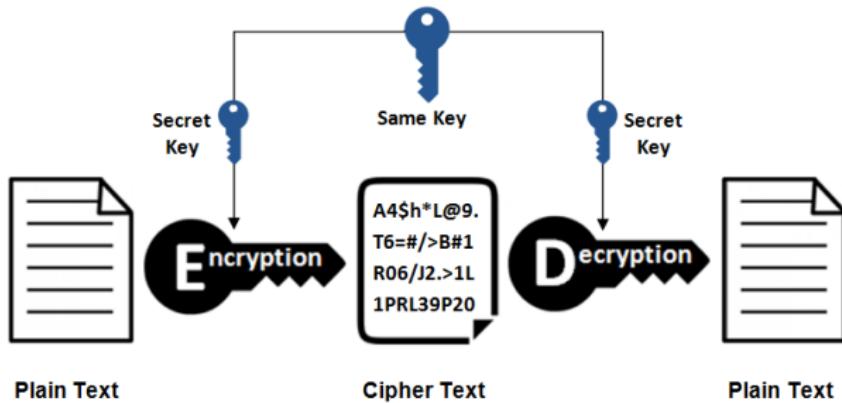


Figure 27: Symmetric-key Encipherment.

Asymmetric-key Encipherment

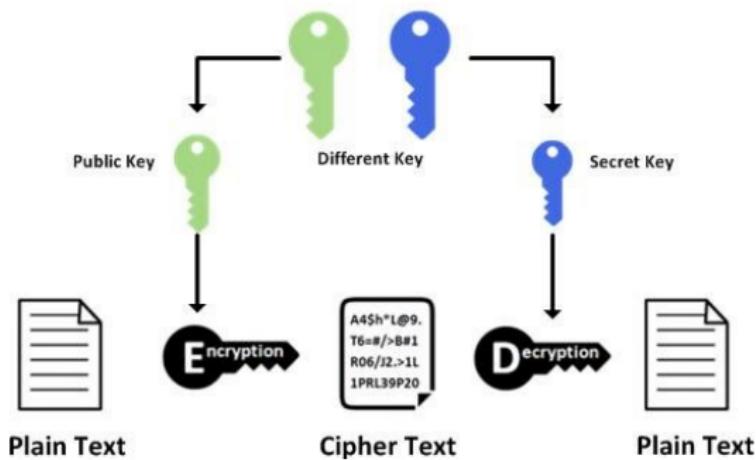


Figure 28: Asymmetric-key Encipherment.

Hashing

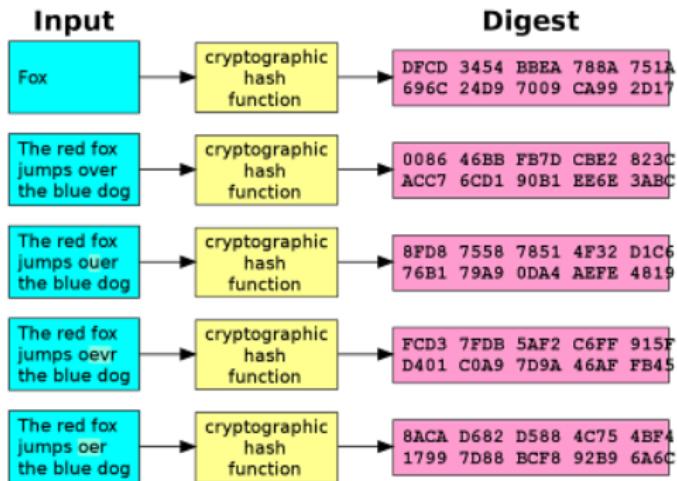


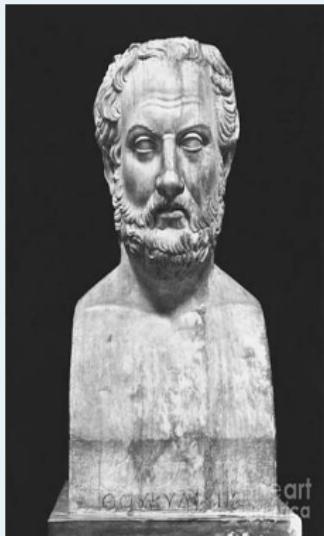
Figure 29: Hashing.

Steganography

- **Steganography:** "*Covered writing*"
- **Cryptography:** "*Secret writing*"
- For example (sent by a German spy during World War I),
Apparently neutral's protest is thoroughly discounted and ignored. Isman hard hit.
Blockade issue affects pretext for embargo on byproducts, ejecting suets and
vegetable oils.
Pershing sails from NY June 1.

Ancient Steganography

(content courtesy: Nasir Memon)



Herodotus (485 – 525 BC) is the first Greek historian. His great work, *The Histories*, is the story of the war between the huge Persian empire and the much smaller Greek citystates.

Herodotus recounts the story of Histiaeus, who wanted to encourage Aristagoras of Miletus to revolt against the Persian king. In order to securely convey his plan, Histiaeus shaved the head of his messenger, wrote the message on his scalp, and then waited for the hair to regrow. The messenger, apparently carrying nothing contentious, could travel freely. Arriving at his destination, he shaved his head and pointed it at the recipient.

Ancient Steganography

(content courtesy: Nasir Memon)

- **Pliny the Elder (AD 23 - 79)** explained how the milk of the thithymallus plant dried to transparency when applied to paper but darkened to brown when subsequently heated, thus recording one of the earliest recipes for invisible ink.



The **Ancient Chinese** wrote notes on small pieces of silk that they then wadded into little balls and coated in wax, to be swallowed by a messenger and retrieved at the messenger's gastrointestinal convenience.

Renaissance Steganography



Giovanni Battista Porta (1535-1615) described how to conceal a message within a hardboiled egg by writing on the shell with a special ink made with an ounce of alum and a pint of vinegar. The solution penetrates the porous shell, leaving no visible trace, but the message is stained on the surface of the hardened egg albumen, so it can be read when the shell is removed..

Modern Steganography: The Prisoners' Problem

- Simmons 1983: Prisoners problem
- Done in the context of USA- USSR non-proliferation treaty compliance checking.
 - Alice and Bob are prisoners, Wendy is a warden. Alice and Bob are allowed to exchange messages, say images, but Wendy checks all messages.
 - Alice and Bob try to hide information in their messages so that Wendy cannot detect it.
 - Wendy cannot arbitrarily suppress all messages; the prisoners' human rights cannot be violated without some proof of illegal activity.

Modern Steganography: The Prisoners' Problem..cont..

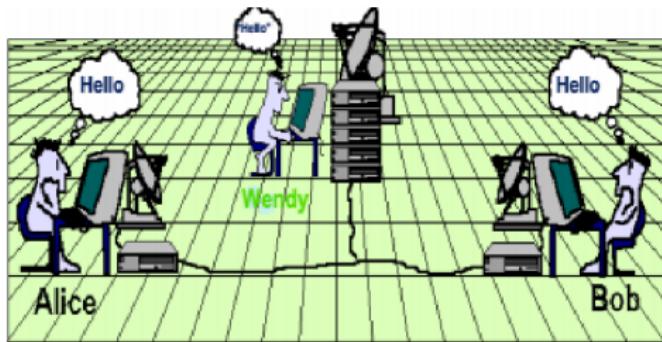


Figure 30: Prisoners' Problem.

Modern Steganography: simplified framework

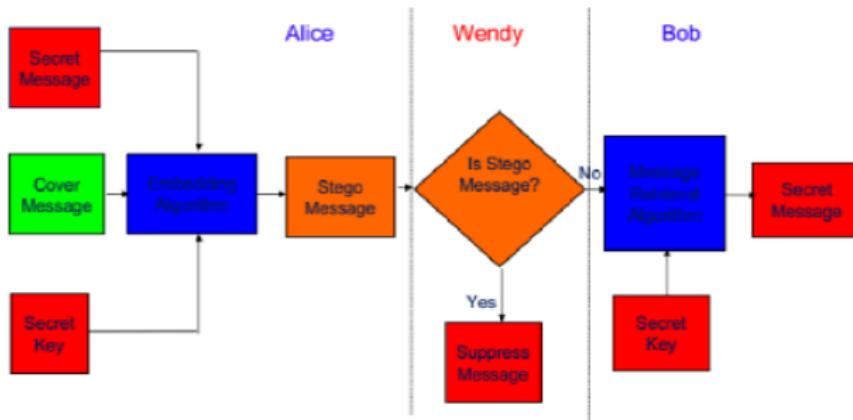


Figure 31: Prisoners' Problem.

Steganography..

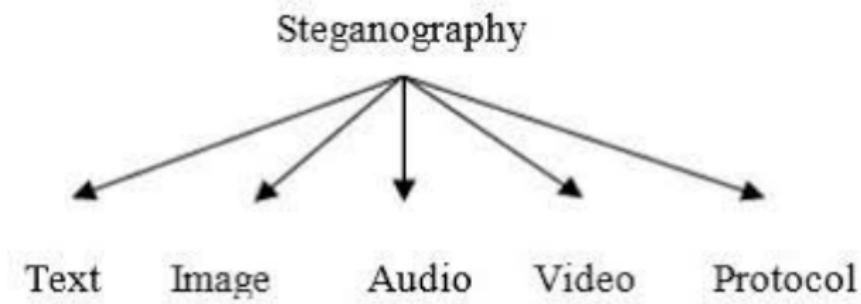


Figure 32: Different types of Steganography: Text, Image, Audio, Video, Protocol/Network.

Bibliography: Books and Resources

- Cryptography and Network Security: Principles and Practice by William Stallings
- Cryptography and Network Security by Behrouz A Forouzan and Debdeep Mukhopadhyay
- Principles of Information Security by Michael E. Whitman and Herbert J. Mattord.
- Cisco platform, and Internet.
- Published research papers, study materials from researchers of security domain.

Unit IV

Syllabus

Intruders, Intrusion Detection and Preventing techniques, Program Security- Threats against programs, Secure programs, Viruses and other malicious code; Introduction to Operating System Security: User Authentication mechanisms, Memory and Address protection, File system protection.

Intrusion Detection & Prevention System(IDPS)

Intrusion: An intrusion occurs when an attacker attempts to gain entry into or disrupt the normal operations of an information system, almost always with the intent to do harm.

Intrusion Prevention System

- Intrusion prevention consists of activities that **deter an intrusion**.
- Some important intrusion prevention activities are **writing and implementing good enterprise information security policy, planning and executing effective information security programs, installing and testing technology-based information security countermeasures** (such as firewalls and intrusion detection systems), and conducting and measuring the effectiveness of employee training and awareness activities.

Intrusion Detection System(IDS)

- Intrusion detection consists of procedures and systems that **identify system intrusions**.
- An IDS works like a **burglar alarm in that it detects a violation** (some system activity analogous to an opened or broken window) and activates an alarm.
- This alarm can be audible and/or visual (producing noise and lights, respectively), or it can be silent (an e-mail message or pager alert).
- With almost all IDSs, system administrators can choose the configuration of the various alerts and the alarm levels associated with each type of alert.

IDPS Terminology

In order to understand IDPS operational behavior, you must first become familiar with some IDPS terminology:

- **Alert or alarm:** An **indication that a system has just been attacked** or is under attack. IDPS alerts and alarms take the form of audible signals, e-mail messages, pager notifications, or pop-up windows.
- **Evasion:** The process by which attackers **change the format and/or timing of their activities** to avoid being detected by the IDPS.
- **False attack stimulus:** An event **that triggers an alarm when no actual attack is in progress**. Scenarios that test the configuration of IDPSs may use false attack stimuli to determine if the IDPSs can distinguish between these stimuli and real attacks.
- **False negative:** The **failure of an IDPS to react to an actual attack event**. This is the most grievous failure, since the purpose of an IDPS is to detect and respond to attacks.
- **False positive:** An alert or alarm that **occurs in the absence of an actual attack**. A false positive can sometimes be produced when an IDPS mistakes normal system activity for an attack. False positives tend to make users insensitive to alarms and thus reduce their reactivity to actual intrusion events.
- **Noise:** Alarm events that are accurate and noteworthy but that **do not pose significant threats** to information security. Unsuccessful attacks are the most common source of IDPS noise, and some of these may in fact be triggered by scanning and enumeration tools deployed by network users without intent to do harm.

- **Site policy:** The rules and configuration guidelines governing the implementation and operation of IDPSs within the organization.
- **Site policy awareness:** An IDPS's ability to dynamically modify its configuration in response to environmental activity. A so-called smart IDPS can adapt its reactions in response to administrator guidance over time and circumstances of the current local environment.
- **True attack stimulus:** An event that triggers alarms and causes an IDPS to react as if a real attack is in progress. The event may be an actual attack, in which an attacker is at work on a system compromise attempt, or it may be a drill, in which security personnel are using hacker tools to conduct tests of a network segment.
- **Tuning:** The process of adjusting an IDPS to maximize its efficiency in detecting true positives, while minimizing both false positives and false negatives.
- **Confidence value:** The measure of an IDPS's ability to correctly detect and identify certain types of attacks. The confidence value an organization places in the IDPS is based on experience and past performance measurements.
- **Alarm filtering:** The process of classifying IDPS alerts so that they can be more effectively managed. An IDPS administrator can set up alarm filtering by running the system for a while to track what types of false positives it generates and then adjusting the alarm classifications.
- **Alarm clustering and compaction:** A process of grouping almost identical alarms that happen at close to the same time into a single higher-level alarm. This consolidation reduces the number of alarms generated, thereby reducing administrative overhead, and also identifies a relationship among multiple alarms.

Why use an IDPS?

According to the NIST documentation on industry best practices, there are several compelling reasons to acquire and use an IDPS:

1. To prevent problem behaviors by increasing the perceived risk of discovery and punishment for those who would attack or otherwise abuse the system.
2. To **detect attacks and other security violations** that are not prevented by other security measures.
3. To **detect and deal with the preambles to attacks** (commonly experienced as network probes and other “doorknob rattling” activities).
4. To **document the existing threat** to an organization.
5. To **act as quality control for security design and administration**, especially in large and complex enterprises.
6. To provide useful information about intrusions that do take place, allowing improved diagnosis, recovery, and correction of causative factors

Types of IDPS

- There are two types of IDPS:
 - **Network-based IDPS(NIDPS)**
 - **Host-based IDPS(HIDPS)**
- A network-based IDPS is focused on **protecting network information assets**. Two specialized subtypes of network-based IDPS are the **wireless IDPS** and the **network behavior analysis (NBA) IDPS**. The wireless IDPS **focuses on wireless networks**, as the name indicates, while the NBA IDPS **examines traffic flow on a network** in an attempt to recognize abnormal patterns like DDoS, malware, and policy violations.
- A host-based IDPS **protects the server or host's information assets**. It monitors both network connection activity and current information states on host servers. The application-based model works on one or more host systems that support a single application and defends that specific application from special forms of attack.

NIDPS

- **Network-Based IDPS** A **network-based IDPS (NIDPS)** resides on a computer or appliance connected to a segment of an organization's network and **monitors network traffic on that network segment, looking for indications of ongoing or successful attacks.**
- When the NIDPS identifies activity that it is programmed to recognize as an attack, it responds by **sending notifications to administrators.**
- When examining incoming packets, an NIDPS looks for patterns within network traffic such as large collections of related items of a certain type—which could indicate that a **denial-of-service attack** is underway—or the exchange of a series of related packets in a certain pattern—which could indicate that a **port scan** is in progress.
- An **NIDPS can detect many more types of attacks than a host-based IDPS**, but it requires a much more complex configuration and maintenance program.
- A NIDPS is installed at a specific place in the network (such as on the inside of an edge router) from where it is possible to monitor the traffic going into and out of a particular network segment.

HIDPS

- **Host-Based IDPS** While a network-based IDPS resides on a network segment and monitors activities across that segment, a host-based IDPS (HIDPS) resides on a particular computer or server, known as the host, and **monitors activity only on that system**.
- HIDPSs are also known as **system integrity verifiers** because they benchmark and monitor the status of key system files and **detect when an intruder creates, modifies, or deletes monitored files**.
- An HIDPS has an advantage over an NIDPS in that it can **access encrypted information traveling over the network** and use it to make decisions about potential or actual attacks.
- Also, since the **HIDPS works on only one computer system**, all the traffic it examines traverses that system.
- An HIDPS is also capable of **monitoring system configuration databases, such as windows registries**, in addition to stored configuration files like .ini, .cfg, and .dat files. Most HIDPSs work on the principle of configuration or change management, which means that they record the sizes, locations, and other attributes of system files.

Terminologies of Malicious Program

Name	Description
Virus	Attaches itself to a program and propagates copies of itself to other programs
Worm	Program that propagates copies of itself to other computers
Logic bomb	Triggers action when condition occurs
Trojan horse	Program that contains unexpected additional functionality
Backdoor (trapdoor)	Program modification that allows unauthorized access to functionality
Exploits	Code specific to a single vulnerability or set of vulnerabilities
Downloaders	Program that installs other items on a machine that is under attack. Usually, a downloader is sent in an e-mail.
Auto-rooter	Malicious hacker tools used to break into new machines remotely
Kit (virus generator)	Set of tools for generating new viruses automatically
Spammer programs	Used to send large volumes of unwanted e-mail
Flooders	Used to attack networked computer systems with a large volume of traffic to carry out a denial of service (DoS) attack
Keyloggers	Captures keystrokes on a compromised system
Rootkit	Set of hacker tools used after attacker has broken into a computer system and gained root-level access
Zombie	Program activated on an infected machine that is activated to launch attacks on other machines

Viruses

- A virus is a piece of software that can **infect other programs by modifying them**; the modification includes a copy of the virus program, which can then go on to infect other programs.
- A virus can do anything that other programs do. The only difference is that it **attaches itself to another program and executes secretly when the host program is run**. Once a virus is executing, it can perform any function, such as **erasing files and programs**.

Contd...

During its lifetime, a typical virus goes through the following **four phases**:

1. Dormant phase: The **virus is idle**. The virus will eventually be **activated by some event**, such as a date, the presence of another program or file, or the capacity of the disk exceeding some limit. Not all viruses have this stage.

2. Propagation phase: The virus **places an identical copy of itself into other programs or into certain system areas on the disk**. Each infected program will now contain a clone of the virus, which will itself enter a propagation phase.

3. Triggering phase: The **virus is activated to perform the function for which it was intended**. As with the dormant phase, the triggering phase can be caused by a variety of system events, including a count of the number of times that this copy of the virus has made copies of itself.

4. Execution phase: The function is performed. The **function may be harmless, such as a message on the screen, or damaging**, such as the destruction of programs and data files.

Most viruses carry out their work in a manner that is specific to a particular operating system and, in some cases, specific to a particular hardware platform. Thus, they are designed to take advantage of the details and weaknesses of particular systems.

Types of Viruses

Parasitic virus: The traditional and still most common form of virus. A **parasitic virus attaches itself to executable files and replicates**, when the infected program is executed, by finding other executable files to infect.

Memory-resident virus: **Lodges in main memory** as part of a resident system program. From that point on, the **virus infects every program that executes**.

Boot sector virus: **Infects a master boot record or boot record** and spreads when a system is booted from the disk containing the virus.

Stealth virus: A form of virus **explicitly designed to hide itself** from detection by antivirus software.

Polymorphic virus: A virus that **mutates with every infection**, making detection by the "signature" of the virus impossible.

Metamorphic virus: As with a polymorphic virus, a metamorphic virus mutates with every infection. The difference is that a **metamorphic virus rewrites itself completely at each iteration**, increasing the difficulty of detection. Metamorphic viruses may **change their behavior as well as their appearance**.

Firewall

- A firewall forms a barrier through which the traffic going in each direction must pass. A firewall security policy dictates which **traffic is authorized to pass** in each direction.
- A firewall may be designed to **operate as a filter at the level of IP packets**, or may operate at a higher protocol layer.

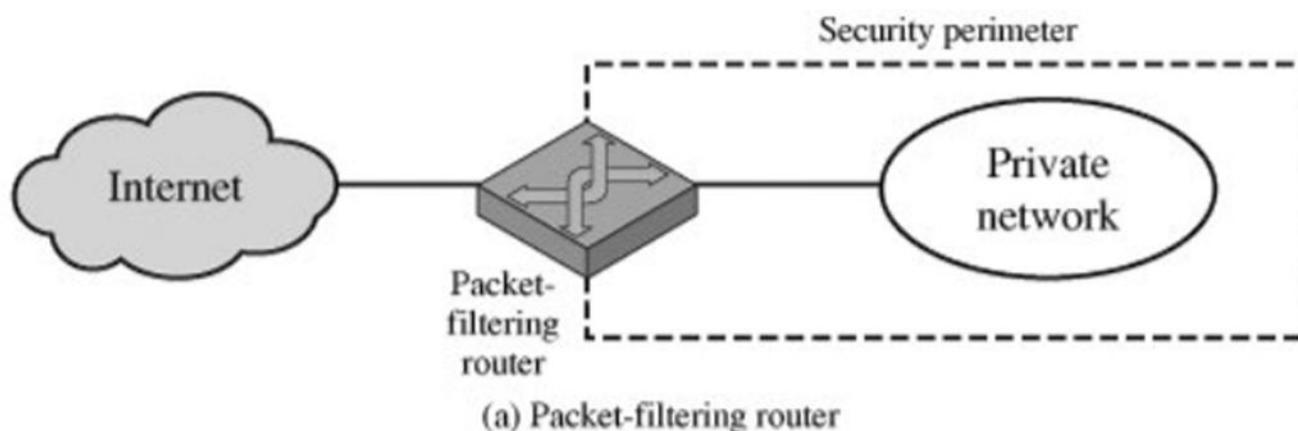
Types of Firewall

There are **three types** of Firewall:

- a. Packet Filtering Firewall
- b. Application Level Gateway
- c. Circuit Level Gateway

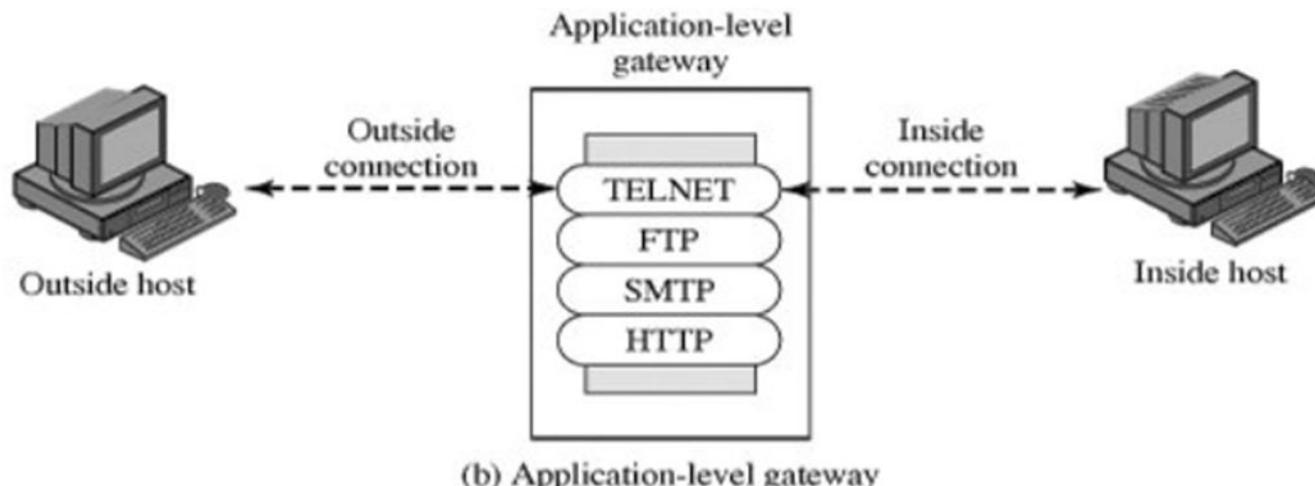
Packet-Filtering Router

- A packet-filtering router applies a **set of rules to each incoming and outgoing IP packet** and then forwards or discards the packet.
- The router is typically configured to filter packets going in both directions (from and to the internal network). Filtering rules are based on information contained in a network packet.



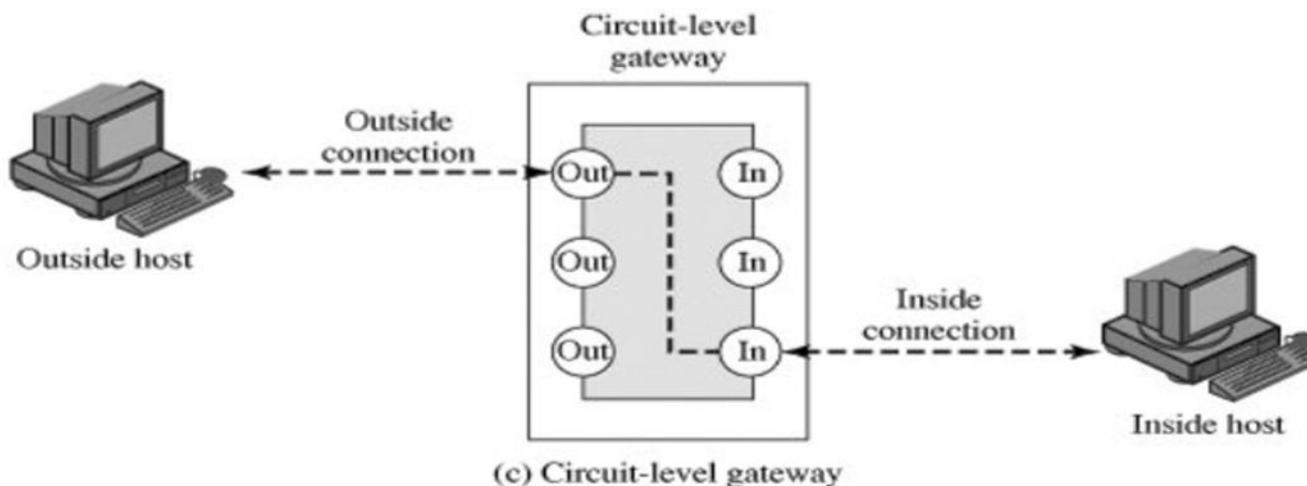
Application-Level Gateway

- An application-level gateway, also called a proxy server, acts as a relay of application-level traffic. The user contacts the gateway using a TCP/IP application, such as Telnet or FTP, and the gateway asks the user for the name of the remote host to be accessed.
- When the user responds and provides a valid user ID and authentication information, the gateway contacts the application on the remote host and relays TCP segments containing the application data between the two endpoints. If the gateway does not implement the proxy code for a specific application, the service is not supported and cannot be forwarded across the firewall.
- Application-level gateways tend to be more secure than packet filters. Rather than trying to deal with the numerous possible combinations that are to be allowed and forbidden at the TCP and IP level, the application-level gateway need only scrutinize a few allowable applications. In addition, it is easy to log and audit all incoming traffic at the application level.
- A prime disadvantage of this type of gateway is the additional processing overhead on each connection.



Circuit-Level Gateway

- A circuit-level gateway can be a stand-alone system or it can be a specialized function performed by an application-level gateway for certain applications.
- A **circuit-level gateway does not permit an end-to-end TCP connection; rather, the gateway sets up two TCP connections, one between itself and a TCP user on an inner host and one between itself and a TCP user on an outside host.**
- Once the two connections are established, the gateway typically relays TCP segments from one connection to the other without examining the contents. The security function consists of determining which connections will be allowed.



Memory and Address Protection

- Memory protection includes **protection for the memory** that the OS itself uses as well as the memory of user processes.
- Major challenge in multi-programming system is to prevent one program from affecting the data and programs in the memory space of other users.

The various methods for memory and address protection are:

i. Fence:

- A fence or fence address is simplest form of memory protection which can be used only for single user operating system.
- **A fence is a particular address that users and their processes cannot cross.** Only the OS can operate on one side of the fence and users are restricted to the other side.
- A fence could be static, in which case there is a fixed fence address. Alternatively, a dynamic fence can be used, which can be implemented using a fence register to specify the current fence address.

ii. Base and Bounds registers:

- This type of protection can be used in multi- user environment where one user's program needs to be protected from the other.
- **Each user has a base register which is the lower address and a Bound register which is the upper address limit.**
- The base and bounds register approach implicitly assumes that the user or process space is contiguous in memory. The OS must determine what protection to apply to a specific memory location.
- In some cases it might be sufficient to apply the same protection to all of a user's memory.
- The disadvantage is that the registers confine access to consecutive range of addresses.

iii. Tagging:

- This specifies the protection for each individual address. In this method of protection every word of machine memory has one or more extra bits to identify the access rights to that word.
- Only privileged instructions can set these access bits. While this is as fine-grained protection as possible, it introduces significant overhead.
- The overhead can be reduced by tagging sections of the address space instead of each individual address. Another drawback to tagging is compatibility, since tagging schemes are not in common use.

iv. Segmentation:

- This method divides the memory into logical units such as individual procedures or the data in one array.
- Once they are divided, appropriate access control can be enforced on each segment.
- A benefit of segmentation is that any segment can be placed in any memory location provided the location is large enough to hold it. The OS must keep track of the locations of all segments, which is accomplished using \langle segment,offset \rangle pairs, where the named segment specifies the segment, and the offset is the starting address of the specified segment.
- With segmentation, all address references must go through the OS, so the OS can, in this respect, achieve complete mediation. Depending on the access control applied to particular segments, users can share access to some segments or users can be restricted to specific segments.

v. Paging:

- Paging discards the disadvantage of segmentation. In paging all segments are of a fixed size called as pages and the memory divided is known as page frames.
- In paging a particular page can be accessed using a pair of the form <page, offset=""> where page is the page number and offset is location within a page.
- The advantages of paging over segmentation include no fragmentation, improved efficiency, and the fact that there are no variable sizes to worry about.
- The disadvantages are that there is, in general, no logical unity to pages, which makes it more difficult to determine the proper access control to apply to a given page.

Scanning and Analysis Tools

- **Port Scanners:** Port scanning utilities, or **port scanners**, are tools used by both attackers and defenders to identify (or fingerprint) the computers that are active on a network, as well as the ports and services active on those computers, the functions and roles the machines are fulfilling, and other useful information. A port is a network channel or connection point in a data communications system. There are 65,536 port numbers in use for TCP and another 65,536 port numbers for UDP. Services using the TCP/IP protocol can run on any port; however, services with reserved ports generally run on ports 1–1023. Port 0 is not used. Ports greater than 1023 are typically referred to as ephemeral ports and may be randomly allocated to server and client processes. An open port can be used by an attacker to send commands to a computer, potentially gain access to a server, and possibly exert control over a networking device. The general rule of thumb is to remove from service or secure any port not absolutely necessary to conducting business. For example, if a business doesn't host Web services, there is no need for port 80 to be available on its servers.
- **Operating System Detection Tools:** Detecting a target computer's operating system is very valuable to an attacker, because once the OS is known, all of the vulnerabilities to which it is susceptible can easily be determined. There are many tools that use networking protocols to determine a remote computer's OS. One specific tool worth mentioning is **XProbe**, which uses ICMP to determine the remote OS.

- **Active vulnerability scanners** scan networks for highly detailed information. An active scanner is one that initiates traffic on the network in order to determine security holes. As a class, this type of scanner identifies exposed usernames and groups, shows open network shares, and exposes configuration problems and other vulnerabilities in servers. An example of a vulnerability scanner is GFI LANguard Network Security Scanner (NSS), which is available as freeware for non-commercial use. Another example of a vulnerability scanner is Nessus, which is a professional freeware utility that uses IP packets to identify the hosts available on the network, the services (ports) they are offering, the operating system and OS version they are running, the type of packet filters and firewalls in use, and dozens of other characteristics of the network.
- **Packet Sniffers:** A **packet sniffer** (sometimes called a network protocol analyzer) is a network tool that collects copies of packets from the network and analyzes them. It can provide a network administrator with valuable information for diagnosing and resolving networking issues. In the wrong hands, however, a sniffer can be used to eavesdrop on network traffic. An excellent free, client-based network protocol analyzer is **Wireshark**, formerly known as Ethereal. Wireshark allows the administrator to examine data from both live network traffic and captured traffic. Wireshark has several features, including a language filter and TCP session reconstruction utility.

Unit III

Syllabus

Security in networks: Threats and Vulnerabilities, IP Security – Overview, Architecture etc., Email Security – PGP, S/MIME; Web Security – Requirements, Security Protocols like SSL, TLS, SET; Firewalls.

- **IP Security - IPSec - Covered**
- **E-mail Security(PGP,S/MIME)**
- **Web Security(SET, SSL,TLS)**

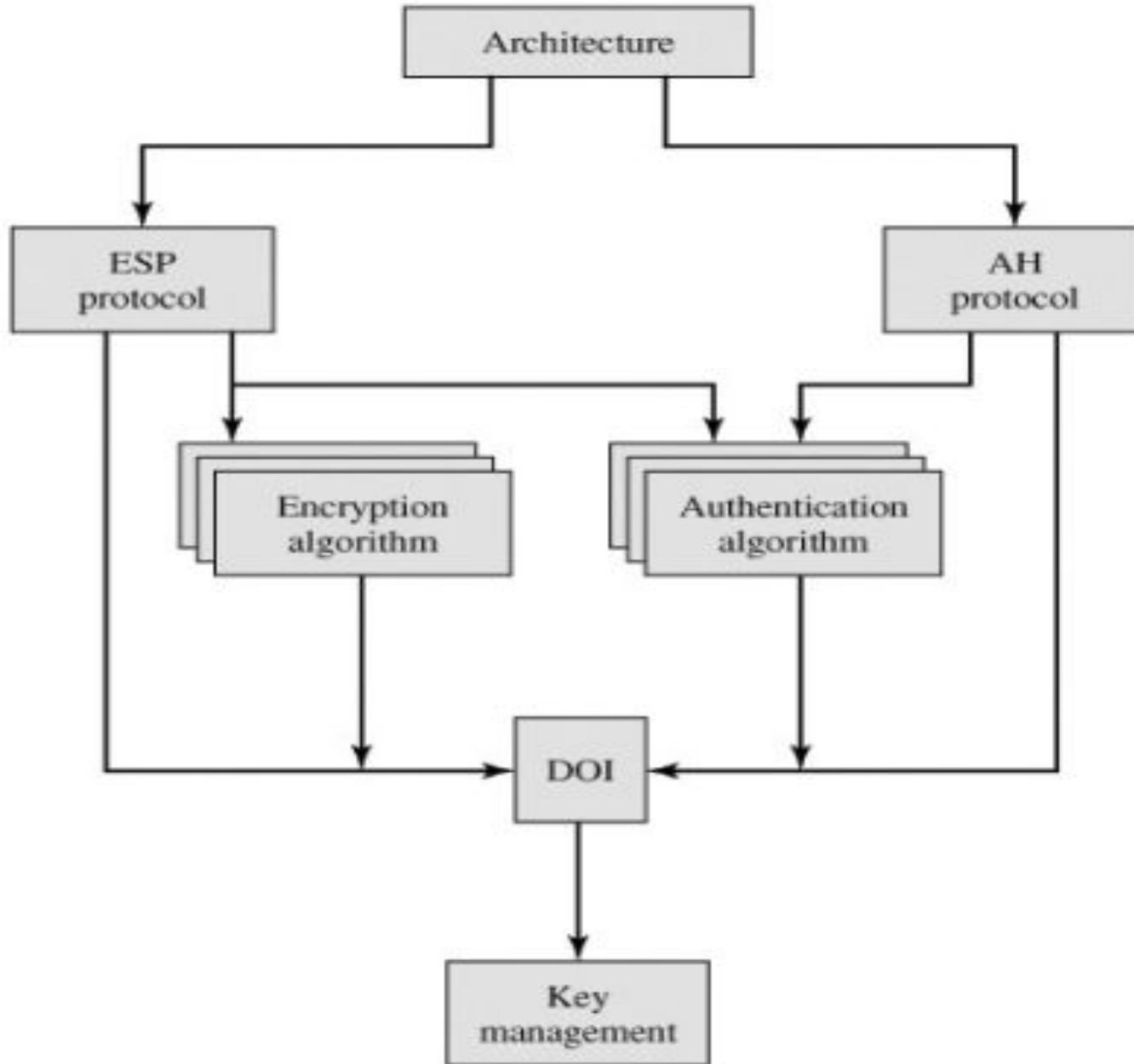
IP Security

- Internet Protocol Security (IPSec) is an open-source protocol framework for security development within the **TCP/IP family of protocol** standards.
- It is used to secure communications across **IP-based networks such as LANs, WANs, and the Internet**.
- The protocol is designed to protect **data integrity, user confidentiality, and authenticity** at the **IP packet level**.

Contd...

- IP-level security encompasses three functional areas: **authentication, confidentiality, and key management.**
- The authentication mechanism assures that a received packet was, in fact, transmitted by the party identified as the source in the packet header. In addition, this mechanism assures that the packet has not been altered in transit.
- The confidentiality facility enables communicating nodes to encrypt messages to prevent eavesdropping by third parties.
- The key management facility is concerned with the secure exchange of keys.

IPSec Documents



IPSec Services

The services are:

- Access control
- Connectionless integrity
- Data origin authentication
- Rejection of replayed packets (a form of partial sequence integrity) - Duplicate
- Confidentiality (encryption)
- Limited traffic flow confidentiality

Protocols

- The **application header (AH) protocol** provides system-to-system **authentication** and data integrity verification, but does not provide secrecy for the content of a network communication.
- The **encapsulating security payload (ESP) protocol** provides **secrecy** for the contents of network communications as well as system-to-system **authentication** and data integrity verification.

	AH	ESP (encryption only)	ESP (encryption plus authentication)
Access control	✓	✓	✓
Connectionless integrity	✓		✓
Data origin authentication	✓		✓
Rejection of replayed packets	✓	✓	✓
Confidentiality		✓	✓
Limited traffic flow confidentiality		✓	✓

Modes

- IPSec operates in two modes: **transport and tunnel**.
- In **transport mode** only the **IP data are encrypted, not the IP headers**. This allows intermediate nodes to read the source and destination addresses.
- In **tunnel mode** the **entire IP packet(Data as well as Header)** is encrypted and is then placed into the content portion of another IP packet. This requires other systems at the beginning and end of the tunnel to act as proxies and to send and receive the encrypted packets.

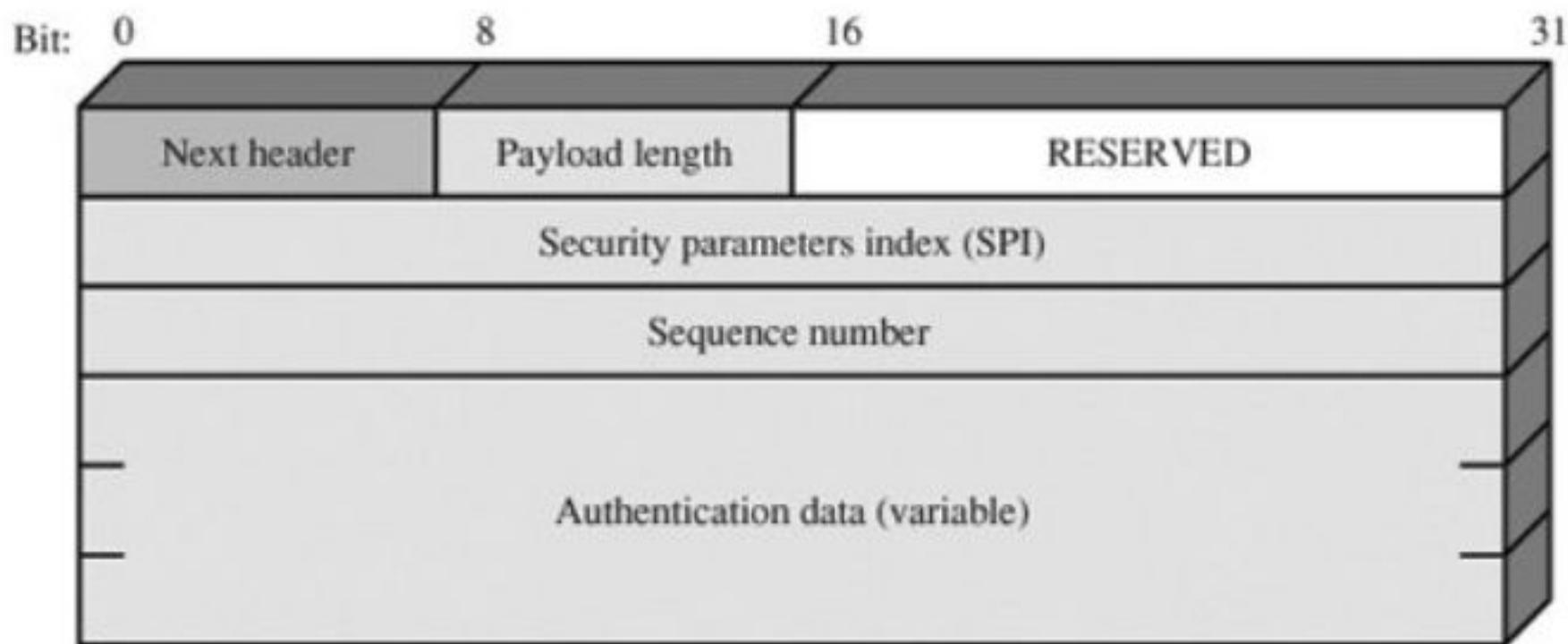
Security Association

Agreement between the sender and receiver

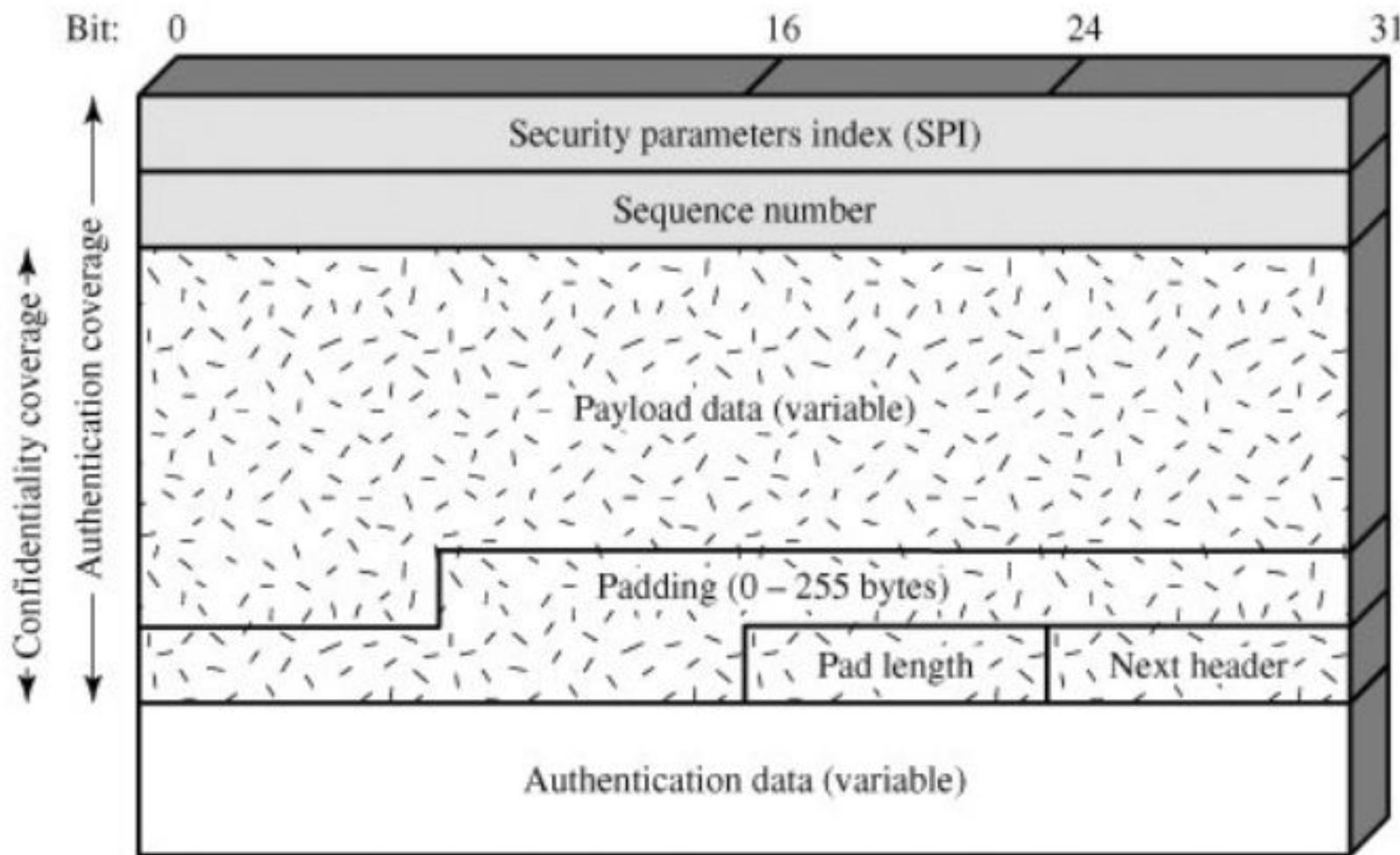
IP Sec - VPN - Private -
C and S - (Parameters) SA Database(SAD)

establish- Connect
transfer
terminate

AH Format



ESP Format



- Security Parameters Index (32 bits): Identifies a security association.
- Sequence Number (32 bits): A monotonically increasing counter value; this provides an anti-replay function, as discussed for AH.
- Payload Data (variable): This is a transport-level segment (transport mode) or IP packet (tunnel mode) that is protected by encryption.

- Padding (0-255 bytes): The purpose of this field is discussed later. for block
- Pad Length (8 bits): Indicates the number of pad bytes immediately preceding this field.
- Next Header (8 bits): Identifies the type of data contained in the payload data field by identifying the first header in that payload (for example, an extension header in IPv6, or an upper-layer protocol such as TCP).
- Authentication Data (variable): A variable length field (must be an integral number of 32-bit words) that contains the Integrity Check Value computed over the ESP packet minus the Authentication Data field.

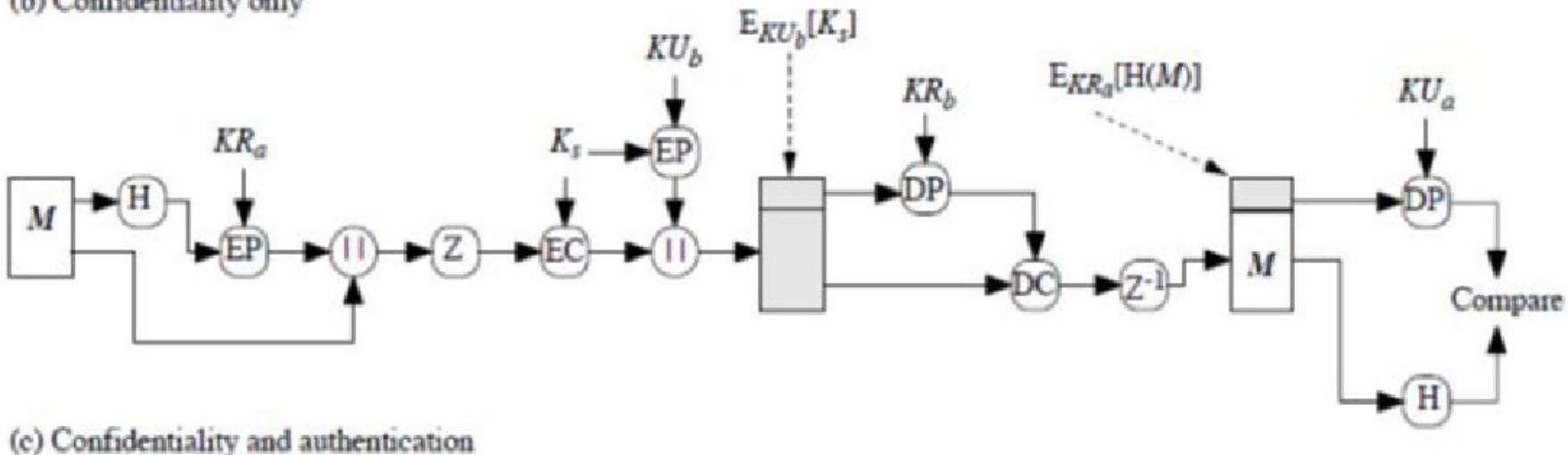
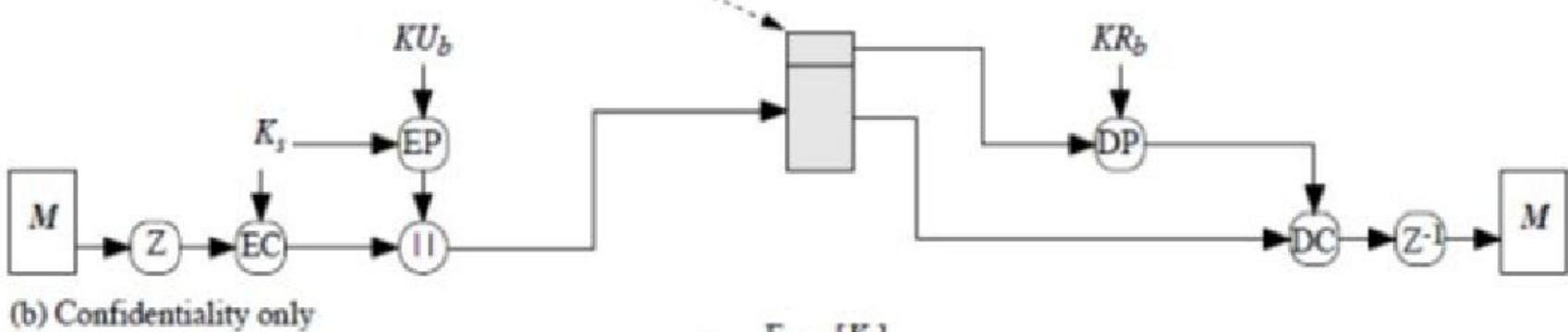
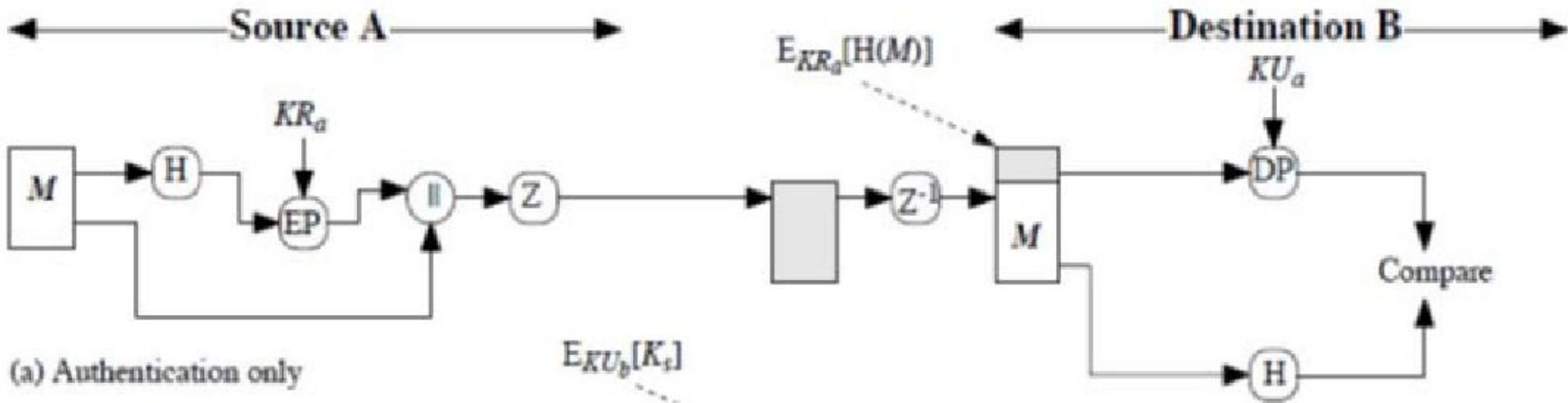
Key Management

- The key management portion of IPSec involves the **determination and distribution of secret keys**.
- A typical requirement is four keys for communication between two applications: transmit and receive pairs for both AH and ESP.
- The IPSec Architecture document mandates support for two types of key management:
 1. **Manual:** A system administrator manually configures each system with its own keys and with the keys of other communicating systems. This is practical for small, relatively static environments.
 2. **Automated:** An automated system enables the on-demand creation of keys for SAs and facilitates the use of keys in a large distributed system with an evolving configuration. The default automated key management protocol for IPSec is referred to as ISAKMP/Oakley.

E-mail Security

- E-mail security – PGP and S/MIME
- PGP is an open-source freely available software package for e-mail security.
- It provides **authentication** through the use of digital signature; **confidentiality** through the use of symmetric block encryption; **compression** using the ZIP algorithm; **e-mail compatibility** using the radix-64 encoding scheme; and **segmentation and reassembly** to accommodate long e-mails.

Function	Algorithms	Used Description
Digital signature	DSS/SHA or RSA/SHA	A hash code of a message is created using SHA-1. This message digest is encrypted using DSS or RSA with the sender's private key and included with the message.
Message encryption	CAST or IDEA or Three-key Triple DES with Diffie-Hellman or RSA	A message is encrypted using CAST-128 or IDEA or 3DES with a one-time session key generated by the sender. The session key is encrypted using Diffie-Hellman or RSA with the recipient's public key and included with the message.
Compression	ZIP	A message may be compressed, for storage or transmission, using ZIP.
Email compatibility	Radix 64 conversion	To provide transparency for email applications, an encrypted message may be converted to an ASCII string using radix 64 conversion.
Segmentation		To accommodate maximum message size limitations, PGP performs segmentation and reassembly.



MIME

- **Multipurpose Internet Mail Extensions**
- MIME is intended to address some of the problems and **limitations of the use of SMTP** (Simple Mail Transfer Protocol) or some other mail transfer protocol.

Limitation of SMTP

- SMTP cannot transmit executable files or other binary objects.
- SMTP cannot transmit text data that includes national language characters.
- SMTP servers may reject mail message over a certain size.
- SMTP not support the transfer of video, images and audio files.

The MIME specification includes the following elements:

1. Five new message header fields are defined, which may be included in an RFC 822 header. These fields provide information about the body of the message.
2. A number of content formats are defined, thus standardizing representations that support multimedia electronic mail.
3. Transfer encodings are defined that enable the conversion of any content format into a form that is protected from alteration by the mail system.

MIME Header

The five header fields defined in MIME are as follows:

- 1. MIME-Version:** Must have the parameter value 1.0. This field indicates that the message conforms to RFCs 2045 and 2046.
- 2. Content-Type:** Describes the **data contained in the body** with sufficient detail that the receiving user agent can pick an appropriate agent or mechanism to represent the data to the user or otherwise deal with the data in an appropriate manner. text, html, audio, video, image, Table
- 3. Content-Transfer-Encoding:** Indicates the type of **transformation that has been used to represent the body of the message** in a way that is acceptable for mail transport.
- 4. Content-ID:** Used to identify MIME entities uniquely in multiple contexts.
- 5. Content-Description:** A text **description of the object with the body**; this is useful when the object is not readable (e.g., audio data).

S/MIME

- S/MIME (Secure/Multipurpose Internet Mail Extension) is a **security enhancement to the MIME** Internet email format standard, based on technology from RSA Data Security.
- Although both PGP and S/MIME are on an IETF standards track, it appears likely that **S/MIME** will emerge as the industry standard for **commercial and organizational** use, while **PGP** will remain the choice for **personal e-mail security** for many users.
- Confidentiality, Integrity, Authentication, Non repudiation

S/MIME Functionality

- In terms of general functionality, S/MIME is very similar to PGP. Both offer the ability to **sign and/or encrypt messages**.
- S/MIME provides the following functions:
 - 1. Enveloped data:** This consists of **encrypted content** of any type and encrypted-content encryption keys for one or more recipients.
 - 2. Signed data:** A **digital signature** is formed by taking the message digest of the content to be signed and then encrypting that with the private key of the signer. The content plus signature are then encoded using base64 encoding. A signed data message can **only be viewed by a recipient with S/MIME capability**.
 - 3. Clear-signed data:** As with signed data, a **digital signature** of the content is formed. However, in this case, only the digital signature is encoded using base64. As a result, recipients **without S/MIME capability can view the message content**, although they cannot verify the signature.
 - 4. Signed and enveloped data:** **Signed-only and encrypted-only** entities may be nested, so that encrypted data may be signed and signed data or clear-signed data may be encrypted.

S/MIME uses the following terminology, taken from RFC 2119 to specify the requirement level:

- 1. MUST:** The definition is an absolute requirement of the specification. An implementation must include this feature or function to be in conformance with the specification.
- 2. SHOULD:** There may exist valid reasons in particular circumstances to ignore this feature or function, but it is recommended that an implementation include the feature or function.

- S/MIME incorporates three public-key algorithms. **The Digital Signature Standard (DSS)** is the preferred algorithm for **digital signature**. (Sender's private message sign)
- S/MIME lists **Diffie-Hellman** Key Exchange as the preferred algorithm for encrypting session keys; in fact, S/MIME uses a variant of Diffie-Hellman that does provide encryption/decryption, known as ElGamal. As an alternative, RSA, can be used for both signatures and session key encryption. For the hash function used to create the digital signature, the specification requires the 160-bit SHA-1 but recommends receiver support for the 128-bit MD5 for backward compatibility with older versions of S/MIME.
- For message encryption, **three-key triple DES (tripleDES)** is recommended, but compliant implementations must support 40-bit RC2.
- S/MIME uses **public-key certificates** that conform to version 3 of X.509.
- Gmail(Gsuite - paid), Outlook(SMIME) : S/MIME Certificate, CA, enable or disable, Sender and receiver, verify
- sgsits, SMIME certificate, global sign, verisign, validity,

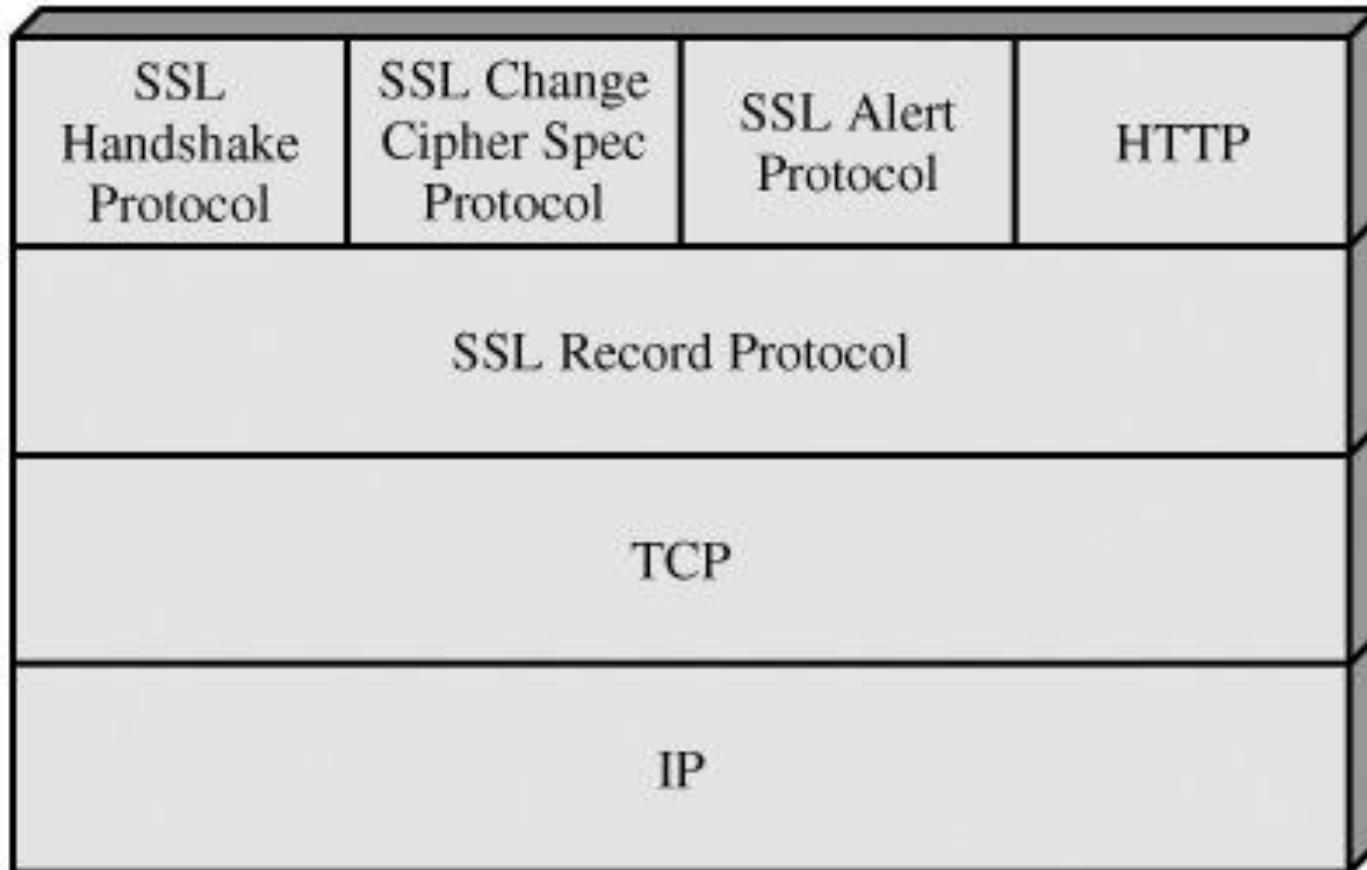
Web Security

- **Secure Socket Layer (SSL)** provide security to the data that is transferred between web browser and server.
- SSL **encrypt** the link between a web server and a browser which ensures that all data passed between them remain private and free from attack.
- SSL is designed to make use of **TCP** to provide a **reliable end-to-end secure service**.

Secure Socket Layer Protocols

- SSL record protocol: Data transfer
- Handshake protocol - create the session, parameters
- Change-cipher spec protocol: encryption
- Alert protocol: warning and error/fatal

SSL Protocol Stack



Two important SSL concepts are the **SSL session** and the **SSL connection**, which are defined in the specification as follows:

Connection: A connection is a **transport** (in the OSI layering model definition) that **provides a suitable type of service**. For SSL, such connections are peer-to-peer relationships. The connections are transient. **Every connection is associated with one session.**

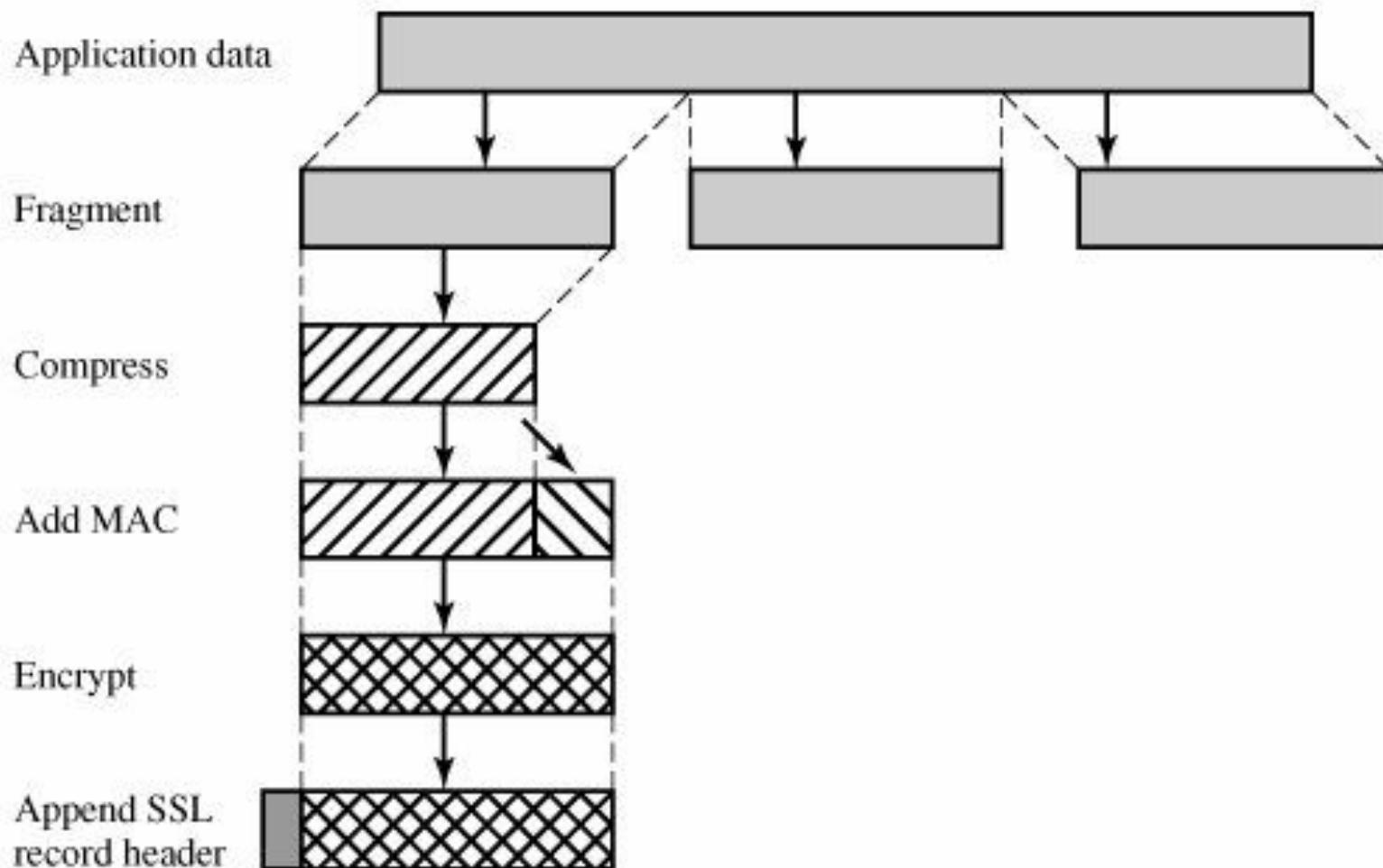
Session: An SSL session is an **association between a client and a server**. Sessions are **created by the Handshake Protocol**. Sessions define a set of cryptographic security parameters, which can be shared among multiple connections. Sessions are used to avoid the expensive negotiation of new security parameters for each connection. A session can have multiple connections.

SSL Record Protocol

The SSL Record Protocol provides two services for SSL connections:

Confidentiality: The Handshake Protocol/Session defines a shared secret key that is used for conventional encryption/DES/AES of SSL payloads.

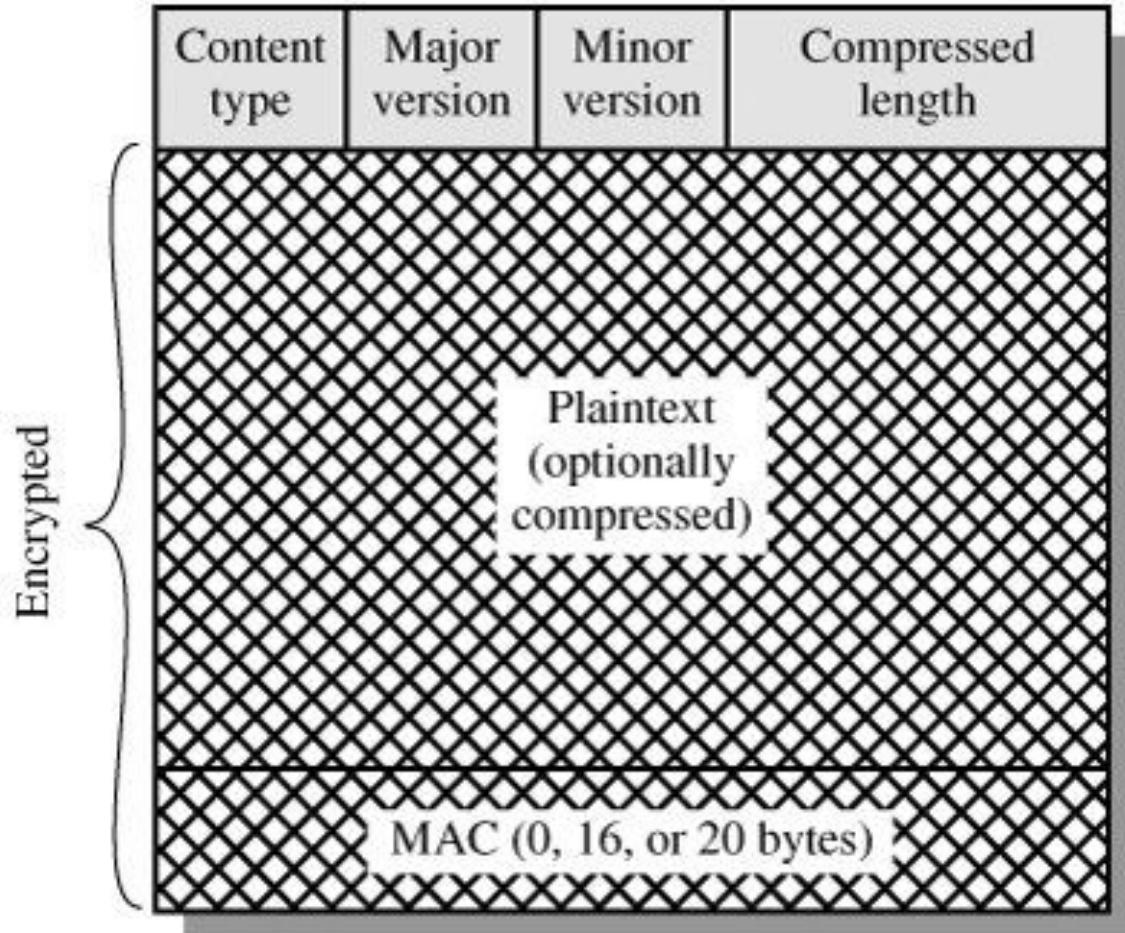
Message Integrity: The Handshake Protocol also defines a shared secret key that is used to form a message authentication code (MAC)/md5/SHA



The Record Protocol takes an application message to be transmitted, **fragments** the data into manageable blocks, optionally **compresses** the data, applies a **MAC**, **encrypts**, adds a **header**, and transmits the resulting unit in a TCP segment. Received data are decrypted, verified(data verification and sender verification), decompressed, and reassembled and then delivered to higher-level users.

The final step of SSL Record Protocol processing is to prepend a header, consisting of the following fields:

- **Content Type (8 bits)**: The higher layer protocol used to process the enclosed fragment.
- **Major Version (8 bits)**: Indicates major version of SSL in use. For SSLv3, the value is 3.
- **Minor Version (8 bits)**: Indicates minor version in use. For SSLv3, the value is 0.
- **Compressed Length (16 bits)**: The length in bytes of the plaintext fragment (or compressed fragment if compression is used). The maximum value is $2^{14} + 2048$.



Change Cipher Spec Protocol

- The Change Cipher Spec Protocol is one of the three SSL-specific protocols that use the SSL Record Protocol, and it is the simplest. This protocol consists of a **single message**, which consists of a **single byte with the value 1**. The sole purpose of this message is to cause the **pending state to be copied into the current state**, which updates the cipher suite/encryption to be used on this connection.
- Which types of encryption algo should be used?

Alert Protocol

- The Alert Protocol is used to convey **SSL-related alerts to the peer entity**. As with other applications that use SSL, **alert messages are compressed and encrypted**, as specified by the current state.
- Each message in this protocol consists of **two bytes**.
- The first byte takes the value **warning(1)** or **fatal(2)** to convey the severity of the message.
- If the level is **fatal**, **SSL immediately terminates the connection**. Other connections on the same session may continue, but no new connections on this session may be established.
- The **second byte contains a code** that indicates the specific alert.

First, we list those alerts that are always fatal (definitions from the SSL specification):

unexpected_message: An inappropriate message was received.

bad_record_mac: An incorrect MAC was received.

decompression_failure: The decompression function received improper input (e.g., unable to decompress or decompress to greater than maximum allowable length).

handshake_failure: Sender was unable to negotiate an acceptable set of security parameters given the options available.

illegal_parameter: A field in a handshake message was out of range or inconsistent with other fields.

The remainder of the alerts are the following:

close_notify: Notifies the recipient that the sender will not send any more messages on this connection. Each party is required to send a close_notify alert before closing the write side of a connection.

no_certificate: May be sent in response to a certificate request if no appropriate certificate is available.

bad_certificate: A received certificate was corrupt (e.g., contained a signature that did not verify).

unsupported_certificate: The type of the received certificate is not supported.

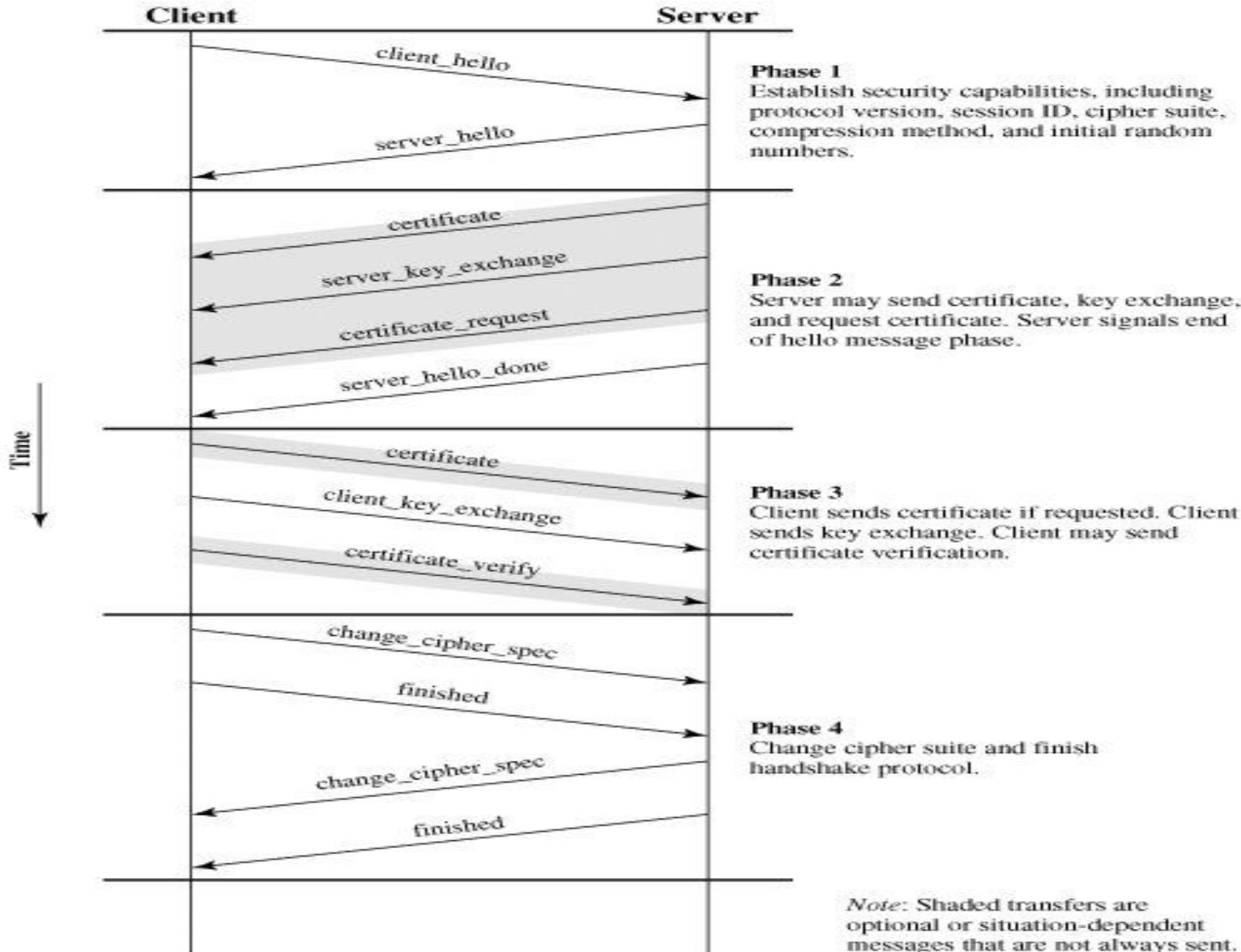
certificate_revoked: A certificate has been revoked by its signer.

certificate_expired: A certificate has expired.

certificate_unknown: Some other unspecified issue arose in processing the certificate, rendering it unacceptable.

Handshake Protocol

- This protocol allows the **server and client to authenticate each other** and to negotiate an **encryption and MAC algorithm** and **cryptographic keys** to be used to protect data sent in an SSL record.
- The Handshake Protocol is used **before any application data/SSL Record is transmitted**.



TLS

TLS means Transport Layer Security, which is a cryptographic protocol successor of SSL 3.0.

Cipher suites:

SSL protocol offers **support for Fortezza cipher suite**. TLS does not offer support. TLS follows a better standardization process that makes defining of new **cipher suites easier like RC4, Triple DES, AES, IDEA, etc.**

Alert messages:

SSL has the **“No certificate” alert message**. TLS protocol **removes the alert message** and replaces it with several other alert messages.

Record Protocol:

SSL uses **Message Authentication Code (MAC)** after encrypting each message while TLS on the other hand uses **HMAC** — a hash-based message authentication code after each message encryption.

Handshake process:

In SSL, the hash calculation also comprises the master secret and pad while in TLS, the hashes are calculated over handshake message.

Message Authentication:

SSL message authentication adjoins the key details and application data in ad-hoc way while TLS version relies on HMAC Hash-based Message Authentication Code.

SET

SET is an open encryption and security specification designed **to protect credit card transactions on the Internet**. SET provides three services:

1. Provides a **secure communications channel among all parties** involved in a transaction.
2. Provides trust by the use of **X.509v3 digital certificates**.
3. **Ensures privacy** because the information is only available to parties in a transaction when and where necessary.

SET incorporates the following features:

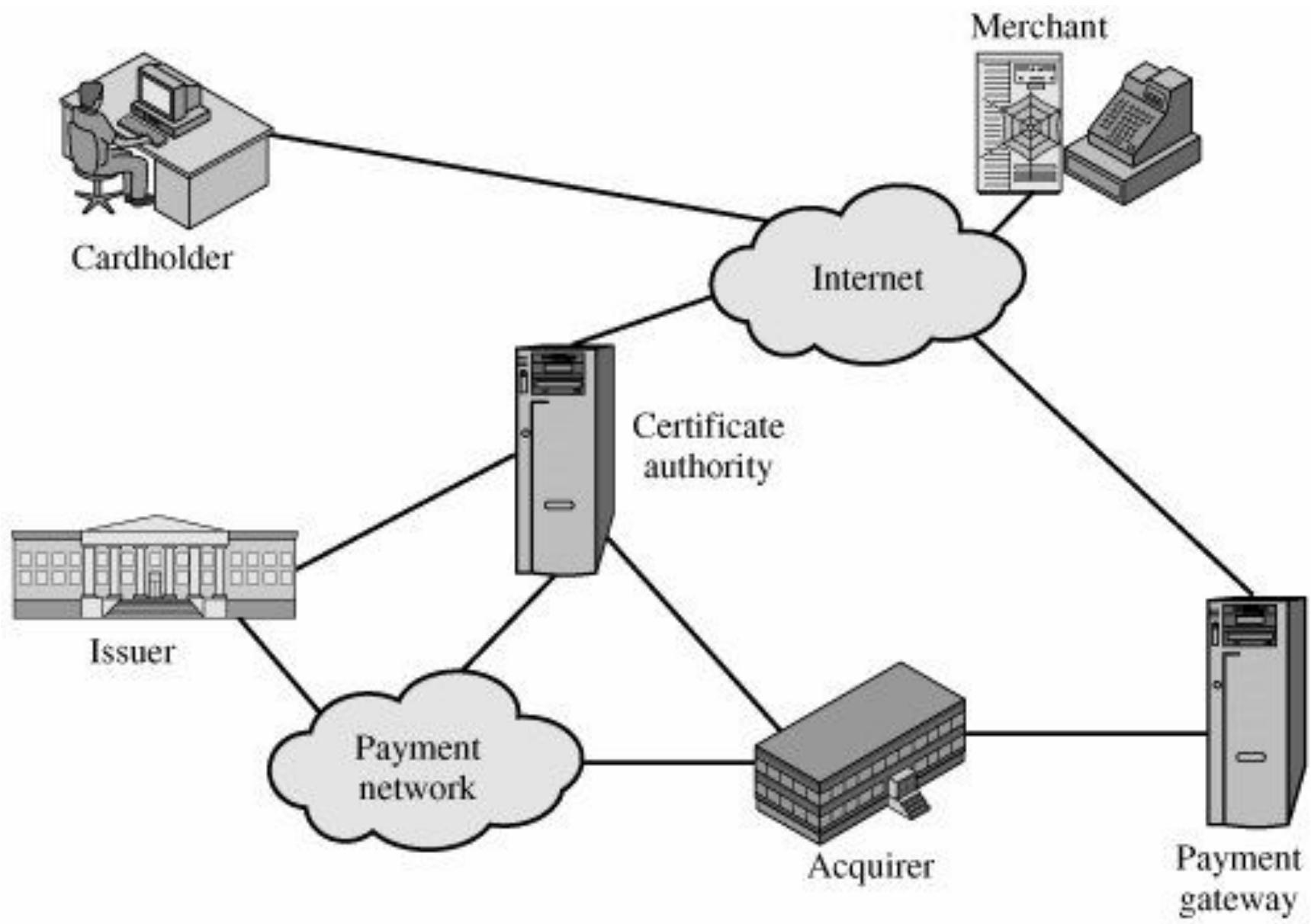
- 1. Confidentiality of information**
- 2. Integrity of data**
- 3. Cardholder account authentication**
- 4. Merchant authentication**

SET Participants

1. **Cardholder:** A cardholder is an authorized **holder of a payment card** (e.g., MasterCard, Visa) that has been issued by an issuer.
2. **Merchant:** A merchant is a **person or organization that has goods** or services to sell to the cardholder. Typically, these goods and services are offered via a Web site or by electronic mail.
3. **Issuer:** This is a financial institution, such as a **bank**, that provides the cardholder with the payment card.
4. **Acquirer:** This is a financial institution that establishes an account with a merchant and **processes payment card authorizations and payments**. Merchants will usually accept more than one credit card brand but do not want to deal with multiple bankcard associations or with multiple individual issuers. The acquirer provides authorization to the merchant that a given card account is active and that the proposed purchase does not exceed the credit limit. The acquirer also provides electronic transfer of payments to the merchant's account. Subsequently, the acquirer is reimbursed by the issuer over some sort of payment network for electronic funds transfer.

5. Payment gateway: This is a function operated by the acquirer or a designated third party that **processes merchant payment messages**. The payment gateway interfaces between SET and the existing bankcard payment networks for authorization and payment functions. The merchant exchanges SET messages with the payment gateway over the Internet, while the payment gateway has some direct or network connection to the acquirer's financial processing system.

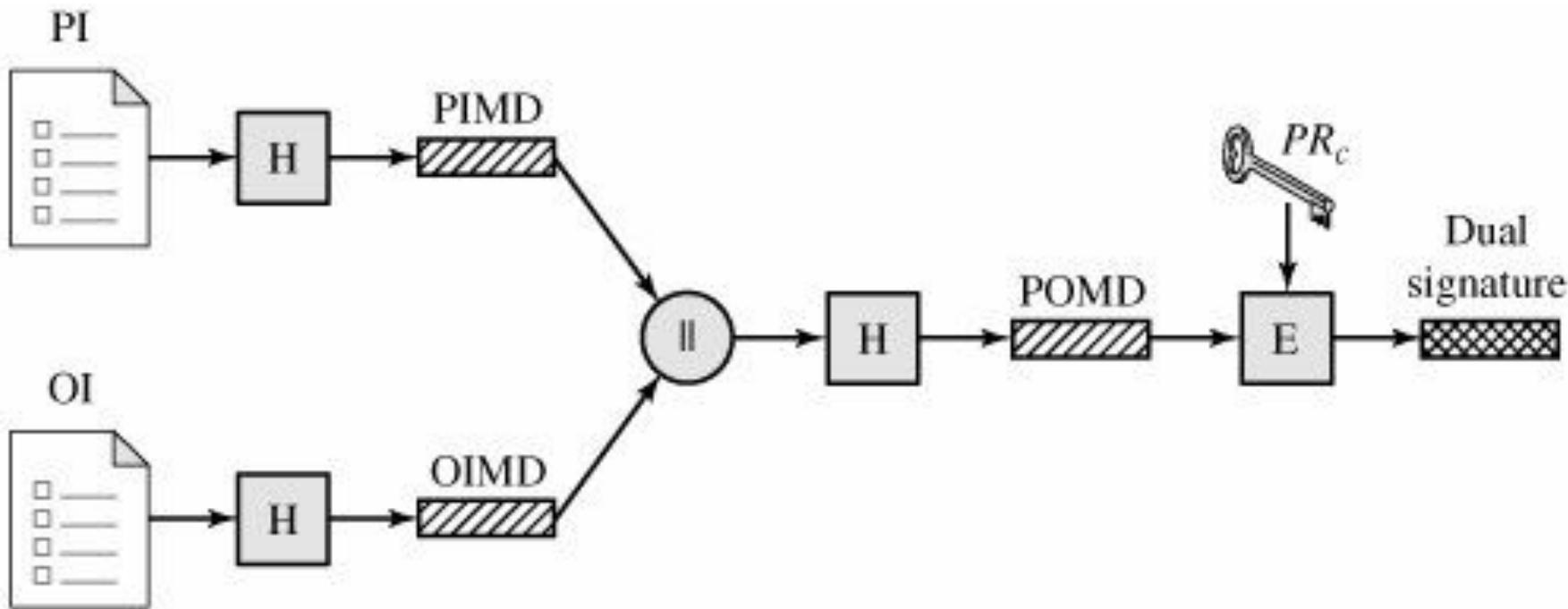
6. Certification authority (CA): This is an entity that is trusted to issue X.509v3 **public-key certificates** for **cardholders, merchants, and payment gateways**. The success of SET will depend on the existence of a CA infrastructure available for this purpose. As was discussed in previous chapters, a hierarchy of CAs is used, so that participants need not be directly certified by a root authority.



Dual Signature

- The purpose of the dual signature is to link two messages that are intended for two different recipients.
- In this case, the customer wants to send the **order information (OI) to the merchant** and the **payment information (PI) to the bank**.
- **The merchant does not need to know the customer's credit card number, and the bank does not need to know the details of the customer's order.**
- The customer is afforded extra protection in terms of privacy by keeping these two items separate.
- However, the two items must be linked in a way that can be used to resolve disputes if necessary.
- The link is needed so that the customer can prove that this payment is intended for this order and not for some other goods or service.

Construction of Dual Signature



PI = Payment information

OI = Order information

H = Hash function (SHA-1)

|| = Concatenation

PIMD = PI message digest

OIMD = OI message digest

POMD = Payment order message digest

E = Encryption (RSA)

PR_c = Customer's private signature key

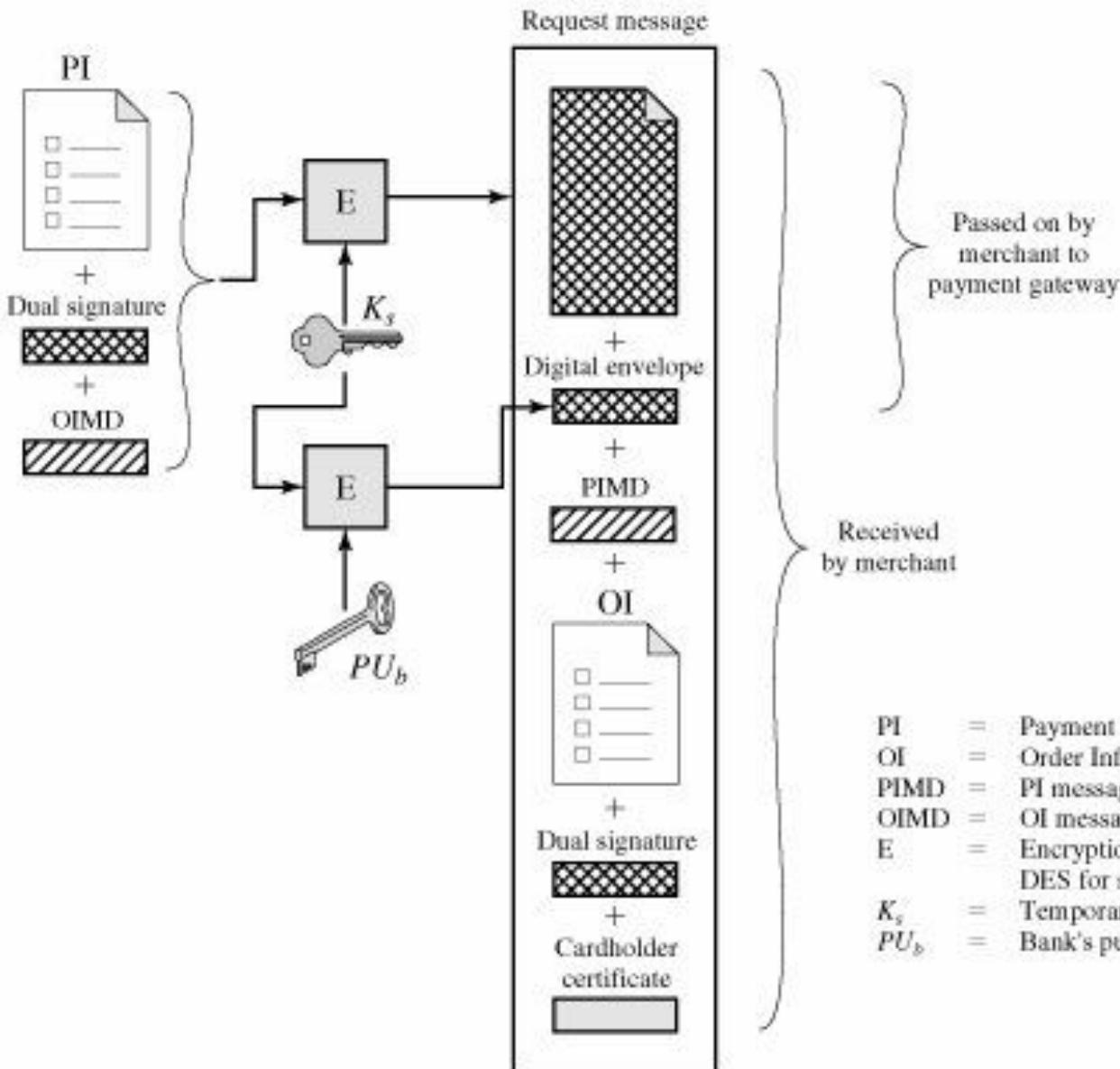
Payment Processing

Purchase request: Message from customer to merchant containing OI for merchant and PI for bank.

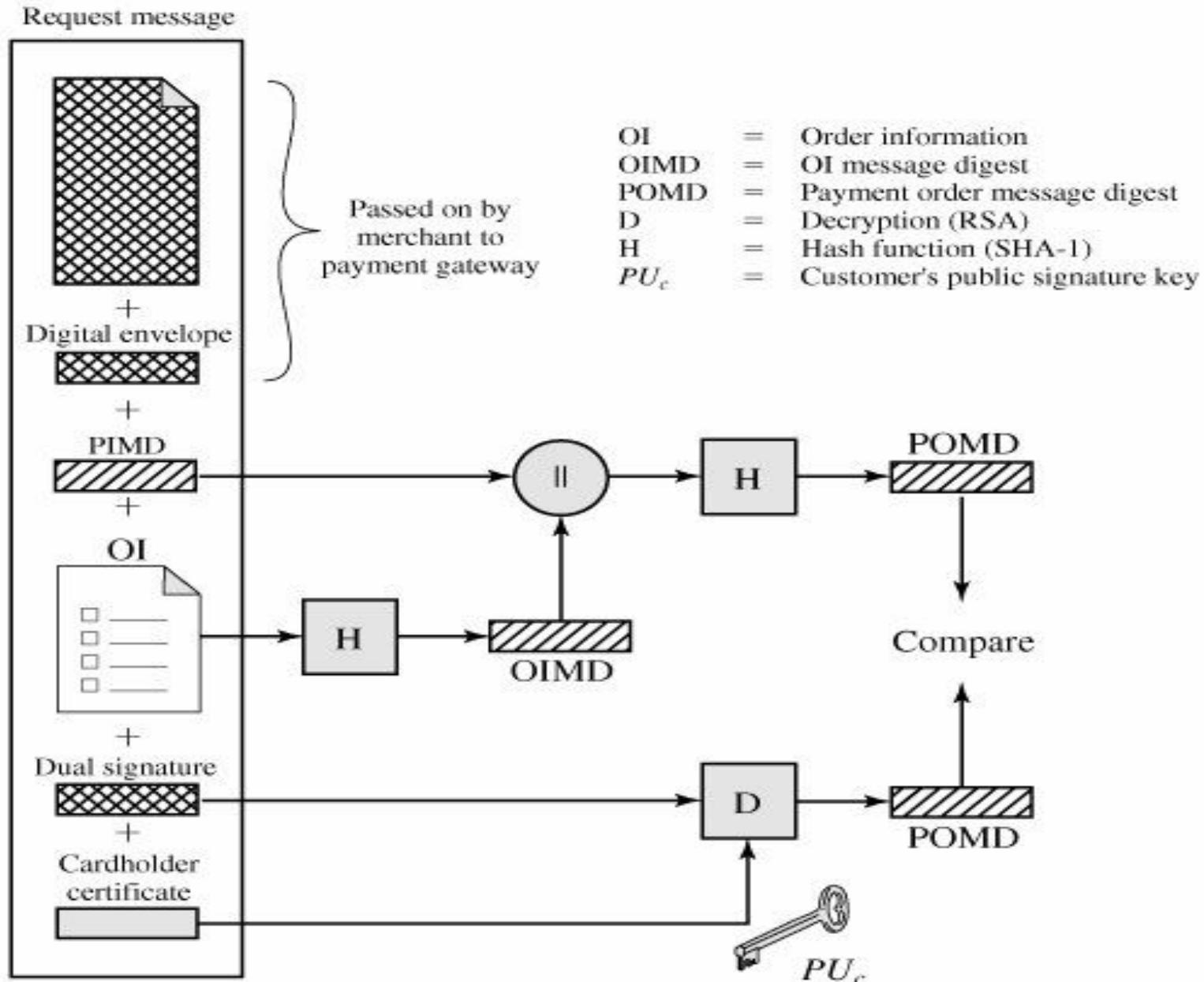
Payment authorization: Exchange between merchant and payment gateway to authorize a given amount for a purchase on a given credit card account.

Payment capture: Allows the merchant to request payment from the payment gateway.

Cardholder Sends Purchase Request



Merchant Verifies Customer Purchase Request





Chapter 7

Advanced Encryption Standard (AES)

Objectives

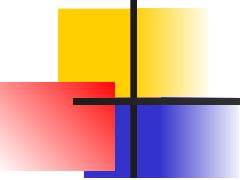
- *To review a short history of AES*
- *To define the basic structure of AES*
- *To define the transformations used by AES*
- *To define the key expansion process*
- *To discuss different implementations*

7-1 INTRODUCTION

The Advanced Encryption Standard (AES) is a symmetric-key block cipher published by the National Institute of Standards and Technology (NIST) in December 2001.

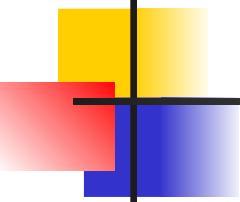
Topics discussed in this section:

- 7.1.1 History**
- 7.1.2 Criteria**
- 7.1.3 Rounds**
- 7.1.4 Data Units**
- 7.1.5 Structure of Each Round**



7.1.1 History.

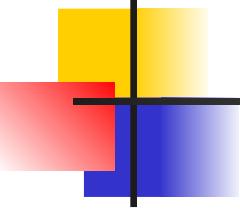
In February 2001, NIST announced that a draft of the Federal Information Processing Standard (FIPS) was available for public review and comment. Finally, AES was published as FIPS 197 in the Federal Register in December 2001.



7.1.2 Criteria

The criteria defined by NIST for selecting AES fall into three areas:

- 1. Security***
- 2. Cost***
- 3. Implementation.***



7.1.3 Rounds.

AES is a non-Feistel cipher that encrypts and decrypts a data block of 128 bits. It uses 10, 12, or 14 rounds. The key size, which can be 128, 192, or 256 bits, depends on the number of rounds.

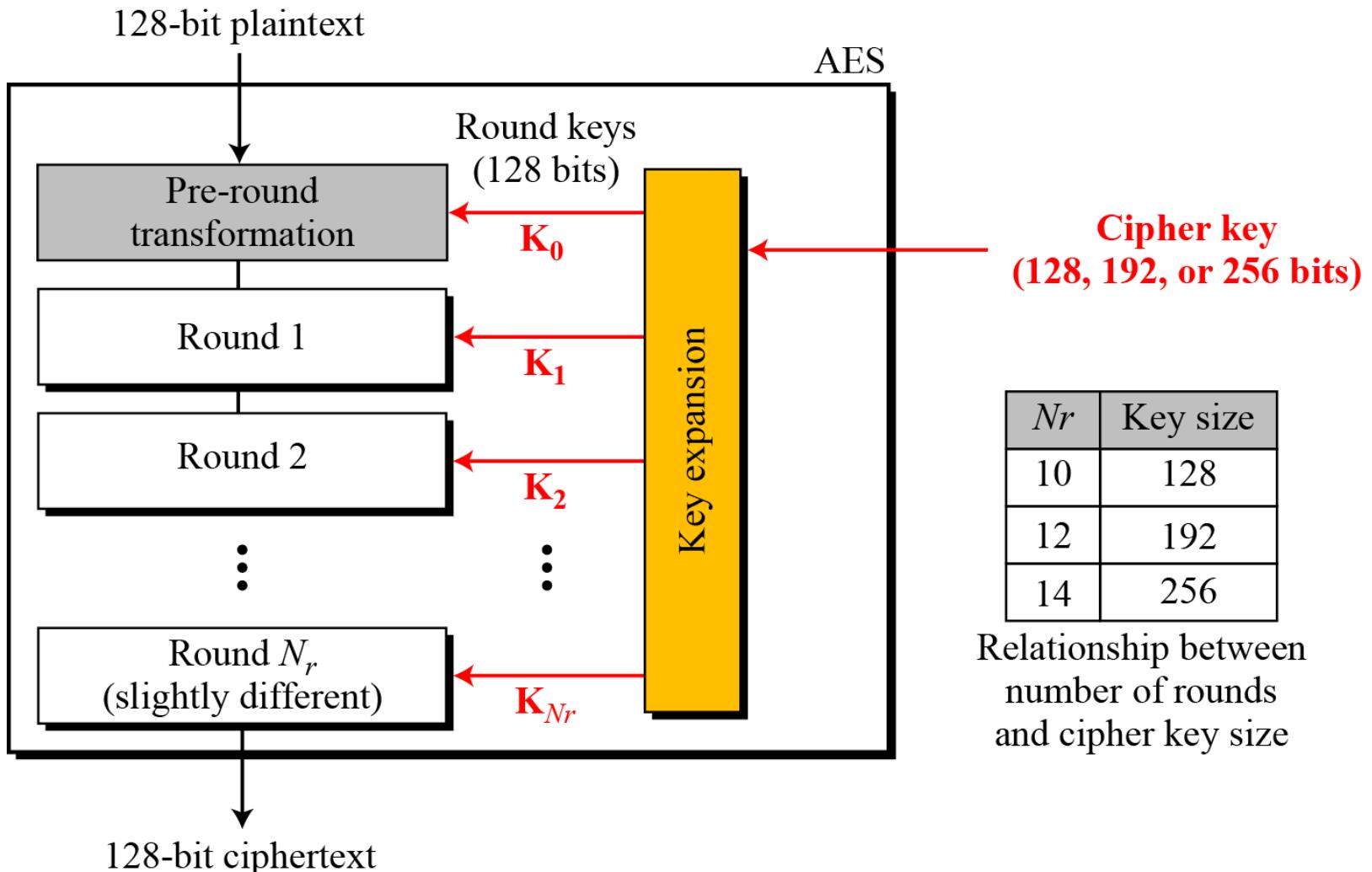
Note

AES has defined three versions, with 10, 12, and 14 rounds.

Each version uses a different cipher key size (128, 192, or 256), but the round keys are always 128 bits.

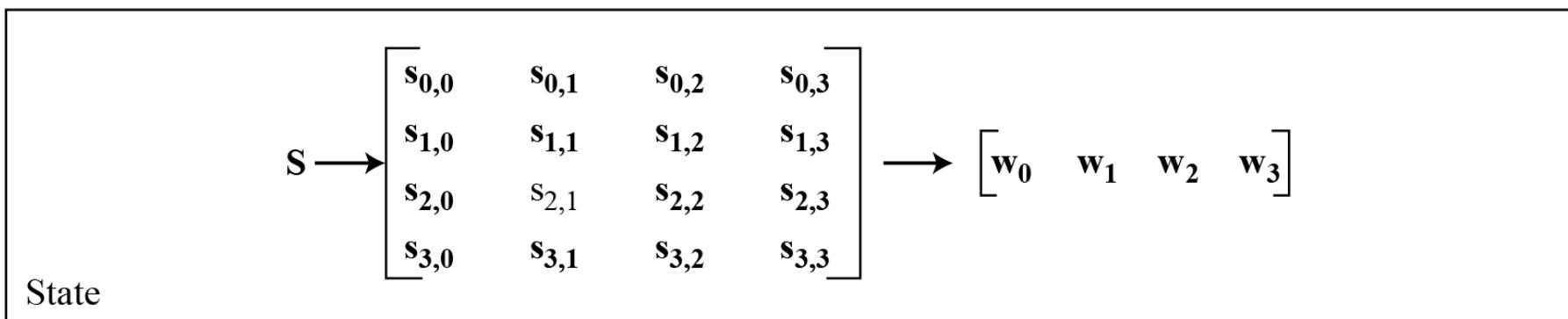
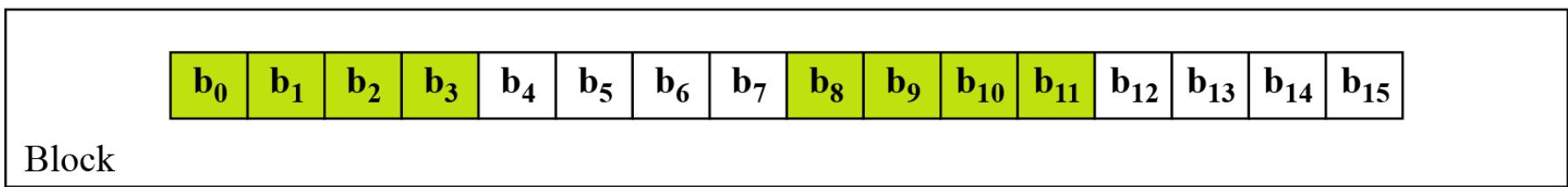
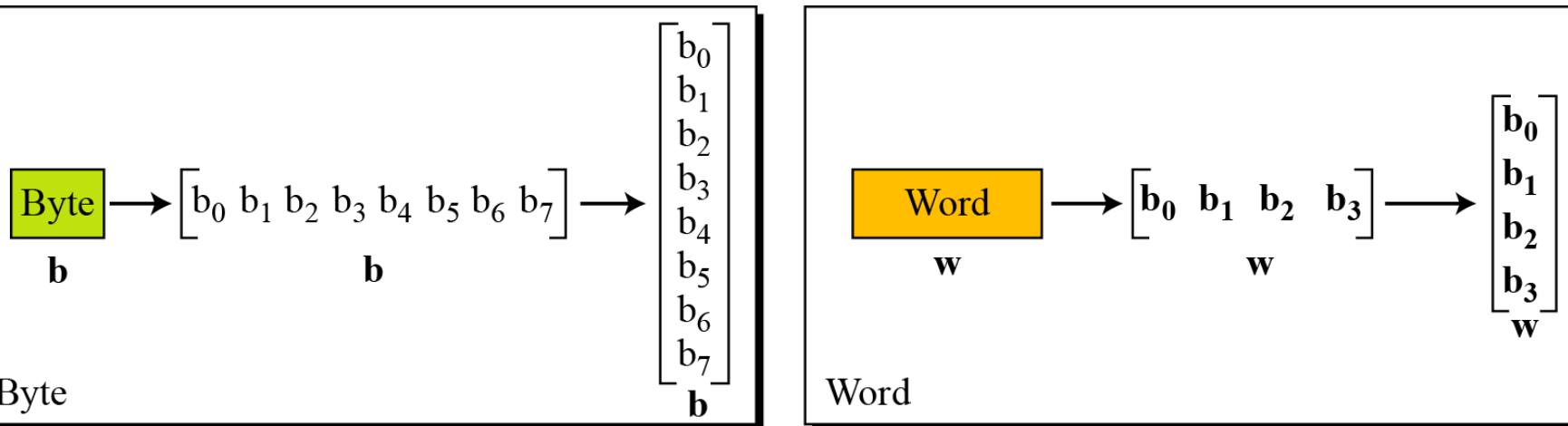
7.1.3 Continue

Figure 7.1 General design of AES encryption cipher



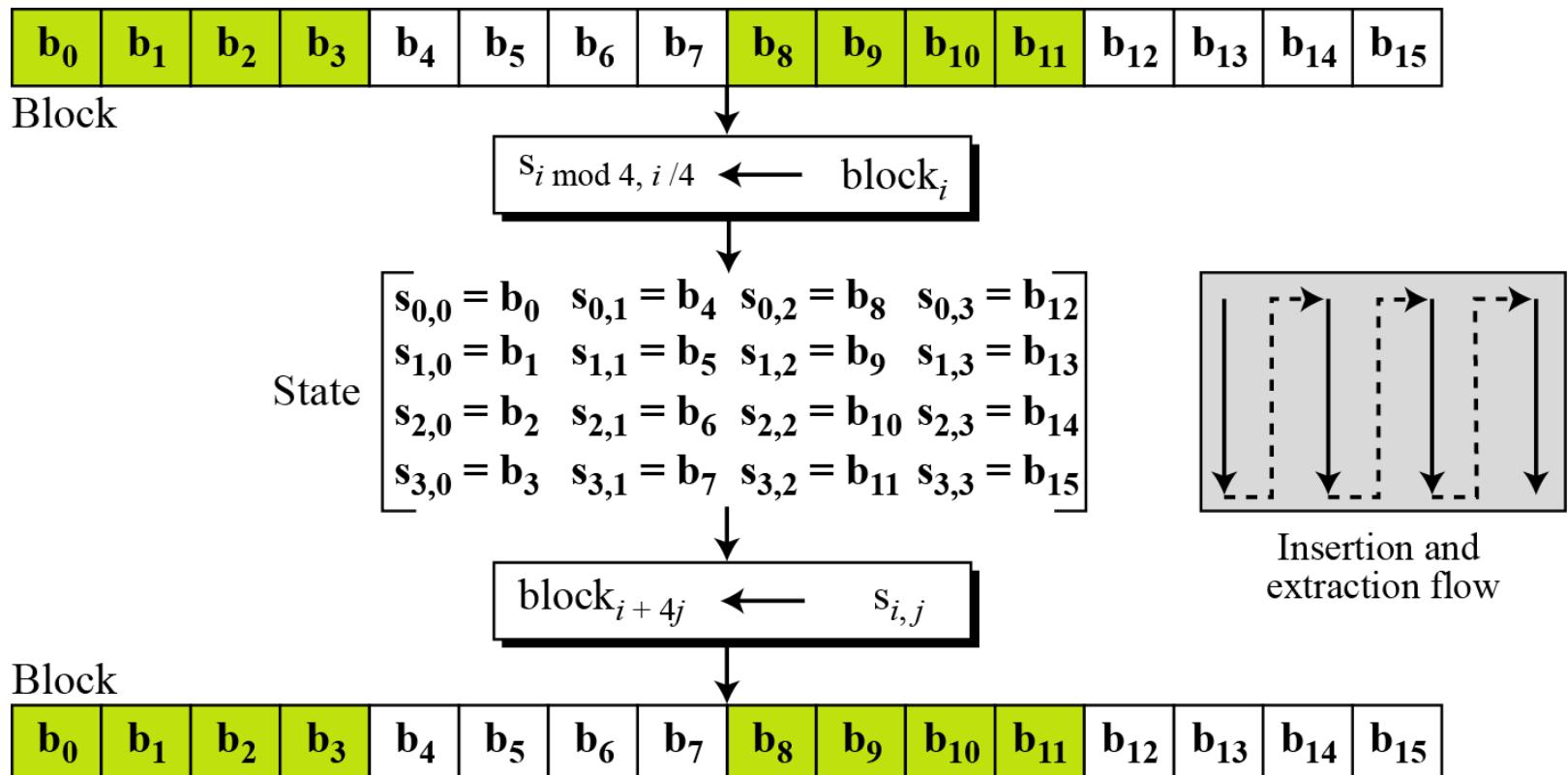
7.1.4 Data Units.

Figure 7.2 Data units used in AES



7.1.4 Continue

Figure 7.3 Block-to-state and state-to-block transformation



7.1.4 Continue

Example 7.1 Continue

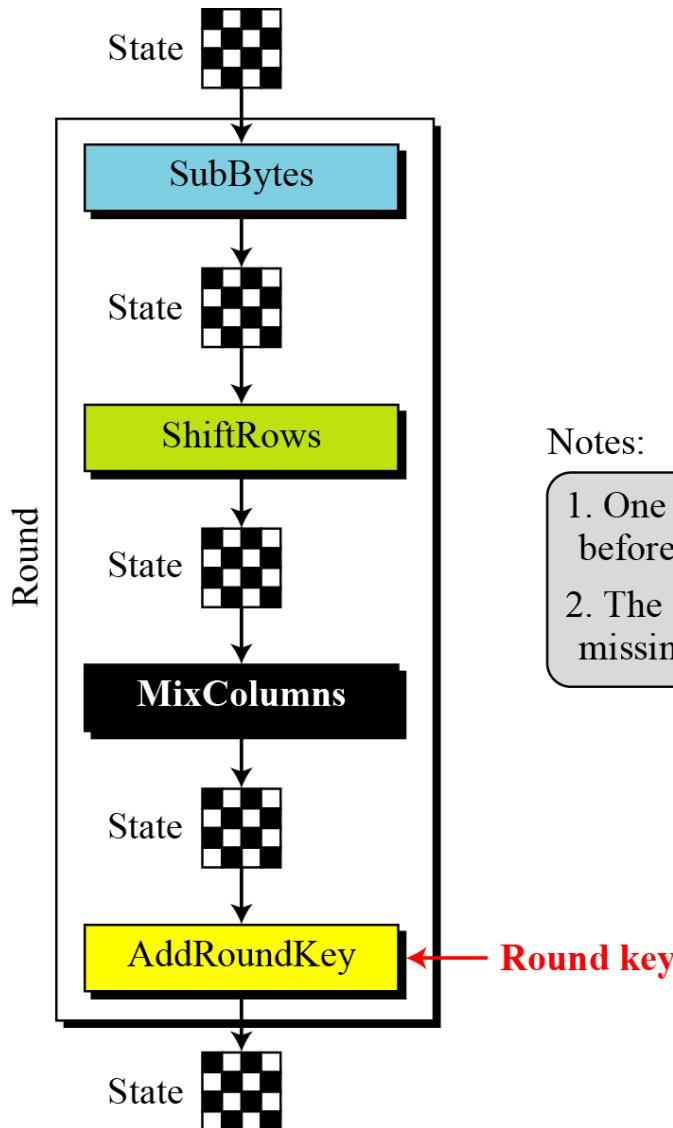
Figure 7.4 Changing plaintext to state

Text	A	E	S	U	S	E	S	A	M	A	T	R	I	X	Z	Z
Hexadecimal	00	04	12	14	12	04	12	00	0C	00	13	11	08	23	19	19
	00	12	0C	08	04	04	00	23	12	12	13	19	14	00	11	19

$$\begin{bmatrix} 00 & 12 & 0C & 08 \\ 04 & 04 & 00 & 23 \\ 12 & 12 & 13 & 19 \\ 14 & 00 & 11 & 19 \end{bmatrix}$$
 State

7.1.5 Structure of Each Round

Figure 7.5 Structure of each round at the encryption site



Notes:

1. One AddRoundKey is applied before the first round.
2. The third transformation is missing in the last round.

7-2 TRANSFORMATIONS

To provide security, AES uses four types of transformations: substitution, permutation, mixing, and key-adding.

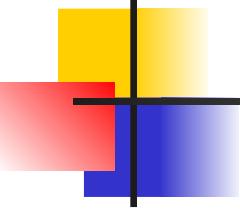
Topics discussed in this section:

7.2.1 Substitution

7.2.2 Permutation

7.2.3 Mixing

7.2.4 Key Adding



7.2.1 Substitution

AES, like DES, uses substitution. AES uses two invertible transformations.

SubBytes

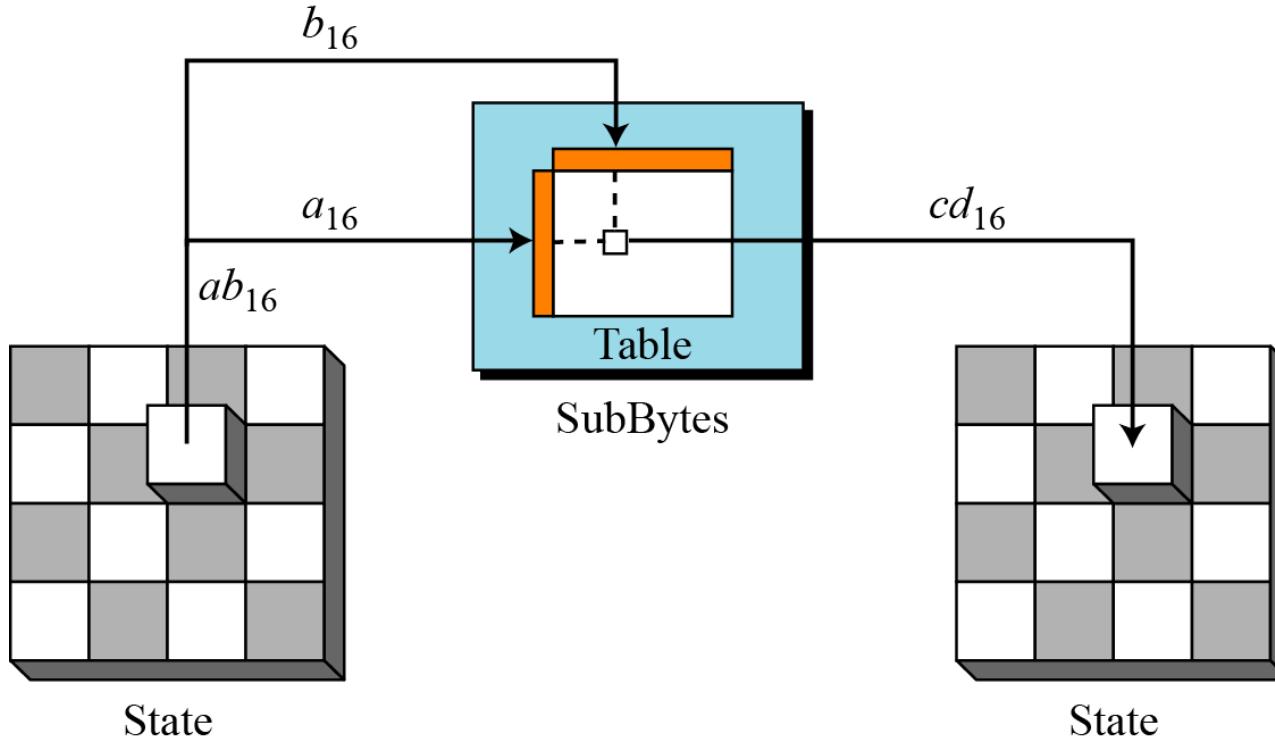
The first transformation, SubBytes, is used at the encryption site. To substitute a byte, we interpret the byte as two hexadecimal digits.

Note

The SubBytes operation involves 16 independent byte-to-byte transformations.

7.2.1 Continue

Figure 7.6 SubBytes transformation



7.2.1 Continue

Table 7.1 SubBytes transformation table

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8

7.2.1 Continue

Table 7.1 SubBytes transformation table (continued)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	CB	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

7.2.1 Continue

InvSubBytes

Table 7.2 InvSubBytes transformation table

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	52	09	6A	D5	30	36	A5	38	BF	40	A3	9E	81	F3	D7	FB
1	7C	E3	39	82	9B	2F	FF	87	34	8E	43	44	C4	DE	E9	CB
2	54	7B	94	32	A6	C2	23	3D	EE	4C	95	0B	42	FA	C3	4E
3	08	2E	A1	66	28	D9	24	B2	76	5B	A2	49	6D	8B	D1	25
4	72	F8	F6	64	86	68	98	16	D4	A4	5C	CC	5D	65	B6	92
5	6C	70	48	50	FD	ED	B9	DA	5E	15	46	57	A7	8D	9D	84
6	90	D8	AB	00	8C	BC	D3	0A	F7	E4	58	05	B8	B3	45	06
7	D0	2C	1E	8F	CA	3F	0F	02	C1	AF	BD	03	01	13	8A	6B

7.2.1 Continue

InvSubBytes (Continued)

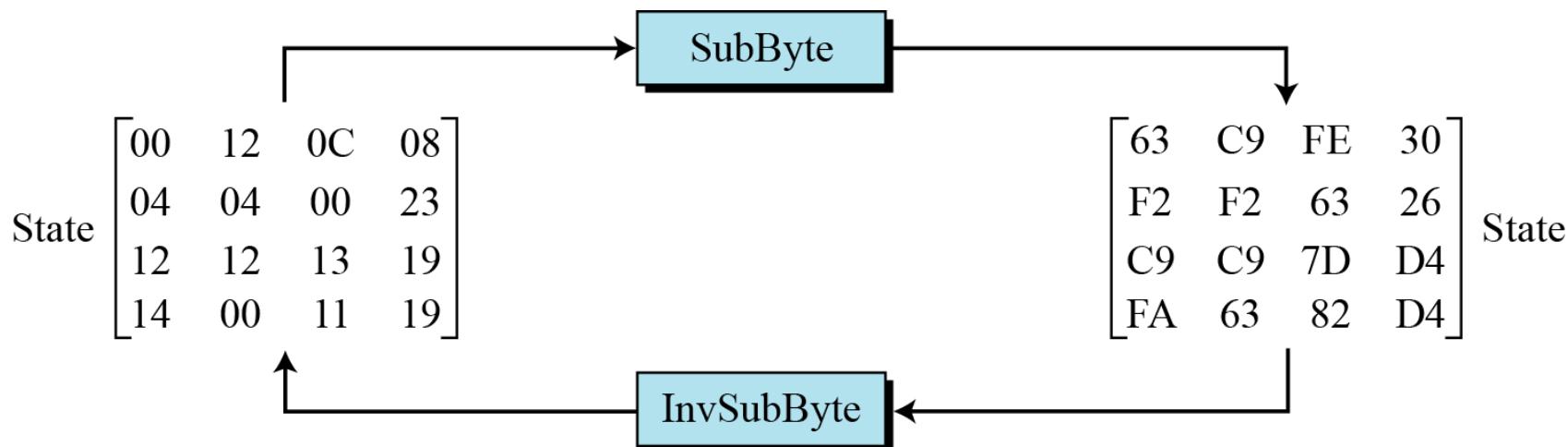
8	3A	91	11	41	4F	67	DC	EA	97	F2	CF	CE	F0	B4	E6	73
9	96	AC	74	22	E7	AD	35	85	E2	F9	37	E8	1C	75	DF	6E
A	47	F1	1A	71	1D	29	C5	89	6F	B7	62	0E	AA	18	BE	1B
B	FC	56	3E	4B	C6	D2	79	20	9A	DB	C0	FE	78	CD	5A	F4
C	1F	DD	A8	33	88	07	C7	31	B1	12	10	59	27	80	EC	5F
D	60	51	7F	A9	19	B5	4A	0D	2D	E5	7A	9F	93	C9	9C	EF
E	A0	E0	3B	4D	AE	2A	F5	B0	C8	EB	BB	3C	83	53	99	61
F	17	2B	04	7E	BA	77	D6	26	E1	69	14	63	55	21	0C	7D

7.2.1 Continue

Example 7.2

Figure 7.7 shows how a state is transformed using the SubBytes transformation. The figure also shows that the InvSubBytes transformation creates the original one. Note that if the two bytes have the same values, their transformation is also the same.

Figure 7.7 SubBytes transformation for Example 7.2



Transformation Using the $GF(2^8)$ Field

AES also defines the transformation algebraically using the $GF(2^8)$ field with the irreducible polynomials $(x^8 + x^4 + x^3 + x + 1)$, as shown in Figure 7.8.

subbyte: $\rightarrow \mathbf{d} = \mathbf{X} (s_{r,c})^{-1} \oplus \mathbf{y}$

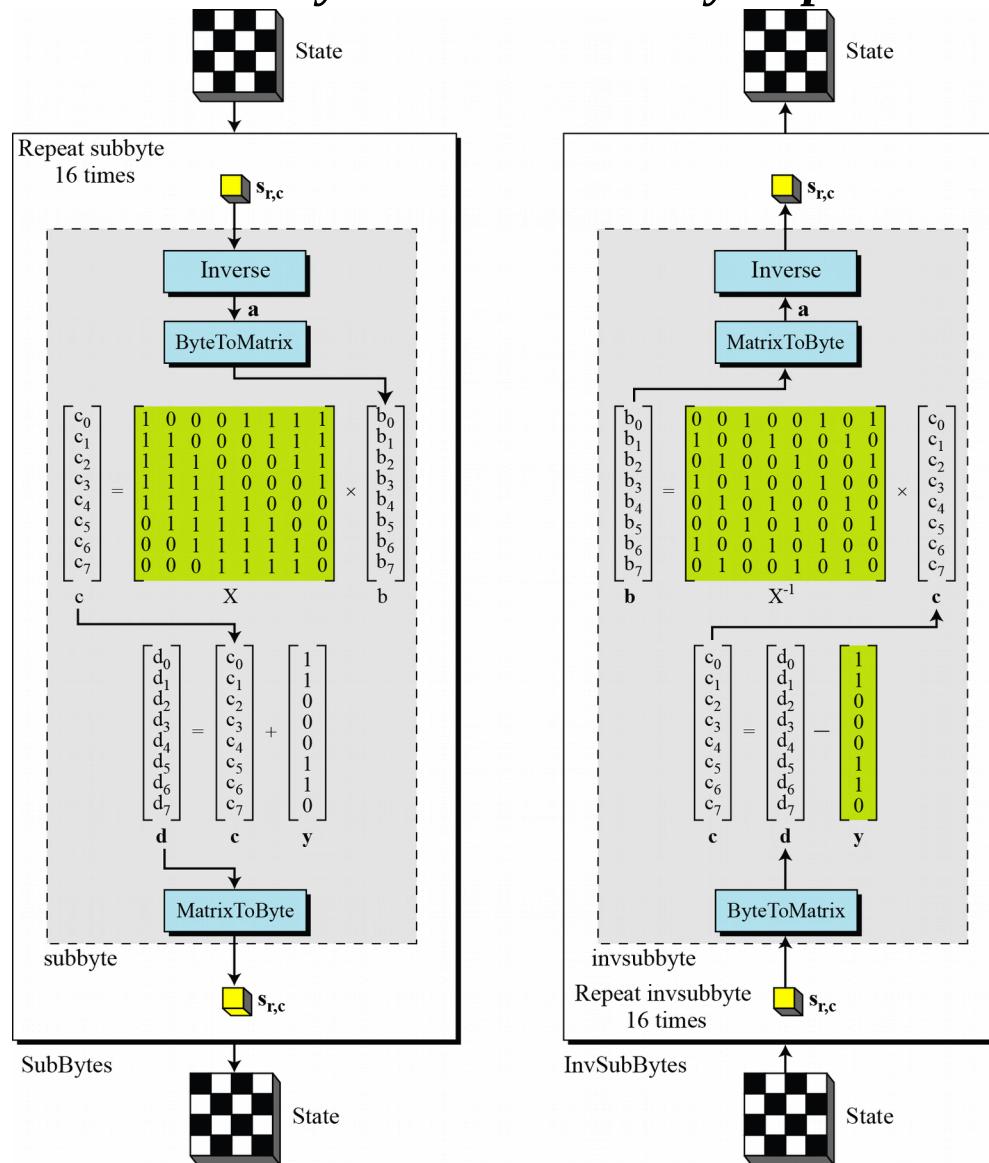
invsubbyte: $\rightarrow [\mathbf{X}^{-1}(\mathbf{d} \oplus \mathbf{y})]^{-1} = [\mathbf{X}^{-1}(\mathbf{X} (s_{r,c})^{-1} \oplus \mathbf{y} \oplus \mathbf{y})]^{-1} = [(s_{r,c})^{-1}]^{-1} = s_{r,c}$

Note

The SubBytes and InvSubBytes transformations are inverses of each other.

7.2.1 Continue

Figure 7.8 SubBytes and InvSubBytes processes



7.2.1 Continue

Example 7.3

Let us show how the byte 0C is transformed to FE by subbyte routine and transformed back to 0C by the invsubbyte routine.

1. *subbyte:*
 - a. The multiplicative inverse of 0C in GF(2⁸) field is B0, which means **b** is (10110000).
 - b. Multiplying matrix **X** by this matrix results in **c** = (10011101)
 - c. The result of XOR operation is **d** = (11111110), which is FE in hexadecimal.

2. *invsubbyte:*
 - a. The result of XOR operation is **c** = (10011101)
 - b. The result of multiplying by matrix **X⁻¹** is (11010000) or B0
 - c. The multiplicative inverse of B0 is 0C.

7.2.1 Continue

Algorithm 7.1 Pseudocode for SubBytes transformation

SubBytes (S)

```
{  
    for (r = 0 to 3)  
        for (c = 0 to 3)  
            Sr,c = subbyte (Sr,c)  
}
```

subbyte (byte)

```
{  
    a ← byte-1           //Multiplicative inverse in GF(28) with inverse of 00 to be 00  
    ByteToMatrix (a, b)  
    for (i = 0 to 7)  
    {  
        ci ← bi ⊕ b(i+4)mod 8 ⊕ b(i+5)mod 8 ⊕ b(i+6)mod 8 ⊕ b(i+7)mod 8  
        di ← ci ⊕ ByteToMatrix (0x63)  
    }  
    MatrixToByte (d, d)  
    byte ← d  
}
```

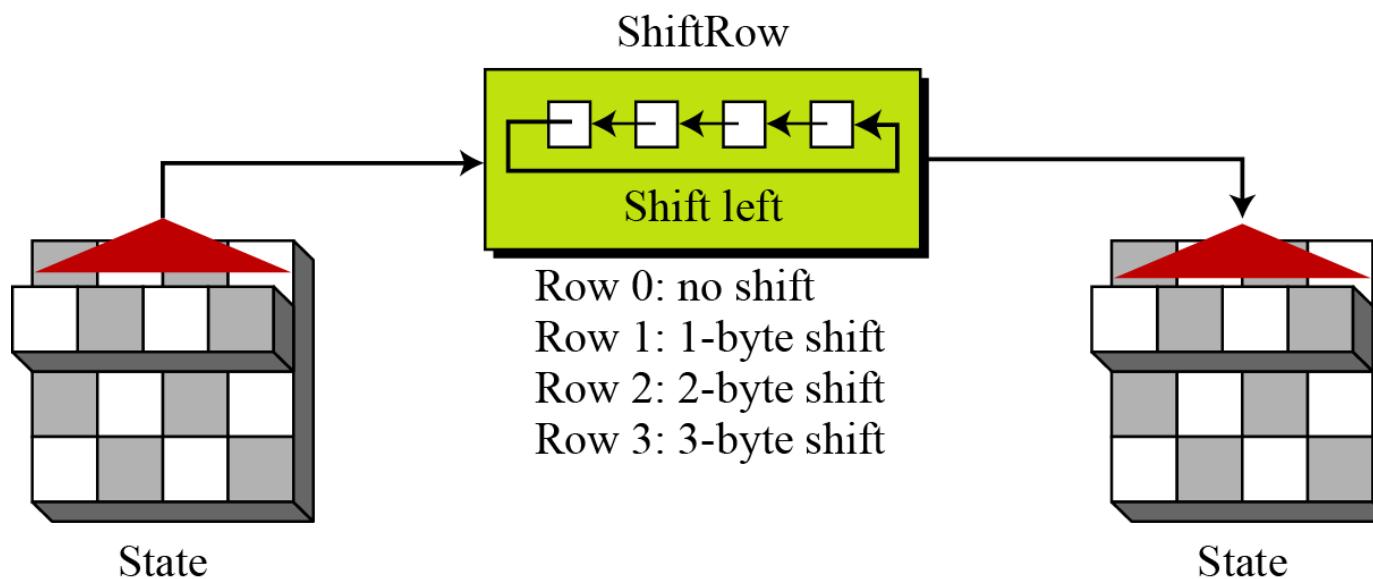
7.2.2 Permutation

Another transformation found in a round is shifting, which permutes the bytes.

ShiftRows

In the encryption, the transformation is called ShiftRows.

Figure 7.9 ShiftRows transformation



7.2.2 Continue

InvShiftRows

In the decryption, the transformation is called InvShiftRows and the shifting is to the right.

Algorithm 7.2 Pseudocode for ShiftRows transformation

ShiftRows (\mathbf{S})

{

 for ($r = 1$ to 3)

 shiftrow (\mathbf{s}_r, r) // s_r is the r th row

}

shiftrow (\mathbf{row}, n) // n is the number of bytes to be shifted

{

 CopyRow (\mathbf{row}, \mathbf{t}) // t is a temporary row

 for ($c = 0$ to 3)

$\mathbf{row}_{(c - n) \text{ mod } 4} \leftarrow \mathbf{t}_c$

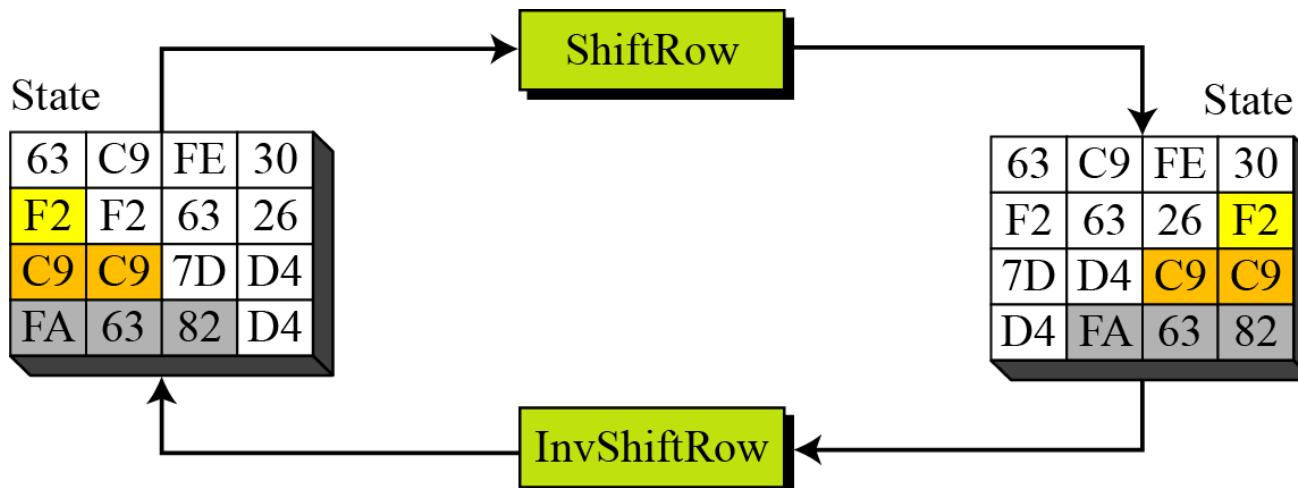
}

7.2.2 Continue

Example 7.4

Figure 7.10 shows how a state is transformed using ShiftRows transformation. The figure also shows that InvShiftRows transformation creates the original state.

Figure 7.10 ShiftRows transformation in Example 7.4



7.2.3 Mixing

We need an interbyte transformation that changes the bits inside a byte, based on the bits inside the neighboring bytes. We need to mix bytes to provide diffusion at the bit level.

Figure 7.11 Mixing bytes using matrix multiplication

$$\begin{array}{l} ax + by + cz + dt \\ ex + fy + gz + ht \\ ix + jy + kz + lt \\ mx + ny + oz + pt \end{array} \xrightarrow{\text{New matrix}} = \begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{bmatrix} \times \begin{bmatrix} x \\ y \\ z \\ t \end{bmatrix}$$

Constant matrix

Old matrix

7.2.3 Continue

Figure 7.12 Constant matrices used by MixColumns and InvMixColumns

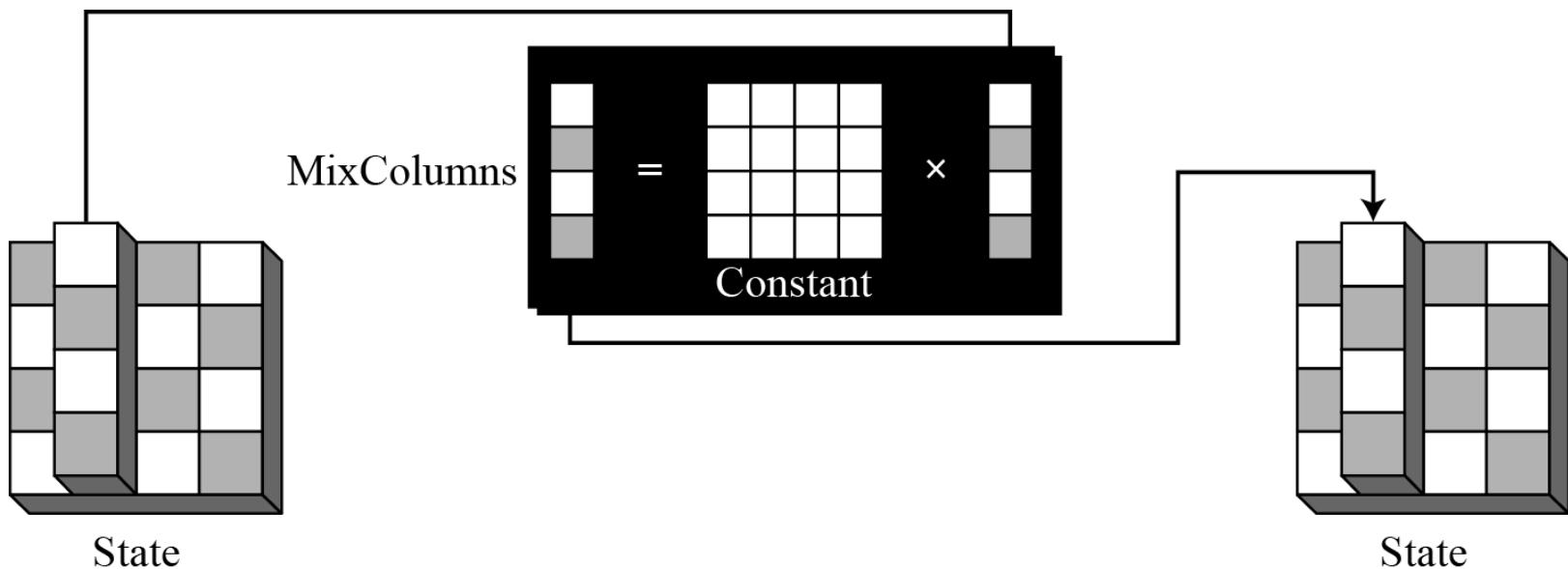
$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \xleftrightarrow{\text{Inverse}} \begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix}$$

C C⁻¹

MixColumns

The MixColumns transformation operates at the column level; it transforms each column of the state to a new column.

Figure 7.13 MixColumns transformation



InvMixColumns

The InvMixColumns transformation is basically the same as the MixColumns transformation.

Note

The MixColumns and InvMixColumns transformations are inverses of each other.

7.2.3 Continue

Algorithm 7.3 Pseudocode for MixColumns transformation

```
MixColumns (S)
{
    for (c = 0 to 3)
        mixcolumn ( $s_c$ )
}

mixcolumn (col)
{
    CopyColumn (col, t)           // t is a temporary column

    col0  $\leftarrow$  (0x02) • t0  $\oplus$  (0x03 • t1)  $\oplus$  t2  $\oplus$  t3

    col1  $\leftarrow$  t0  $\oplus$  (0x02) • t1  $\oplus$  (0x03) • t2  $\oplus$  t3

    col2  $\leftarrow$  t0  $\oplus$  t1  $\oplus$  (0x02) • t2  $\oplus$  (0x03) • t3

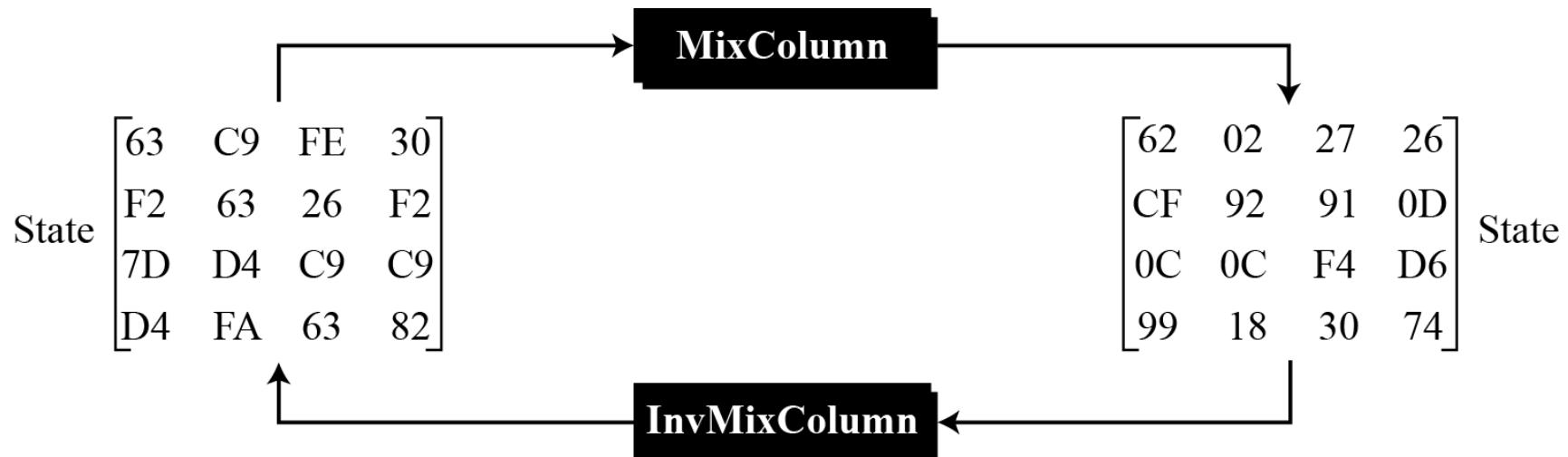
    col3  $\leftarrow$  (0x03 • t0)  $\oplus$  t1  $\oplus$  t2  $\oplus$  (0x02) • t3
}
```

7.2.3 Continue

Example 7.5

Figure 7.14 shows how a state is transformed using the MixColumns transformation. The figure also shows that the InvMixColumns transformation creates the original one.

Figure 7.14 The MixColumns transformation in Example 7.5



7.2.4 Key Adding

AddRoundKey

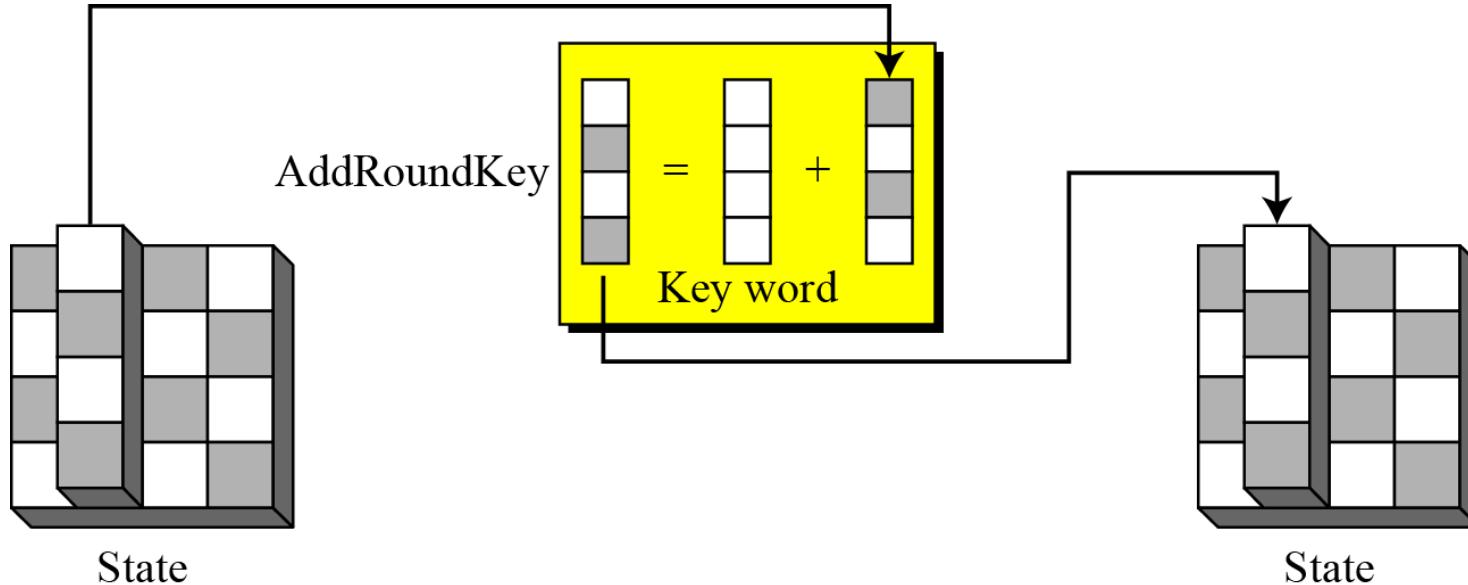
AddRoundKey proceeds one column at a time. AddRoundKey adds a round key word with each state column matrix; the operation in AddRoundKey is matrix addition.

Note

The AddRoundKey transformation is the inverse of itself.

7.2.4 Continue

Figure 7.15 AddRoundKey transformation



Algorithm 7.4 *Pseudocode for AddRoundKey transformation*

AddRoundKey (S)

{

 for ($c = 0$ to 3)

$s_c \leftarrow s_c \oplus w_{\text{round} + 4c}$

}

7-3 KEY EXPANSION

To create round keys for each round, AES uses a key-expansion process. If the number of rounds is N_r , the key-expansion routine creates $N_r + 1$ 128-bit round keys from one single 128-bit cipher key.

Topics discussed in this section:

- 7.3.1 Key Expansion in AES-128**
- 7.3.2 Key Expansion in AES-192 and AES-256**
- 7.3.3 Key-Expansion Analysis**

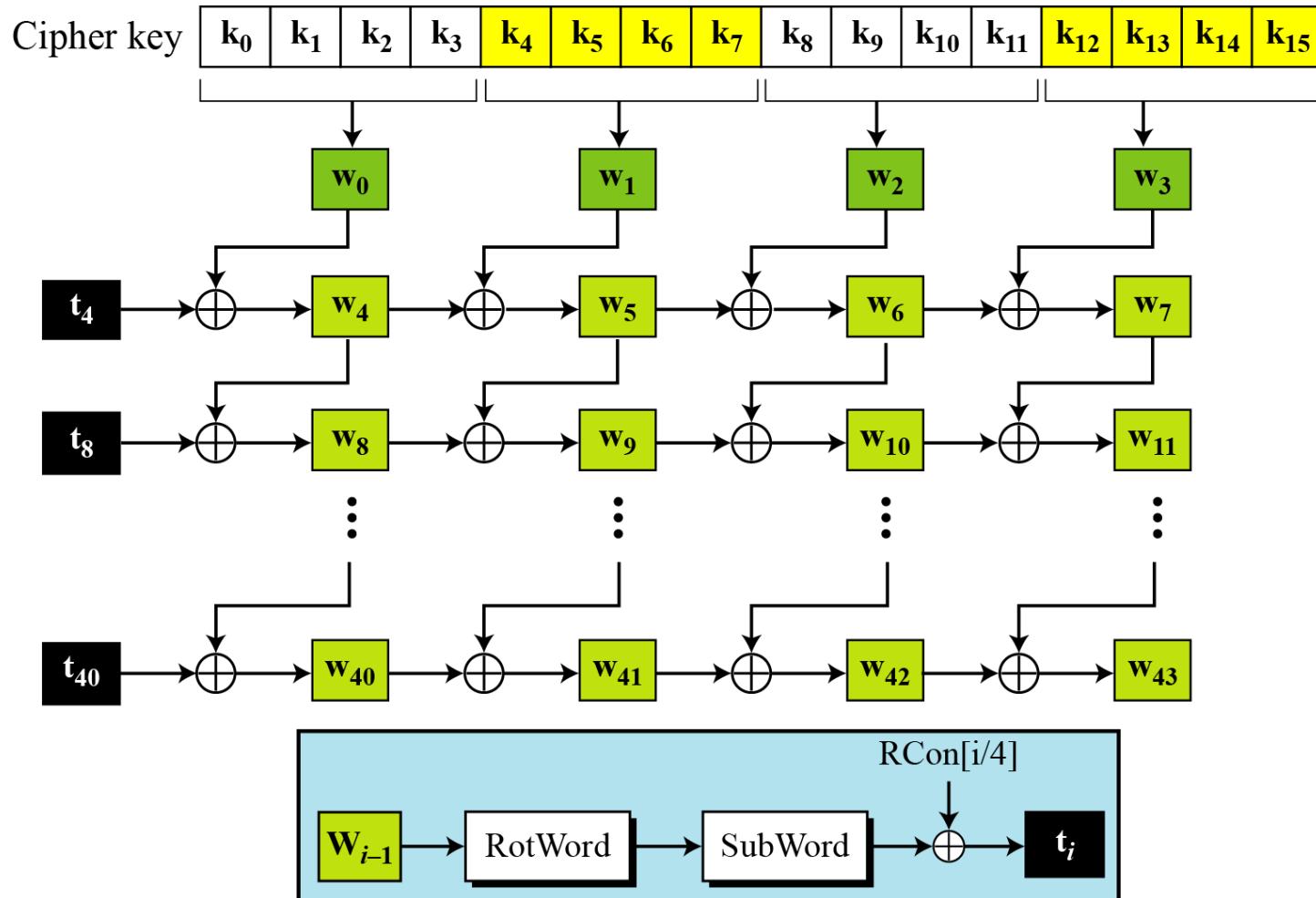
7-3 *Continued*

Table 7.3 *Words for each round*

<i>Round</i>	<i>Words</i>			
Pre-round	\mathbf{w}_0	\mathbf{w}_1	\mathbf{w}_2	\mathbf{w}_3
1	\mathbf{w}_4	\mathbf{w}_5	\mathbf{w}_6	\mathbf{w}_7
2	\mathbf{w}_8	\mathbf{w}_9	\mathbf{w}_{10}	\mathbf{w}_{11}
...	...			
N_r	\mathbf{w}_{4N_r}	\mathbf{w}_{4N_r+1}	\mathbf{w}_{4N_r+2}	\mathbf{w}_{4N_r+3}

7.3.1 Key Expansion in AES-128

Figure 7.16 Key expansion in AES



Making of t_i (temporary) words $i = 4 N_r$

7.3.1 Continue

Table 7.4 *RCon constants*

<i>Round</i>	<i>Constant (RCon)</i>	<i>Round</i>	<i>Constant (RCon)</i>
1	$(\underline{01} \text{ } 00 \text{ } 00 \text{ } 00)_{16}$	6	$(\underline{20} \text{ } 00 \text{ } 00 \text{ } 00)_{16}$
2	$(\underline{02} \text{ } 00 \text{ } 00 \text{ } 00)_{16}$	7	$(\underline{40} \text{ } 00 \text{ } 00 \text{ } 00)_{16}$
3	$(\underline{04} \text{ } 00 \text{ } 00 \text{ } 00)_{16}$	8	$(\underline{80} \text{ } 00 \text{ } 00 \text{ } 00)_{16}$
4	$(\underline{08} \text{ } 00 \text{ } 00 \text{ } 00)_{16}$	9	$(\underline{1B} \text{ } 00 \text{ } 00 \text{ } 00)_{16}$
5	$(\underline{10} \text{ } 00 \text{ } 00 \text{ } 00)_{16}$	10	$(\underline{36} \text{ } 00 \text{ } 00 \text{ } 00)_{16}$

7.3.1 Continue

The key-expansion routine can either use the above table when calculating the words or use the GF(2⁸) field to calculate the leftmost byte dynamically, as shown below (prime is the irreducible polynomial):

RC ₁	$\rightarrow x^{1-1}$	$=x^0$	mod prime	= 1	$\rightarrow 00000001$	$\rightarrow 01_{16}$
RC ₂	$\rightarrow x^{2-1}$	$=x^1$	mod prime	= x	$\rightarrow 00000010$	$\rightarrow 02_{16}$
RC ₃	$\rightarrow x^{3-1}$	$=x^2$	mod prime	= x^2	$\rightarrow 00000100$	$\rightarrow 04_{16}$
RC ₄	$\rightarrow x^{4-1}$	$=x^3$	mod prime	= x^3	$\rightarrow 00001000$	$\rightarrow 08_{16}$
RC ₅	$\rightarrow x^{5-1}$	$=x^4$	mod prime	= x^4	$\rightarrow 00010000$	$\rightarrow 10_{16}$
RC ₆	$\rightarrow x^{6-1}$	$=x^5$	mod prime	= x^5	$\rightarrow 00100000$	$\rightarrow 20_{16}$
RC ₇	$\rightarrow x^{7-1}$	$=x^6$	mod prime	= x^6	$\rightarrow 01000000$	$\rightarrow 40_{16}$
RC ₈	$\rightarrow x^{8-1}$	$=x^7$	mod prime	= x^7	$\rightarrow 10000000$	$\rightarrow 80_{16}$
RC ₉	$\rightarrow x^{9-1}$	$=x^8$	mod prime	= $x^4 + x^3 + x + 1$	$\rightarrow 00011011$	$\rightarrow 1B_{16}$
RC ₁₀	$\rightarrow x^{10-1}$	$=x^9$	mod prime	= $x^5 + x^4 + x^2 + x$	$\rightarrow 00110110$	$\rightarrow 36_{16}$

7.3.1 Continue

Algorithm 7.5 Pseudocode for key expansion in AES-128

KeyExpansion ([key₀ to key₁₅], [w₀ to w₄₃])

{

for ($i = 0$ to 3)

$w_i \leftarrow key_{4i} + key_{4i+1} + key_{4i+2} + key_{4i+3}$

for ($i = 4$ to 43)

{

if ($i \bmod 4 \neq 0$) $w_i \leftarrow w_{i-1} + w_{i-4}$

else

{

$t \leftarrow \text{SubWord}(\text{RotWord}(w_{i-1})) \oplus RCon_{i/4}$ // t is a temporary word

$w_i \leftarrow t + w_{i-4}$

}

}

}

7.3.1 Continue

Example 7.6

Table 7.5 shows how the keys for each round are calculated assuming that the 128-bit cipher key agreed upon by Alice and Bob is $(24\ 75\ A2\ B3\ 34\ 75\ 56\ 88\ 31\ E2\ 12\ 00\ 13\ AA\ 54\ 87)_{16}$.

Table 7.5 Key expansion example

Round	Values of t 's	First word in the round	Second word in the round	Third word in the round	Fourth word in the round
—		$w_{00} = 2475A2B3$	$w_{01} = 34755688$	$w_{02} = 31E21200$	$w_{03} = 13AA5487$
1	AD20177D	$w_{04} = 8955B5CE$	$w_{05} = BD20E346$	$w_{06} = 8CC2F146$	$w_{07} = 9F68A5C1$
2	470678DB	$w_{08} = CE53CD15$	$w_{09} = 73732E53$	$w_{10} = FFB1DF15$	$w_{11} = 60D97AD4$
3	31DA48D0	$w_{12} = FF8985C5$	$w_{13} = 8CFAAB96$	$w_{14} = 734B7483$	$w_{15} = 2475A2B3$
4	47AB5B7D	$w_{16} = B822deb8$	$w_{17} = 34D8752E$	$w_{18} = 479301AD$	$w_{19} = 54010FFA$
5	6C762D20	$w_{20} = D454F398$	$w_{21} = E08C86B6$	$w_{22} = A71F871B$	$w_{23} = F31E88E1$
6	52C4F80D	$w_{24} = 86900B95$	$w_{25} = 661C8D23$	$w_{26} = C1030A38$	$w_{27} = 321D82D9$
7	E4133523	$w_{28} = 62833EB6$	$w_{29} = 049FB395$	$w_{30} = C59CB9AD$	$w_{31} = F7813B74$
8	8CE29268	$w_{32} = EE61ACDE$	$w_{33} = EAFFE1F4B$	$w_{34} = 2F62A6E6$	$w_{35} = D8E39D92$
9	0A5E4F61	$w_{36} = E43FE3BF$	$w_{37} = 0EC1FCF4$	$w_{38} = 21A35A12$	$w_{39} = F940C780$
10	3FC6CD99	$w_{40} = DBF92E26$	$w_{41} = D538D2D2$	$w_{42} = F49B88C0$	$w_{43} = 0DDDB4F40$

7.3.1 Continue

Example 7.7

*Each round key in AES depends on the previous round key. The dependency, however, is **nonlinear** because of SubWord transformation. The addition of the round constants also guarantees that each round key will be different from the previous one.*

Example 7.8

The two sets of round keys can be created from two cipher keys that are different only in one bit.

Cipher Key 1: 12 45 A2 A1 23 31 A4 A3 B2 CC AA 34 C2 BB 77 23

Cipher Key 2: 12 45 A2 A1 23 31 A4 A3 B2 CC AB 34 C2 BB 77 23

7.3.1 Continue

Example 7.8 Continue

Table 7.6 Comparing two sets of round keys

R.	Round keys for set 1	Round keys for set 2	B. D.
—	1245A2A1 2331A4A3 B2CCAA <u>3</u> 4 C2BB7723	1245A2A1 2331A4A3 B2CC <u>A</u> B34 C2BB7723	01
1	F9B08484 DA812027 684D8 <u>A</u> 13 AAF6 <u>FD</u> 30	F9B08484 DA812027 684D8 <u>B</u> 13 AAF6 <u>FC</u> 30	02
2	B9E48028 6365A00F 0B282A1C A1DED72C	B9008028 6381A00F 0BCC2B1C A13AD72C	17
3	A0EAF11A C38F5115 C8A77B09 6979AC25	3D0EF11A 5E8F5115 55437A09 F479AD25	30
4	1E7BCEE3 DDF49FF6 1553E4FF 7C2A48DA	839BCEA5 DD149FB0 8857E5B9 7C2E489C	31
5	EB2999F3 36DD0605 238EE2FA 5FA4AA20	A2C910B5 7FDD8F05 F78A6ABC 8BA42220	34
6	82852E3C B4582839 97D6CAC3 C87260E3	CB5AA788 B487288D 430D4231 C8A96011	56
7	82553FD4 360D17ED A1DBDD2E 69A9BD D	588A2560 EC0D0DED AF004FDC 67A92FCD	50
8	D12F822D E72295C0 46F948EE 2F50F523	0B9F98E5 E7929508 4892DAD4 2F3BF519	44
9	99C9A438 7EEB31F8 38127916 17428C35	F2794CF0 15EBD9F8 5D79032C 7242F635	51
10	83AD32C8 FD460330 C5547A26 D216F613	E83BDAB0 FDD00348 A0A90064 D2EBF651	52

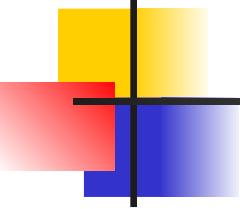
7.3.1 Continue

Example 7.9

The concept of weak keys, as we discussed for DES in INS class (+ Chapter 6 of the book), does not apply to AES. Assume that all bits in the cipher key are 0s. The following shows the words for some rounds:

Pre-round:	00000000	00000000	00000000	00000000
Round 01:	62636363	62636363	62636363	62636363
Round 02:	9B9898C9	F9FBFBAA	9B9898C9	F9FBFBAA
Round 03:	90973450	696CCFFA	F2F45733	0B0FAC99
...
Round 10:	B4EF5BCB	3E92E211	23E951CF	6F8F188E

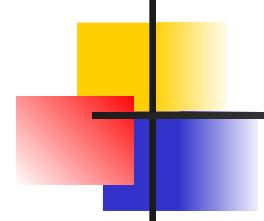
The words in the pre-round and the first round are all the same. In the second round, the first word matches with the third; the second word matches with the fourth. However, after the second round the pattern disappears; every word is different.



7.3.2 Key Expansion in AES-192

Key expansion algorithms in AES-192 and AES-256 versions are very similar to the key expansion algorithm in AES-128.

- 1. In AES-192, the words are generated in groups of six instead of four.**
 - a. The cipher key creates the first six words (w_0 to w_5).**
 - b. If $i \bmod 6 \neq 0$, $w_i = w_{i-1} + w_{i-6}$; otherwise $w_i = t + w_{i-6}$**



7.3.2 Key Expansion in AES-256

2. In AES-256, the words are generated in groups of eight instead of four.
 - a. the cipher key creates the first eight words (w_0 to w_7).
 - b. If $i \bmod 8 \neq 0$, $w_i = w_{i-1} + w_{i-8}$; otherwise $w_i = t + w_{i-8}$.
 - c. If $i \bmod 4 = 0$, but $i \bmod 8 \neq 0$, then $w_i = \text{SubWord}(w_{i-1}) + w_{i-8}$.

7-4 CIPHERS

AES uses four types of transformations for encryption and decryption. In the standard, the encryption algorithm is referred to as the cipher and the decryption algorithm as the inverse cipher.

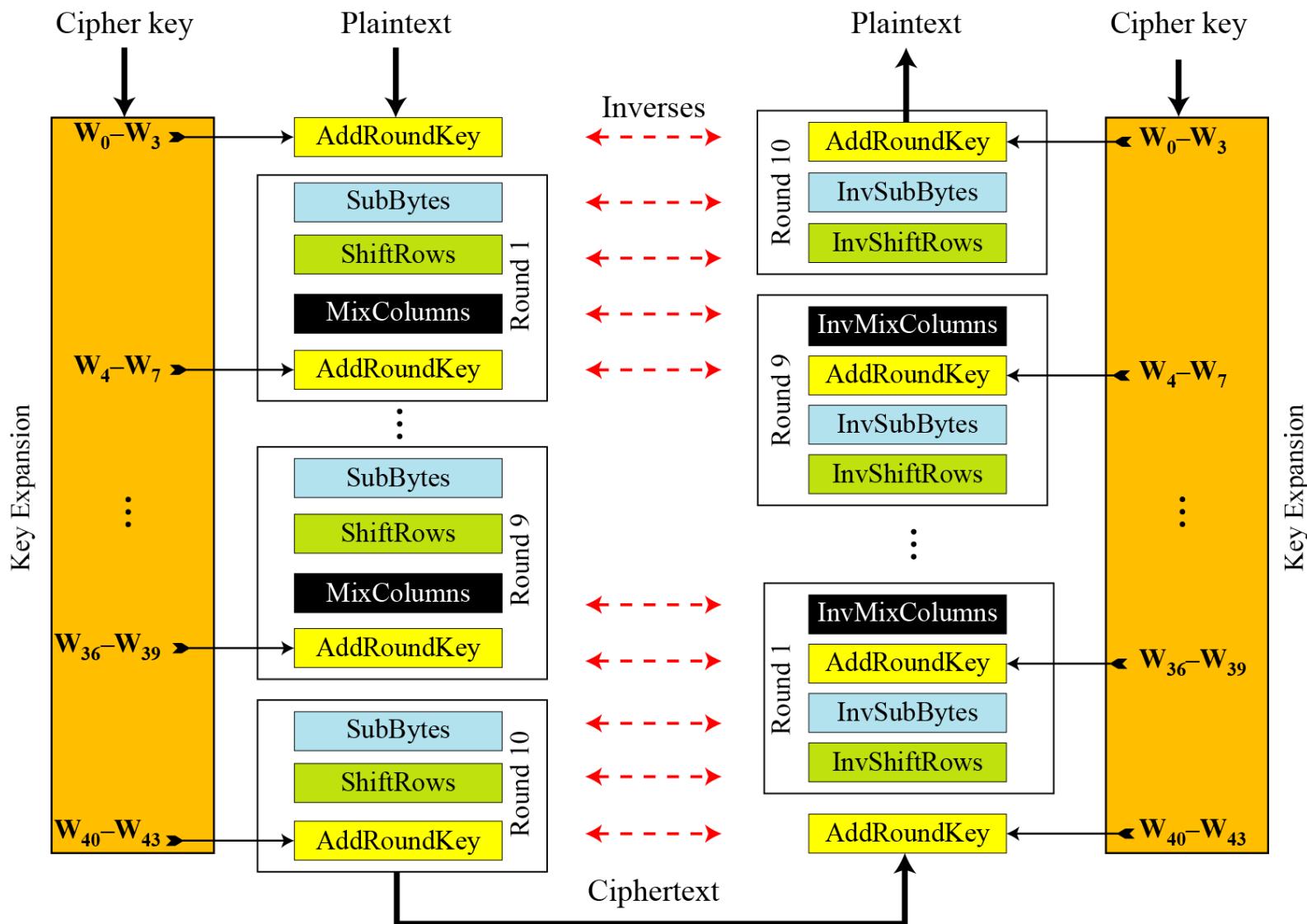
Topics discussed in this section:

7.4.1 Original Design

7.4.2 Alternative Design

7.4.1 Original Design

Figure 7.17 Ciphers and inverse ciphers of the original design



7.4.1 Continue

Algorithm

The code for the AES-128 version of this design is shown in Algorithm 7.6.

Algorithm 7.6 Pseudocode for cipher in the original design

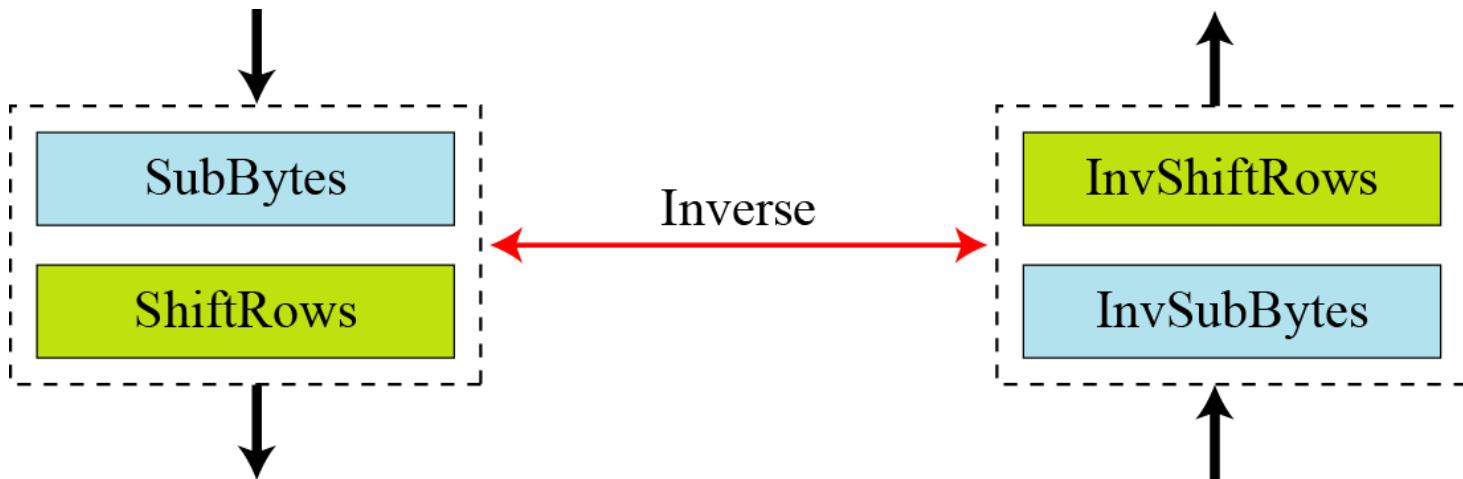
```
Cipher (InBlock [16], OutBlock[16], w[0 ... 43])
{
    BlockToState (InBlock, S)

    S ← AddRoundKey (S, w[0...3])
    for (round = 1 to 10)
    {
        S ← SubBytes (S)
        S ← ShiftRows (S)
        if (round ≠ 10)  S ← MixColumns (S)
        S ← AddRoundKey (S, w[4 × round, 4 × round + 3])
    }

    StateToBlock (S, OutBlock);
}
```

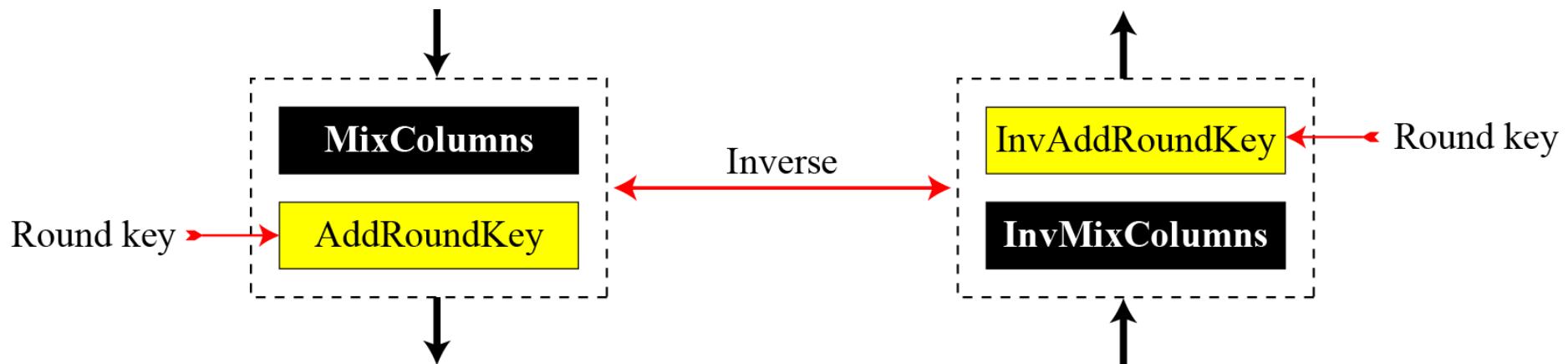
7.4.2 Alternative Design

Figure 7.18 Invertibility of SubBytes and ShiftRows combinations



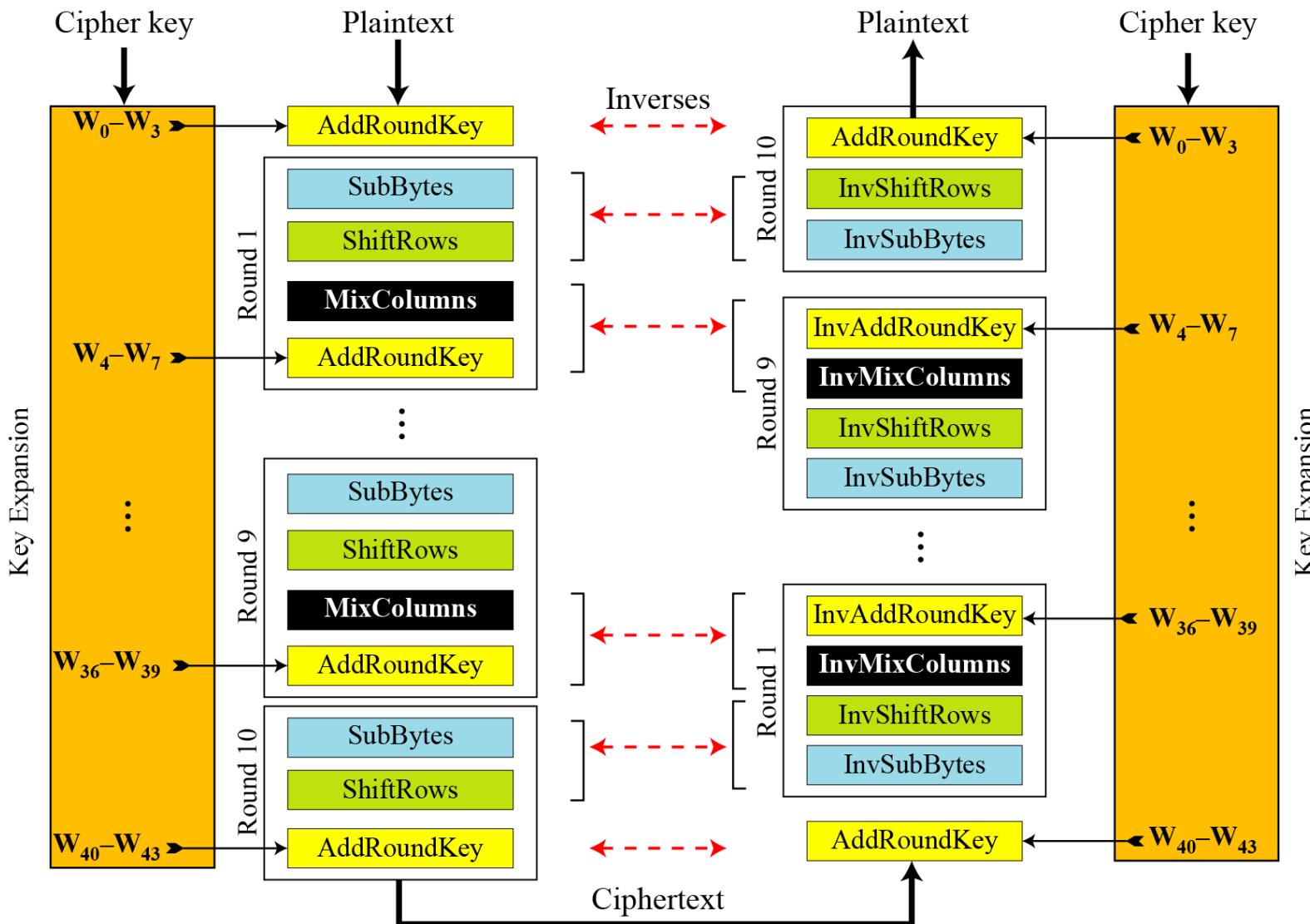
7.4.2 Continue

Figure 7.19 Invertibility of MixColumns and AddRoundKey combination



7.4.2 Continue

Figure 7.20 Cipher and reverse cipher in alternate design



Changing Key-Expansion Algorithm

Instead of using InvRoundKey transformation in the reverse cipher, the key-expansion algorithm can be changed to create a different set of round keys for the inverse cipher.

7-5 Examples

In this section, some examples of encryption/decryption and key generation are given to emphasize some points discussed in the two previous sections.

Example 7.10

The following shows the ciphertext block created from a plaintext block using a randomly selected cipher key.

Plaintext:	00	04	12	14	12	04	12	00	0C	00	13	11	08	23	19	19
Cipher Key:	24	75	A2	B3	34	75	56	88	31	E2	12	00	13	AA	54	87
Ciphertext:	BC	02	8B	D3	E0	E3	B1	95	55	0D	6D	FB	E6	F1	82	41

Example 7.10 *Continued*

Table 7.7 Example of encryption

Round	Input State				Output State				Round Key			
Pre-round	00	12	0C	08	24	26	3D	1B	24	34	31	13
	04	04	00	23	71	71	E2	89	75	75	E2	AA
	12	12	13	19	B0	44	01	4D	A2	56	12	54
	14	00	11	19	A7	88	11	9E	B3	88	00	87
1	24	26	3D	1B	6C	44	13	BD	89	BD	8C	9F
	71	71	E2	89	B1	9E	46	35	55	20	C2	68
	B0	44	01	4D	C5	B5	F3	02	B5	E3	F1	A5
	A7	88	11	9E	5D	87	FC	8C	CE	46	46	C1
2	6C	44	13	BD	1A	90	15	B2	CE	73	FF	60
	B1	9E	46	35	66	09	1D	FC	53	73	B1	D9
	C5	B5	F3	02	20	55	5A	B2	CD	2E	DF	7A
	5D	87	FC	8C	2B	CB	8C	3C	15	53	15	D4

7-5 *Continued*

Example 7.10 *Continued*

3	1A 90 15 B2 66 09 1D FC 20 55 5A B2 2B CB 8C 3C	F6 7D A2 B0 1B 61 B4 B8 67 09 C9 45 4A 5C 51 09	FF 8C 73 13 89 FA 4B 92 85 AB 74 0E C5 96 83 57
4	F6 7D A2 B0 1B 61 B4 B8 67 09 C9 45 4A 5C 51 09	CA E5 48 BB D8 42 AF 71 D1 BA 98 2D 4E 60 9E DF	B8 34 47 54 22 D8 93 01 DE 75 01 OF B8 2E AD FA
5	CA E5 48 BB D8 42 AF 71 D1 BA 98 2D 4E 60 9E DF	90 35 13 60 2C FB 82 3A 9E FC 61 ED 49 39 CB 47	D4 E0 A7 F3 54 8C 1F 1E F3 86 87 88 98 B6 1B E1
6	90 35 13 60 2C FB 82 3A 9E FC 61 ED 49 39 CB 47	18 0A B9 B5 64 68 6A FB 5A EF D7 79 8E B2 10 4D	86 66 C1 32 90 1C 03 1D 0B 8D 0A 82 95 23 38 D9

7-5 *Continued*

Example 7.10 *Continued*

7	18 0A B9 B5 64 68 6A FB 5A EF D7 79 8E B2 10 4D	01 63 F1 96 55 24 3A 62 F4 8A DE 4D CC BA 88 03	62 04 C5 F7 83 9F 9C 81 3E B3 B9 3B B6 95 AD 74
8	01 63 F1 96 55 24 3A 62 F4 8A DE 4D CC BA 88 03	2A 34 D8 46 2D 6B A2 D6 51 64 CF 5A 87 A8 F8 28	EE EA 2F D8 61 FE 62 E3 AC 1F A6 9D DE 4B E6 92
9	2A 34 D8 46 2D 6B A2 D6 51 64 CF 5A 87 A8 F8 28	0A D9 F1 3C 95 63 9F 35 2A 80 29 00 16 76 09 77	E4 0E 21 F9 3F C1 A3 40 E3 FC 5A C7 BF F4 12 80
10	0A D9 F1 3C 95 63 9F 35 2A 80 29 00 16 76 09 77	BC E0 55 E6 02 E3 0D F1 8B B1 6D 82 D3 95 F8 41	DB D5 F4 0D F9 38 9B DB 2E D2 88 4F 26 D2 C0 40

Example 7.11

Figure 7.21 shows the state entries in one round, round 7, in Example 7.10.

Figure 7.21 States in a single round



Example 7.12

One may be curious to see the result of encryption when the plaintext is made of all 0s. Using the cipher key in Example 7.10 yields the ciphertext.

Plaintext:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00			
Cipher Key:	24	75	A2	B3	34	75	56	88	31	E2	12	00	13	AA	54	87														
Ciphertext:	63	2C	D4	5E	5D	56	ED	B5	62	04	01	A0	AA	9C	2D	8D														

Example 7.13

Let us check the avalanche effect that we discussed in Chapter 6. Let us change only one bit in the plaintext and compare the results. We changed only one bit in the last byte. The result clearly shows the effect of diffusion and confusion. Changing a single bit in the plaintext has affected many bits in the ciphertext.

Plaintext 1: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00

Plaintext 2: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 01

Ciphertext 1: 63 2C D4 5E 5D 56 ED B5 62 04 01 A0 AA 9C 2D 8D

Ciphertext 2: 26 F3 9B BC A1 9C 0F B7 C7 2E 7E 30 63 92 73 13

Example 7.14

The following shows the effect of using a cipher key in which all bits are 0s.

Plaintext:	00	04	12	14	12	04	12	00	0c	00	13	11	08	23	19	19
Cipher Key:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
Ciphertext:	5A	6F	4B	67	57	B7	A5	D2	C4	30	91	ED	64	9A	42	72

7-6 ANALYSIS OF AES

This section is a brief review of the three characteristics of AES.

Topics discussed in this section:

- 7.6.1 Security**
- 7.6.2 Implementation**
- 7.6.3 Simplicity and Cost**

AES was designed after DES. Most of the known attacks on DES were already tested on AES.

Brute-Force Attack

AES is definitely more secure than DES due to the larger-size key.

Statistical Attacks

Numerous tests have failed to do statistical analysis of the ciphertext.

Differential and Linear Attacks

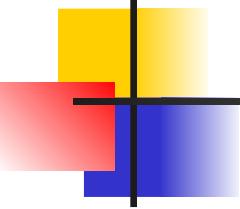
There are no differential and linear attacks on AES as yet.

Statistical Attacks

Numerous tests have failed to do statistical analysis of the ciphertext.

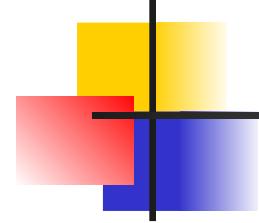
Differential and Linear Attacks

There are no differential and linear attacks on AES as yet.



7.6.2 Implementation

AES can be implemented in software, hardware, and firmware. The implementation can use table lookup process or routines that use a well-defined algebraic structure.



7.6.3 Simplicity and Cost

The algorithms used in AES are so simple that they can be easily implemented using cheap processors and a minimum amount of memory.

Chapter 9

Mathematics of Cryptography

*Part III: Primes and Related
Congruence Equations*

Chapter 9

Objectives

- ❑ To introduce prime numbers and their applications in cryptography.
- ❑ To discuss some primality test algorithms and their efficiencies.
- ❑ To discuss factorization algorithms and their applications in cryptography.
- ❑ To describe the Chinese remainder theorem and its application.
- ❑ To introduce quadratic congruence.
- ❑ To introduce modular exponentiation and logarithm.

9-1 PRIMES

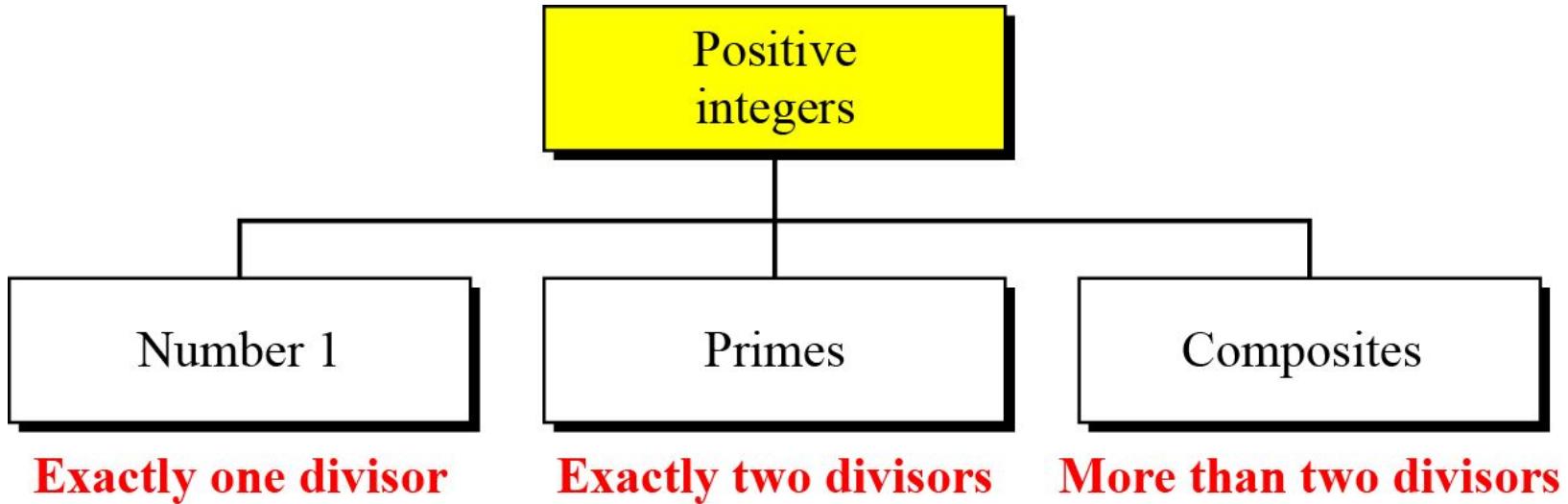
Asymmetric-key cryptography uses primes extensively. The topic of primes is a large part of any book on number theory. This section discusses only a few concepts and facts to pave the way for Chapter 10.

Topics discussed in this section:

- 9.1.1 Definition**
- 9.1.2 Cardinality of Primes**
- 9.1.3 Checking for Primeness**
- 9.1.4 Euler's Phi-Function**
- 9.1.5 Fermat's Little Theorem**
- 9.1.6 Euler's Theorem**
- 9.1.7 Generating Primes**

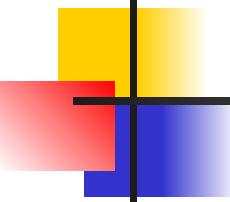
9.1.1 Definition

Figure 9.1 *Three groups of positive integers*



Note

A prime is divisible only by itself and 1.



9.1.1 *Continued*

Example 9.1

What is the smallest prime?

Solution

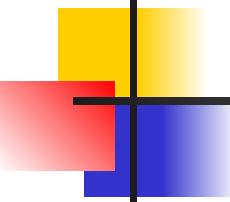
The smallest prime is 2, which is divisible by 2 (itself) and 1.

Example 9.2

List the primes smaller than 10.

Solution

There are four primes less than 10: 2, 3, 5, and 7. It is interesting to note that the percentage of primes in the range 1 to 10 is 40%. The percentage decreases as the range increases.



9.1.2 *Cardinality of Primes*

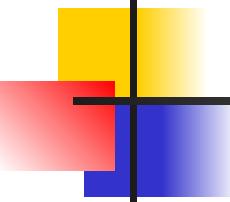
Infinite Number of Primes

Note

There is an infinite number of primes.

Number of Primes

$$[n / (\ln n)] < \pi(n) < [n / (\ln n - 1.08366)]$$



9.1.2 *Continued*

Example 9.3

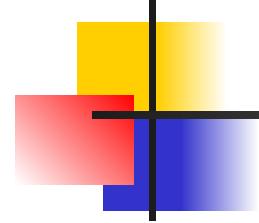
As a trivial example, assume that the only primes are in the set $\{2, 3, 5, 7, 11, 13, 17\}$. Here $P = 510510$ and $P + 1 = 510511$. However, $510511 = 19 \times 97 \times 277$; none of these primes were in the original list. Therefore, there are three primes greater than 17.

Example 9.4

Find the number of primes less than 1,000,000.

Solution

The approximation gives the range 72,383 to 78,543. The actual number of primes is 78,498.



9.1.3 Checking for Primeness

*Given a number n , how can we determine if n is a prime?
The answer is that we need to see if the number is
divisible by all primes less than*

$$\sqrt{n}$$

We know that this method is inefficient, but it is a good start.

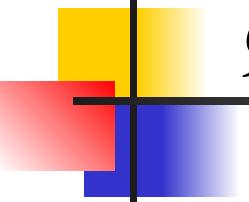
Theorem

If n is composite, then n has a prime divisor less than or equal to \sqrt{n} .

Proof.

- Let $n = ab$, $1 < a < n$, $1 < b < n$.
- We can't have both $a > \sqrt{n}$ and $b > \sqrt{n}$ since this would lead to $ab > n$.
- Therefore, n must have a prime divisor less than or equal to \sqrt{n} .





9.1.3 *Continued*

Example 9.5

Is 97 a prime?

Solution

The floor of $\sqrt{97} = 9$. The primes less than 9 are 2, 3, 5, and 7. We need to see if 97 is divisible by any of these numbers. It is not, so 97 is a prime.

Example 9.6

Is 301 a prime?

Solution

The floor of $\sqrt{301} = 17$. We need to check 2, 3, 5, 7, 11, 13, and 17. The numbers 2, 3, and 5 do not divide 301, but 7 does. Therefore 301 is not a prime.

9.1.3 *Continued*

Sieve of Eratosthenes

Table 9.1 Sieve of Eratosthenes

	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

9.1.4 Euler's Phi-Function

Euler's phi-function, $\phi(n)$, which is sometimes called the Euler's totient function plays a very important role in cryptography.

1. $\phi(1) = 0$.
2. $\phi(p) = p - 1$ if p is a prime.
3. $\phi(m \times n) = \phi(m) \times \phi(n)$ if m and n are relatively prime.
4. $\phi(p^e) = p^e - p^{e-1}$ if p is a prime.

9.1.4 Continued

We can combine the above four rules to find the value of $\phi(n)$. For example, if n can be factored as

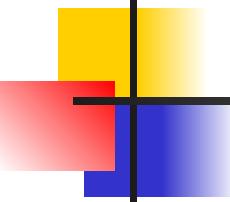
$$n = p_1^{e_1} \times p_2^{e_2} \times \dots \times p_k^{e_k}$$

then we combine the third and the fourth rule to find

$$\phi(n) = (p_1^{e_1} - p_1^{e_1-1}) \times (p_2^{e_2} - p_2^{e_2-1}) \times \dots \times (p_k^{e_k} - p_k^{e_k-1})$$

Note

The difficulty of finding $\phi(n)$ depends on the difficulty of finding the factorization of n .



9.1.4 *Continued*

Example 9.7

What is the value of $\phi(13)$?

Solution

Because 13 is a prime, $\phi(13) = (13 - 1) = 12$.

Example 9.8

What is the value of $\phi(10)$?

Solution

We can use the third rule: $\phi(10) = \phi(2) \times \phi(5) = 1 \times 4 = 4$, because 2 and 5 are primes.

9.1.4 *Continued*

Example 9.9

What is the value of $\phi(240)$?

Solution

We can write $240 = 2^4 \times 3^1 \times 5^1$. Then

$$\phi(240) = (2^4 - 2^3) \times (3^1 - 3^0) \times (5^1 - 5^0) = 64$$

Example 9.10

Can we say that $\phi(49) = \phi(7) \times \phi(7) = 6 \times 6 = 36$?

Solution

No. The third rule applies when m and n are relatively prime. Here $49 = 7^2$. We need to use the fourth rule: $\phi(49) = 7^2 - 7^1 = 42$.

9.1.4 *Continued*

Example 9.11

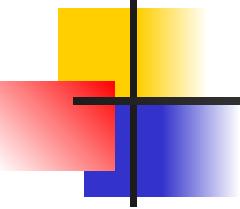
What is the number of elements in Z_{14}^* ?

Solution

The answer is $\phi(14) = \phi(7) \times \phi(2) = 6 \times 1 = 6$. The members are 1, 3, 5, 9, 11, and 13.

Note

Interesting point: If $n > 2$, the value of $\phi(n)$ is even.



9.1.5 Fermat's Little Theorem

First Version

$$a^{p-1} \equiv 1 \pmod{p}$$

Second Version

$$a^p \equiv a \pmod{p}$$

9.1.5 *Continued*

Example 9.12

Find the result of $6^{10} \bmod 11$.

Solution

We have $6^{10} \bmod 11 = 1$. This is the first version of Fermat's little theorem where $p = 11$.

Example 9.13

Find the result of $3^{12} \bmod 11$.

Solution

Here the exponent (12) and the modulus (11) are not the same. With substitution this can be solved using Fermat's little theorem.

$$3^{12} \bmod 11 = (3^{11} \times 3) \bmod 11 = (3^{11} \bmod 11)(3 \bmod 11) = (3 \times 3) \bmod 11 = 9$$

9.1.5 *Continued*

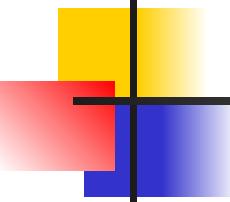
Multiplicative Inverses

$$a^{-1} \bmod p = a^{p-2} \bmod p$$

Example 9.14

The answers to multiplicative inverses modulo a prime can be found without using the extended Euclidean algorithm:

- a. $8^{-1} \bmod 17 = 8^{17-2} \bmod 17 = 8^{15} \bmod 17 = 15 \bmod 17$
- b. $5^{-1} \bmod 23 = 5^{23-2} \bmod 23 = 5^{21} \bmod 23 = 14 \bmod 23$
- c. $60^{-1} \bmod 101 = 60^{101-2} \bmod 101 = 60^{99} \bmod 101 = 32 \bmod 101$
- d. $22^{-1} \bmod 211 = 22^{211-2} \bmod 211 = 22^{209} \bmod 211 = 48 \bmod 211$



9.1.6 Euler's Theorem

First Version

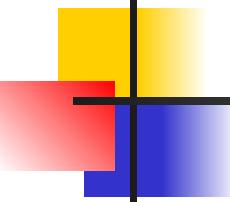
$$a^{\varphi(n)} \equiv 1 \pmod{n}$$

Second Version

$$a^{k \times \varphi(n) + 1} \equiv a \pmod{n}$$

Note

The second version of Euler's theorem is used in the RSA cryptosystem in Chapter 10.



9.1.5 *Continued*

Example 9.15

Find the result of $6^{24} \bmod 35$.

Solution

We have $6^{24} \bmod 35 = 6^{\phi(35)} \bmod 35 = 1$.

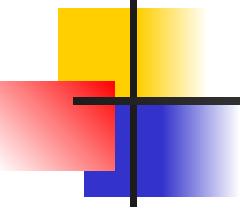
Example 9.16

Find the result of $20^{62} \bmod 77$.

Solution

If we let $k = 1$ on the second version, we have

$$\begin{aligned}20^{62} \bmod 77 &= (20 \bmod 77) (20^{\phi(77)+1} \bmod 77) \bmod 77 \\&= (20)(20) \bmod 77 = 15.\end{aligned}$$

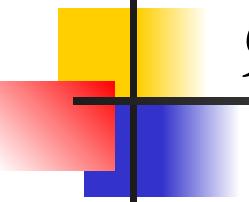


9.1.6 Continued

Multiplicative Inverses

Euler's theorem can be used to find multiplicative inverses modulo a composite.

$$a^{-1} \bmod n = a^{\phi(n)-1} \bmod n$$



9.1.5 *Continued*

Example 9.17

The answers to multiplicative inverses modulo a composite can be found without using the extended Euclidean algorithm if we know the factorization of the composite:

- a. $8^{-1} \text{ mod } 77 = 8^{\phi(77)-1} \text{ mod } 77 = 8^{59} \text{ mod } 77 = 29 \text{ mod } 77$
- b. $7^{-1} \text{ mod } 15 = 7^{\phi(15)-1} \text{ mod } 15 = 7^7 \text{ mod } 15 = 13 \text{ mod } 15$
- c. $60^{-1} \text{ mod } 187 = 60^{\phi(187)-1} \text{ mod } 187 = 60^{159} \text{ mod } 187 = 53 \text{ mod } 187$
- d. $71^{-1} \text{ mod } 100 = 71^{\phi(100)-1} \text{ mod } 100 = 71^{39} \text{ mod } 100 = 31 \text{ mod } 100$

9.1.7 Generating Primes

Mersenne Primes

$$M_p = 2^p - 1$$

$$M_2 = 2^2 - 1 = 3$$

$$M_3 = 2^3 - 1 = 7$$

$$M_5 = 2^5 - 1 = 31$$

$$M_7 = 2^7 - 1 = 127$$

$$M_{11} = 2^{11} - 1 = 2047$$

Not a prime ($2047 = 23 \times 89$)

$$M_{13} = 2^{13} - 1 = 8191$$

$$M_{17} = 2^{17} - 1 = 131071$$

Note

A number in the form $M_p = 2^p - 1$ is called a Mersenne number and may or may not be a prime.

9.1.7 Continued

Fermat Primes

$$F_n = 2^{2^n} + 1$$

$$\begin{aligned} F_0 &= 3 & F_1 &= 5 & F_2 &= 17 & F_3 &= 257 & F_4 &= 65537 \\ F_5 &= 4294967297 = 641 \times 6700417 \end{aligned} \quad \textcolor{red}{\text{Not a prime}}$$

9-2 PRIMALITY TESTING

Finding an algorithm to correctly and efficiently test a very large integer and output a prime or a composite has always been a challenge in number theory, and consequently in cryptography. However, recent developments look very promising.

Topics discussed in this section:

- 9.2.1 Deterministic Algorithms**
- 9.2.2 Probabilistic Algorithms**
- 9.2.3 Recommended Primality Test**

9.2.1 Deterministic Algorithms

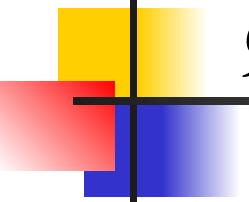
Divisibility Algorithm

Algorithm 9.1 Pseudocode for the divisibility test

```
Divisibility_Test ( $n$ ) //  $n$  is the number to test for primality
{
     $r \leftarrow 2$ 
    while ( $r < \sqrt{n}$ )
    {
        if ( $r \mid n$ ) return "a composite"
         $r \leftarrow r + 1$ 
    }
    return "a prime"
}
```

Note

The bit-operation complexity of the divisibility test is exponential.



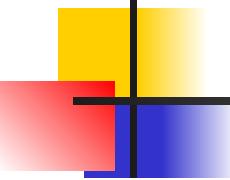
9.2.1 *Continued*

Example 9.18

Assume n has 200 bits. What is the number of bit operations needed to run the divisibility-test algorithm?

Solution

The bit-operation complexity of this algorithm is $2^{n_b/2}$. This means that the algorithm needs 2^{100} bit operations. On a computer capable of doing 2^{30} bit operations per second, the algorithm needs 2^{70} seconds to do the testing (**forever**).



9.2.1 *Continued*

AKS Algorithm

$$O((\log_2 n_b)^{12})$$

Example 9.19

Assume n has 200 bits. What is the number of bit operations needed to run the AKS algorithm?

Solution

This algorithm needs only $(\log_2 200)^{12} = 39,547,615,483$ bit operations. On a computer capable of doing 1 billion bit operations per second, the algorithm needs only 40 seconds.

9.2.2 Probabilistic Algorithms

Fermat Test

If n is a prime, then $a^{n-1} \equiv 1 \pmod{n}$.

If n is a prime, $a^{n-1} \equiv 1 \pmod{n}$

If n is a composite, it is possible that $a^{n-1} \equiv 1 \pmod{n}$

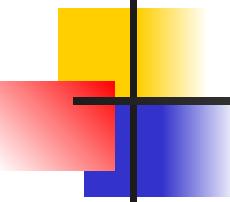
Example 9.20

Does the number 561 pass the Fermat test?

Solution

Use base 2

The number passes the Fermat test, but it is not a prime, because $561 = 33 \times 17$.



9.2.2 *Continued*

Example 9.20

Does the number 561 pass the Fermat test?

Solution

Use base 2

$$2^{561-1} = 1 \bmod 561$$

The number passes the Fermat test, but it is not a prime, because $561 = 33 \times 17$.

9.2.2 *Continued*

Square Root Test

If n is a prime, $\sqrt{1} \bmod n = \pm 1$.

If n is a composite, $\sqrt{1} \bmod n = \pm 1$ and possibly other values.

Example 9.21

What are the square roots of $1 \bmod n$ if n is 7 (a prime)?

Solution

The only square roots are 1 and -1 . We can see that

$$1^2 = 1 \bmod 7$$

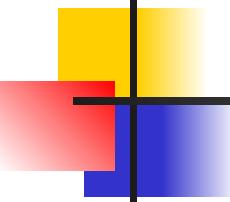
$$2^2 = 4 \bmod 7$$

$$3^2 = 2 \bmod 7$$

$$(-1)^2 = 1 \bmod 7$$

$$(-2)^2 = 4 \bmod 7$$

$$(-3)^2 = 2 \bmod 7$$



9.2.2 *Continued*

Example 9.21

What are the square roots of $1 \bmod n$ if n is 7 (a prime)?

Solution

The only square roots are 1 and -1 . We can see that

$$1^2 = 1 \bmod 7$$

$$(-1)^2 = 1 \bmod 7$$

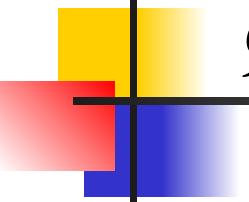
$$2^2 = 4 \bmod 7$$

$$(-2)^2 = 4 \bmod 7$$

$$3^2 = 2 \bmod 7$$

$$(-3)^2 = 2 \bmod 7$$

Note that we don't have to test 4, 5 and 6 because $4 = -3 \bmod 7$, $5 = -2 \bmod 7$ and $6 = -1 \bmod 7$.



9.2.2 *Continued*

Example 9.22

What are the square roots of $1 \bmod n$ if n is 8 (a composite)?

Solution

There are four solutions: 1, 3, 5, and 7 (which is -1). We can see that

$$1^2 = 1 \bmod 8$$

$$3^2 = 1 \bmod 8$$

$$(-1)^2 = 1 \bmod 8$$

$$5^2 = 1 \bmod 8$$

9.2.2 *Continued*

Example 9.23

What are the square roots of $1 \bmod n$ if n is 17 (a prime)?

Solution

There are only two solutions: 1 and -1

$$1^2 = 1 \bmod 17$$

$$2^2 = 4 \bmod 17$$

$$3^2 = 9 \bmod 17$$

$$4^2 = 16 \bmod 17$$

$$5^2 = 8 \bmod 17$$

$$6^2 = 2 \bmod 17$$

$$(7)^2 = 15 \bmod 17$$

$$(8)^2 = 13 \bmod 17$$

$$(-1)^2 = 1 \bmod 17$$

$$(-2)^2 = 4 \bmod 17$$

$$(-3)^2 = 9 \bmod 17$$

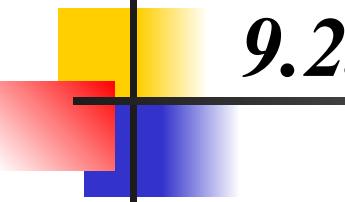
$$(-4)^2 = 16 \bmod 17$$

$$(-5)^2 = 8 \bmod 17$$

$$(-6)^2 = 2 \bmod 17$$

$$(-7)^2 = 15 \bmod 17$$

$$(-8)^2 = 13 \bmod 17$$



9.2.2 *Continued*

Example 9.24

What are the square roots of $1 \bmod n$ if n is 22 (a composite)?

Solution

Surprisingly, there are only two solutions, +1 and -1, although 22 is a composite.

$$\begin{aligned} 1^2 &= 1 \bmod 22 \\ (-1)^2 &= 1 \bmod 22 \end{aligned}$$

9.2.2 Continued

Miller-Rabin Test

$$n - 1 = m \times 2^k$$

Figure 9.2 Idea behind Fermat primality test

$$a^{n-1} = a^{m \times 2^k} = [a^m]^{2^k} = [a^m]^{\underbrace{2 \cdot 2 \cdot \dots \cdot 2}_{\text{k times}}}$$

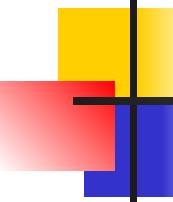
Note

The Miller-Rabin test needs from step 0 to step $k - 1$.

9.2.2 *Continued*

Algorithm 9.2 Pseudocode for Miller-Rabin test

```
Miller_Rabin_Test ( $n, a$ ) //  $n$  is the number;  $a$  is the base.  
{  
    Find  $m$  and  $k$  such that  $n - 1 = m \times 2^k$   
     $T \leftarrow a^m \bmod n$   
    if ( $T = \pm 1$ ) return "a prime"  
    for ( $i \leftarrow 1$  to  $k - 1$ ) //  $k - 1$  is the maximum number of steps.  
    {  
         $T \leftarrow T^2 \bmod n$   
        if ( $T = +1$ ) return "a composite"  
        if ( $T = -1$ ) return "a prime"  
    }  
    return "a composite"  
}
```



9.2.2 *Continued*

Example 9.25

Does the number 561 pass the Miller-Rabin test?

Solution

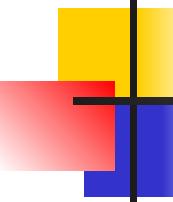
Using base 2, let $561 - 1 = 35 \times 2^4$, which means $m = 35$, $k = 4$, and $a = 2$.

Initialization: $T = 2^{35} \bmod 561 = 263 \bmod 561$

$k = 1$: $T = 263^2 \bmod 561 = 166 \bmod 561$

$k = 2$: $T = 166^2 \bmod 561 = 67 \bmod 561$

$k = 3$: $T = 67^2 \bmod 561 = +1 \bmod 561$ → a composite



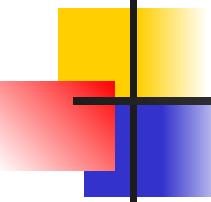
9.2.2 *Continued*

Example 9.26

We already know that 27 is not a prime. Let us apply the Miller-Rabin test.

Solution

With base 2, let $27 - 1 = 13 \times 2^1$, which means that $m = 13$, $k = 1$, and $a = 2$. In this case, because $k - 1 = 0$, we should do only the initialization step: $T = 2^{13} \bmod 27 = 11 \bmod 27$. However, because the algorithm never enters the loop, it returns a composite.



9.2.2 *Continued*

Example 9.27

We know that 61 is a prime, let us see if it passes the Miller-Rabin test.

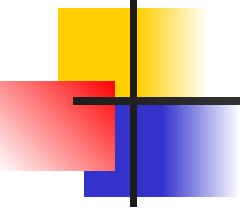
Solution

We use base 2.

$$61 - 1 = 15 \times 2^2 \rightarrow m = 15 \quad k = 2 \quad a = 2$$

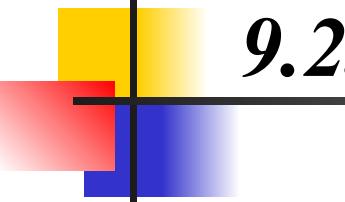
$$\text{Initialization: } T = 2^{15} \bmod 61 = 11 \bmod 61$$

$$k = 1 \qquad \qquad T = 11^2 \bmod 61 = -1 \bmod 61 \quad \rightarrow \text{ a prime}$$



9.2.3 Recommended Primality Test

Today, one of the most popular primality test is a combination of the divisibility test and the Miller-Rabin test.



9.2.3 *Continued*

Example 9.28

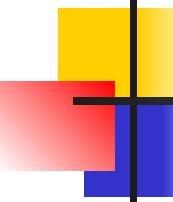
The number 4033 is a composite (37×109). Does it pass the recommended primality test?

Solution

1. Perform the divisibility tests first. The numbers 2, 3, 5, 7, 11, 17, and 23 are not divisors of 4033.
2. Perform the Miller-Rabin test with a base of 2, $4033 - 1 = 63 \times 26$, which means m is 63 and k is 6.

Initialization: $T \equiv 2^{63} \pmod{4033} \equiv 3521 \pmod{4033}$

$k = 1$ $T \equiv T^2 \equiv 3521^2 \pmod{4033} \equiv -1 \pmod{4033} \rightarrow \text{Passes}$



9.2.3 *Continued*

Example 9.28 Continued

3. But we are not satisfied. We continue with another base, 3.

Initialization: $T \equiv 3^{63} \pmod{4033} \equiv 3551 \pmod{4033}$

$$k = 1 \quad T \equiv T^2 \equiv 3551^2 \pmod{4033} \equiv 2443 \pmod{4033}$$

$$k = 2 \quad T \equiv T^2 \equiv 2443^2 \pmod{4033} \equiv 3442 \pmod{4033}$$

$$k = 3 \quad T \equiv T^2 \equiv 3442^2 \pmod{4033} \equiv 2443 \pmod{4033}$$

$$k = 4 \quad T \equiv T^2 \equiv 2443^2 \pmod{4033} \equiv 3442 \pmod{4033}$$

$$k = 5 \quad T \equiv T^2 \equiv 3442^2 \pmod{4033} \equiv 2443 \pmod{4033} \rightarrow \text{Failed (composite)}$$

9-3 FACTORIZATION

Factorization has been the subject of continuous research in the past; such research is likely to continue in the future. Factorization plays a very important role in the security of several public-key cryptosystems (see Chapter 10).

Topics discussed in this section:

- 9.3.1 Fundamental Theorem of Arithmetic**
- 9.3.2 Factorization Methods**
- 9.3.3 Fermat Method**
- 9.3.4 Pollard $p - 1$ Method**
- 9.3.5 Pollard rho Method**
- 9.3.6 More Efficient Methods**

9.3.1 Fundamental Theorem of Arithmetic

$$n = p_1^{e1} \times p_2^{e2} \times \dots \times p_k^{ek}$$

Greatest Common Divisor

$$a = p_1^{a1} \times p_2^{a2} \times \dots \times p_k^{ak}$$

$$b = p_1^{b1} \times p_2^{b2} \times \dots \times p_k^{bk}$$

$$\gcd(a, b) = p_1^{\min(a_1, b_1)} \times p_2^{\min(a_2, b_2)} \times \dots \times p_k^{\min(a_k, b_k)}$$

Least Common Multiplier

$$a = p_1^{a1} \times p_2^{a2} \times \dots \times p_k^{ak}$$

$$b = p_1^{b1} \times p_2^{b2} \times \dots \times p_k^{bk}$$

$$\text{lcm}(a, b) = p_1^{\max(a_1, b_1)} \times p_2^{\max(a_2, b_2)} \times \dots \times p_k^{\max(a_k, b_k)}$$

$$\text{lcm}(a, b) \times \gcd(a, b) = a \times b$$

9.3.2 Factorization Methods

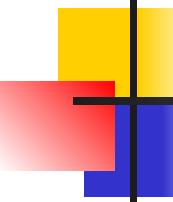
Trial Division Method

Algorithm 9.3 *Pseudocode for trial-division factorization*

```

Trial_Division_Factorization (n) // n is the number to be factored
{
    a ← 2
    while ( $a \leq \sqrt{n}$ )
    {
        while ( $(n \text{ mod } a = 0)$ )
        {
            output a // Factors are output one by one
             $n = n / a$ 
        }
        a ← a + 1
    }
    if ( $(n > 1)$ ) output n // n has no more factors
}

```



9.3.2 *Continued*

Example 9.29

Use the trial division algorithm to find the factors of 1233.

Solution

We run a program based on the algorithm and get the following result.

$$1233 = 3^2 \times 137$$

Example 9.30

Use the trial division algorithm to find the factors of 1523357784.

Solution

We run a program based on the algorithm and get the following result.

$$1523357784 = 2^3 \times 3^2 \times 13 \times 37 \times 43987$$

9.3.3 Fermat Method

$$n = x^2 - y^2 = a \times b \quad \text{with } a = (x + y) \text{ and } b = (x - y)$$

Algorithm 9.4 Pseudocode for Fermat factorization

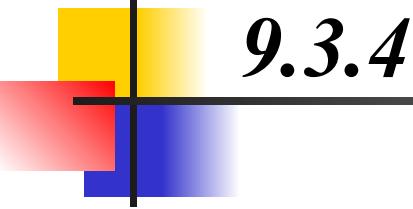
```
Feramat_Factorization (n)                                // n is the number to be factored
{
    x ← √n                                              // smallest integer greater than √n
    while (x < n)
    {
        w ← x2 - n
        if(w is perfect square)  y ← √w;  a ← x+y;  b ← x-y;  return a and b
        x ← x + 1
    }
}
```

9.3.4 Pollard $p - 1$ Method

$$p = \gcd(2^{B!} - 1, n)$$

Algorithm 9.5 Pseudocode for Pollard $p - 1$ factorization

```
Pollard_(p - 1)_Factorization (n, B) // n is the number to be factored
{
    a ← 2
    e ← 2
    while (e ≤ B)
    {
        a ←  $a^e \text{ mod } n$ 
        e ← e + 1
    }
    p ← gcd (a - 1, n)
    if  $1 < p < n$  return p
    return failure
}
```



9.3.4 *Continued*

Example 9.31

Use the Pollard $p - 1$ method to find a factor of 57247159 with the bound $B = 8$.

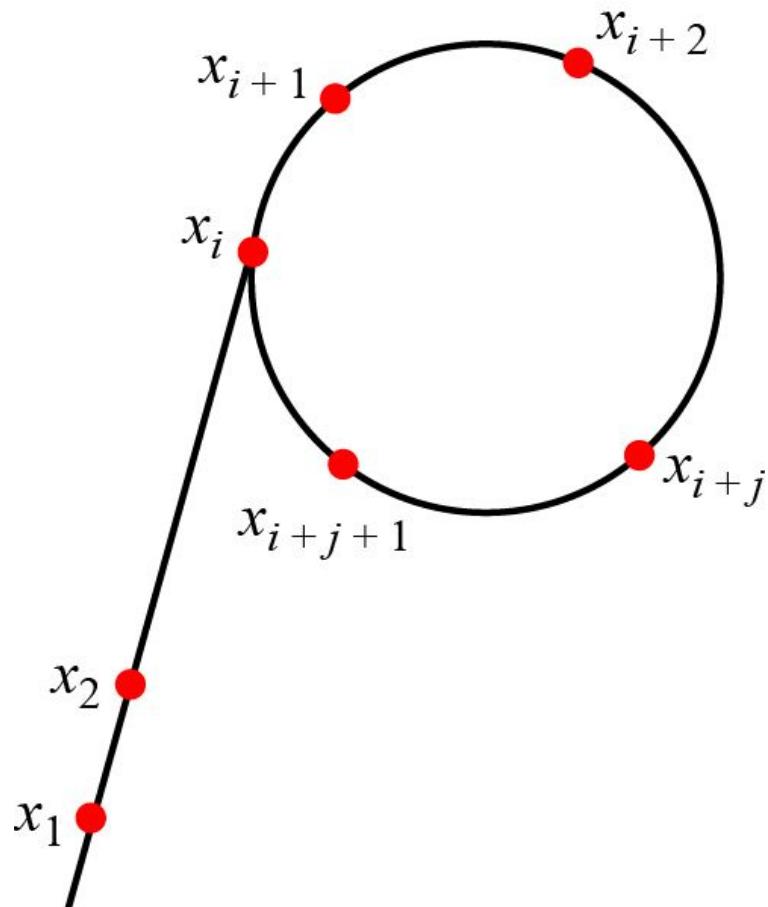
Solution

We run a program based on the algorithm and find that $p = 421$. As a matter of fact $57247159 = 421 \times 135979$. Note that 421 is a prime and $p - 1$ has no factor greater than 8

$$421 - 1 = 2^2 \times 3 \times 5 \times 7$$

9.3.5 Pollard rho Method

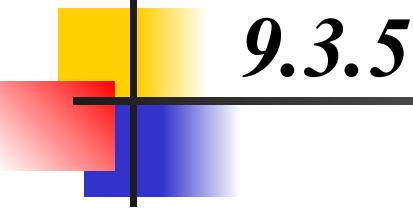
Figure 9.3 Pollard rho successive numbers



9.3.5 Continued

Algorithm 9.6 *Pseudocode for Pollard rho method*

```
Pollard_rho_Factorization ( $n, B$ ) //  $n$  is the number to be factored
{
     $x \leftarrow 2$ 
     $y \leftarrow 2$ 
     $p \leftarrow 1$ 
    while ( $p = 1$ )
    {
         $x \leftarrow f(x) \bmod n$ 
         $y \leftarrow f(f(y) \bmod n) \bmod n$ 
         $p \leftarrow \gcd(x - y, n)$ 
    }
    return  $p$  // if  $p = n$ , the program has failed
}
```



9.3.5 *Continued*

Example 9.32

Assume that there is a computer that can perform 2^{30} (almost 1 billion) bit operations per second. What is the approximation time required to factor an integer of size

- a. 60 decimal digits? b. 100 decimal digits?

Solution

- a. A number of 60 decimal digits has almost 200 bits. The complexity is then or 2^{50} . With 2^{30} operations per second, the algorithm can be computed in 2^{20} seconds, or almost 12 days.
- b. A number of 100 decimal digits has almost 300 bits. The complexity is 2^{75} . With 2^{30} operations per second, the algorithm can be computed in 2^{45} seconds, many years.

9.3.5 *Continued*

Example 9.33

We have written a program to calculate the factors of 434617. The result is 709 ($434617 = 709 \times 613$).

Table 9.2 Values of x , y , and p in Example 9.33

x	y	p
2	2	1
5	26	1
26	23713	1
677	142292	1
23713	157099	1
346589	52128	1
142292	41831	1
380320	68775	1
157099	427553	1
369457	2634	1
52128	63593	1
102901	161353	1
41831	64890	1
64520	21979	1
68775	16309	709

9.3.6 More Efficient Methods

Quadratic Sieve

The method uses a sieving procedure to find the value of $x^2 \bmod n$.

$$O(e^C), \text{ where } C \approx (\ln n \ln \ln n)^{1/2}$$

Number Field Sieve

The method uses a sieving procedure in an algebraic ring structure to find $x^2 \equiv y^2 \bmod n$.

$$O(e^C) \text{ where } C \approx 2 (\ln n)^{1/3} (\ln \ln n)^{2/3}$$

9.3.6 *Continued*

Example 9.34

Assume that there is a computer that can perform 230 (almost 1 billion) bit operations per second. What is the approximate time required for this computer to factor an integer of 100 decimal digits using one of the following methods?

- a. Quadratic sieve method b. Number field sieve method

Solution

A number with 100 decimal digits has almost 300 bits ($n = 2^{300}$).
 $\ln(2^{300}) = 207$ and $\ln \ln(2^{300}) = 5$.

$$\text{a. } (207)^{1/2} \times (5)^{1/2} = 14 \times 2.23 \approx 32 \quad e^{32} \quad (e^{32}) / (2^{30}) \approx 20 \text{ hours.}$$

$$\text{b. } (207)^{1/3} \times (5)^{2/2} = 6 \times 3 \approx 18. \quad e^{18} \quad (e^{18}) / (2^{30}) \approx 6 \text{ seconds.}$$

9-4 CHINESE REMAINDER THEOREM

The Chinese remainder theorem (CRT) is used to solve a set of congruent equations with one variable but different moduli, which are relatively prime, as shown below:

$$x \equiv a_1 \pmod{m_1}$$

$$x \equiv a_2 \pmod{m_2}$$

...

$$x \equiv a_k \pmod{m_k}$$

9-4 Continued

Example 9.35

The following is an example of a set of equations with different moduli:

$$x \equiv 2 \pmod{3}$$

$$x \equiv 3 \pmod{5}$$

$$x \equiv 2 \pmod{7}$$

The solution to this set of equations is given in the next section; for the moment, note that the answer to this set of equations is $x = 23$. This value satisfies all equations: $23 \equiv 2 \pmod{3}$, $23 \equiv 3 \pmod{5}$, and $23 \equiv 2 \pmod{7}$.

9-4 Continued

Solution To Chinese Remainder Theorem

1. Find $M = m_1 \times m_2 \times \dots \times m_k$. This is the common modulus.
2. Find $M_1 = M/m_1, M_2 = M/m_2, \dots, M_k = M/m_k$.
3. Find the multiplicative inverse of M_1, M_2, \dots, M_k using the corresponding moduli (m_1, m_2, \dots, m_k). Call the inverses $M_1^{-1}, M_2^{-1}, \dots, M_k^{-1}$.
4. The solution to the simultaneous equations is

$$x = (a_1 \times M_1 \times M_1^{-1} + a_2 \times M_2 \times M_2^{-1} + \dots + a_k \times M_k \times M_k^{-1}) \bmod M$$

9-4 Continued

Example 9.36

Find the solution to the simultaneous equations:

$$x \equiv 2 \pmod{3}$$

$$x \equiv 3 \pmod{5}$$

$$x \equiv 2 \pmod{7}$$

Solution

We follow the four steps.

1. $M = 3 \times 5 \times 7 = 105$

2. $M_1 = 105 / 3 = 35, M_2 = 105 / 5 = 21, M_3 = 105 / 7 = 15$

3. The inverses are $M_1^{-1} = 2, M_2^{-1} = 1, M_3^{-1} = 1$

4. $x = (2 \times 35 \times 2 + 3 \times 21 \times 1 + 2 \times 15 \times 1) \pmod{105} = 23 \pmod{105}$

9-4 Continued

Example 9.37

Find an integer that has a remainder of 3 when divided by 7 and 13, but is divisible by 12.

Solution

This is a CRT problem. We can form three equations and solve them to find the value of x.

$$x = 3 \bmod 7$$

$$x = 3 \bmod 13$$

$$x = 0 \bmod 12$$

If we follow the four steps, we find $x = 276$. We can check that $276 = 3 \bmod 7$, $276 = 3 \bmod 13$ and 276 is divisible by 12 (the quotient is 23 and the remainder is zero).

9-4 Continued

Example 9.38

Assume we need to calculate $z = x + y$ where $x = 123$ and $y = 334$, but our system accepts only numbers less than 100.

$$\begin{array}{ll} x \equiv 24 \pmod{99} & y \equiv 37 \pmod{99} \\ x \equiv 25 \pmod{98} & y \equiv 40 \pmod{98} \\ x \equiv 26 \pmod{97} & y \equiv 43 \pmod{97} \end{array}$$

Adding each congruence in x with the corresponding congruence in y gives

$$\begin{array}{ll} x + y \equiv 61 \pmod{99} & \rightarrow z \equiv 61 \pmod{99} \\ x + y \equiv 65 \pmod{98} & \rightarrow z \equiv 65 \pmod{98} \\ x + y \equiv 69 \pmod{97} & \rightarrow z \equiv 69 \pmod{97} \end{array}$$

Now three equations can be solved using the Chinese remainder theorem to find z . One of the acceptable answers is $z = 457$.

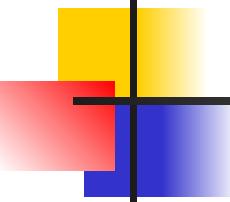
9-5 QUADRATIC CONGRUENCE

In cryptography, we also need to discuss quadratic congruence—that is, equations of the form $a_2x^2 + a_1x + a_0 \equiv 0 \pmod{n}$. We limit our discussion to quadratic equations in which $a_2 = 1$ and $a_1 = 0$, that is equations of the form

$$x^2 \equiv a \pmod{n}.$$

Topics discussed in this section:

- 9.5.1 Quadratic Congruence Modulo a Prime**
- 9.5.2 Quadratic Congruence Modulo a Composite**



9.5.1 Quadratic Congruence Modulo a Prime

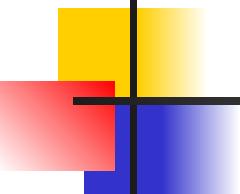
We first consider the case in which the modulus is a prime.

Example 9.39

The equation $x^2 \equiv 3 \pmod{11}$ has two solutions, $x \equiv 5 \pmod{11}$ and $x \equiv -5 \pmod{11}$. But note that $-5 \equiv 6 \pmod{11}$, so the solutions are actually 5 and 6. Also note that these two solutions are incongruent.

Example 9.40

The equation $x^2 \equiv 2 \pmod{11}$ has no solution. No integer x can be found such that its square is 2 mod 11.



9.5.1 *Continued*

Quadratic Residues and Nonresidue

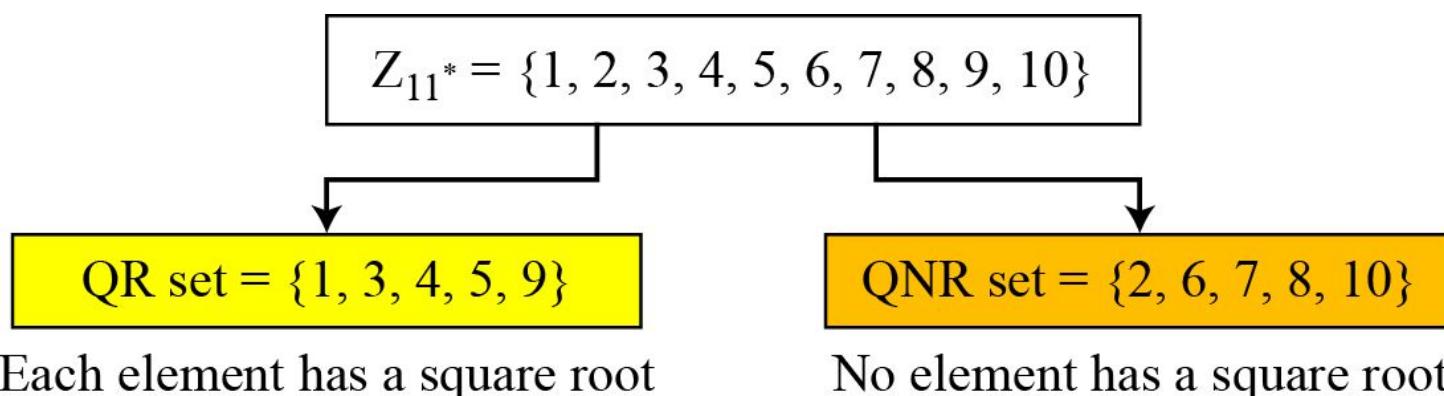
In the equation $x^2 \equiv a \pmod{p}$, a is called a quadratic residue (QR) if the equation has two solutions; a is called quadratic nonresidue (QNR) if the equation has no solutions.

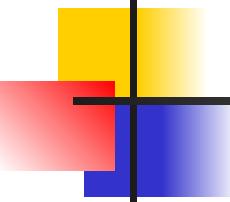
9.5.1 *Continued*

Example 9.41

There are 10 elements in Z_{11}^* . Exactly five of them are quadratic residues and five of them are nonresidues. In other words, Z_{11}^* is divided into two separate sets, QR and QNR, as shown in Figure 9.4.

Figure 9.4 *Division of Z_{11}^* elements into QRs and QNRs*





9.5.1 Continued

Euler's Criterion

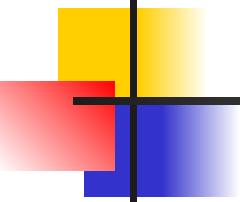
- a. If $a^{(p-1)/2} \equiv 1 \pmod{p}$, a is a quadratic residue modulo p .
- b. If $a^{(p-1)/2} \equiv -1 \pmod{p}$, a is a quadratic nonresidue modulo p .

Example 9.42

To find out if 14 or 16 is a QR in \mathbb{Z}_{23}^* , we calculate:

$$14^{(23-1)/2} \pmod{23} \rightarrow 22 \pmod{23} \rightarrow -1 \pmod{23} \text{ nonresidue}$$

$$16^{(23-1)/2} \pmod{23} \rightarrow 16^{11} \pmod{23} \rightarrow 1 \pmod{23} \text{ residue}$$

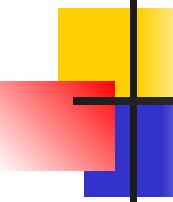


9.5.1 *Continued*

Solving Quadratic Equation Modulo a Prime

Special Case: $p = 4k + 3$

$$x \equiv a^{(p+1)/4} \pmod{p} \quad \text{and} \quad x \equiv -a^{(p+1)/4} \pmod{p}$$



9.5.1 Continued

Example 9.43

Solve the following quadratic equations:

a. $x^2 \equiv 3 \pmod{23}$

b. $x^2 \equiv 2 \pmod{11}$

c. $x^2 \equiv 7 \pmod{19}$

Solutions

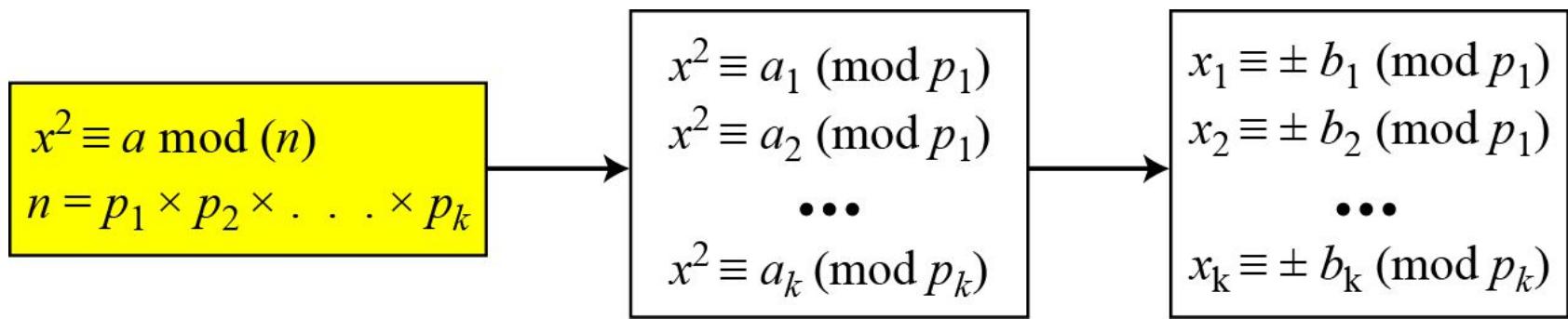
a. $x \equiv \pm 16 \pmod{23}$ $\sqrt{3} \equiv \pm 16 \pmod{23}$.

b. There is no solution for $\sqrt{2}$ in \mathbb{Z}_{11} .

c. $x \equiv \pm 11 \pmod{19}$. $\sqrt{7} \equiv \pm 11 \pmod{19}$.

9.5.2 Quadratic Congruence Modulo a Composite

Figure 9.5 Decomposition of congruence modulo a composite



9.5.2 *Continued*

Example 9.44

Assume that $x^2 \equiv 36 \pmod{77}$. We know that $77 = 7 \times 11$. We can write

$$x^2 \equiv 36 \pmod{7} \equiv 1 \pmod{7} \quad \text{and} \quad x^2 \equiv 36 \pmod{11} \equiv 3 \pmod{11}$$

The answers are $x \equiv +1 \pmod{7}$, $x \equiv -1 \pmod{7}$, $x \equiv +5 \pmod{11}$, and $x \equiv -5 \pmod{11}$. Now we can make four sets of equations out of these:

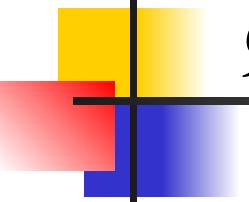
Set 1: $x \equiv +1 \pmod{7} \quad x \equiv +5 \pmod{11}$

Set 2: $x \equiv +1 \pmod{7} \quad x \equiv -5 \pmod{11}$

Set 3: $x \equiv -1 \pmod{7} \quad x \equiv +5 \pmod{11}$

Set 4: $x \equiv -1 \pmod{7} \quad x \equiv -5 \pmod{11}$

The answers are $x = \pm 6$ and ± 27 .



9.5.2 *Continued*

Complexity

How hard is it to solve a quadratic congruence modulo a composite? The main task is the factorization of the modulus. In other words, the complexity of solving a quadratic congruence modulo a composite is the same as factorizing a composite integer. As we have seen, if n is very large, factorization is infeasible.

Note

Solving a quadratic congruence modulo a composite
is as hard as factorization
of the modulus.

9-6 EXPONENTIATION AND LOGARITHM

Exponentiation: $y = a^x$ \rightarrow **Logarithm:** $x = \log_a y$

Topics discussed in this section:

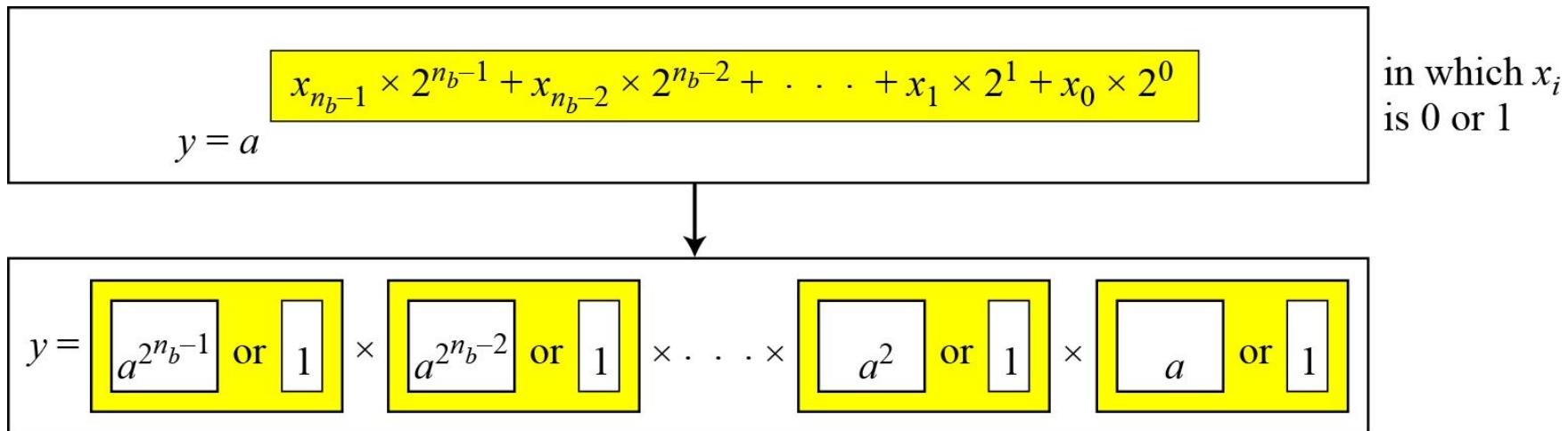
9.6.1 Exponentiation

9.6.2 Logarithm

9.6.1 Exponentiation

Fast Exponentiation

Figure 9.6 *The idea behind the square-and-multiply method*



Example:

$$y = a^9 = a^{1001_2} = a^8 \times 1 \times 1 \times a$$

9.6.1 Continued

Algorithm 9.7 Pseudocode for square-and-multiply algorithm

Square_and_Multiply (a, x, n)

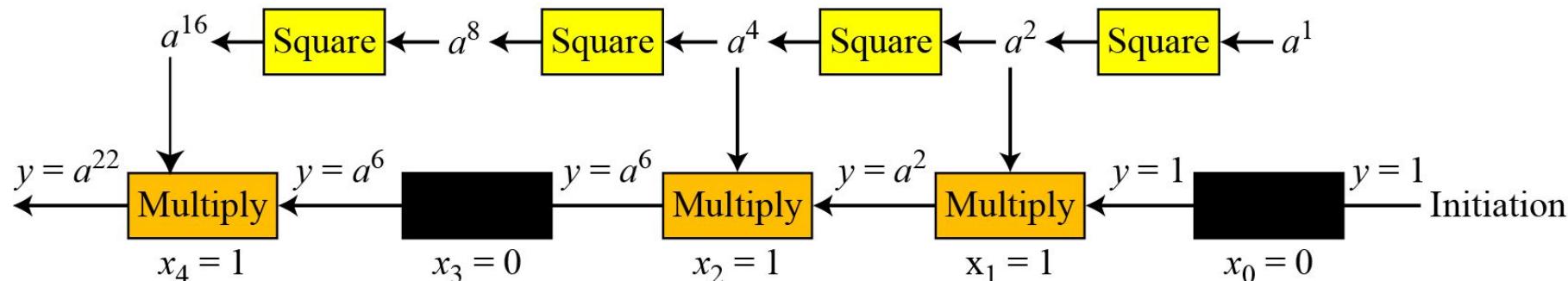
```
{  
    y ← 1  
    for (i ← 0 to  $n_b - 1$ ) //  $n_b$  is the number of bits in  $x$   
    {  
        if ( $x_i = 1$ ) y ←  $a \times y \bmod n$  // multiply only if the bit is 1  
         $a \leftarrow a^2 \bmod n$  // squaring is not needed in the last iteration  
    }  
    return y  
}
```

9.6.1 Continued

Example 9.45

Figure 9.7 shows the process for calculating $y = a^x$ using the Algorithm 9.7 (for simplicity, the modulus is not shown). In this case, $x = 22 = (10110)_2$ in binary. The exponent has five bits.

Figure 9.7 Demonstration of calculation of a^{22} using square-and-multiply method



9.6.1 Continued

Table 9.3 Calculation of $17^{22} \bmod 21$

i	x_i	<i>Multiplication (Initialization: $y = 1$)</i>	<i>Squaring (Initialization: $a = 17$)</i>
0	0		$a = 17^2 \bmod 21 = 16$
1	1	$y = 1 \times 16 \bmod 21 = 16$	$a = 16^2 \bmod 21 = 4$
2	1	$y = 16 \times 4 \bmod 21 = 1$	$a = 4^2 \bmod 21 = 16$
3	0		$a = 16^2 \bmod 21 = 4$
4	1	$y = 1 \times 4 \bmod 21 = 4$	

How about $21^{24} \bmod 8$?

9.6.2 Logarithm

In cryptography, we also need to discuss modular logarithm.

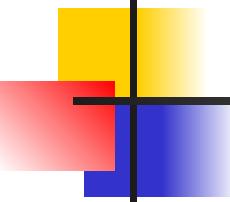
Exhaustive Search

Algorithm 9.8 Exhaustive search for modular logarithm

Modular_Logarithm (a, y, n)

```
{  
    for ( $x = 1$  to  $n - 1$ ) // k is the number of bits in x  
    {  
        if ( $y \equiv a^x \pmod{n}$ ) return  $x$   
    }  
    return failure  
}
```

Table 8.3 Powers of Integers, Modulo 19

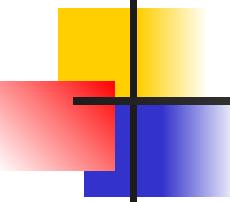


9.6.2 *Continued*

Order of the Group.

Example 9.46

What is the order of group $G = \langle Z_{21}^*, \times \rangle$? $|G| = \phi(21) = \phi(3) \times \phi(7) = 2 \times 6 = 12$. There are 12 elements in this group: 1, 2, 4, 5, 8, 10, 11, 13, 16, 17, 19, and 20. All are relatively prime with 21.



9.6.2 *Continued*

Order of an Element

Example 9.47

Find the order of all elements in $G = \langle \mathbb{Z}_{10}^*, \times \rangle$.

Solution

This group has only $\phi(10) = 4$ elements: 1, 3, 7, 9. We can find the order of each element by trial and error.

- a. $1^1 \equiv 1 \pmod{10} \rightarrow \text{ord}(1) = 1.$
- b. $3^4 \equiv 1 \pmod{10} \rightarrow \text{ord}(3) = 4.$
- c. $7^4 \equiv 1 \pmod{10} \rightarrow \text{ord}(7) = 4.$
- d. $9^2 \equiv 1 \pmod{10} \rightarrow \text{ord}(9) = 2.$

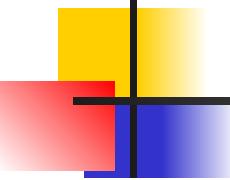
9.6.2 Continued

Euler's Theorem

Example 9.48

Table 9.4 Finding the orders of elements in Example 9.48

	$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$	$i = 6$	$i = 7$
$a = 1$	x: 1						
$a = 3$	x: 3	x: 1	x: 3	x: 1	x: 3	x: 1	x: 3
$a = 5$	x: 5	x: 1	x: 5	x: 1	x: 5	x: 1	x: 5
$a = 7$	x: 7	x: 1	x: 7	x: 1	x: 7	x: 1	x: 7



9.6.2 Continued

Primitive Roots In the group $G = \langle \mathbb{Z}_n^*, \times \rangle$, when the order of an element is the same as $\varphi(n)$, that element is called the primitive root of the group.

Example 9.49

Table 9.4 shows that there are no primitive roots in $G = \langle \mathbb{Z}_8^*, \times \rangle$ because no element has the order equal to $\varphi(8) = 4$. The order of elements are all smaller than 4.

9.6.2 Continued

Example 9.50

Table 9.5 shows the result of $a^i \equiv x \pmod{7}$ for the group $G = \langle \mathbb{Z}_7^*, \times \rangle$. In this group, $\phi(7) = 6$.

Table 9.5 Example 9.50

	$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$	$i = 6$
$a = 1$	x: 1					
$a = 2$	x: 2	x: 4	x: 1	x: 2	x: 4	x: 1
$a = 3$	x: 3	x: 2	x: 6	x: 4	x: 5	x: 1
$a = 4$	x: 4	x: 2	x: 1	x: 4	x: 2	x: 1
$a = 5$	x: 5	x: 4	x: 6	x: 2	x: 3	x: 1
$a = 6$	x: 6	x: 1	x: 6	x: 1	x: 6	x: 1

Primitive root →

Primitive root →

9.6.2 *Continued*

Note

The group $G = \langle \mathbb{Z}_n^*, \times \rangle$ has primitive roots only if n is 2, 4, p^t , or $2p^t$.

Example 9.51

For which value of n , does the group $G = \langle \mathbb{Z}_n^*, \times \rangle$ have primitive roots: 17, 20, 38, and 50?

Solution

- $G = \langle \mathbb{Z}_{17}^*, \times \rangle$ has primitive roots, 17 is a prime.
- $G = \langle \mathbb{Z}_{20}^*, \times \rangle$ has no primitive roots.
- $G = \langle \mathbb{Z}_{38}^*, \times \rangle$ has primitive roots, $38 = 2 \times 19$ prime.
- $G = \langle \mathbb{Z}_{50}^*, \times \rangle$ has primitive roots, $50 = 2 \times 5^2$ and 5 is a prime.

9.6.2 *Continued*

Note

If the group $G = \langle \mathbb{Z}_n^*, \times \rangle$ has any primitive root,
the number of primitive roots is $\phi(\phi(n))$.

9.6.2 Continued

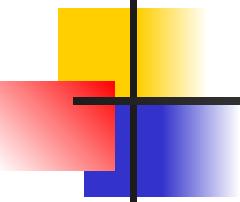
Cyclic Group If g is a primitive root in the group, we can generate the set Z_n^* as $Z_n^* = \{g^1, g^2, g^3, \dots, g^{\varphi(n)}\}$

Example 9.52

The group $G = \langle Z_{10}^*, \times \rangle$ has two primitive roots because $\varphi(10) = 4$ and $\varphi(\varphi(10)) = 2$. It can be found that the primitive roots are 3 and 7. The following shows how we can create the whole set Z_{10}^* using each primitive root.

$$\begin{array}{lllll} g = 3 \rightarrow & g^1 \bmod 10 = 3 & g^2 \bmod 10 = 9 & g^3 \bmod 10 = 7 & g^4 \bmod 10 = 1 \\ g = 7 \rightarrow & g^1 \bmod 10 = 7 & g^2 \bmod 10 = 9 & g^3 \bmod 10 = 3 & g^4 \bmod 10 = 1 \end{array}$$

The group $G = \langle Z_n^*, \times \rangle$ is a cyclic group if it has primitive roots. The group $G = \langle Z_p^*, \times \rangle$ is always cyclic.



9.6.2 Continued

The idea of Discrete Logarithm

Properties of $G = \langle \mathbb{Z}_p^, \times \rangle$:*

- 1.** *Its elements include all integers from 1 to $p - 1$.*
- 2.** *It always has primitive roots.*
- 3.** *It is cyclic. The elements can be created using g^x where x is an integer from 1 to $\phi(n) = p - 1$.*
- 4.** *The primitive roots can be thought as the base of logarithm.*

9.6.2 Continued

Solution to Modular Logarithm Using Discrete Logs

Tabulation of Discrete Logarithms

Table 9.6 Discrete logarithm for $\mathbf{G} = \langle \mathbf{Z}_7^*, \times \rangle$

y	1	2	3	4	5	6
$x = L_3 y$	6	2	1	4	5	3
$x = L_5 y$	6	4	5	2	1	3

9.6.2 *Continued*

Example 9.53

Find x in each of the following cases:

a. $4 \equiv 3^x \pmod{7}$.

b. $6 \equiv 5^x \pmod{7}$.

Solution

We can easily use the tabulation of the discrete logarithm in Table 9.6.

a. $4 \equiv 3^x \pmod{7} \rightarrow x = L_3 4 \pmod{7} = 4 \pmod{7}$

b. $6 \equiv 5^x \pmod{7} \rightarrow x = L_5 6 \pmod{7} = 3 \pmod{7}$

9.6.2 Continued

Using Properties of Discrete Logarithms

Table 9.7 Comparison of traditional and discrete logarithms

Traditional Logarithm	Discrete Logarithms
$\log_a 1 = 0$	$L_g 1 \equiv 0 \pmod{\phi(n)}$
$\log_a (x \times y) = \log_a x + \log_a y$	$L_g(x \times y) \equiv (L_g x + L_g y) \pmod{\phi(n)}$
$\log_a x^k = k \times \log_a x$	$L_g x^k \equiv k \times L_g x \pmod{\phi(n)}$

Using Algorithms Based on Discrete

Note

The discrete logarithm problem has the same complexity as the factorization problem.

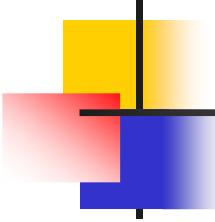


Chapter 10

Asymmetric-Key Cryptography



Copyright © The McGraw-Hill Companies, Inc. Permission required for reproduction or display.



Chapter 10

Objectives

- Present asymmetric-key cryptography.**
- Distinguish between symmetric-key cryptography and asymmetric-key cryptography.**
- Introduce trapdoor one-way functions and their use in asymmetric-key cryptosystems**
- Discuss the RSA cryptosystem**

10-1 INTRODUCTION

The advent of asymmetric-key cryptography does not eliminate the need for symmetric-key cryptography. Symmetric and asymmetric-key cryptography will exist in parallel and continue to serve the community. We actually believe that they are complements of each other; the advantages of one can compensate for the disadvantages of the other.

Topics discussed in this section:

Keys

General Idea

Asymmetric Cryptography Practices

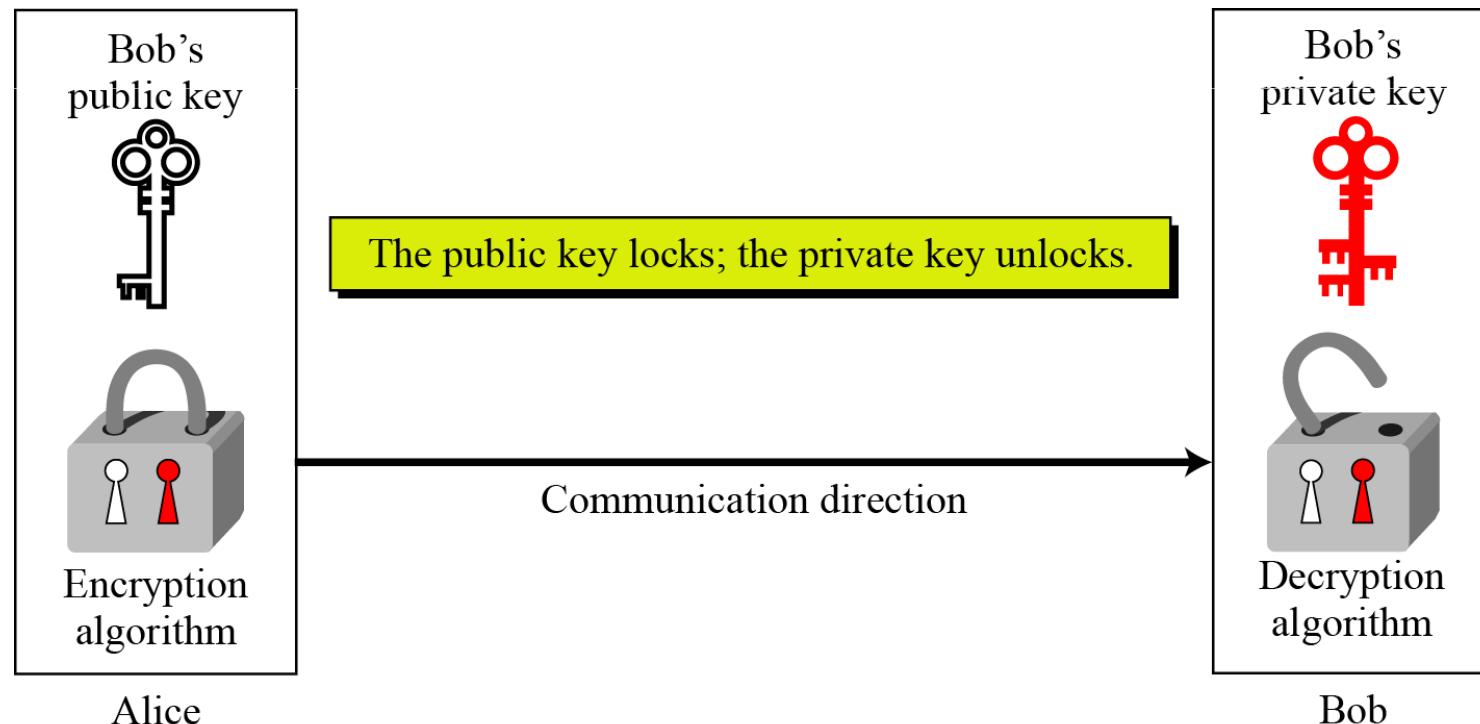
Symmetric Cryptography Versus Asymmetric Cryptography

Trapdoor One-Way Function

10.1.1 Keys

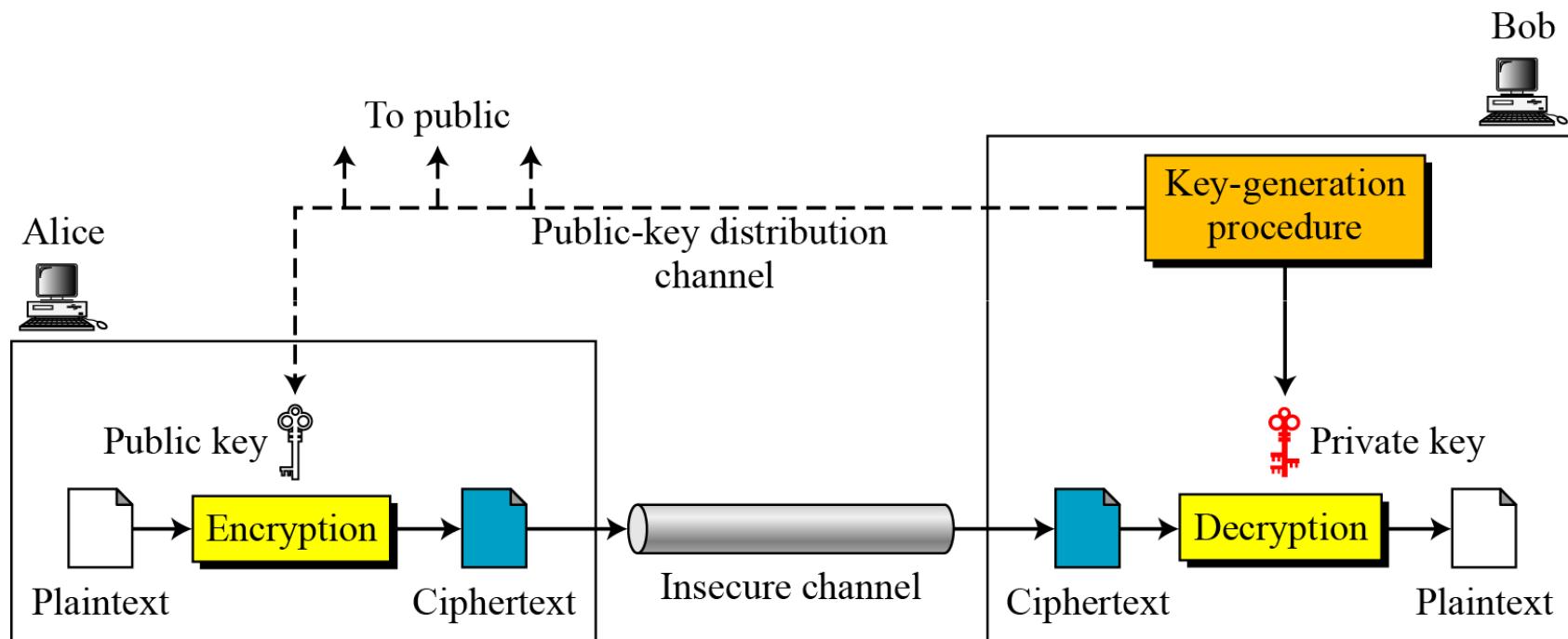
Asymmetric key cryptography, known as public key cryptography, uses two separate keys: one private and one public.

Figure 10.1 Locking and unlocking in asymmetric-key cryptosystem

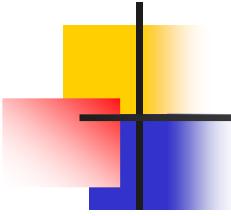


10.1.2 General Idea

Figure 10.2 General idea of asymmetric-key cryptosystem

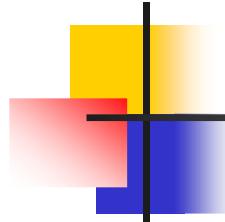


$$C = f(K_{public}, P) \quad P = g(K_{private}, C)$$



Asymmetric Cryptography Practices

Action	Whose Key to Use	Which Key to Use	Explanation
Bob wants to send Alice an encrypted message	Alice's key	Public key	Whenever an encrypted message is to be sent the recipient's key is always used and never the sender's keys.
Alice wants to read an encrypted message sent by Bob	Alice's key	Private key	An encrypted message can only be read by using the recipient's private key.
Bob wants to send a copy to himself of the encrypted message that he sent to Alice	Bob's key	Public key to encrypt Private key to decrypt	An encrypted message can only be read by the recipient's private key. Bob would need to encrypt it with his own public key and then use his private key to decrypt it.
Bob receives an encrypted reply message from Alice	Bob's key	Private key	The recipient's private key is used to decrypt received messages.
Bob wants Susan to read Alice's reply message that he received	Susan's key	Public key	The message should be encrypted with Susan's key for her to decrypt and read it with her private key.



Symmetric Cryptography Versus Asymmetric Cryptography

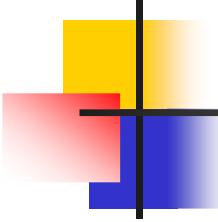
Note-1

**Symmetric-key cryptography is based on sharing secrecy;
asymmetric-key cryptography is based on personal secrecy.**

Note-2

In symmetric-key cryptography system, the number of keys needed for each user is 1.

In asymmetric-key cryptography system, the number of keys needed for each user is 2.



Symmetric Cryptography Versus Asymmetric Cryptography

Note-3

In symmetric-key cryptography, symbols in plaintext and ciphertext are permuted or substituted.

In asymmetric-key cryptography, plaintext and ciphertext are treated as integers.

Note-4

Symmetric-key cryptography is appropriate for long messages, and the speed of encryption/decryption is fast.

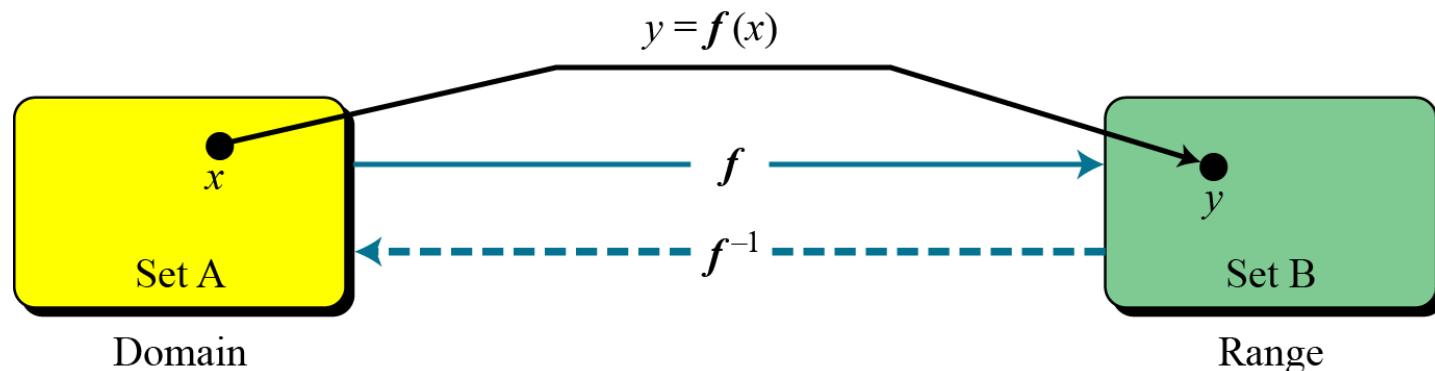
Asymmetric-key cryptography is appropriate for short messages, and the speed of encryption/decryption is slow.

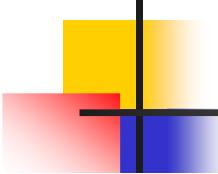
10.1.4 Trapdoor One-Way Function

The main idea behind asymmetric-key cryptography is the concept of the trapdoor one-way function.

Functions

Figure 10.3 A function as rule mapping a domain to a





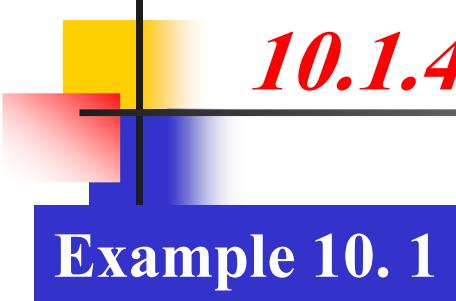
10.1.4 Continued

One-Way Function (OWF)

1. f is easy to compute $\rightarrow y=f(x)$
2. f^{-1} is difficult to compute $\rightarrow x=f^{-1}(y)$

Trapdoor One-Way Function (TOWF)

3. Given y and a trapdoor, x can be computed easily.



10.1.4 Continued

Example 10. 1

When n is large, $n = p \times q$ is a one-way function. Given p and q , it is always easy to calculate n ; given n , it is very difficult to compute p and q . This is the factorization problem.

Example 10. 2

When n is large, the function $y = x^k \bmod n$ is a trapdoor one-way function. Given x , k , and n , it is easy to calculate y . Given y , k , and n , it is very difficult to calculate x . This is the discrete logarithm problem. However, if we know the trapdoor, k' such that $k \times k' = 1 \bmod \Phi(n)$, we can use $x = y^{k'} \bmod n$ to find x .

10-2 RSA CRYPTOSYSTEM

The most common public-key algorithm is the RSA cryptosystem, named for its inventors (Rivest, Shamir, and Adleman).

Topics discussed in this section:

10.2.1 Introduction

10.2.2 Procedure

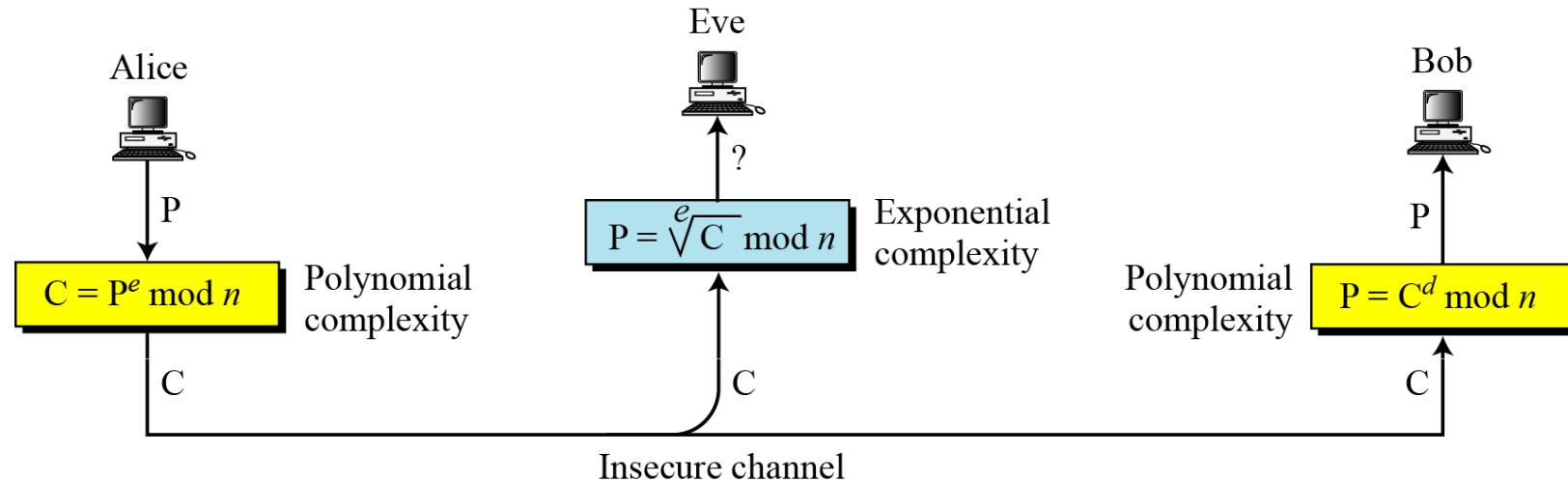
10.2.3 Some Trivial Examples

10.2.4 Attacks on RSA

10.2.5 Recommendations

10.2.1 Introduction

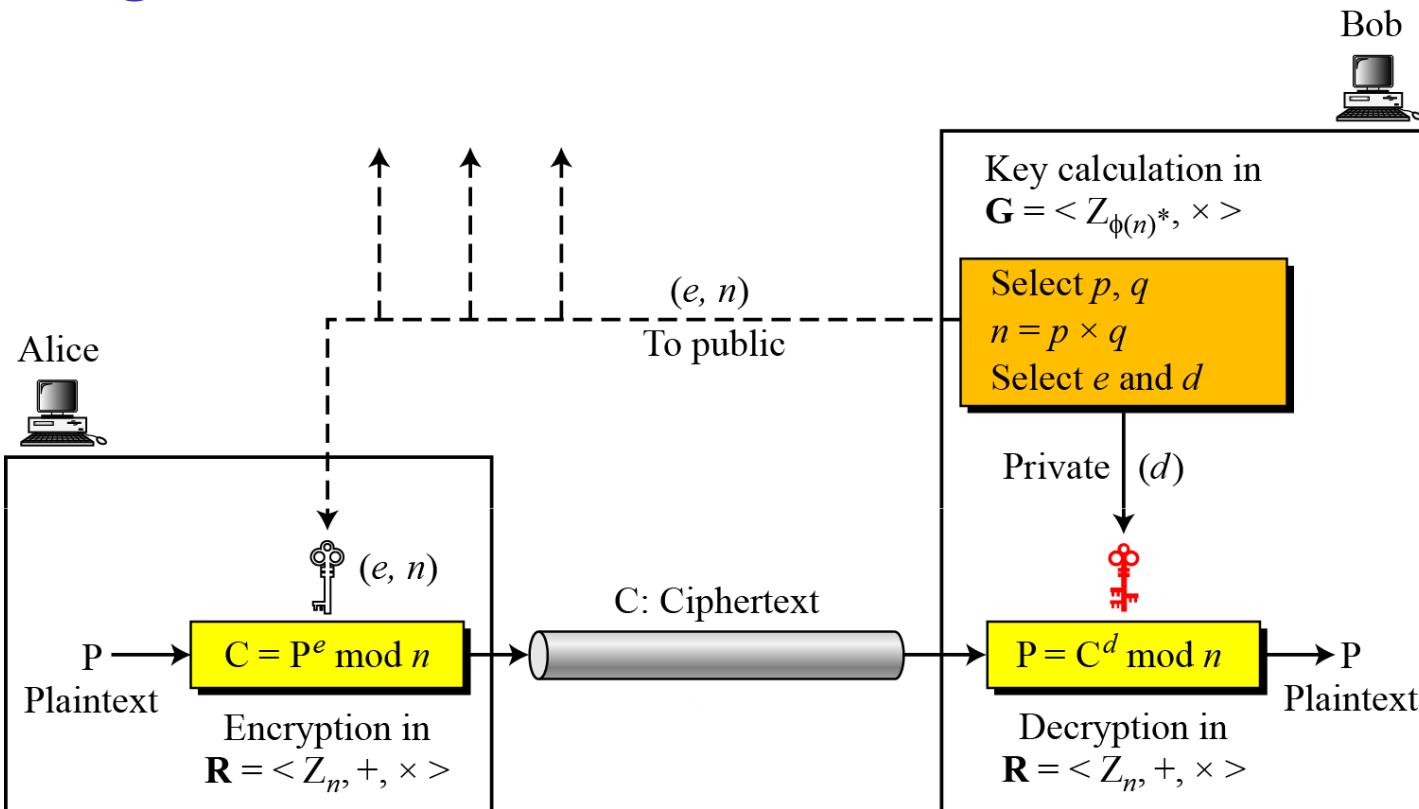
Figure 10.5 Complexity of operations in RSA



**RSA uses modular exponentiation for encryption/decryption;
To attack it, Eve needs to calculate $\sqrt[e]{C} \text{ mod } n$.**

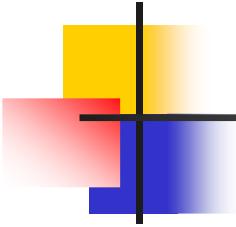
10.2.2 Procedure

Figure 10.6 Encryption, decryption, and key generation in RSA



RSA uses two algebraic structures:
a public ring $\mathbf{R} = \langle \mathbb{Z}_n, +, \times \rangle$ and a private group $\mathbf{G} = \langle \mathbb{Z}_{\phi(n)}^*, \times \rangle$.

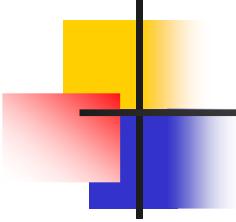
In RSA, the tuple (e, n) is the public key; the integer d is the private key.



10.2.2 Continued

Algorithm 10.2 RSA Key Generation

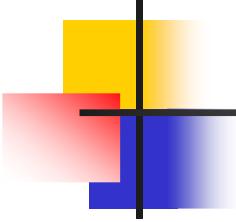
```
RSA_Key_Generation
{
    Select two large primes  $p$  and  $q$  such that  $p \neq q$ .
     $n \leftarrow p \times q$ 
     $\phi(n) \leftarrow (p - 1) \times (q - 1)$ 
    Select  $e$  such that  $1 < e < \phi(n)$  and  $e$  is coprime to  $\phi(n)$ 
     $d \leftarrow e^{-1} \bmod \phi(n)$                                 //  $d$  is inverse of  $e$  modulo  $\phi(n)$ 
    Public_key  $\leftarrow (e, n)$                                 // To be announced publicly
    Private_key  $\leftarrow d$                                     // To be kept secret
    return Public_key and Private_key
}
```



10.2.2 Continued

When Alice wants Bob to send her a message, she:

- Selects two (large) primes p, q , **TOP SECRET**,
- Computes $n = pq$ and $\phi(n) = (p-1)(q-1)$. $\phi(n)$ is **TOP SECRET**.
- Selects an integer e , $1 < e < \phi(n)$, such that $\gcd(e, \phi(n)) = 1$,
- Computes d , such that $d * e \pmod{\phi(n)} = 1$, d also **TOP SECRET**,
- Gives **public key (e, n)** to Bob, and keeps her **private key (d, n)**.



10.2.2 Continued

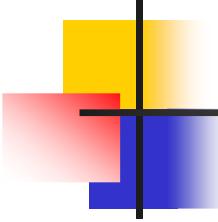
Encryption

Algorithm 10.3 RSA encryption

```
RSA_Encryption (P, e, n)          // P is the plaintext in  $Z_n$  and  $P < n$ 
{
    C ← Fast_Exponentiation (P, e, n)    // Calculation of  $(P^e \bmod n)$ 
    return C
}
```

In RSA, p and q must be at least 512 bits; n must be at least 1024 bits.

If the plaintext P is larger than n , then P has to be encrypted letter by letter.



10.2.2 Continued

Decryption

Algorithm 10.4 RSA decryption

```
RSA_Decryption (C, d, n)           // C is the ciphertext in  $Z_n$ 
{
    P ← Fast_Exponentiation (C, d, n)    // Calculation of  $(C^d \bmod n)$ 
    return P
}
```



10.2.3 Some Trivial Examples

Example 10.5

Bob chooses 7 and 11 as p and q and calculates $n = 77$. The value of $\phi(n) = (7 - 1)(11 - 1)$ or 60. Now he chooses two exponents, e and d , from Z_{60}^* . If he chooses e to be 13, then d is 37. Note that $e \times d \bmod 60 = 1$ (they are inverses of each other). Now imagine that Alice wants to send the plaintext 5 to Bob. She uses the public exponent 13 to encrypt 5.

Plaintext: 5

$$C = 5^{13} = 26 \bmod 77$$

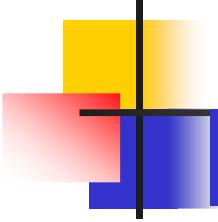
Ciphertext: 26

Bob receives the ciphertext 26 and uses the private key 37 to decipher the ciphertext:

Ciphertext: 26

$$P = 26^{37} = 5 \bmod 77$$

Plaintext: 5



10.2.2 Continued

Example 10.5 (cont.)

Calculate $5^{13} \bmod 77$:

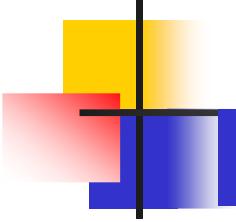
$$5^1 = 5 \bmod 77 = 5$$

$$5^2 = 25 \bmod 77 = 25$$

$$5^4 = 625 \bmod 77 = 9$$

$$5^8 = 390625 \bmod 77 = 4$$

$$5^{13} = 5^1 * 5^4 * 5^8 = 180 \bmod 77 = 26$$



10.2.3 Some Trivial Examples

Example 10. 6

Now assume that another person, John, wants to send a message to Bob. John can use the same public key announced by Bob (probably on his website), 13; John's plaintext is 63. John calculates the following:

Plaintext: 63

$$C = 63^{13} \equiv 28 \pmod{77}$$

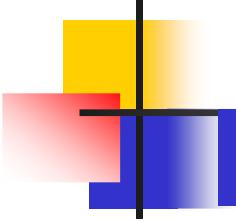
Ciphertext: 28

Bob receives the ciphertext 28 and uses his private key 37 to decipher the ciphertext:

Ciphertext: 28

$$P = 28^{37} \equiv 63 \pmod{77}$$

Plaintext: 63



10.2.3 Some Trivial Examples

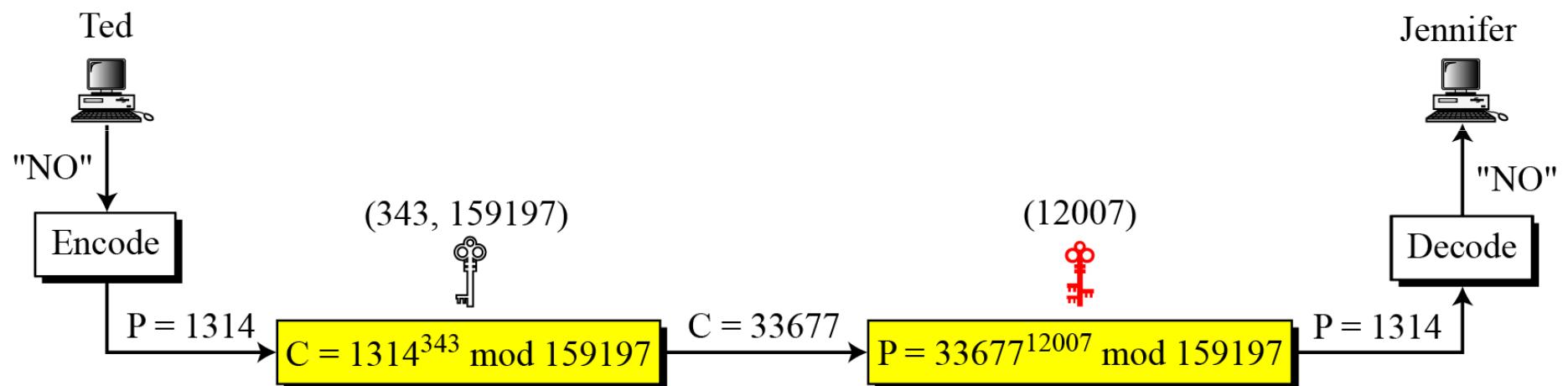
Example 10.7

Jennifer creates a pair of keys for herself. She chooses $p = 397$ and $q = 401$. She calculates $n = 159197$. She then calculates $\phi(n) = 158400$. She then chooses $e = 343$ and $d = 12007$. Show how Ted can send a message to Jennifer if he knows e and n .

Suppose Ted wants to send the message “NO” to Jennifer. He changes each character to a number (from 00 to 25), with each character coded as two digits. He then concatenates the two coded characters and gets a four-digit number. The plaintext is 1314. Figure 10.7 shows the process.

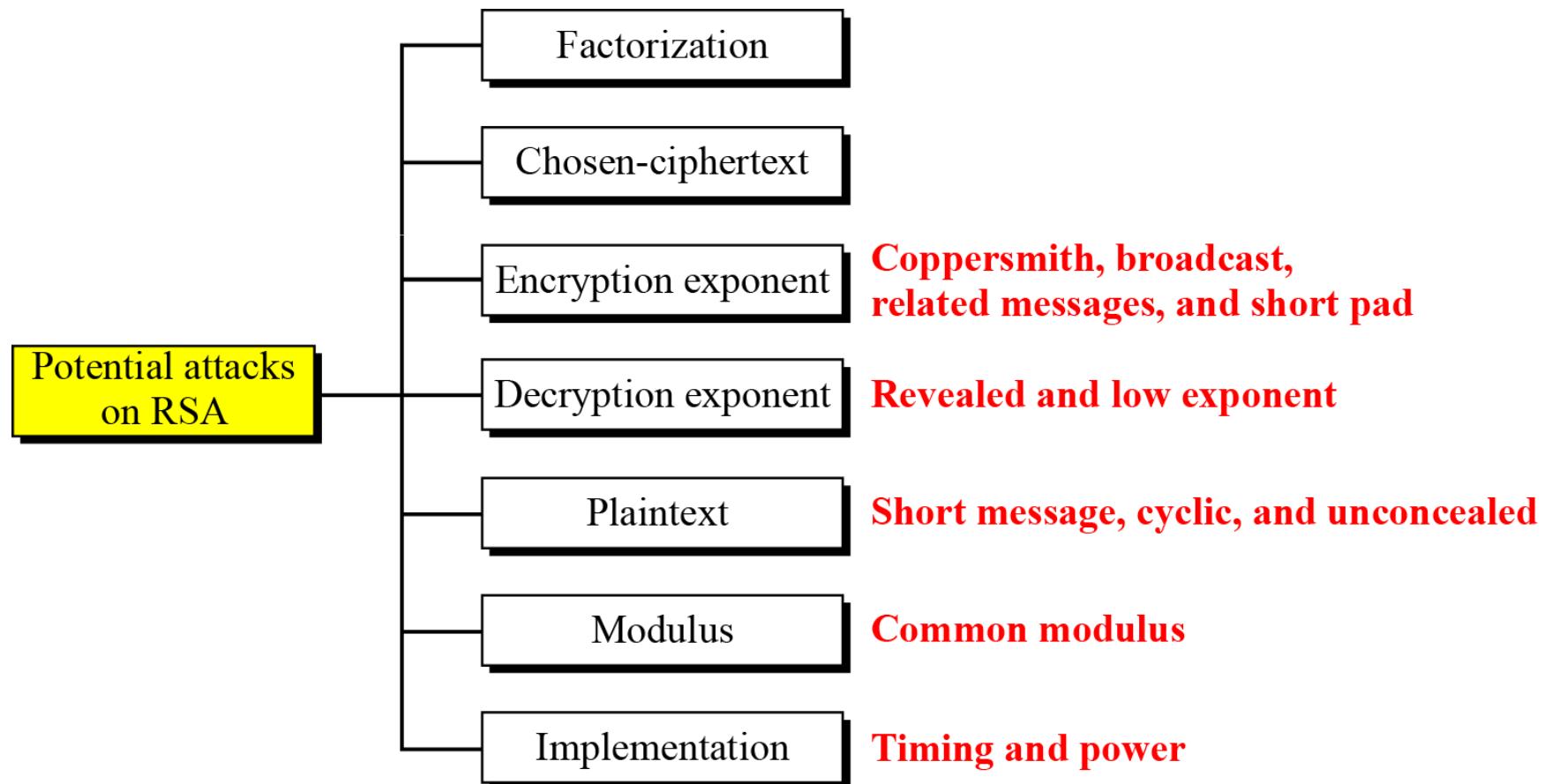
10.2.3 Continued

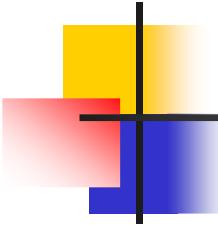
Figure 10.7 Encryption and decryption in Example 10.7



10.2.4 Attacks on RSA

Figure 10.8 Taxonomy of potential attacks on RSA

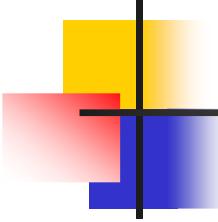




10.2.4 Continued

Factorization Attack

- The security of RSA is based on the idea that the modulus is so large that it is infeasible to factor it in reasonable time.
- Even though n is public, p & q are secret. If Eve can factor n and get p & q , she can calculate $\Phi(n)$. Then she can calculate $d = e \text{ mod } \Phi(n)$ because e is public.



Recommendations

- The number of bits in n should be at least 1024.
- Two primes p & q must be 512 bit at least.
- p & q should not be close to each other.
- Modulus n must not be shared.
- If d is leaked, immediately change n, e and d.
- Message must be padded by OAEP.