# Cryptography & Network Security

- Cryptography : A word in Greek origin means " Secret writing"

- Cryptography (Basic definition)
  - It is the science and art of transforming messages to make them secure and immune to attacks.

- Different types of Cryptography
  - Symmetric Key Cryptography : Same key will be used for encryption & decryption.
  - Asymmetric Key Cryptography: One key will be used for encryption and another key will be used for decryption.

# Steganography

- Steganography : A word in Greek origin means “ Covered writing”
- Steganography (Basic definition)
  - Steganography means concealing the message itself by covering it with something else.
- Historical Use:
  - Messages were carved on pieces of wood and later dipped on the wax to covered writing.
  - Messages were written on the thin piece of silk and later swallowed by the messenger.

# Steganography

- Modern Use
  - Text cover: The cover of secret data is Text

    0 1 0 1 0 1

    We have to study about cryptography not steganography
  - Image cover: Secret data can also be covered under a color.

    01101110 00001111 10101101 00010101

# Goals of Network Security

- Privacy or Confidentiality:
  - It is used to protect data from disclosure attack.
- Authentication:
  - It is used to provide authentication of the party at either end of the line.
- Integrity:
  - It is used to protect data from modification, insertion, deletion etc.
- Non-Repudiation:
  - This service protects against repudiation either by sender or the receiver.

# ITU-T Security Services

- International Telecommunication Union- Telecommunication Standardized Sector (ITU-T)

- Security Services:
  - Data Confidentiality
  - Data Integrity
  - Authentication
  - Non-repudiation
  - Access control: It provides protection against unauthorized access to data.

# ITU-T Security Mechanisms

- ITU-T recommends some security mechanisms to provide the various security services.
- **Encipherment**: It can be used to provide confidentiality. It can also be used to complement other mechanisms. (Cryptography & Steganography)
- **Data Integrity**: It appends a short check value created for the message to be transmitted.
- **Digital Signature**: Sender can electronically sign the data and receiver can electronically verify the signature.
- **Authentication exchange**: Two entities exchange some messages to prove their identity to each other.

# ITU-T Security Mechanisms

- **Traffic Padding**: It means inserting some bogus data into the data traffic to thwart traffic analysis.
- **Routing Control**: Selecting and changing different available routes between the sender and the receiver.
- **Notarization**: It means selecting a trusted third party to control the communication between two entities.
- **Access Control**: It gives methods to prove that a user has access right to the data or resources of the system.

# Security Attacks

- Passive attacks:
  - Attacker's goal is to just obtain the information.
  - Attack dose not modify the data or harm the system.
  - Snooping: Unauthorized access to or interception of data.
  - Traf fic analysis: Attempts of analyzing encrypted messages to come up with likely patterns.
- Active attacks:
  - Attack may change the data or harm the system.
  - Modification: The attacker modifies the information.
  - Masquerading or Spoofing: It happens when the attacker is posing the identity of somebody else.

# Security attacks

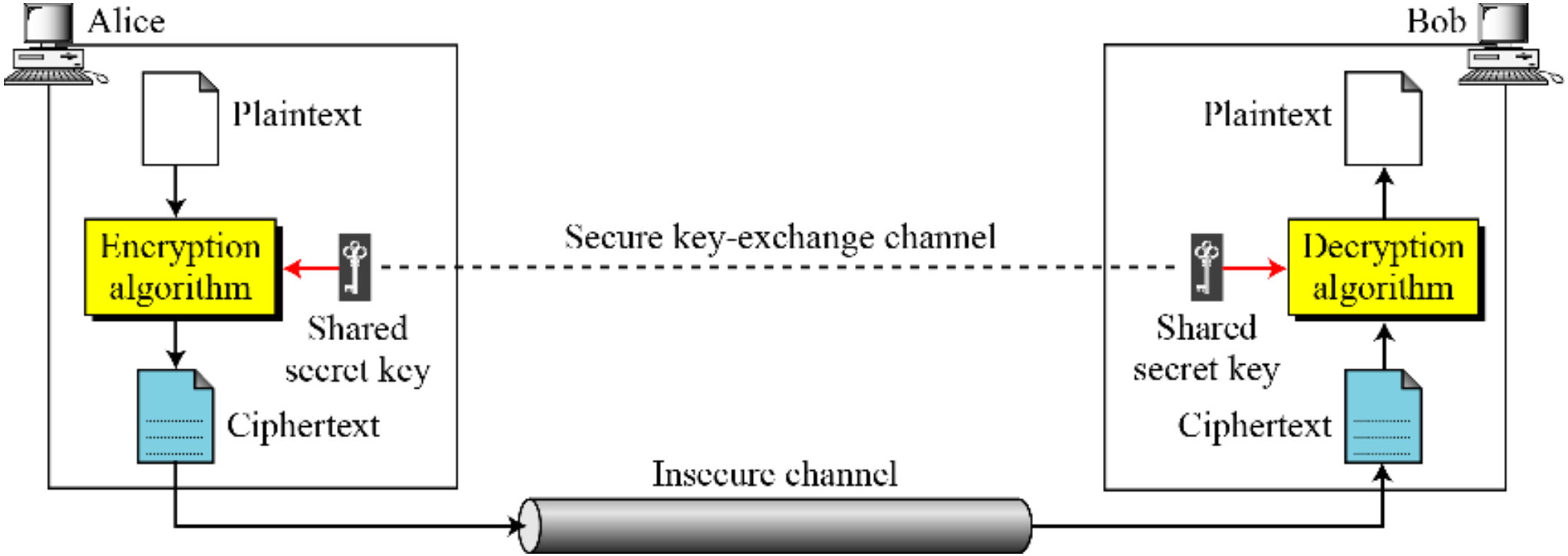
- Replaying: The attacker obtains the copy of the message and later tries to replay it.
- Repudiation: Sender denies that he has send the message or Receiver denies that he has received the message.
- Denial of Service: It may slow down or totally interrupt the service of a system.

| Attacks | Passive/Active | Threatening |
|---|---|---|
| Snooping, Traffic Analysis | Passive | Confidentiality |
| Modification, Masqyerading, Replaying, Repudiation | Active | Integrity |
| Denial of service | Active | Availability |

# General idea of Symmetric Key Cipher



General idea of symmetric-key cipher



# General idea of Symmetric Key Cipher



If P is the plaintext, C is the ciphertext, and K is the key.

Encryption: $C = E_k(P)$ Decryption: $P = D_k(C)$

**Alice:** $C = E_k(P)$

**Bob:** $P_1 = D_k(C) = D_k(E_k(P)) = P$

# Kerckhoff's Principle



- It appears that a cipher is more secure if we hide both the encryption(E)/decryption(D) algorithm and the secret key.
- Based on the Kerckhoff's principle, adversary always know the E/D algorithm.
- Resistance of the cipher to attack must be based on the secrecy of the key.
- Guessing of the key should be so difficult that there is no need to hide the E/D algorithm.
- For modern ciphers, it is require that there key domain should be very large.

# Traditional Ciphers

- Substitution & Transposition ciphers

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

- Substitution ciphers can be categorized to:
  Mono-alphabetic & Poly-alphabetic ciphers.

- Mono-alphabetic: The relationship between a character in the plaintext to a character in the ciphertext is one to one.
- Poly-alphabetic: The relationship between a character in the plaintext to a character in the ciphertext is one to many.

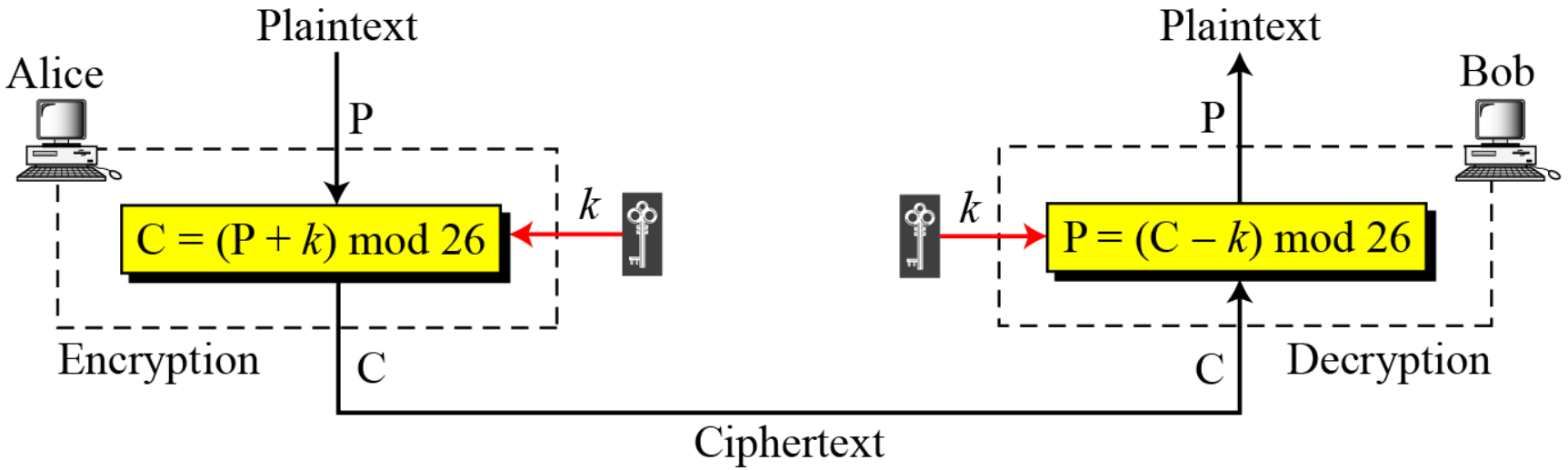
# Monoalphabetic cipher- Additive

The simplest mono-alphabetic cipher is the additive cipher. This cipher is sometimes called a shift cipher and sometimes a Caesar cipher(with key 3), but the term additive cipher better reveals its mathematical nature.

Plaintext and Ciphertext in $Z_{26}$

| | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 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 Cipher




When the cipher is additive, the plaintext, ciphertext and key are integers in $Z_{26}$.

$Z$: Set of integers.

$Z_n$: Set of nonnegative integers less than n. Ex: $Z_5$ ={0,1,2,3,4}, $Z_2$ ={0,1}

# Additive Cipher

Q: Use the additive cipher with key = 15 to encrypt the message "hello".

We apply the encryption algorithm to the plaintext, character by character:

| | | |
|---|---|---|
| Plaintext: h → 07 | Encryption: (07 + 15) mod 26 | Ciphertext: 22 → W |
| Plaintext: e → 04 | Encryption: (04 + 15) mod 26 | Ciphertext: 19 → T |
| Plaintext: l → 11 | Encryption: (11 + 15) mod 26 | Ciphertext: 00 → A |
| Plaintext: l → 11 | Encryption: (11 + 15) mod 26 | Ciphertext: 00 → A |
| Plaintext: o → 14 | Encryption: (14 + 15) mod 26 | Ciphertext: 03 → D |

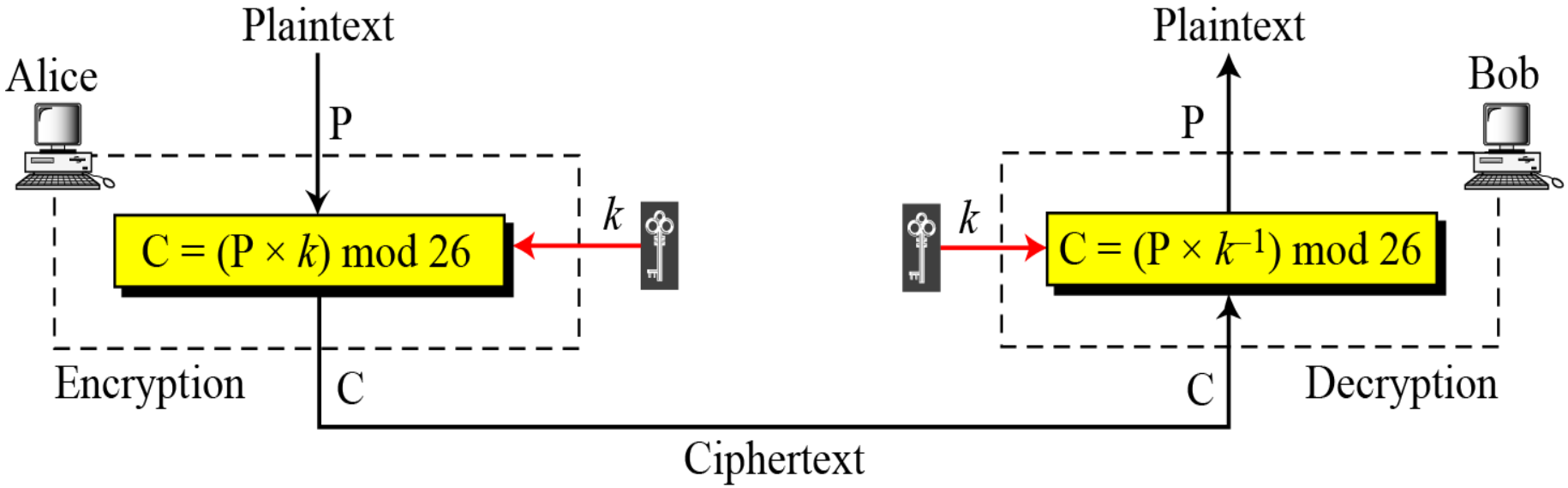
# Additive Cipher

Q. Use the additive cipher with key = 15 to decrypt the message "WTAAD".

We apply the decryption algorithm to the plaintext character by character:

| | | |
|---|---|---|
| Ciphertext: W → 22 | Decryption: (22 – 15) mod 26 | Plaintext: 07 → h |
| Ciphertext: T → 19 | Decryption: (19 – 15) mod 26 | Plaintext: 04 → e |
| Ciphertext: A → 00 | Decryption: (00 – 15) mod 26 | Plaintext: 11 → l |
| Ciphertext: A → 00 | Decryption: (00 – 15) mod 26 | Plaintext: 11 → l |
| Ciphertext: D → 03 | Decryption: (03 – 15) mod 26 | Plaintext: 14 → o |

# Multiplicative Cipher




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

# Multiplicative Cipher

$Z_{n*}$: Set of nonnegative integers less than n and co-prime to n.

The calculation of $Z_{n*}$ starts after finding the prime factors for n. For ex: When n=26, the prime factors are 2 & 13.

Next, drop the elements which are multiple of prime factor in the range of 1 to (n-1) for given n.

For ex: If n=26, then within a range of 1-26, drop elements which are multiple of 2 & 13.

# Multiplicative Cipher

Ques: What is the key domain for any multiplicative cipher?

Ans: 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.

The set $Z_8^*$ has only 4 members: 1,3,5,7. The key pairs are

(1,1) as 1x1 mod 8=1
(3,3) as 3x3 mod 8=1
(5,5) as 5x5 mod 8=1
(7,7) as 7x7 mod 8=1

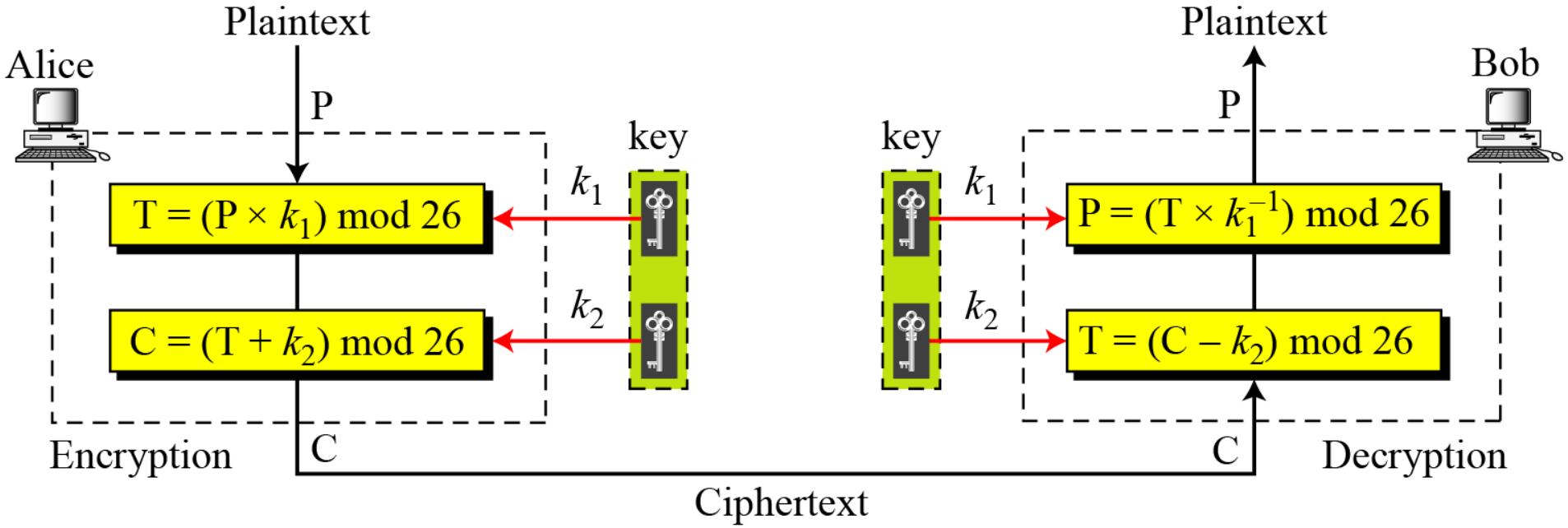
# Multiplicative Cipher

Ques: Use a multiplicative cipher to encrypt the message “hello” with a key of 7. The ciphertext is “XCZZU”.

| | | |
|---|---|---|
| Plaintext: h → 07 | Encryption: (07 × 07) mod 26 | ciphertext: 23 → X |
| Plaintext: e → 04 | Encryption: (04 × 07) mod 26 | ciphertext: 02 → C |
| Plaintext: l → 11 | Encryption: (11 × 07) mod 26 | ciphertext: 25 → Z |
| Plaintext: l → 11 | Encryption: (11 × 07) mod 26 | ciphertext: 25 → Z |
| Plaintext: o → 14 | Encryption: (14 × 07) mod 26 | ciphertext: 20 → U |

# Affine Cipher






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

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

# Affine Cipher



The affine cipher uses a pair of keys in which the first key is from $Z_{26}^*$ and the second is from $Z_{26}$. The size of the key domain is $26 \times 12 = 312$.

**Ques**: Use an affine cipher to encrypt the message "hello" with the key pair (7, 2).

| | | |
|---|---|---|
| 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 |

# Affine Cipher

**Ques**: Use the af ne cipher to decrypt the message "ZEBBW" with the key pair (7, 2) in modulus 26.

| | | |
|---|---|---|
| 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 |

# Poly-alphabetic Cipher (Auto Key Cipher)



Here the key is a stream of sub-keys where the first sub-key is predetermined value secretly agreed between sender and receiver.

The second sub-key is the value of the first plaintext, third sub-key is the value of the second plaintext and so on.

# Auto Key Cipher



**Ques**: Alice and Bob agreed to use an autokey cipher with initial key value k1 = 12 to encrypt the message “Attack is today”.

| | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 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** |

# Vigenere Cipher

Here the key stream is the repetition of the initial secret key.

**Ques**: Encrypt the message “She is listening” using the 6-character keyword “PASCAL”.

Initial key stream is (15,0,18,2,0,11)

| **Plaintext:** | s | h | e | i | s | l | i | s | t | e | n | i | n | g |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **P’s values:** | 18 | 07 | 04 | 08 | 18 | 11 | 08 | 18 | 19 | 04 | 13 | 08 | 13 | 06 |
| **Key stream:** | *15* | *00* | *18* | *02* | *00* | *11* | *15* | *00* | *18* | *02* | *00* | *11* | *15* | *00* |
| **C’s values:** | **07** | **07** | **22** | **10** | **18** | **22** | **23** | **18** | **11** | **6** | **13** | **19** | **02** | **06** |
| **Ciphertext:** | **H** | **H** | **W** | **K** | **S** | **W** | **X** | **S** | **L** | **G** | **N** | **T** | **C** | **G** |

# Playfair Cipher



The secret key is made up of 25 characters arranged in 5x5 matrix. (I and J assumed same meaning).

**Before encryption:**

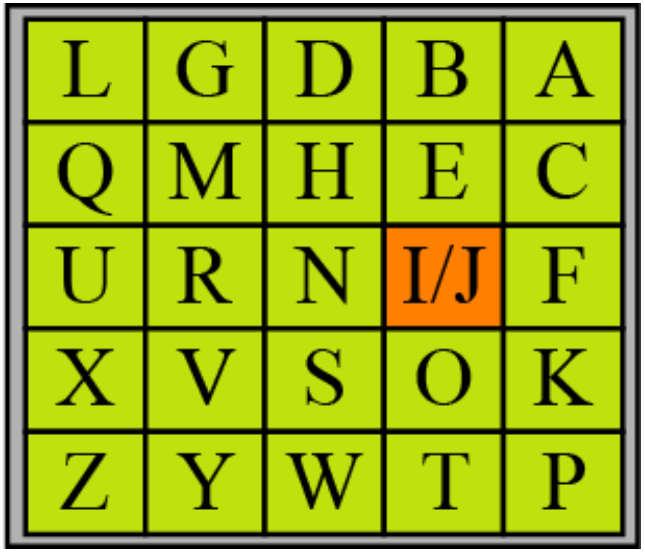
The entire message is divided into a pair of characters. If two letters in a pair are the same, a bogus letter is inserted to separate them. Finally number of characters should be even.

**Encryption Rules:**

- If two letters are in same row, encrypted letters are right to it.
- If two letters are in column, encrypted letter are beneath to it.
- If two letters are not in same row or column, the corresponding encrypted for each letter is in its row but in the same column as other.

# Playfair Cipher

Secret Key =




**Ques**: Encrypt the plaintext "hello" using the key given above.

he → EC  lx → QZ  lo → BX

Plaintext: hello  Ciphertext: ECQZBX

# Hill Cipher

The plaintext is divided into equal size blocks.

The blocks are encrypted one at a time, such that each character in the block contributes to the encryption of the other characters of the block.

In Hill cipher, the key is a square matrix of size m x m in which m is the size of the block.

$$K = \begin{bmatrix} k_{11} & k_{12} & \cdots & k_{1m} \\ k_{21} & k_{22} & \cdots & k_{2m} \\ \vdots & \vdots & & \vdots \\ k_{m1} & k_{m2} & \cdots & k_{mm} \end{bmatrix}$$

$$C_1 = P_1 k_{11} + P_2 k_{21} + \cdots + P_m k_{m1}$$
$$C_2 = P_1 k_{12} + P_2 k_{22} + \cdots + P_m k_{m2}$$
$$\cdots$$
$$C_m = P_1 k_{1m} + P_2 k_{2m} + \cdots + P_m k_{mm}$$

# Hill Cipher

For example, the plaintext “code is ready” can make a 3 × 4 matrix when adding extra bogus character “z” to the last block and removing the spaces. The ciphertext is “OHKNIHGKLISS”.

$$\underset{C}{\begin{bmatrix} 14 & 07 & 10 & 13 \\ 08 & 07 & 06 & 11 \\ 11 & 08 & 18 & 18 \end{bmatrix}} = \underset{P}{\begin{bmatrix} 02 & 14 & 03 & 04 \\ 08 & 18 & 17 & 04 \\ 00 & 03 & 24 & 25 \end{bmatrix}} \underset{K}{\begin{bmatrix} 09 & 07 & 11 & 13 \\ 04 & 07 & 05 & 06 \\ 02 & 21 & 14 & 09 \\ 03 & 23 & 21 & 08 \end{bmatrix}}$$

**a. Encryption**

$$\underset{P}{\begin{bmatrix} 02 & 14 & 03 & 04 \\ 08 & 18 & 17 & 04 \\ 00 & 03 & 24 & 25 \end{bmatrix}} = \underset{C}{\begin{bmatrix} 14 & 07 & 10 & 13 \\ 08 & 07 & 06 & 11 \\ 11 & 08 & 18 & 18 \end{bmatrix}} \underset{K^{-1}}{\begin{bmatrix} 02 & 15 & 22 & 03 \\ 15 & 00 & 19 & 03 \\ 09 & 09 & 03 & 11 \\ 17 & 00 & 04 & 07 \end{bmatrix}}$$

**b. Decryption**

# Hill Cipher

## Encryption

Plaintext Key Ciphertext

$$\begin{bmatrix} a & b & c \\ a & b & c \\ a & b & c \end{bmatrix} = \begin{bmatrix} 00 & 01 & 02 \\ 00 & 01 & 02 \\ 00 & 01 & 02 \end{bmatrix} \quad X \quad \begin{bmatrix} 2 & 1 & 2 \\ 0 & 2 & 1 \\ 5 & 2 & 3 \end{bmatrix} = \quad : \begin{bmatrix} 10 & 06 & 7 \\ 10 & 06 & 7 \\ 10 & 06 & 7 \end{bmatrix}$$

## Decryption

Ciphertext $\text{Key}^{-1}$ Plaintext

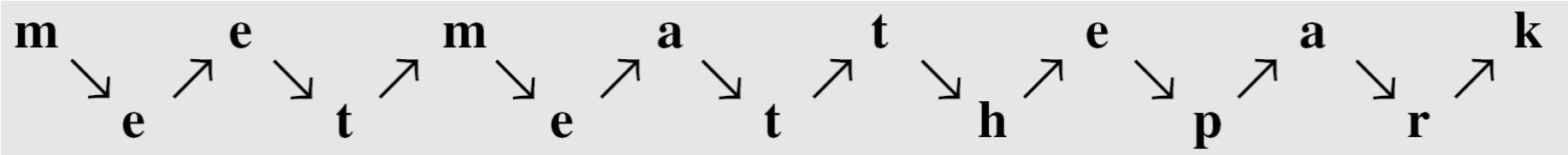
$$\begin{bmatrix} 10 & 06 & 7 \\ 10 & 06 & 7 \\ 10 & 06 & 7 \end{bmatrix} \quad X \quad \begin{bmatrix} -4/7 & -1/7 & 3/7 \\ -5/7 & 4/7 & 2/7 \\ 10/7 & -1/7 & -4/7 \end{bmatrix} \quad = \quad \begin{bmatrix} 00 & 01 & 02 \\ 00 & 01 & 02 \\ 00 & 01 & 02 \end{bmatrix}$$

There are two methods for permutation of characters.

Text is written column by column and then transmitted row by row.
Text is written row by row and then transmitted column by column.

A good example of the first method is the **Rail Fence Cipher.**

**For example: Meet me at the park**




She then creates the ciphertext "MEMATEAKETETHPR".

# Keyless Transposition Cipher

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 "MMTAEEHREAEKTTPZ".

# Keyed Transposition Cipher

Another method is to divide the plaintext into groups of predetermined size called blocks.

Then key is used to permute the characters in each block separately.
Alice needs to send the message “Enemy attacks tonight” to Bob.

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

Encryption ↓

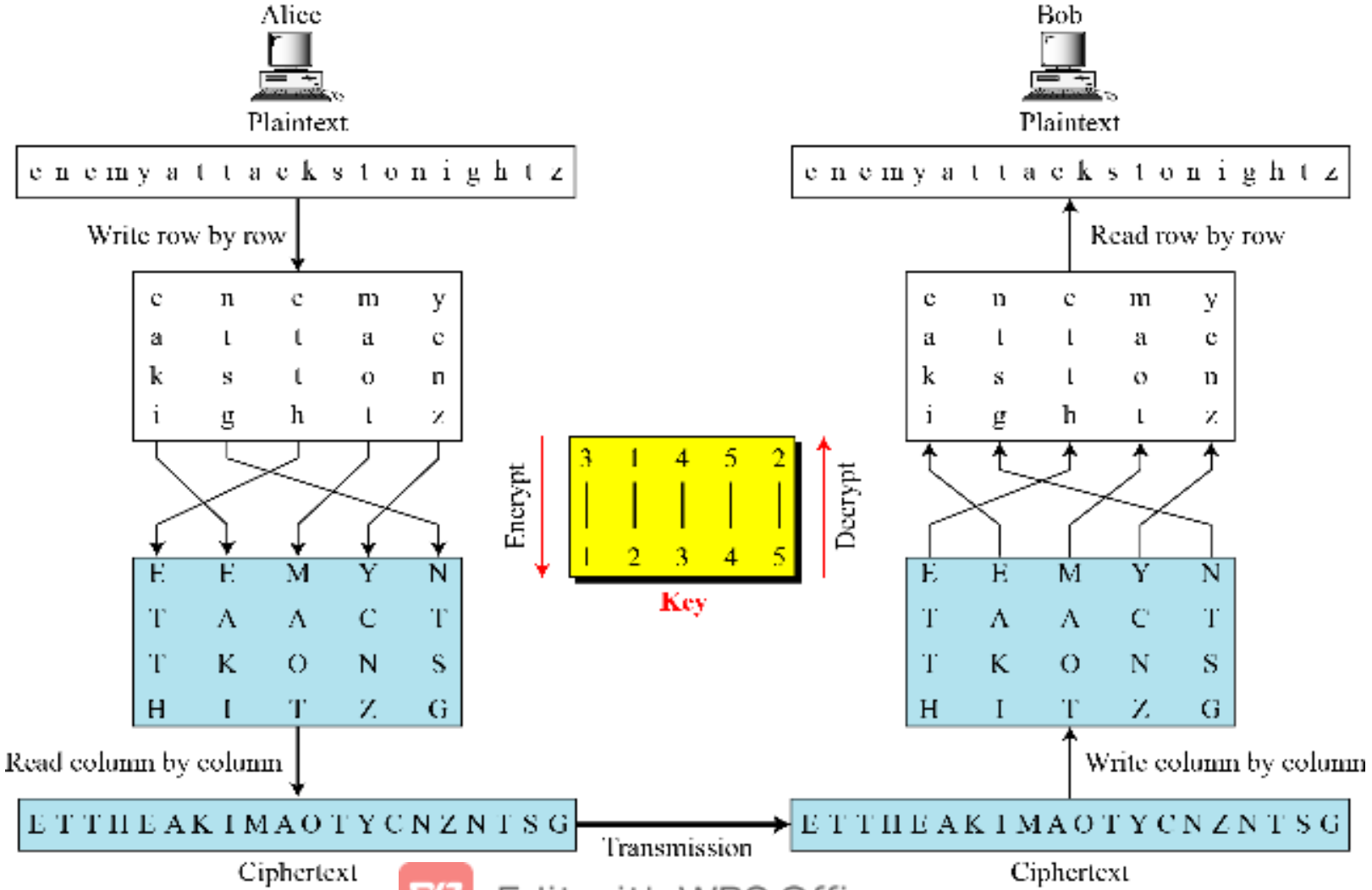
| 3 | 1 | 4 | 5 | 2 |
|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 |

↑ Decryption

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

# Combining Two Approaches

Most recent transposition ciphers combine the two approaches to achieve better scramble.

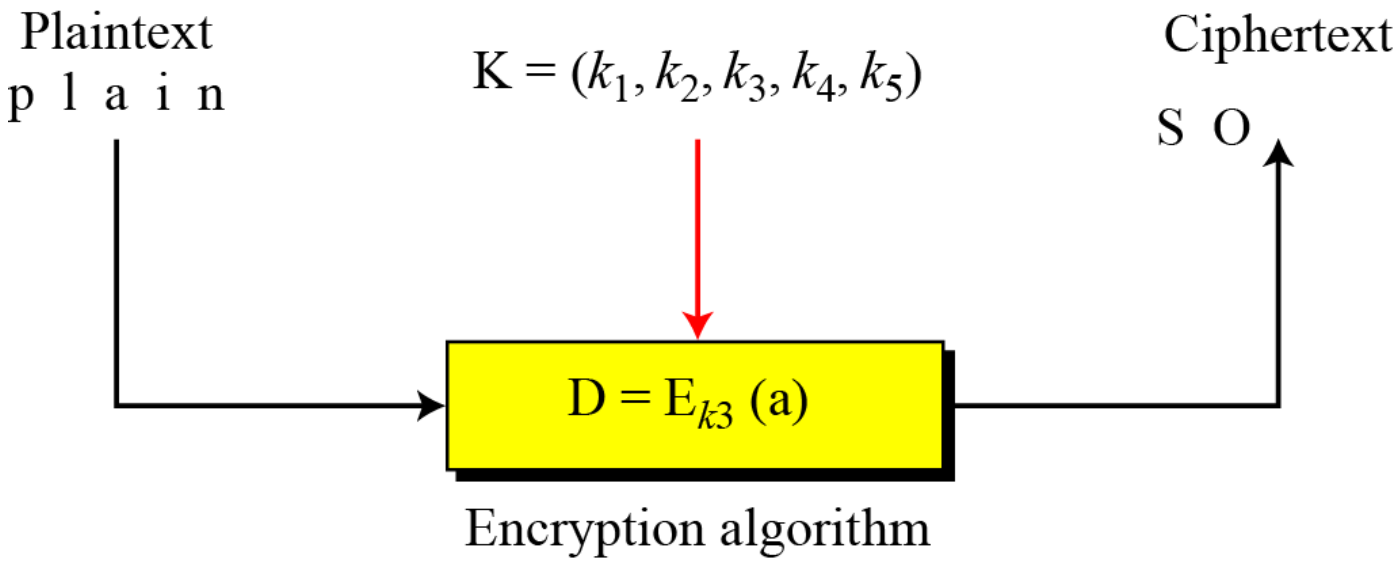


# Stream Ciphers

Plaintext stream (P), Ciphertext stream(C), and the key stream (K)

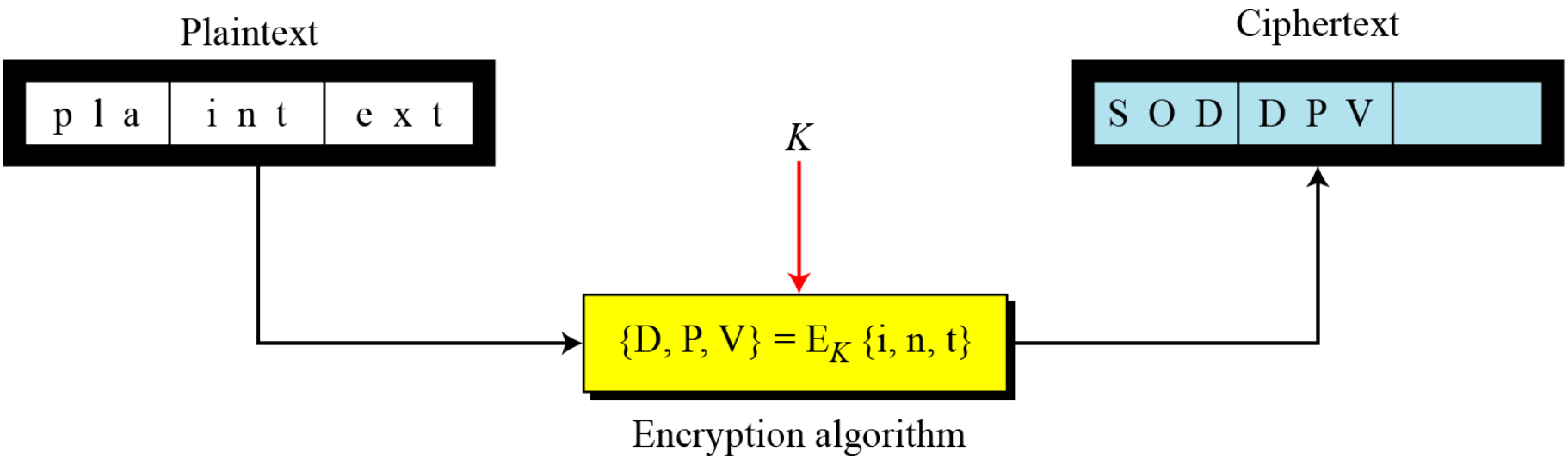
$$P = P_1P_2P_3, \ldots \qquad C = C_1C_2C_3, \ldots \qquad K = (k_1, k_2, k_3, \ldots)$$

$$C_1 = E_{k1}(P_1) \qquad C_2 = E_{k2}(P_2) \qquad C_3 = E_{k3}(P_3) \ldots$$



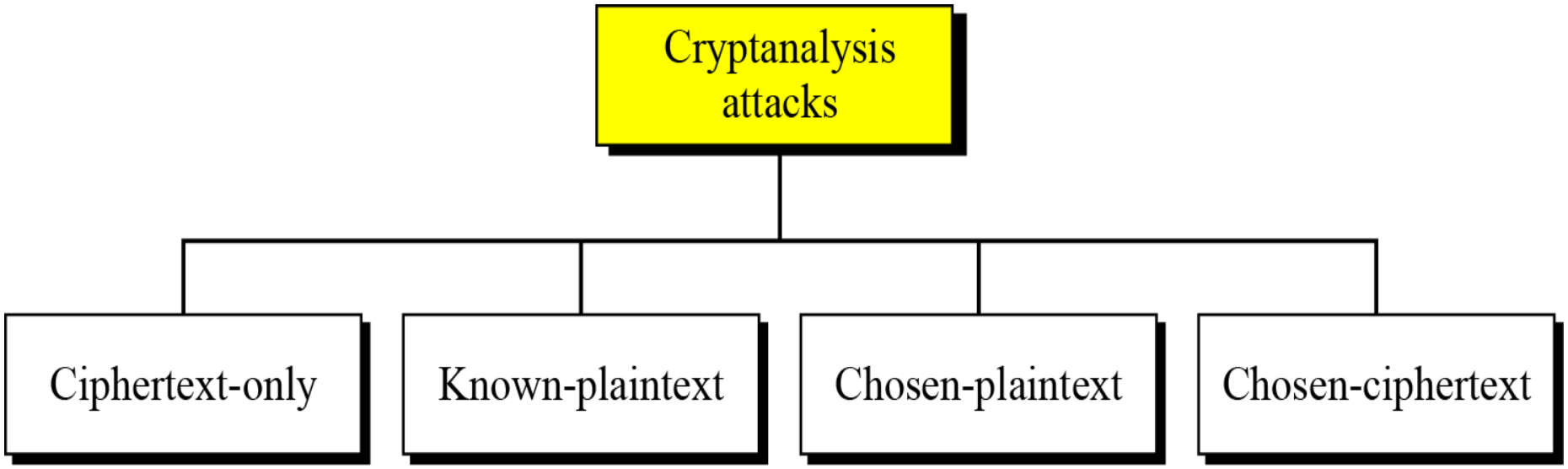
# Block Ciphers

In a block cipher, a group of plaintext symbols of size m ( m > 1) are encrypted together creating a group of ciphertext of the same size.



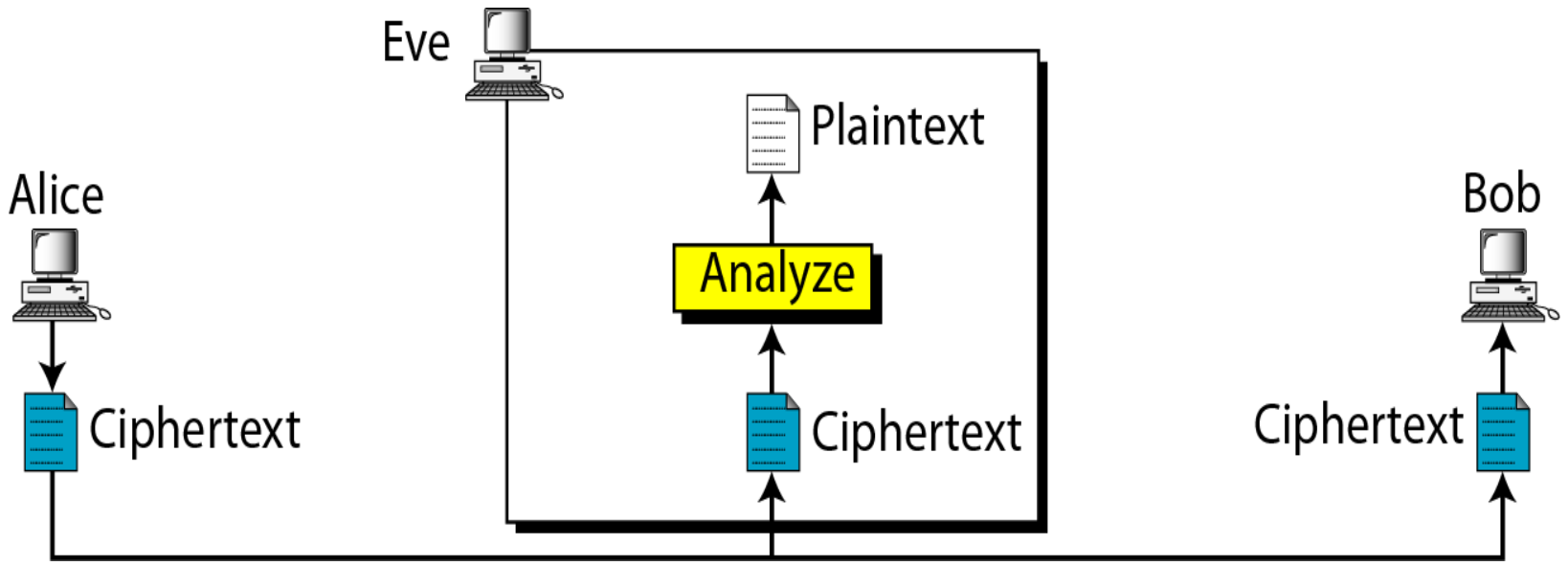
# Cryptanalysis

As cryptography is the science and art of creating secret codes, **Cryptanalysis** is the science and art of breaking those codes.



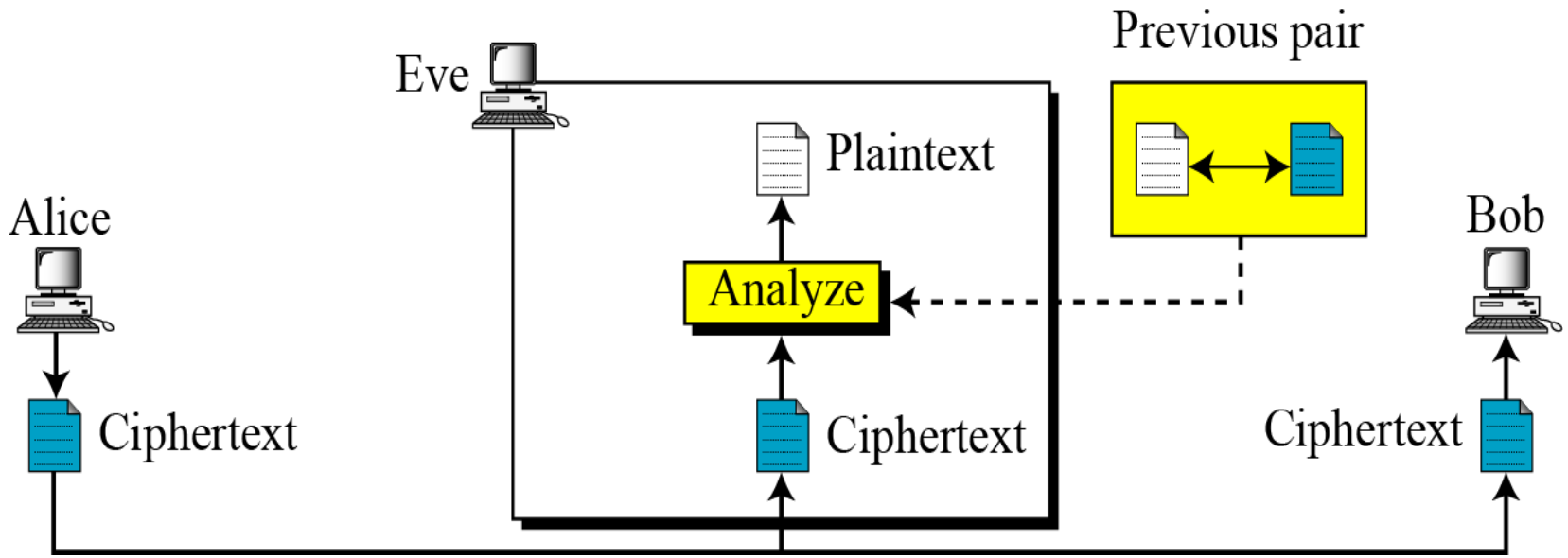
# Ciphertext Only Attack

Here the assumption is that Eve knows the algorithm and can intercept the ciphertext.



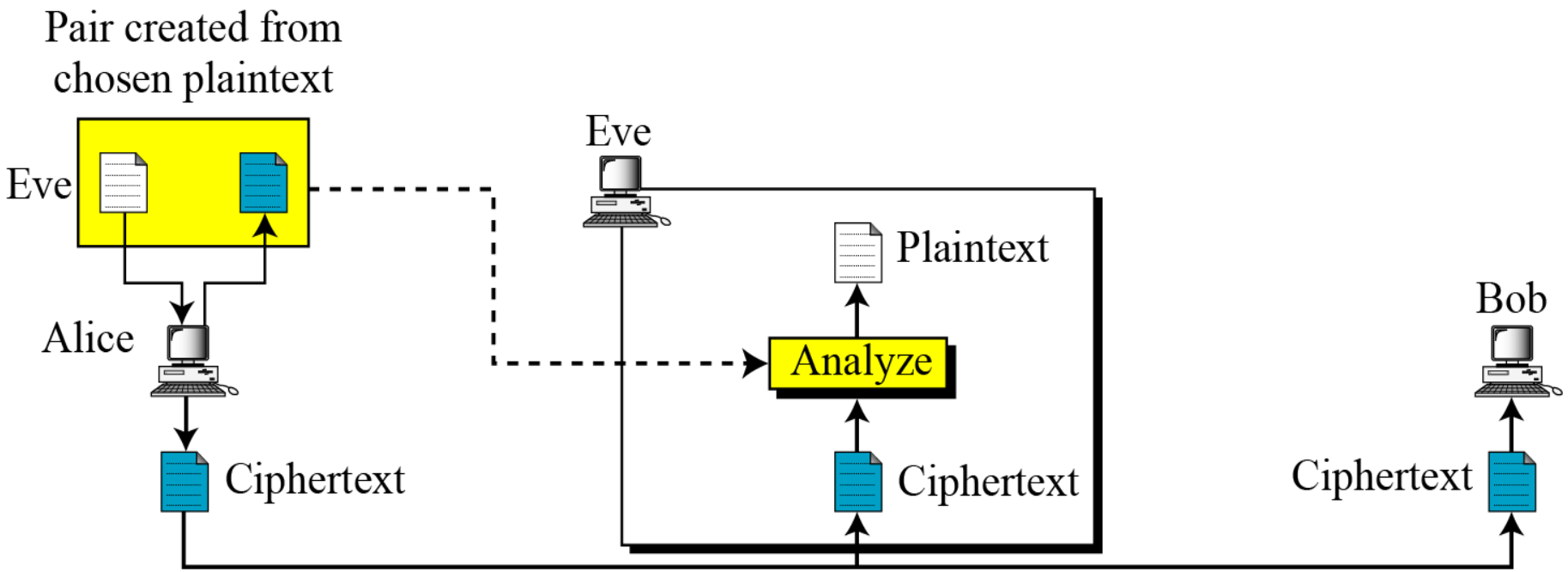
# Known-Plaintext Attack

Eve has access to some plaintext/ciphertext pairs in addition to the intercepted ciphertext. (Less likely to happen)



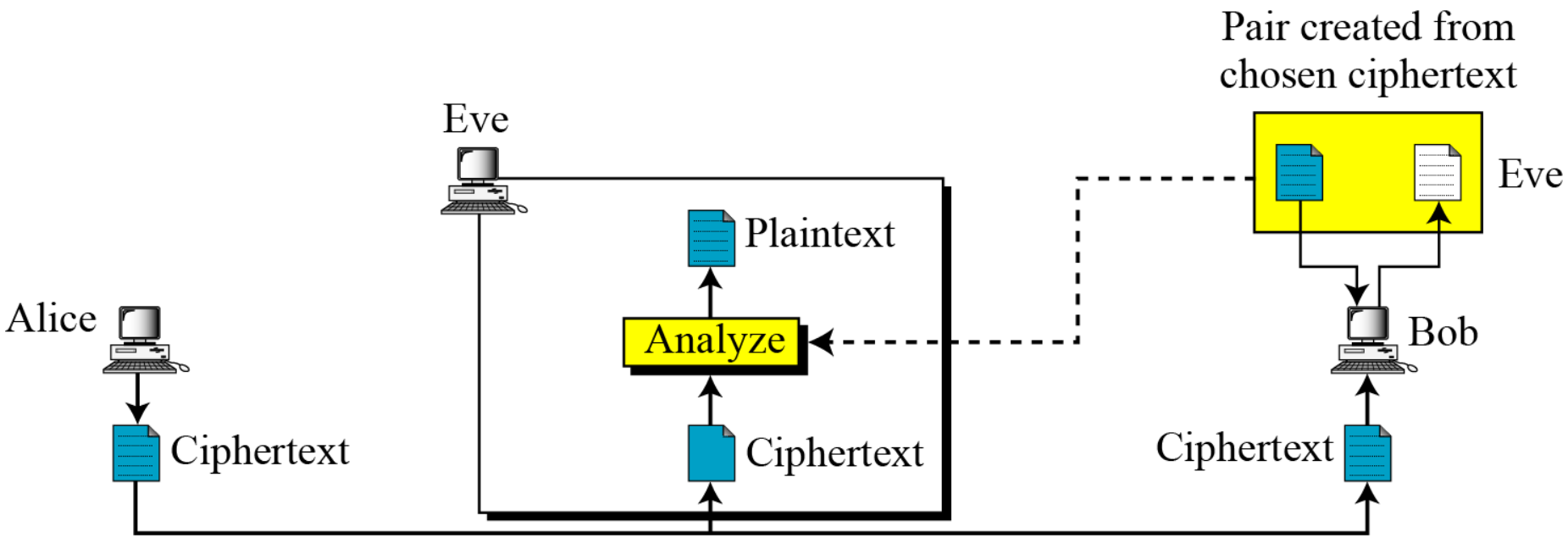
# Chosen- Plaintext Attack

Similar to known-plaintext attack except the plaintext/ciphertext pairs have been chosen by the attacker. (Less likely to happen)



# Chosen-Cipher Attack

Similar to known-plaintext attack except the ciphertext/plaintext pairs have been chosen by the attacker. (Less likely to happen)



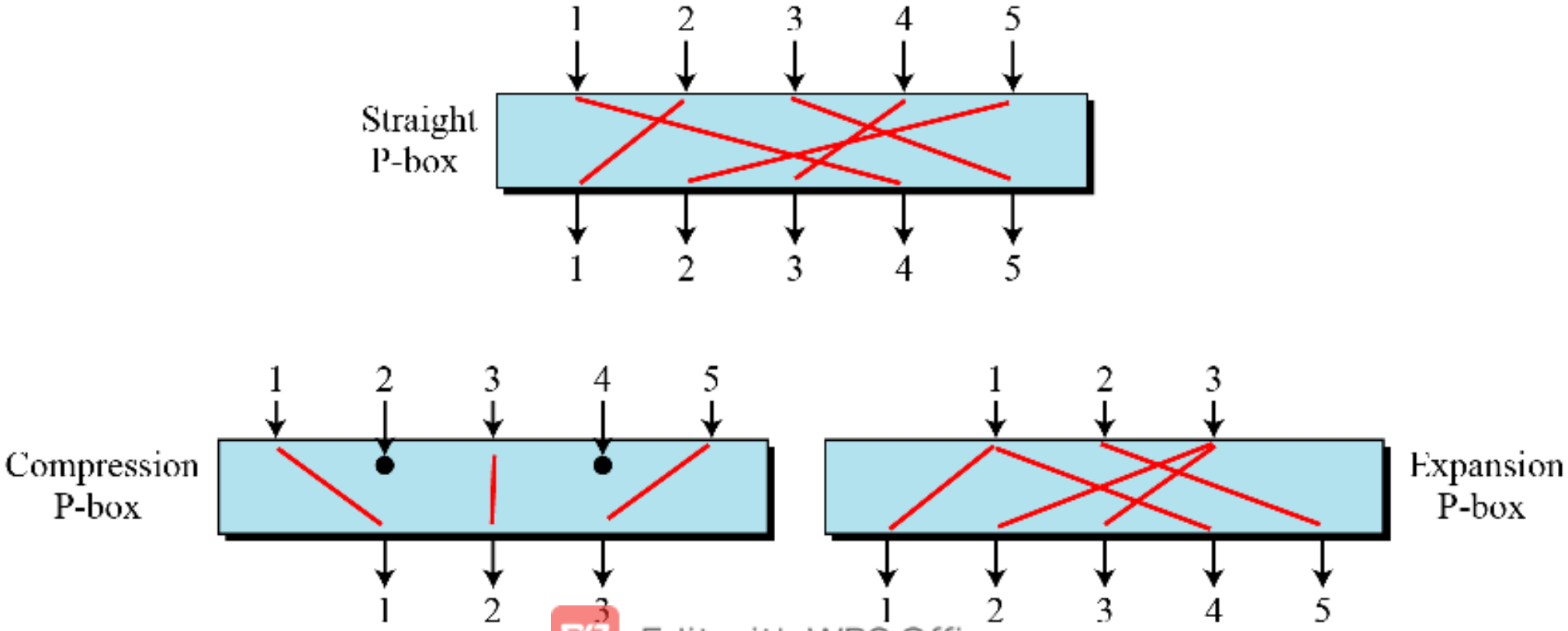
# Modern Block Cipher

- Modern block ciphers can be Stream or Block.
- Modern block ciphers are based on bit level data.
- However traditional ciphers are all based on character level data.
- Stream ciphers takes each data bit by bit for encryption.
- Block ciphers takes block of data for encryption/decryption.

# Components of Modern Block Cipher

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

*Three types of P-boxes*



# Few examples for P-box

*Example of a 32 × 24 compression P-box*

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

*Example of a 64 x 64 straight P-box*

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 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 |

# Substitution box (S-box)

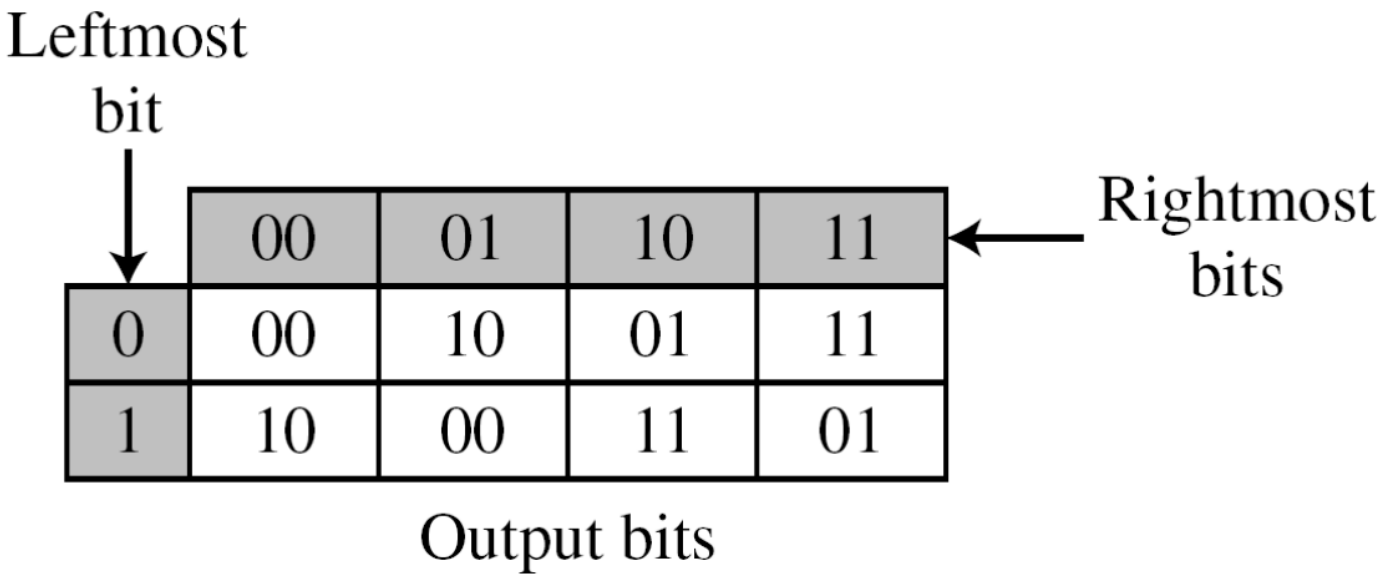
An S-box is an m × n substitution unit, where m and n are not necessarily the same.

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.

# Substitution box (S-box)




Based on the table, an input of 010 yields the output 01. An input of 101 yields the output of 00.

# Product cipher

A product cipher is a complex cipher combining substitution, permutation, and other components.

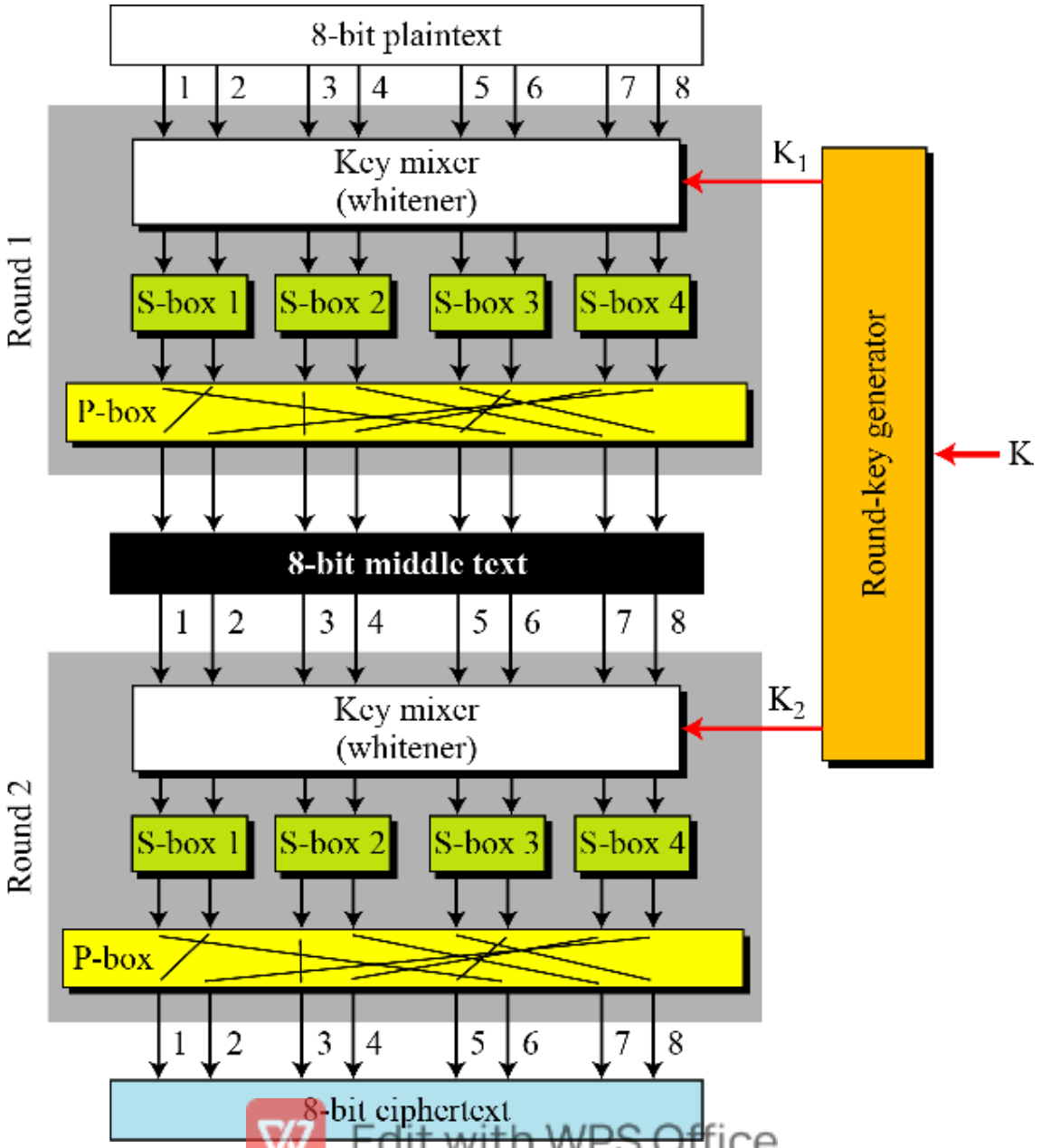
***Diffusion***

The idea of diffusion is to hide the relationship between the ciphertext and the plaintext.

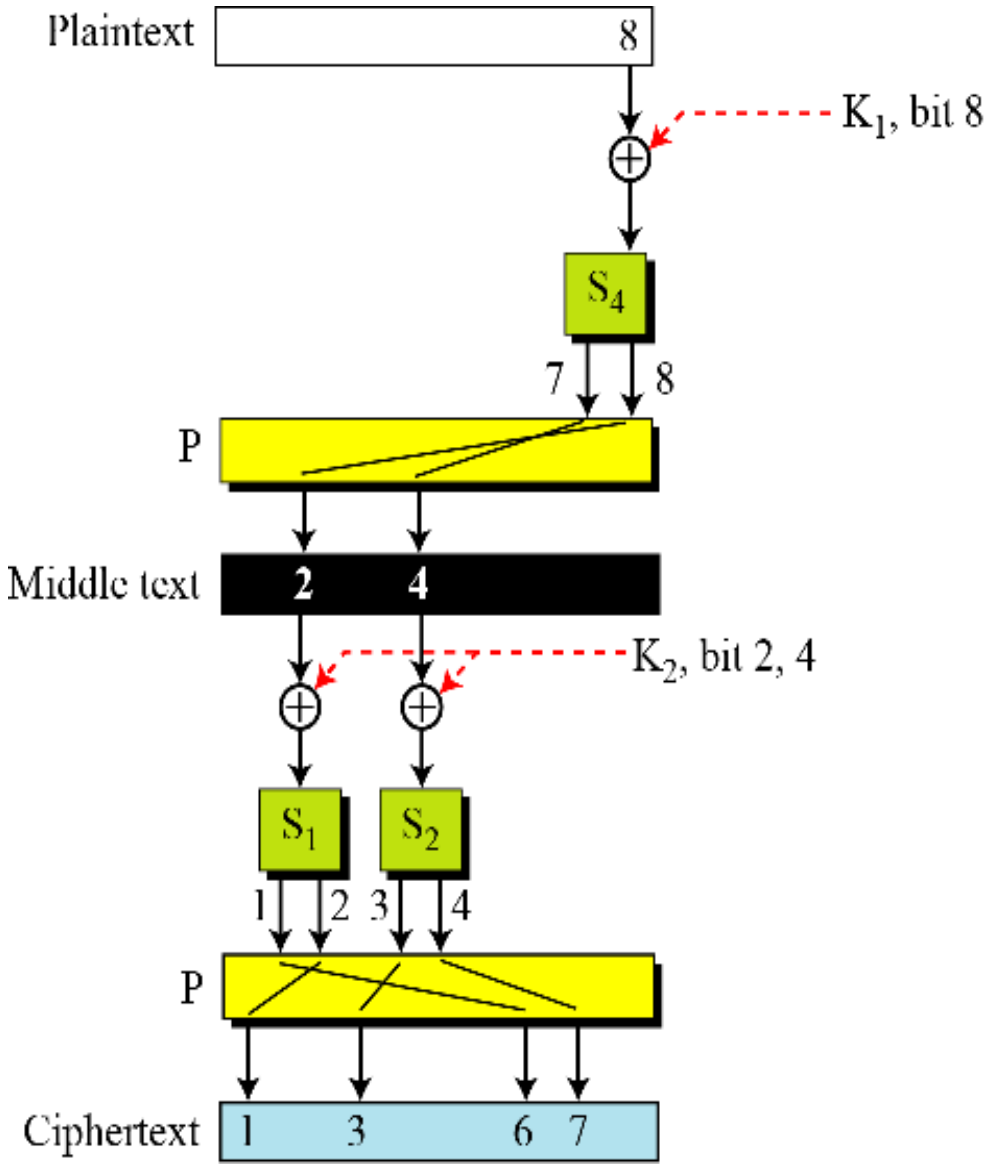
***Confusion***

The idea of confusion is to hide the relationship between the ciphertext and the key.

# Product cipher of two rounds



# Diffusion and confusion in a block cipher

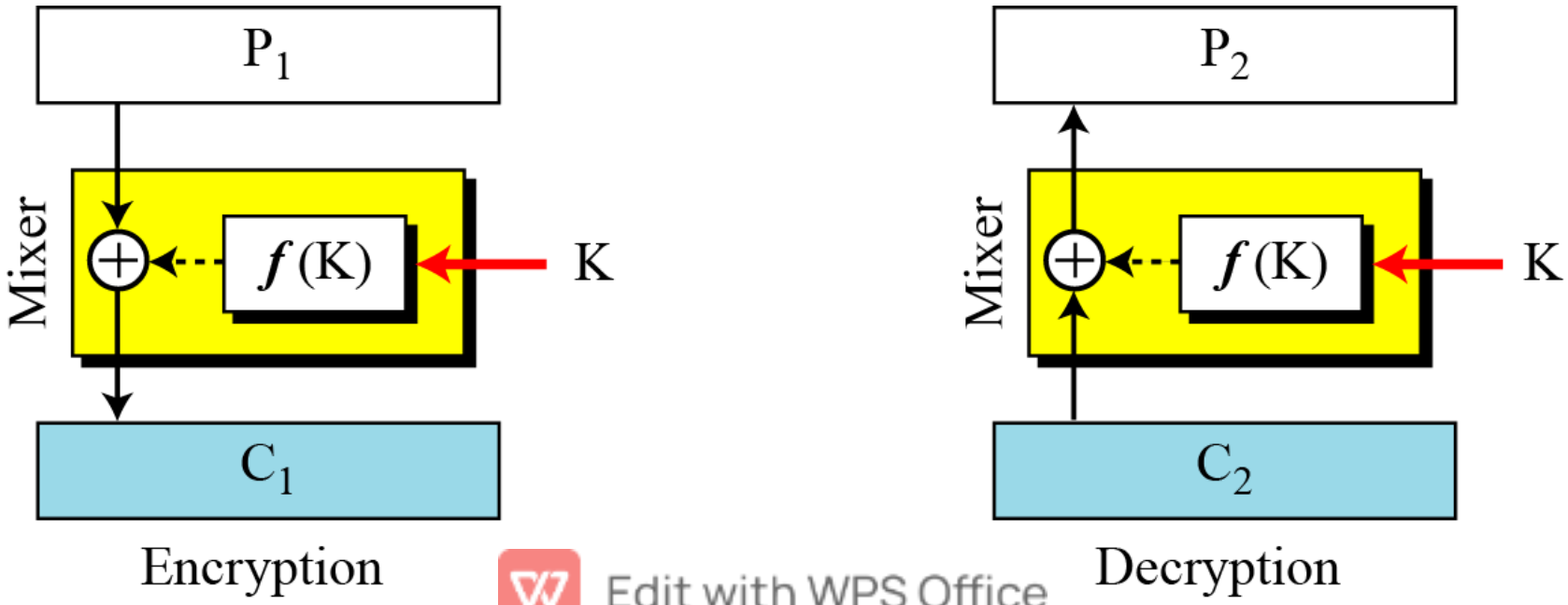


# Class of Product ciphers

1. Feistel ciphers – DES algorithm
2. Non-Feistel ciphers- AES algorithm

A Feistel cipher make uses of both invertible, self-invertible and non-invertible components.

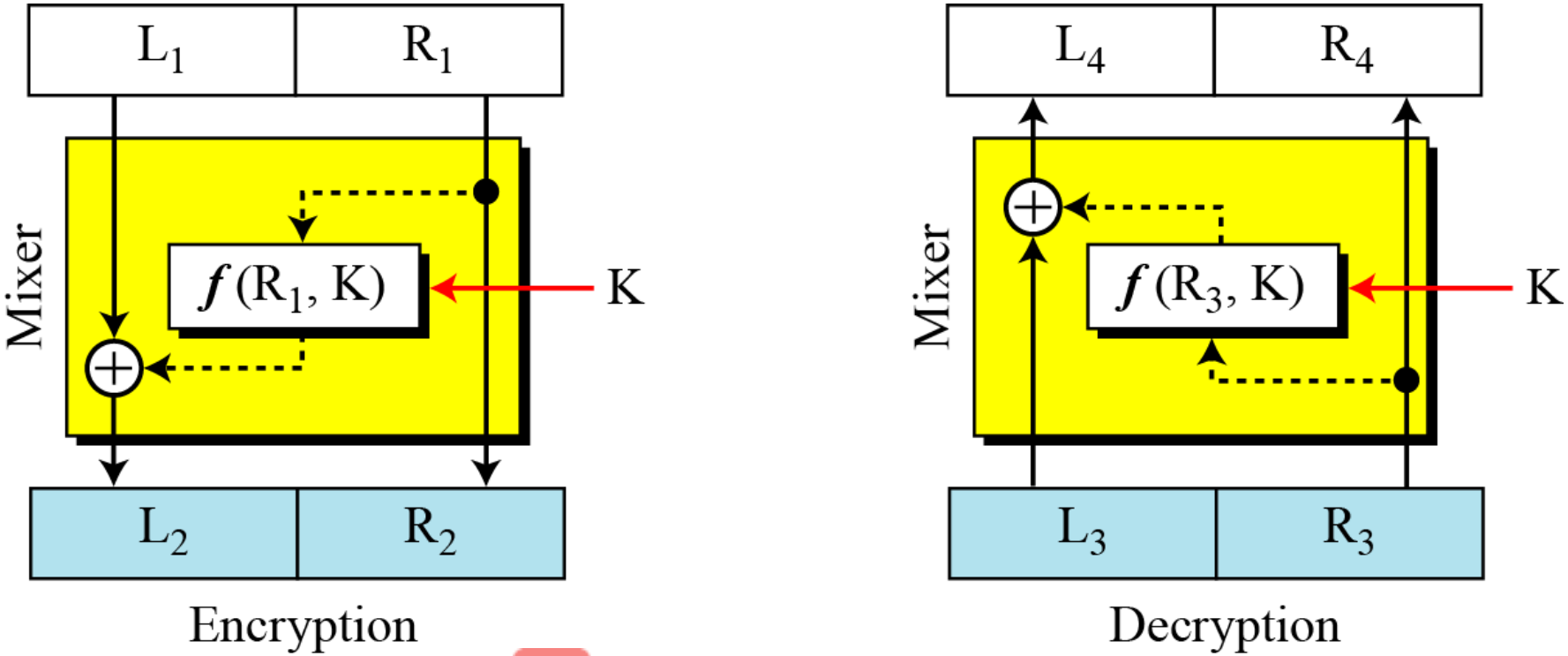
First thought



# Feistel Cipher

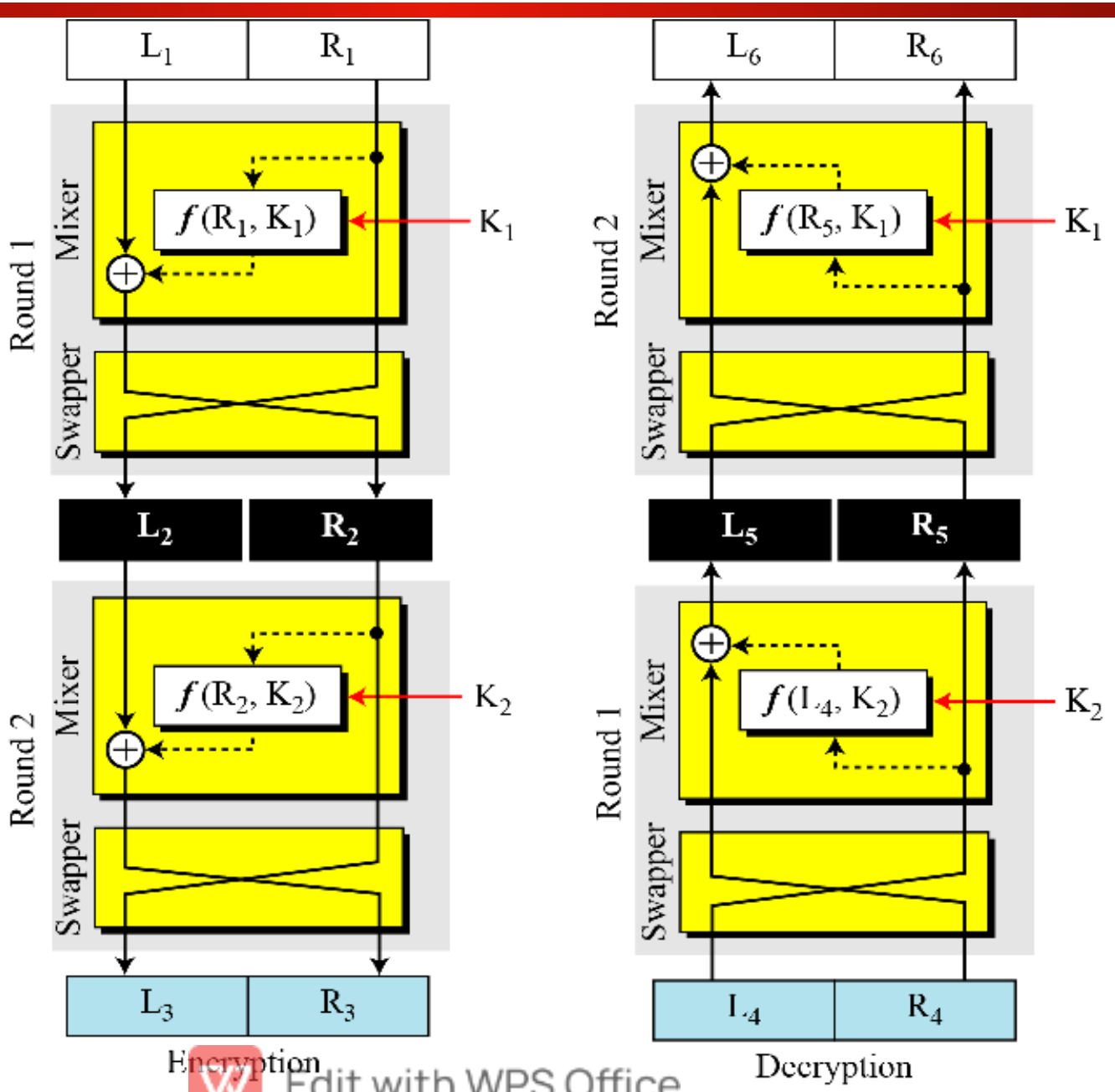
Here we want the input to the function to be also be a part of plaintext/ciphertext apart from key.

## Improvement



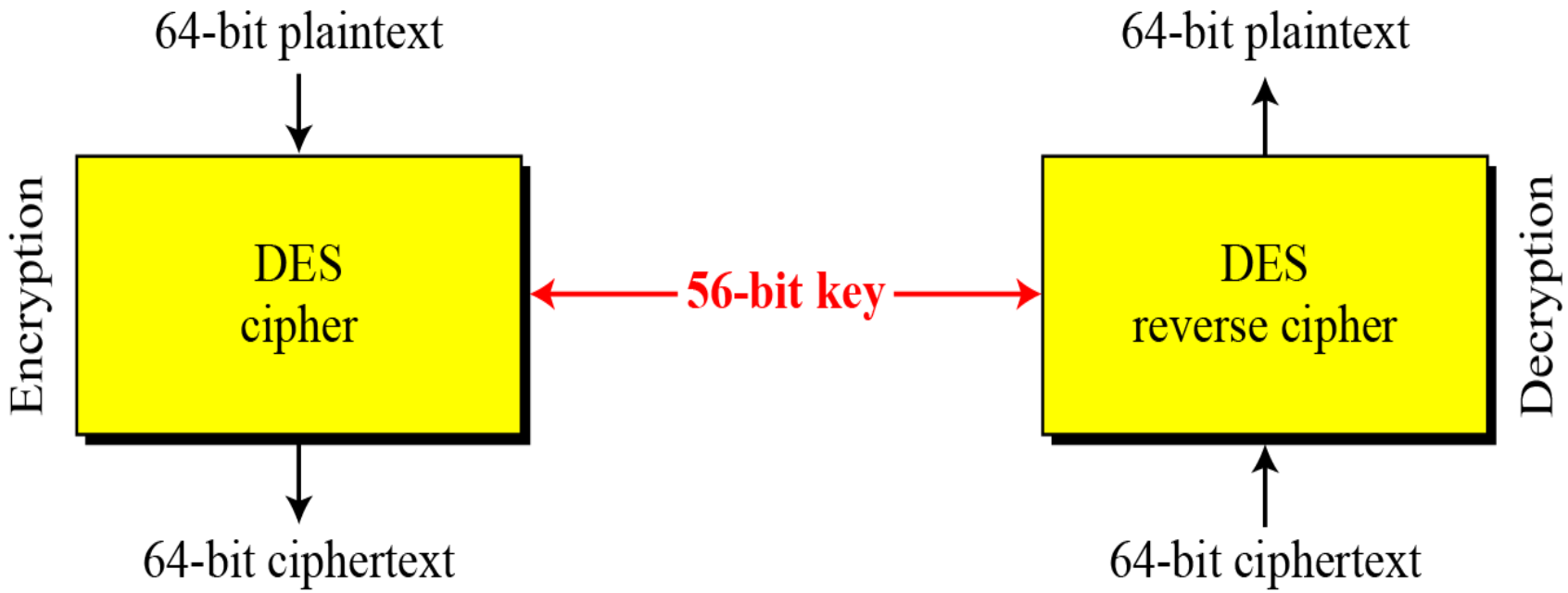
# Feistel Cipher

## Final design

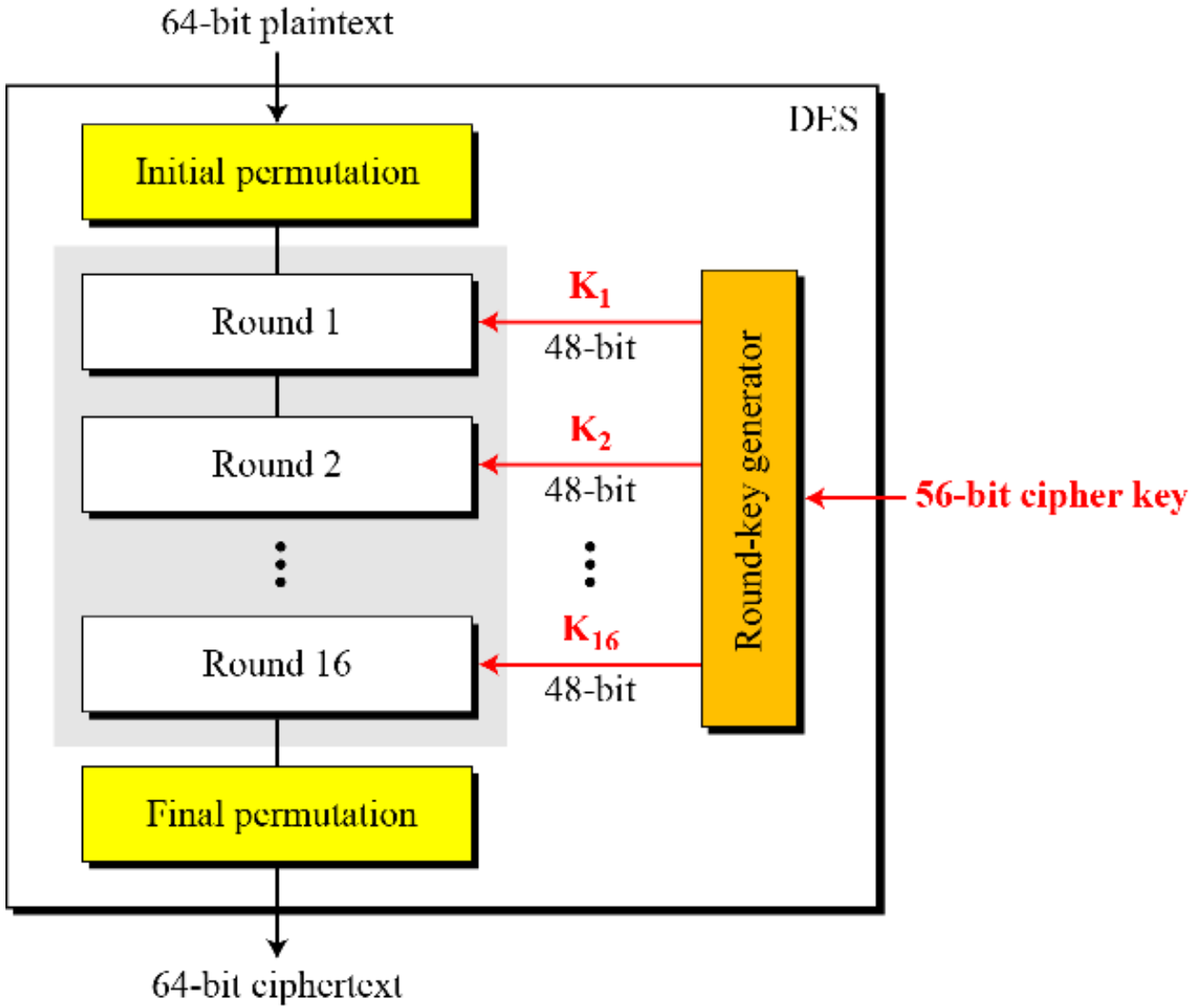


# DES (Data Encryption Standard) cipher

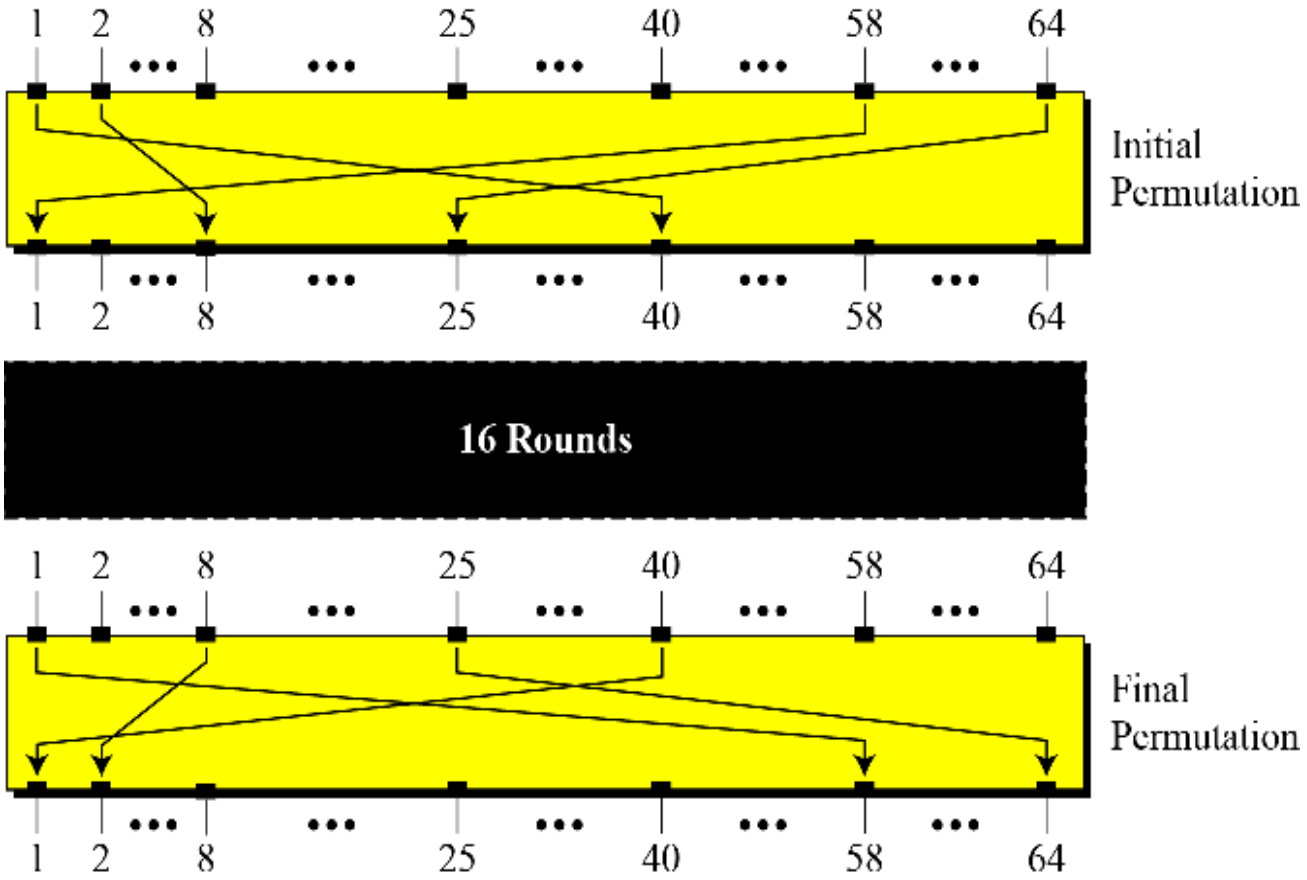
*DES is a block cipher, as shown in Figure.*



# General structure of DES



# Initial and Final Permutation



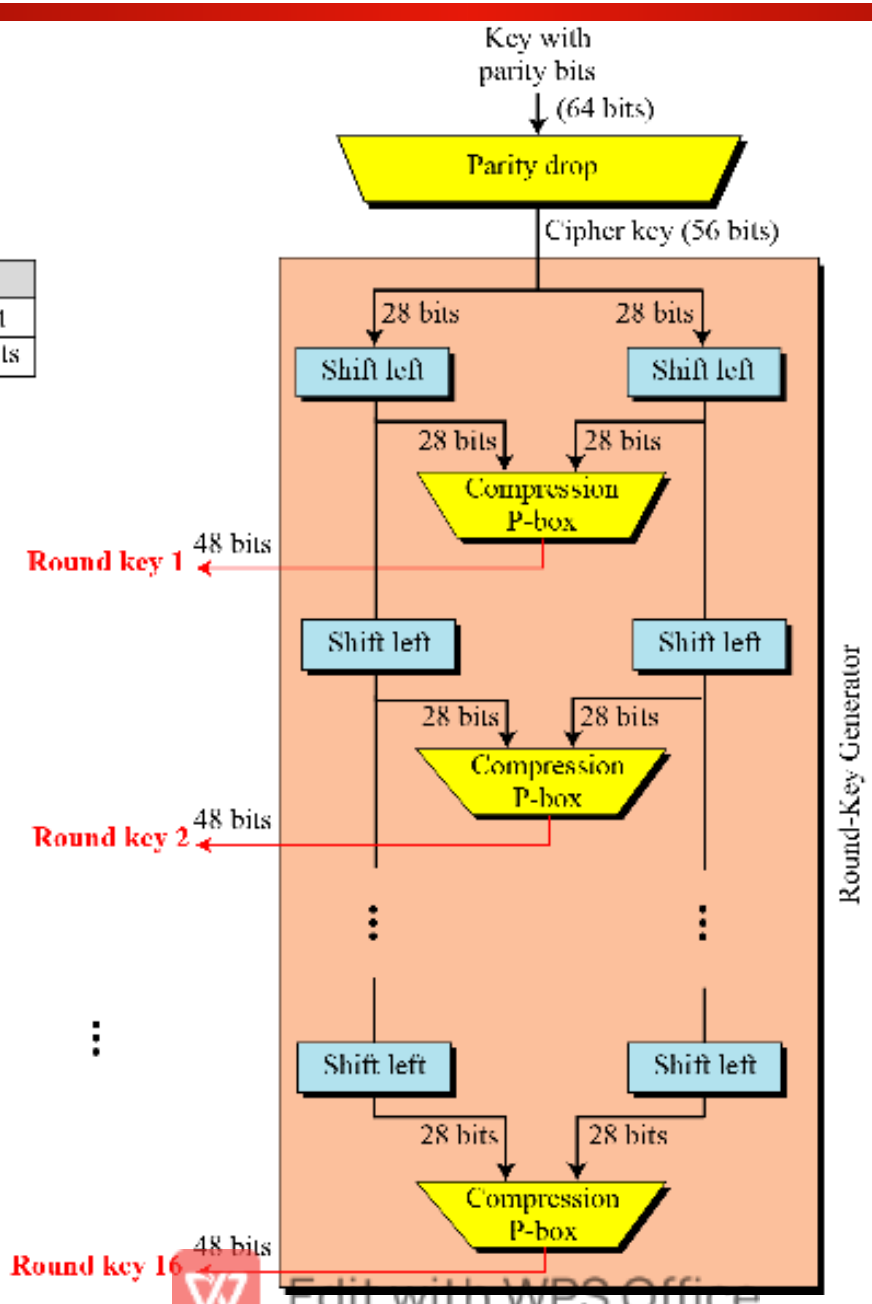
# Initial and Final Permutation Table

| *Initial Permutation* | *Final Permutation* |
|---|---|
| 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 |

# DES Key generation

Shifting

| Rounds | Shift |
|---|---|
| 1, 2, 9, 16 | one bit |
| Others | two bits |



# DES Key generation

*Parity-bit drop 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 |

*Number of bits shifts*

| 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 |

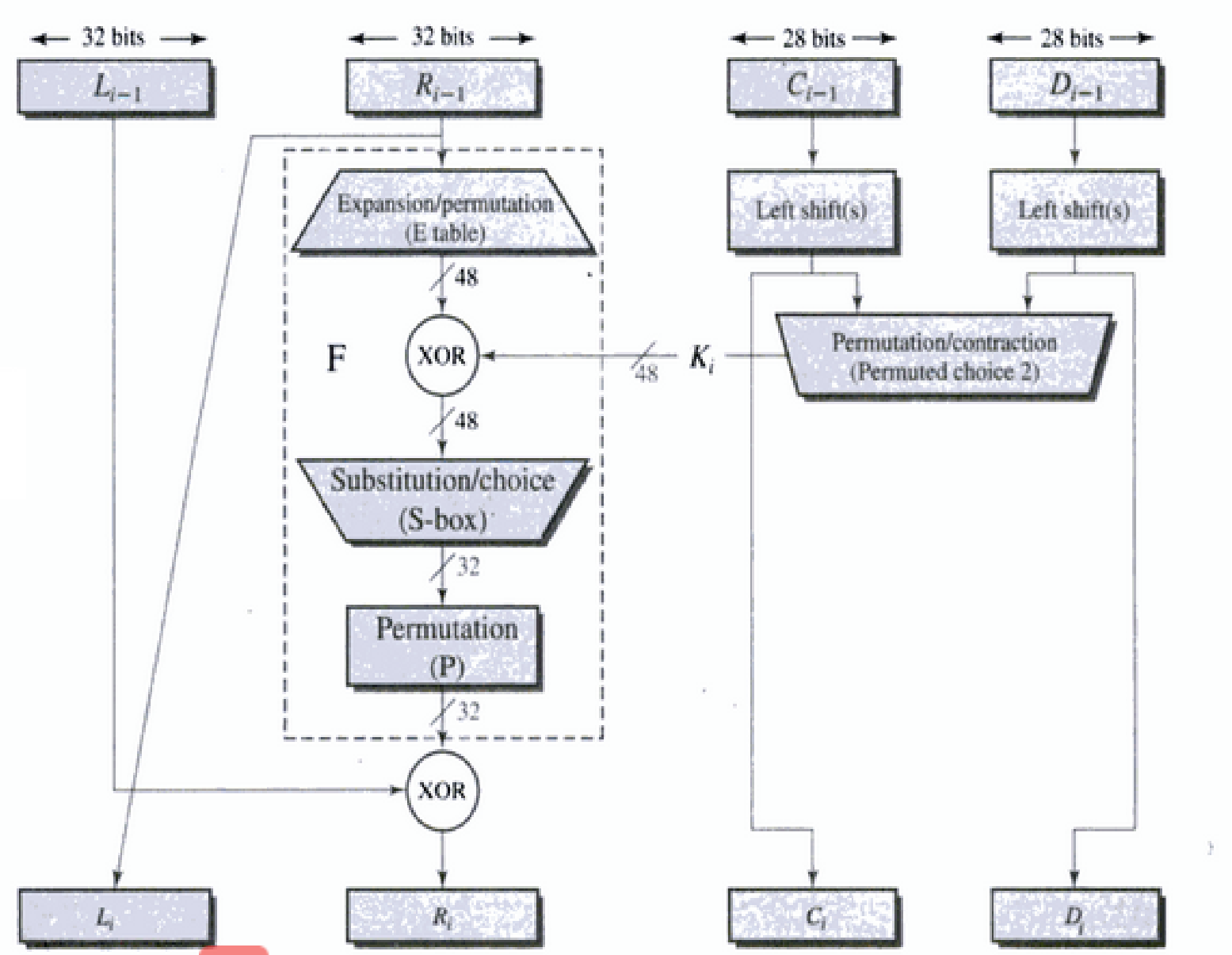
# DES Key generation

*Key-compression table*

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 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 |

# DES Round Structure

$$L_i = R_{i-1}$$
$$R_i = L_{i-1} \times F(R_{i-1}, K_i)$$



# DES Round Structure

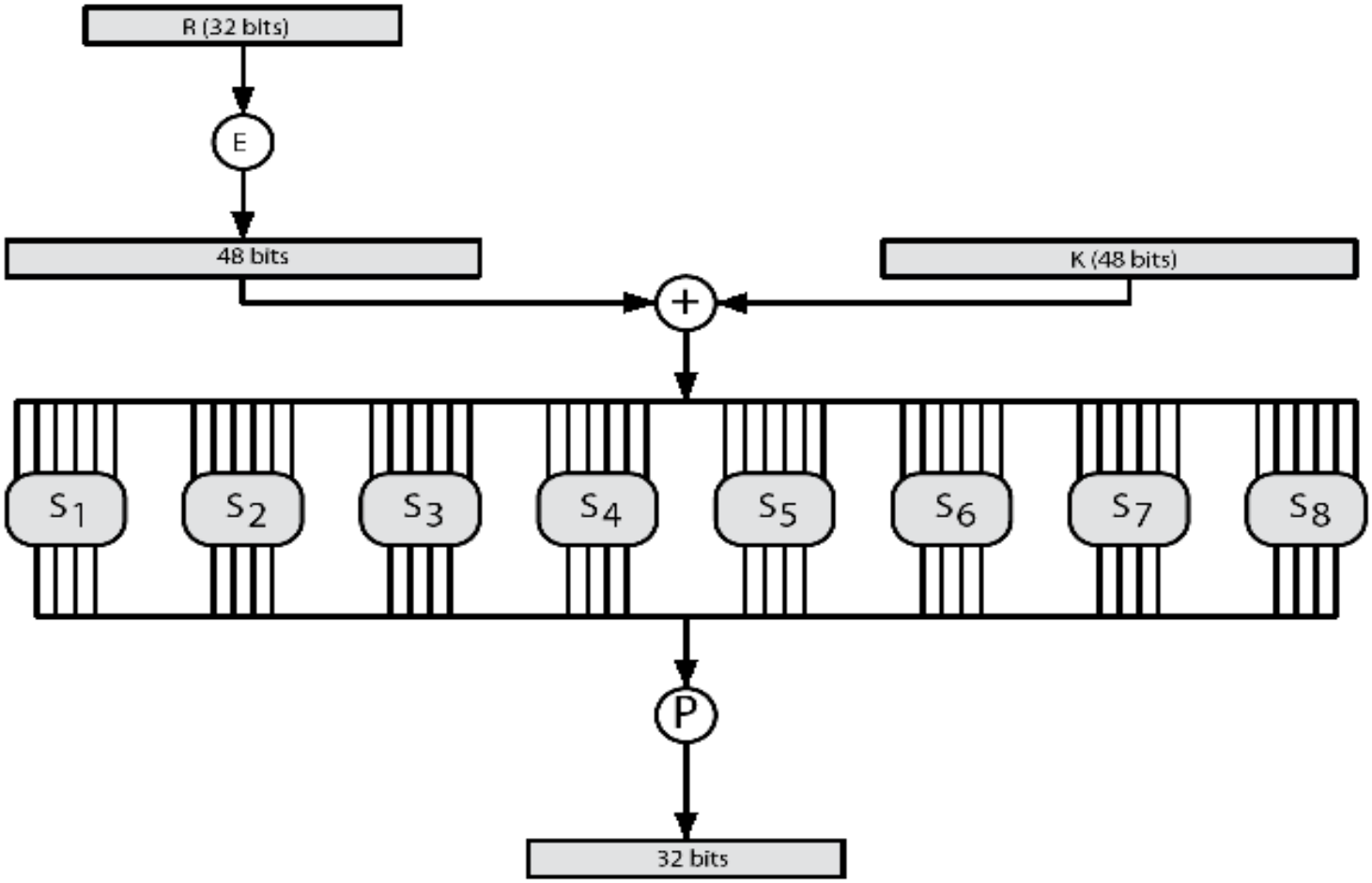
## *Expansion P-box table*

| | | | | | |
|---|---|---|---|---|---|
| 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 |

## *Straight permutation table*

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 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 |

# DES S-box design



# DES S-box

*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 |

The input to S-box 1 is 100011. What is the output?

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 of the Table of S-box 1. The result is 12 in decimal, which in binary is 1100. So the input 100011 yields the output 1100.

# Avalanche effect

A small change in either the plaintext or the key, should produce the significant change in the ciphertext.

**State 1:**

Two plaintext that differ by one bit were used:

00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
10000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

with the key

0000001 1001011 0100100 1100010 0011100 0011000 0011100 0110010

**State 2:**

Single plaintext input with two different key :

01101000 10000101 00101111 01111010 00010011 01110110 11101011 10100100

with two keys that differ in only one bit position:

1110010 1111011 1101111 0011000 0011101 0000100 0110001 11011100
0110010 1111011 1101111 0011000 0011101 0000100 0110001 11011100

# Avalanche effect

Table 3.5 Avalanche Effect in DES

| (a) Change in Plaintext | | (b) Change in Key | |
|---|---|---|---|
| Round | Number of bits that differ | Round | Number of bits that differ |
| 0 | 1 | 0 | 0 |
| 1 | 6 | 1 | 2 |
| 2 | 21 | 2 | 14 |
| 3 | 35 | 3 | 28 |
| 4 | 39 | 4 | 32 |
| 5 | 34 | 5 | 30 |
| 6 | 32 | 6 | 32 |
| 7 | 31 | 7 | 35 |
| 8 | 29 | 8 | 34 |
| 9 | 42 | 9 | 40 |
| 10 | 44 | 10 | 38 |
| 11 | 32 | 11 | 31 |
| 12 | 30 | 12 | 33 |
| 13 | 30 | 13 | 28 |
| 14 | 26 | 14 | 26 |
| 15 | 29 | 15 | 34 |
| 16 | 34 | 16 | 35 |

# Strength of DES

- The use of 56-bit keys:

  With a key length of 56 bits, there ar $2^{56}$ possible keys,

  which is equivalent t $7.2 \times 10^{16}$ keys.

- Nature of DES algorithm

  The design of algorithm specially the design of S-boxes are not made public.

- Timing Attacks

  Timing attack is one in which information about the key or plaintext is obtained by observing how long it takes to perform decryption on various ciphertexts.

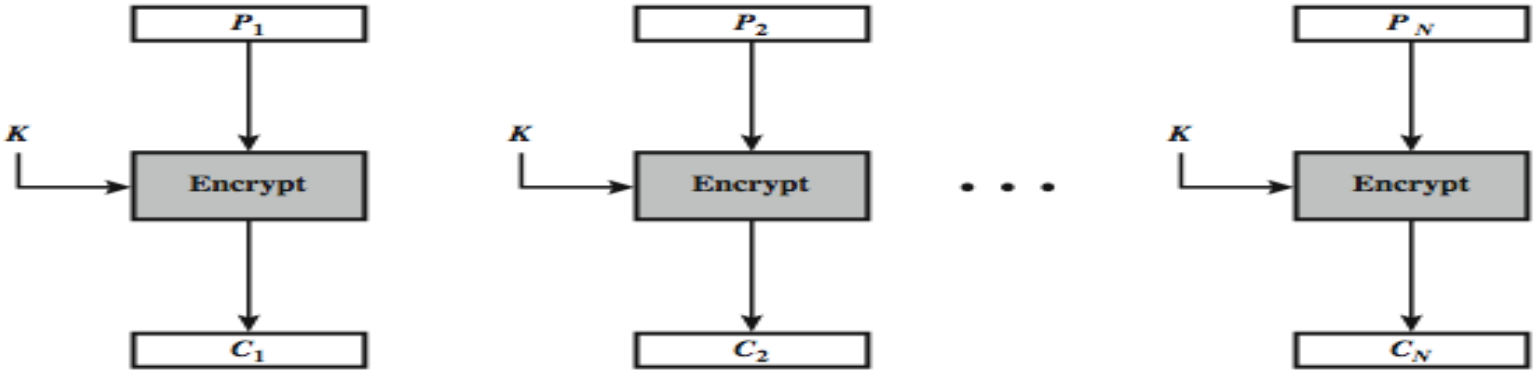
# Block Cipher Modes of Operation

- To apply DES in variety of applications, different modes of operation have been defined.

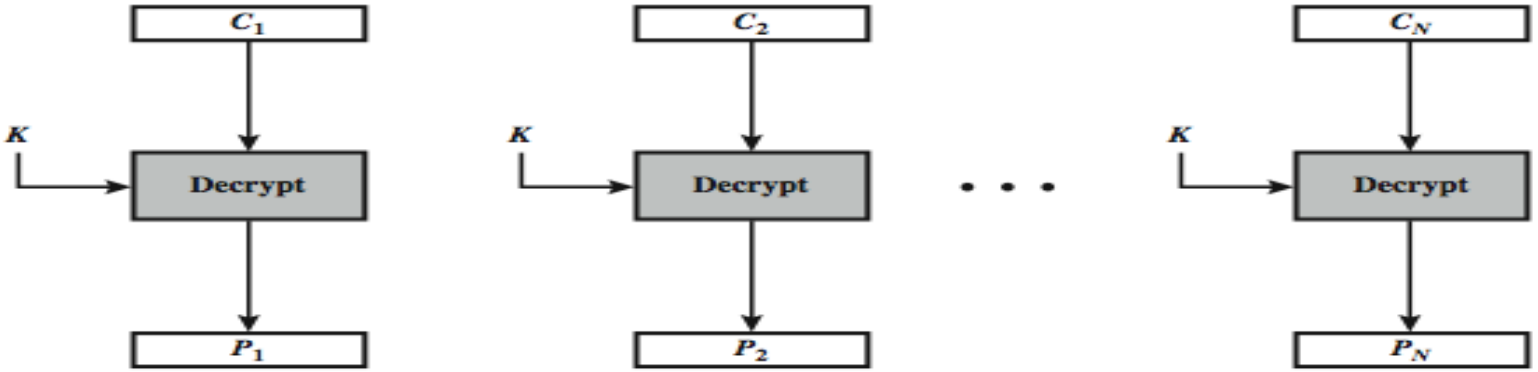
- Electronic Codebook(ECB)
- Cipher Block Chaining(CBC)
- Cipher Feedback(CFB)
- Output Feedback(OFB)
- Counter (CTR)

# Electronic Codebook Mode

- The term codebook is used because for a given key, there is a unique ciphertext for every 64 bit of plaintext.



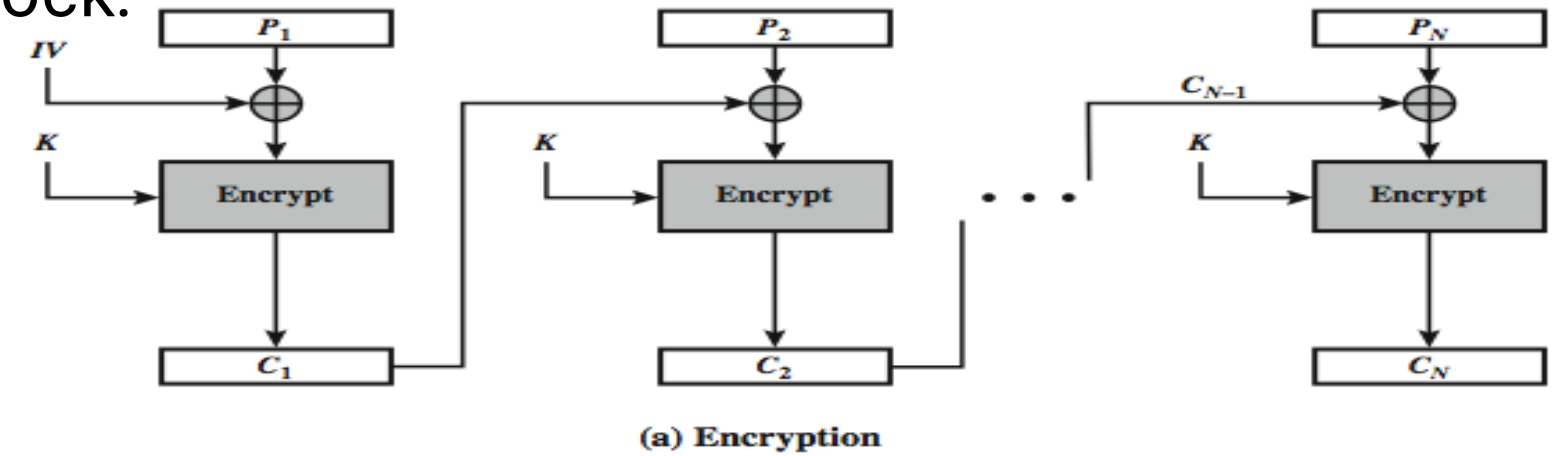

(a) Encryption



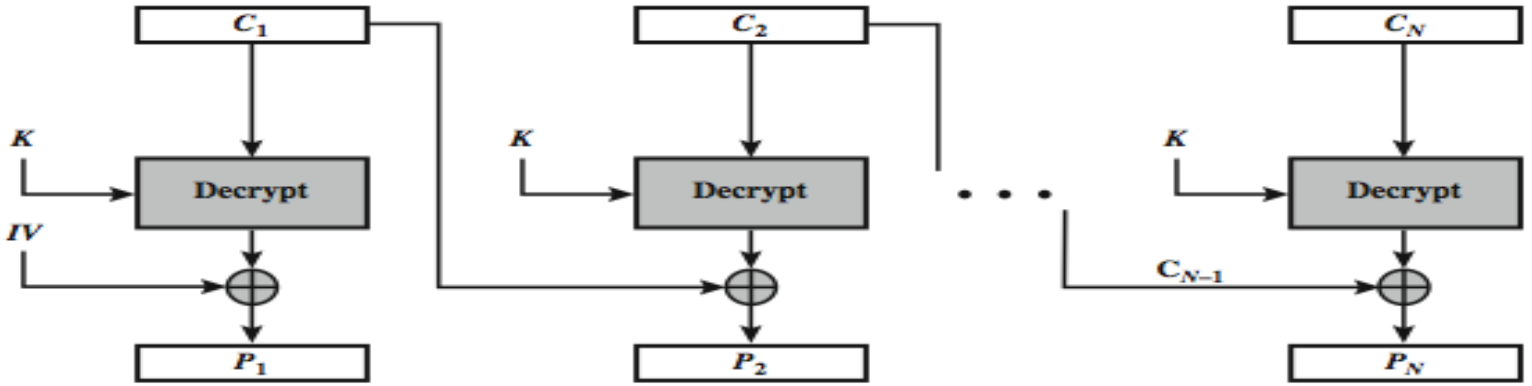

(b) Decryption

# Cipher Block Chaining Mode

- Input to the encryption algorithm is the XOR of the current plaintext block and the preceding ciphertext block.



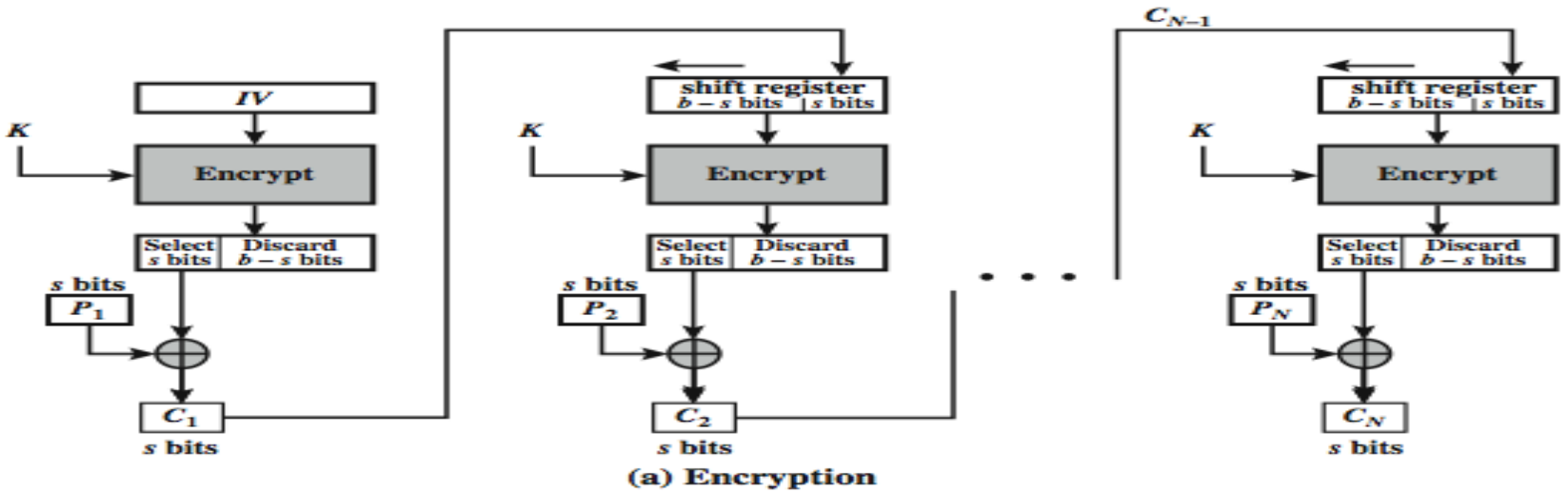

(a) Encryption



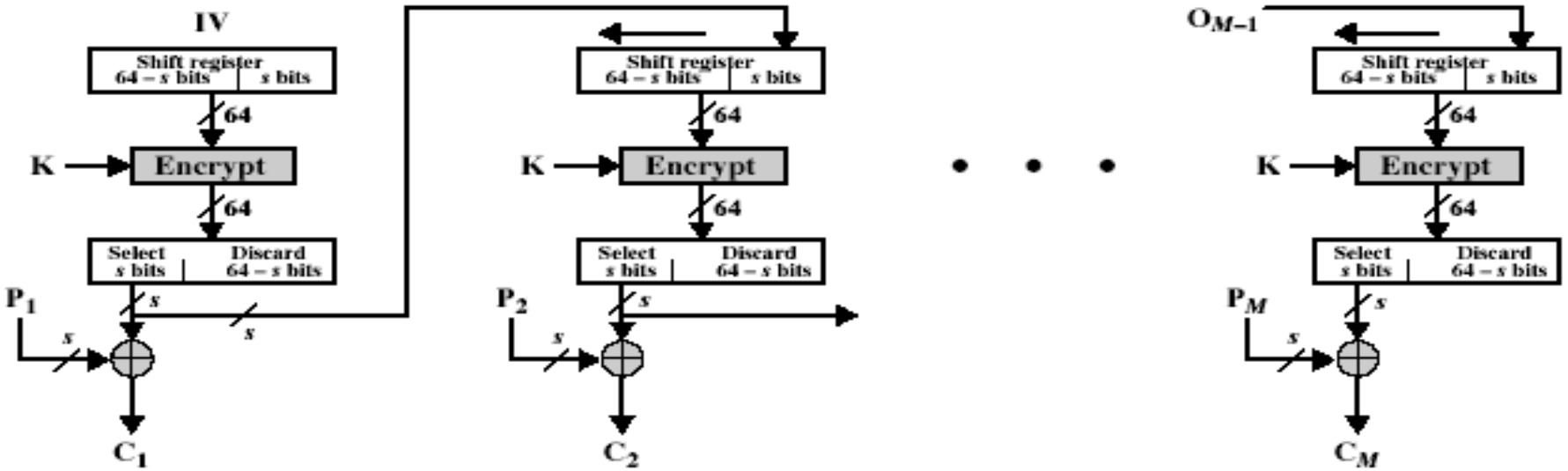

(b) Decryption

# Cipher Feedback Mode

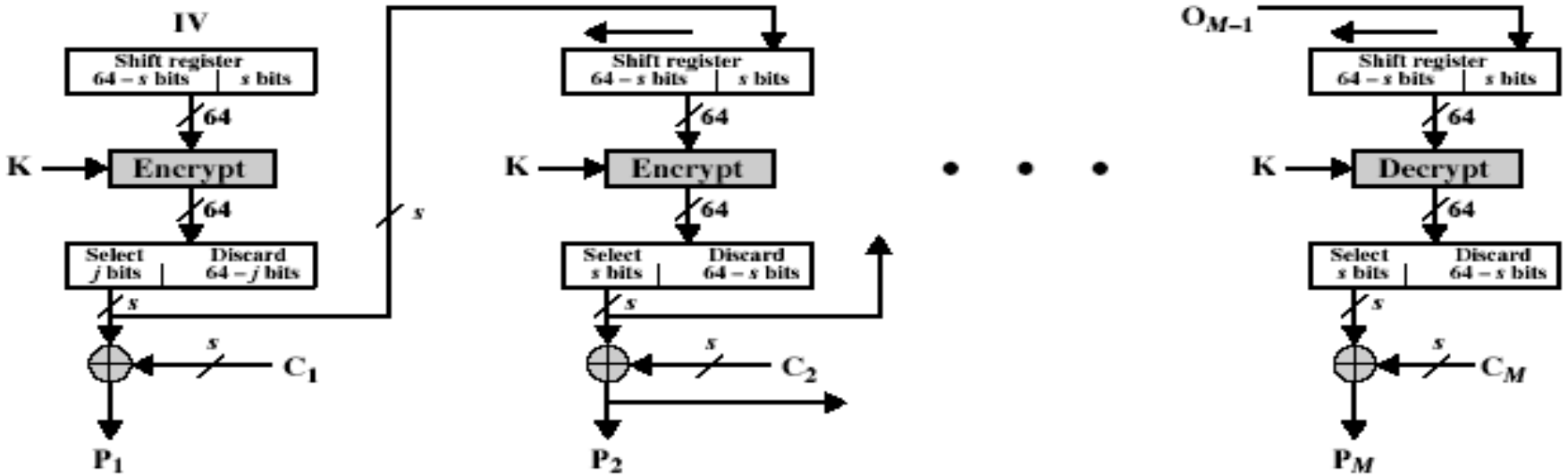
- To convert DES into stream cipher , Cipher feedback or output feedback mode is used.



# Output Feedback Mode



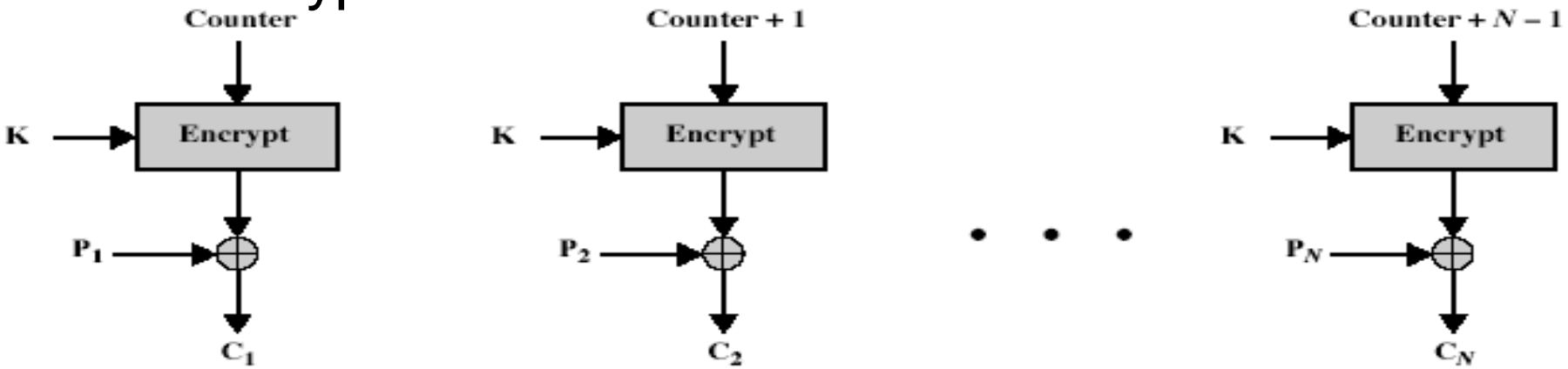

(a) Encryption



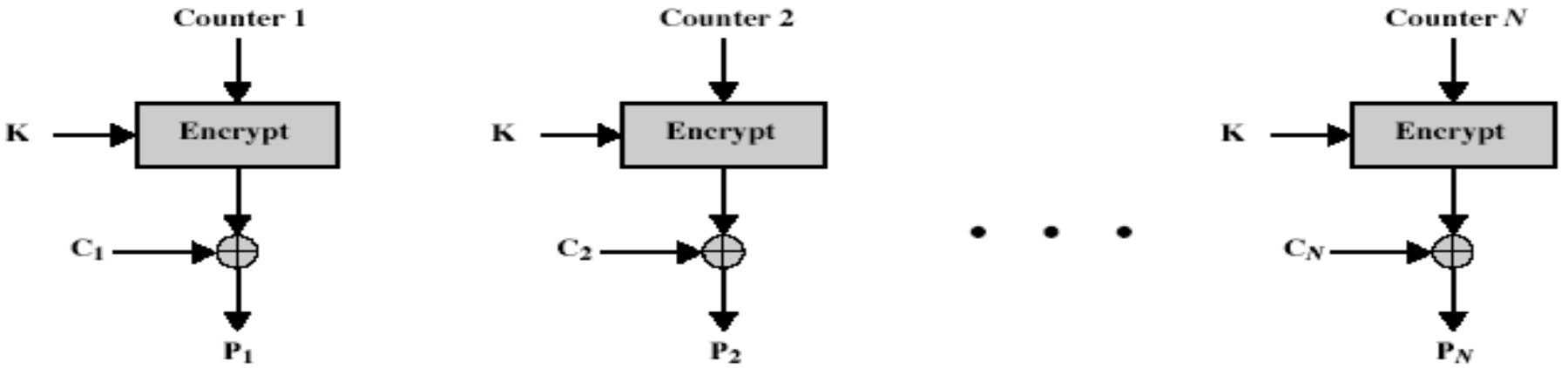

(b) Decryption

# Counter Mode

- A counter equal to plaintext block size is used and counter value must be different for each plaintext block that is encrypted.




(a) Encryption




(b) Decryption

# Triple DES

- Triple DES with Two keys:

$$C = E_{K1}(D_{K2}(E_{K1}(P)))$$

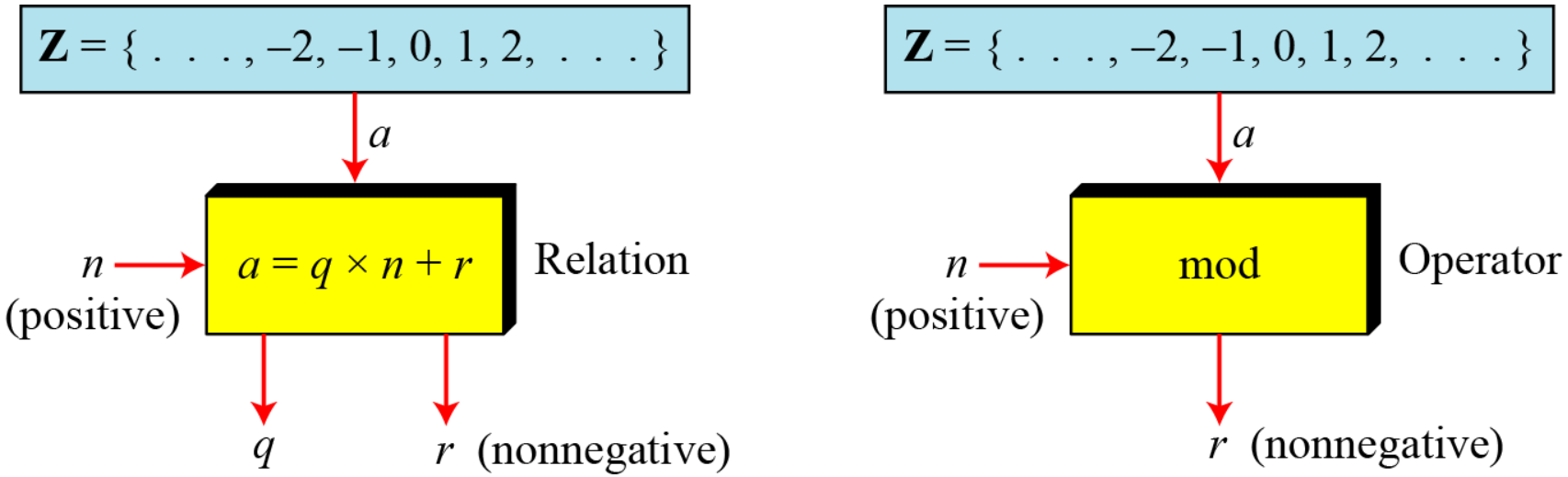
- Triple DES with Three keys:

$$C = E_{K3}(D_{K2}(E_{K1}(P)))$$

# Modulo operator

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

Fig: Division relation and modulo operator



# Modulo operator

Find the result of the following operations:

a. 27 mod 5
b. 36 mod 12
c. −18 mod 14
d. −7 mod 10

Set of Residues: The modulo operation creates a set, which in modular arithmetic is referred to as the set of least residues modulo n, or $Z_n$.

Figure: Some $Z_n$ sets

$$\mathbf{Z}_n = \{ 0, 1, 2, 3, \ldots, (n-1) \}$$

$$\mathbf{Z}_2 = \{ 0, 1 \}$$

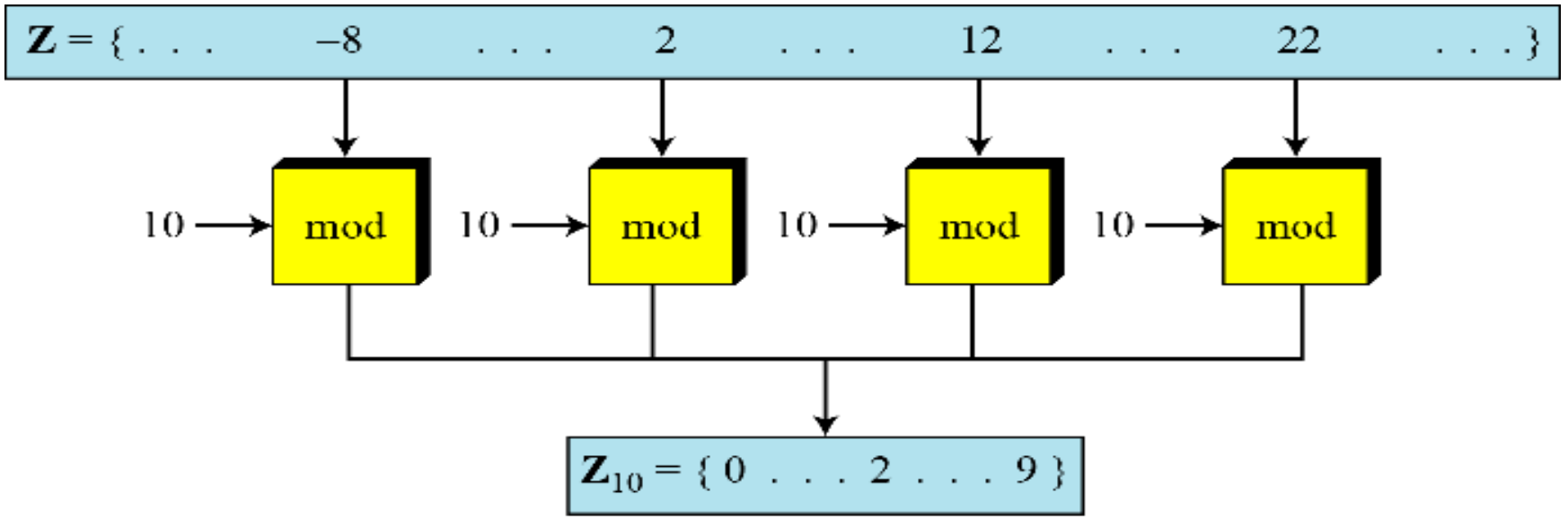
$$\mathbf{Z}_6 = \{ 0, 1, 2, 3, 4, 5 \}$$

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

# Congruence

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

Figure: Concept of congruence




$$-8 \equiv 2 \equiv 12 \equiv 22 \pmod{10}$$

Congruence Relationship

# Additive Inverse

In $Z_n$, two numbers a and b are additive inverses of each other if

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

Find all additive inverse pairs in Z10.

The ten pairs of additive inverses are (0, 0), (1, 9), (2, 8), (3, 7), (4, 6), (5, 5), (6, 4), (7, 3) (8, 2), (9, 1).

# Multiplicative Inverse

In $Z_n$, two numbers a and b are the multiplicative inverse of each other if

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

Find all multiplicative inverse pairs in $Z_7$.

We have six pairs: (1, 1), (2, 4), (3, 5), (4, 2), (5, 3), and (6, 6).

Two numbers a & b are relative prime or co-prime to each other if

**Gcd (a, b) = 1**

# Euclidean Algorithm

Euclidean algorithm is used to find the GCD of two numbers.
It is based on two facts:

**Fact 1**: gcd(a,0)=a

**Fact 2**: gcd(a,b)=gcd(b,r) where r is the remainder after dividing a by b.

# Euclidean Algorithm

Find the greatest common divisor of 2740 and 1760.

We have gcd (2740, 1760) = 20.

| $q$ | $r_1$ | $r_2$ | $r$ |
|---|---|---|---|
| 1 | 2740 | 1760 | 980 |
| 1 | 1760 | 980 | 780 |
| 1 | 980 | 780 | 200 |
| 3 | 780 | 200 | 180 |
| 1 | 200 | 180 | 20 |
| 9 | 180 | 20 | 0 |
| | **20** | 0 | |

# Extended Euclidean Algorithm

Given two integers a and b, we often need to f nd other two integers, s and t, such that

$$s \times a + t \times b = \gcd(a, b)$$

The extended Euclidean algorithm can calculate the gcd (a, b) and at the same time calculate the value of s and t.

# Extended Euclidean Algorithm

$r_1 \leftarrow a;$ $r_2 \leftarrow b;$
$s_1 \leftarrow 1;$ $s_2 \leftarrow 0;$
$t_1 \leftarrow 0;$ $t_2 \leftarrow 1;$ (Initialization)

while $(r_2 > 0)$

{

$q \leftarrow r_1 / r_2;$

$r \leftarrow r_1 - q \times r_2;$
$r_1 \leftarrow r_2;$ $r_2 \leftarrow r;$ (Updating $r$'s)

$s \leftarrow s_1 - q \times s_2;$
$s_1 \leftarrow s_2;$ $s_2 \leftarrow s;$ (Updating $s$'s)

$t \leftarrow t_1 - q \times t_2;$
$t_1 \leftarrow t_2;$ $t_2 \leftarrow t;$ (Updating $t$'s)

}

gcd $(a, b) \leftarrow r_1;$ $s \leftarrow s_1;$ $t \leftarrow t_1$

b. Algorithm

# Extended Euclidean Algorithm

Given a = 161 and b = 28, find gcd (a, b) and the values of s and t.

We get gcd (161, 28) = 7, s = −1 and t = 6.

| $q$ | $r_1$ | $r_2$ | $r$ | $s_1$ | $s_2$ | $s$ | $t_1$ | $t_2$ | $t$ |
|---|---|---|---|---|---|---|---|---|---|
| 5 | 161 | 28 | 21 | 1 | 0 | 1 | 0 | 1 | −5 |
| 1 | 28 | 21 | 7 | 0 | 1 | −1 | 1 | −5 | 6 |
| 3 | 21 | 7 | 0 | 1 | −1 | 4 | −5 | 6 | −23 |
| | **7** | 0 | | **−1** | 4 | | **6** | −23 | |

# Extended Euclidean Algorithm

The extended Euclidean algorithm finds the multiplicative inverses of b in $Z_n$ when n and b are given and gcd (n, b) = 1.

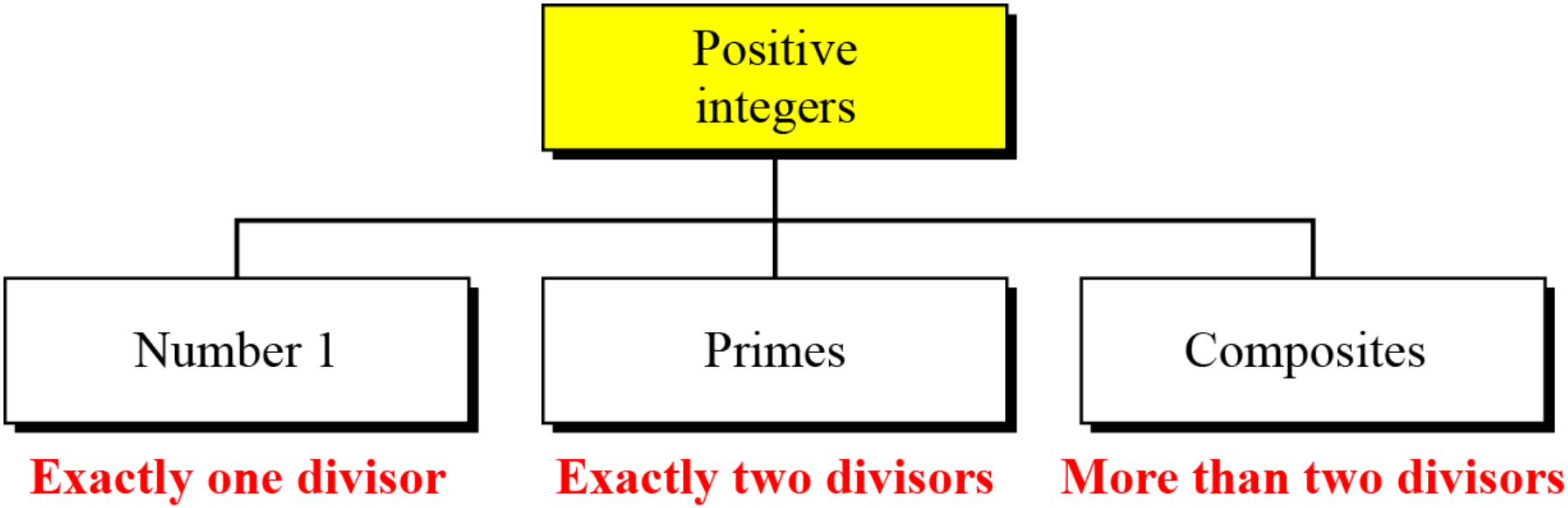
Find the multiplicative inverse of 11 in $Z_{26}$.

| $q$ | $r_1$ | $r_2$ | $r$ | $t_1$ | $t_2$ | $t$ |
|---|---|---|---|---|---|---|
| 2 | 26 | 11 | 4 | 0 | 1 | –2 |
| 2 | 11 | 4 | 3 | 1 | –2 | 5 |
| 1 | 4 | 3 | 1 | –2 | 5 | –7 |
| 3 | 3 | 1 | 0 | 5 | –7 | 26 |
| | 1 | 0 | | –7 | 26 | |

The gcd (26, 11) is 1; the inverse of 11 is –7 or 19.

# Prime number

Figure : *Three groups of positive integers*




A prime is divisible only by itself and 1.

# Euler's Phi Function

*Euler's phi-function, $\phi(n)$, which is sometimes called the*
*Euler's totient function finds the number of integers that are both smaller than n and relatively prime to n.*

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.

**What is the value of $\phi(13)$?**

13 is a prime, $\phi(13) = (13 - 1) = 12$.

# Euler's Phi Function

**What is the value of $\phi(10)$?**

We can use the third rule: $\phi(10) = \phi(2) \times \phi(5) = 1 \times 4 = 4$, because 2 and 5 are primes.

**What is the value of $\phi(240)$?**

We can write $240 = 2^4 \times 3^1 \times 5^1$. Then

$\phi(240) = (2^4 - 2^3) \times (3^1 - 3^0) \times (5^1 - 5^0) = 64$

**What is the value of $\phi(49)$ ?**

The third rule applies when *m* and *n* are relatively prime. Here $49 = 7^2$. We need to use the fourth rule: $\phi(49) = 7^2 - 7^1 = 42$.

What is the number of elements in $Z_{14}^*$?

The answer is $\phi(14) = \phi(7) \times \phi(2) = 6 \times 1 = 6$. The members are 1, 3, 5, 9, 11, and 13.

## Fermat's Little Theorem

*First Version: If p is any prime and a is an integer such that p does not divide a, then*

$$a^{p-1} \equiv 1 \bmod p$$

*Second Version: If p is any prime and a is an integer, then*

$$a^p \equiv a \bmod p$$

# Fermat's Little Theorem

Find the result of $6^{10} \bmod 11$.

We have $6^{10} \bmod 11 = 1$. This is the first version of Fermat's little theorem where $p = 11$.

Find the result of $3^{12} \bmod 11$.

Here second version of Fermat's little theorem can be used where $p = 11$.

$$3^{12} \bmod 11 = (3^{11} \times 3) \bmod 11 = (3^{11} \bmod 11)\,(3 \bmod 11) = (3 \times 3) \bmod 11 = 9$$

# Multiplicative Inverse

If p is a prime and a is an integer such that p does not divide a, then

$$a^{-1} \bmod p = a^{p-2} \bmod p$$

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. $22^{-1} \bmod 211 = 22^{211-2} \bmod 211 = 22^{209} \bmod 211 = 48 \bmod 211$

# Euler's Theorem

If a and n are co-prime, then

$$a^{\phi(n)} \equiv 1 \pmod{n}$$

Find the result of $6^{24}$ mod 35.

Solution

We have $6^{24} \bmod 35 = 6^{\phi(35)} \bmod 35 = 1$.

# Primality Testing

To test whether given number is prime or not, algorithms can be categorized to Deterministic or Probabilistic category.

**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 × 2^k
    T ← a^m mod n
    if (T = ± 1)  return "a prime"
    for (i ← 1 to k − 1)                    // k − 1 is the maximum number of steps.
    {
        T ← T^2 mod n
        if (T = +1) return "a composite"
        if (T = −1) return "a prime"
    }
    return "a composite"
}
```

# Rabin-Miller Primality Testing

Does the number 561 pass the Miller-Rabin test ?

Using base 2, let $561 - 1 = 35 \times 2^4$, which means $m = 35$, $k = 4$, and $a = 2$.

| | | |
|---|---|---|
| **Initialization:** | $T = 2^{35} \bmod 561 = 263 \bmod 561$ | |
| $k = 1$: | $T = 263^2 \bmod 561 = 166 \bmod 561$ | |
| $k = 2$: | $T = 166^2 \bmod 561 = 67 \bmod 561$ | |
| $k = 3$: | $T = 67^2 \bmod 561 = +1 \bmod 561$ | $\rightarrow$ **a composite** |

Does the number 61 pass the Miller-Rabin test ?

We use base 2.

| | | |
|---|---|---|
| $61 - 1 = 15 \times 2^2$ | $\rightarrow$ $m = 15$ $k = 2$ $a = 2$ | |
| *Initialization:* | $T = 2^{15} \bmod 61 = 11 \bmod 61$ | |
| $k = 1$ | $T = 11^2 \bmod 61 = -1 \bmod 61$ | $\rightarrow$ **a prime** |

# 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:

$$\begin{aligned} x &\equiv a_1 \pmod{m_1} \\ x &\equiv a_2 \pmod{m_2} \\ &\cdots \\ x &\equiv a_k \pmod{m_k} \end{aligned}$$

1. Find $M = m_1 \times m_2 \times \ldots \times m_k$. This is the common modulus.
2. Find $M_1 = M/m_1$, $M_2 = M/m_2$, …, $M_k = M/m_k$.
3. Find the multiplicative inverse of $M_1, M_2, \ldots, M_k$ using the corresponding moduli $(m_1, m_2, \ldots, m_k)$. Call the inverses $M_1^{-1}, M_2^{-1}, \ldots, M_k^{-1}$.
4. The solution to the simultaneous equations is

$$x = (a_1 \times M_1 \times M_1^{-1} + a_2 \times M_2 \times M_2^{-1} + \cdots + a_k \times M_k \times M_k^{-1}) \bmod M$$

# Chinese Remainder Theorem

Find the solution to the simultaneous equati

$$x \equiv 2 \pmod 3$$
$$x \equiv 3 \pmod 5$$
$$x \equiv 2 \pmod 7$$

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. $x = (2 \times 35 \times 2 + 3 \times 21 \times 1 + 2 \times 15 \times 1) \bmod 105 = 23 \bmod 105$

# Chinese Remainder Theorem

Find an integer that has a remainder of 3 when divided by 7 and 13, but is divisible by 12.

S o l u t i o n

This is a CRT problem. We can form three equations and solve them to find the value of x.

$$x \equiv 3 \bmod 7$$
$$x \equiv 3 \bmod 13$$
$$x \equiv 0 \bmod 12$$

If we follow the four steps, we f ind x = 276. We can check that 276 = 3 mod 7, 276 = 3 mod 13 and 276 is divisible by 12 (the quotient is 23 and the remainder is zero).

# Primitive Root

In the group G = $<Z_n^*, \times>$, when the order of an element is the same as $\phi(n)$, that element is called the primitive root of the group.

## *Order of the Group.*

What is the order of group G = $<Z_{21}^*, \times>$? $|G| = \phi(21) = \phi(3) \times \phi(7) = 2 \times 6 = 12$. There are 12 elements in this group: 1, 2, 4, 5, 8, 10, 11, 13, 16, 17, 19, and 20.

## *Order of an Element*

The order of an element, a, is the smallest integer i such that

$$a^i \equiv 1 \pmod{n}$$

# Primitive Root

Find the order of the group and element of G = $<Z_8^*, \times>$.

Here $\phi(8) = 4$. Order of the group is 4.
The elements of the group are 1,3,5,7.

| | $i = 1$ | $i = 2$ | $i = 3$ | $i = 4$ | $i = 5$ | $i = 6$ | $i = 7$ |
|---|---|---|---|---|---|---|---|
| $a = 1$ | **x: 1** | x: 1 | x: 1 | x: 1 | x: 1 | x: 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 |

ord(1)=1, ord(3)=2, ord(5)=2, ord(7)=2

# Primitive Root

Table shows the result of $a^i \equiv x \pmod 7$ for the group $G = \langle Z_7^*, \times \rangle$. In this group, $\phi(7) = 6$.

| | $i = 1$ | $i = 2$ | $i = 3$ | $i = 4$ | $i = 5$ | $i = 6$ |
|---|---|---|---|---|---|---|
| $a = 1$ | $x$: 1 | $x$: 1 | $x$: 1 | $x$: 1 | $x$: 1 | $x$: 1 |
| $a = 2$ | $x$: 2 | $x$: 4 | $x$: 1 | $x$: 2 | $x$: 4 | $x$: 1 |
| Primitive root → $a = 3$ | $x$: 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 |
| Primitive root → $a = 5$ | $x$: 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

The group G = $<Z_n^*, \times>$ has primitive roots only if $n$ is 2, 4, $p^t$, or $2p^t$ where $p$ is a odd prime & t is an integer.

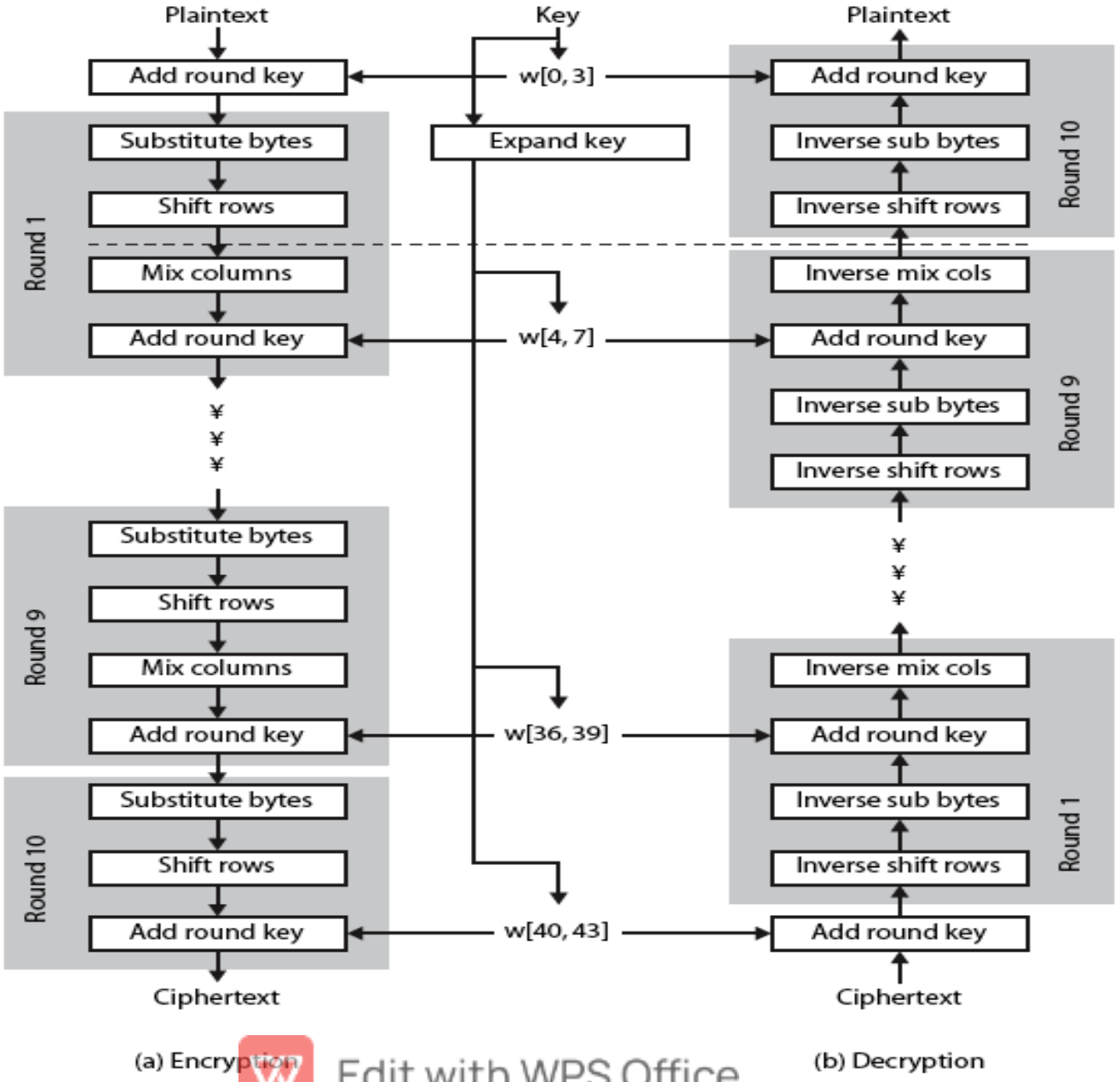
If the group G = $<Z_n^*, \times>$ has any primitive root, the number of primitive roots is $\phi(\phi(n))$.

# AES Versions

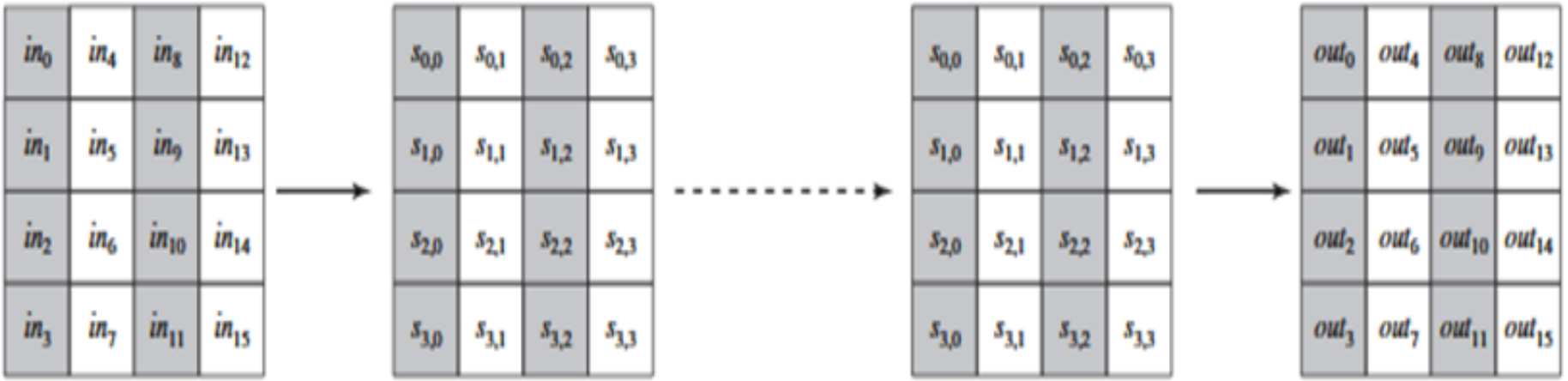
AES Parameters

| Key Size (words/bytes/bits) | 4/16/128 | 6/24/192 | 8/32/256 |
|---|---|---|---|
| Plaintext Block Size (words/bytes/bits) | 4/16/128 | 4/16/128 | 4/16/128 |
| Number of Rounds | 10 | 12 | 14 |
| Round Key Size (words/bytes/bits) | 4/16/128 | 4/16/128 | 4/16/128 |
| Expanded Key Size (words/bytes) | 44/176 | 52/208 | 60/240 |

# AES Basic Structure



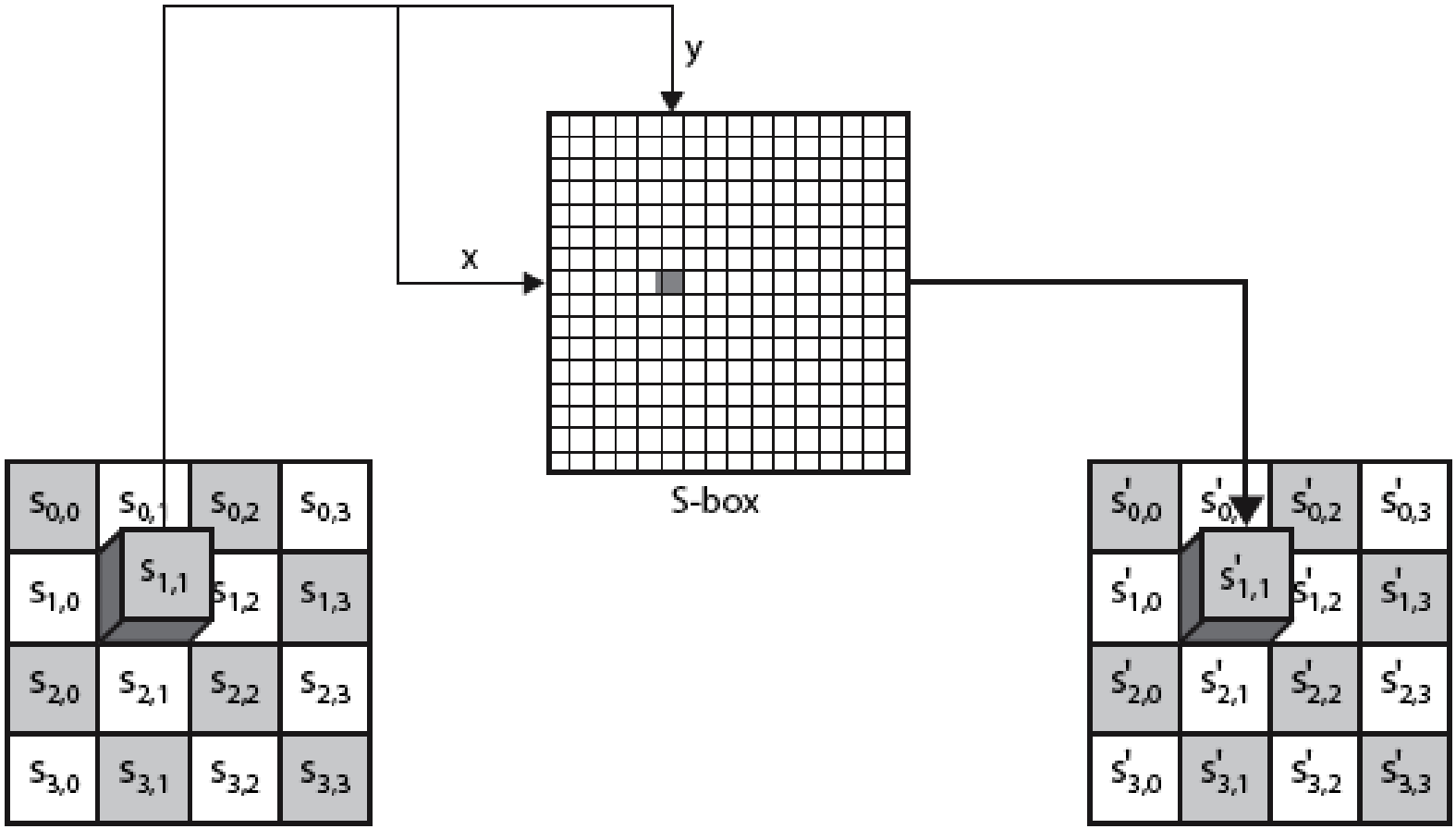
# AES Data Structure




(a) Input, state array, and output


(b) Key and expanded key

# Substitute Bytes



# Substitute Bytes

AES S-Boxes

| x \ y | 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 | 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 | C8 | 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 |

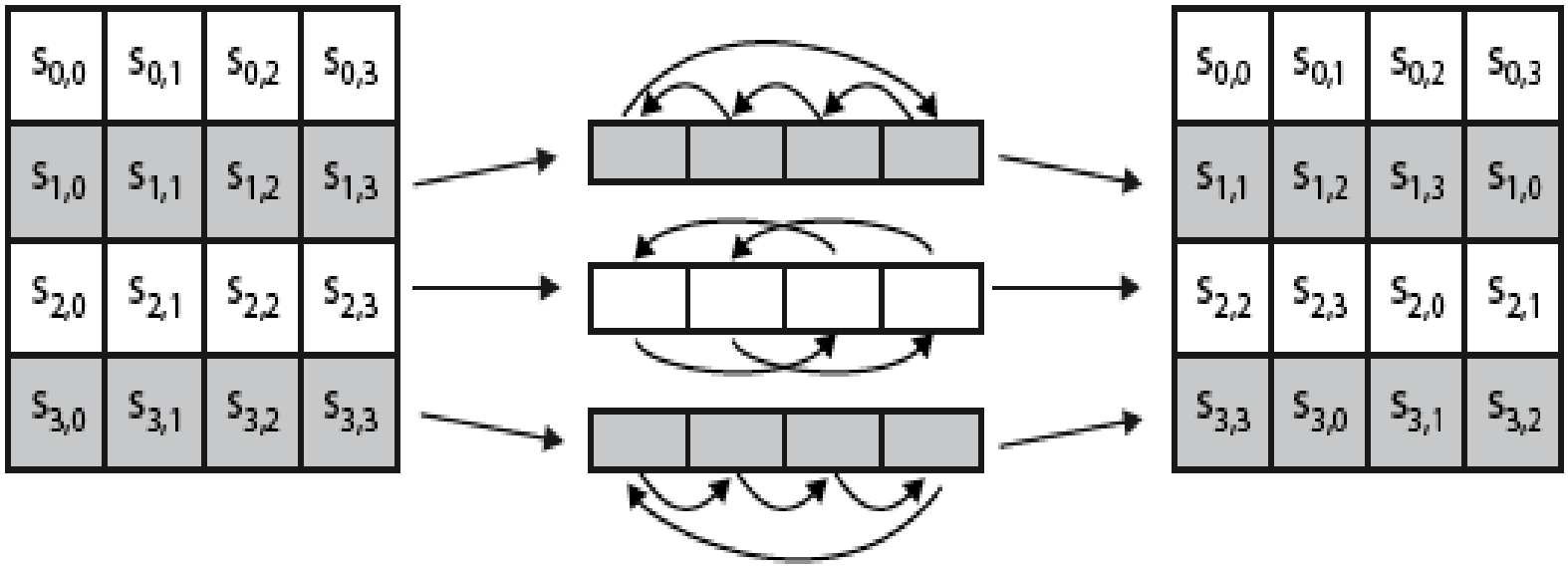
(a) S-box

# Substitute Bytes

| x \ y | 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 |
| 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 |

**(b) Inverse S-box**

# Shift Rows



# Shift Rows

## ShiftRows Transformation

*Forward and Inverse Transformations* The **forward shift row transformation**, called ShiftRows, is depicted in Figure 5.7a. The first row of **State** is not altered. For the second row, a 1-byte circular left shift is performed. For the third row, a 2-byte circular left shift is performed. For the fourth row, a 3-byte circular left shift is performed. The following is an example of ShiftRows.

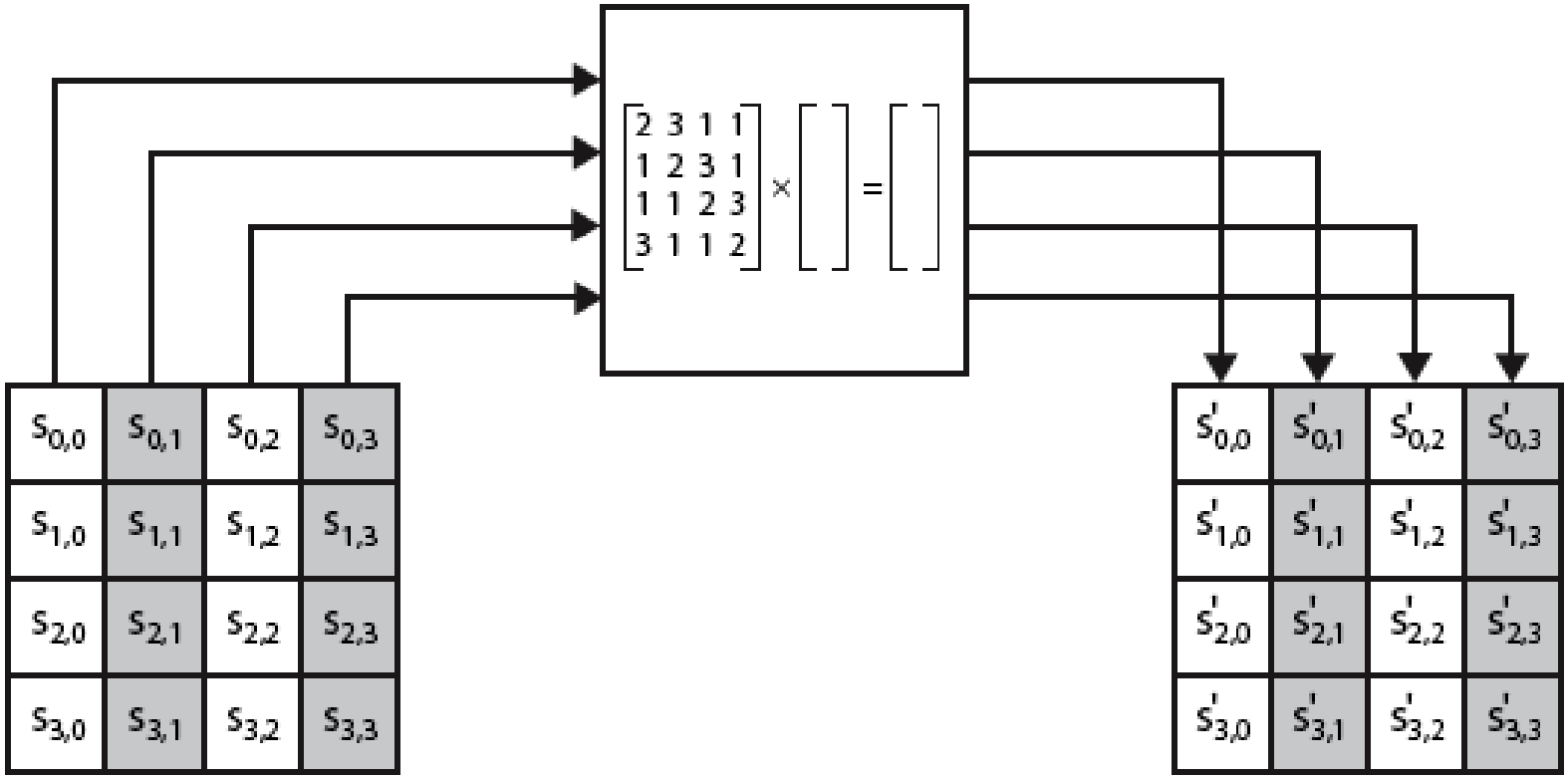
| | | | |
|---|---|---|---|
| 87 | F2 | 4D | 97 |
| EC | 6E | 4C | 90 |
| 4A | C3 | 46 | E7 |
| 8C | D8 | 95 | A6 |

→

| | | | |
|---|---|---|---|
| 87 | F2 | 4D | 97 |
| 6E | 4C | 90 | EC |
| 46 | E7 | 4A | C3 |
| A6 | 8C | D8 | 95 |

The **inverse shift row transformation**, called InvShiftRows, performs the circular shifts in the opposite direction for each of the last three rows, with a 1-byte circular right shift for the second row, and so on.

# Mix Columns



# Mix Columns

## MixColumns Transformation

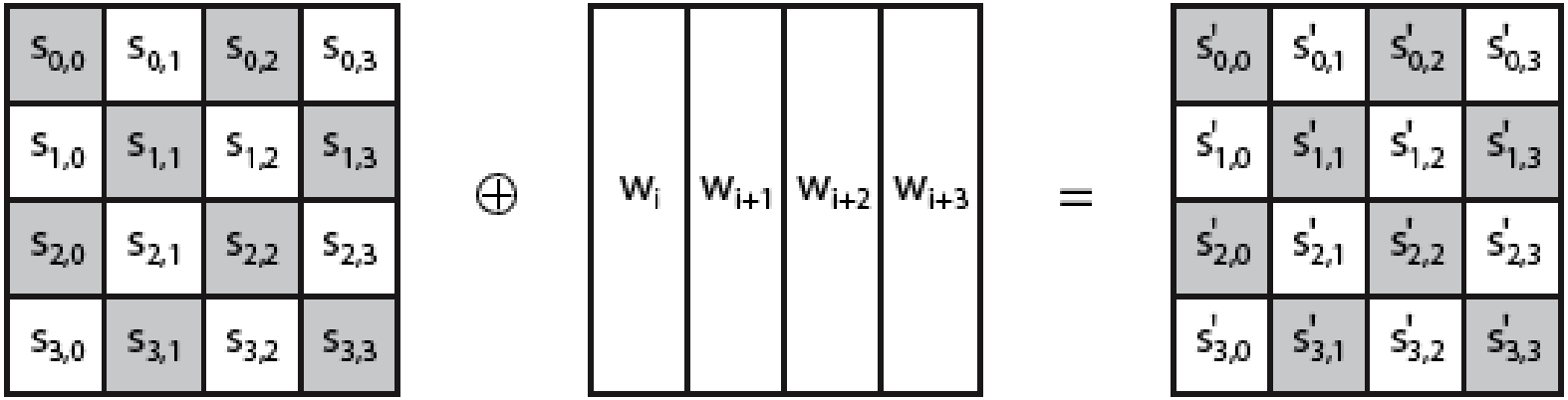
*Forward and Inverse Transformations* The **forward mix column transformation**, called MixColumns, operates on each column individually. Each byte of a column is mapped into a new value that is a function of all four bytes in that column. The transformation can be defined by the following matrix multiplication on **State** (Figure 5.7b):

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} = \begin{bmatrix} s'_{0,0} & s'_{0,1} & s'_{0,2} & s'_{0,3} \\ s'_{1,0} & s'_{1,1} & s'_{1,2} & s'_{1,3} \\ s'_{2,0} & s'_{2,1} & s'_{2,2} & s'_{2,3} \\ s'_{3,0} & s'_{3,1} & s'_{3,2} & s'_{3,3} \end{bmatrix}$$

The **inverse mix column transformation**, called InvMixColumns, is defined by the following matrix multiplication:

$$\begin{bmatrix} 0E & 0B & 0D & 09 \\ 09 & 0E & 0B & 0D \\ 0D & 09 & 0E & 0B \\ 0B & 0D & 09 & 0E \end{bmatrix} \begin{bmatrix} s_{0,0} & s_{0,1} & s_{0,2} & s_{0,3} \\ s_{1,0} & s_{1,1} & s_{1,2} & s_{1,3} \\ s_{2,0} & s_{2,1} & s_{2,2} & s_{2,3} \\ s_{3,0} & s_{3,1} & s_{3,2} & s_{3,3} \end{bmatrix} = \begin{bmatrix} s'_{0,0} & s'_{0,1} & s'_{0,2} & s'_{0,3} \\ s'_{1,0} & s'_{1,1} & s'_{1,2} & s'_{1,3} \\ s'_{2,0} & s'_{2,1} & s'_{2,2} & s'_{2,3} \\ s'_{3,0} & s'_{3,1} & s'_{3,2} & s'_{3,3} \end{bmatrix}$$

# Add Round Key Transformation




## AddRoundKey Transformation

*Forward and Inverse Transformations* **In the forward add round key transformation**, called AddRoundKey, the 128 bits of **State** are bitwise XORed with the 128 bits of the round key. .

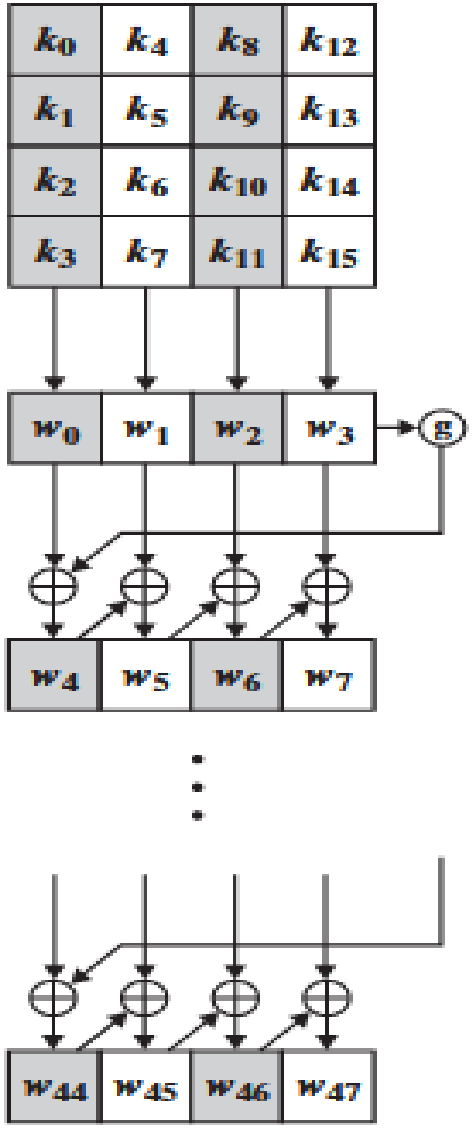
The following is an example of AddRoundKey:

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 47 | 40 | A3 | 4C | ⊕ | AC | 19 | 28 | 57 | = | EB | 59 | 8B | 1B |
| 37 | D4 | 70 | 9F | | 77 | FA | D1 | 5C | | 40 | 2E | A1 | C3 |
| 94 | E4 | 3A | 42 | | 66 | DC | 29 | 00 | | F2 | 38 | 13 | 42 |
| ED | A5 | A6 | BC | | F3 | 21 | 41 | 6A | | 1E | 84 | E7 | D6 |

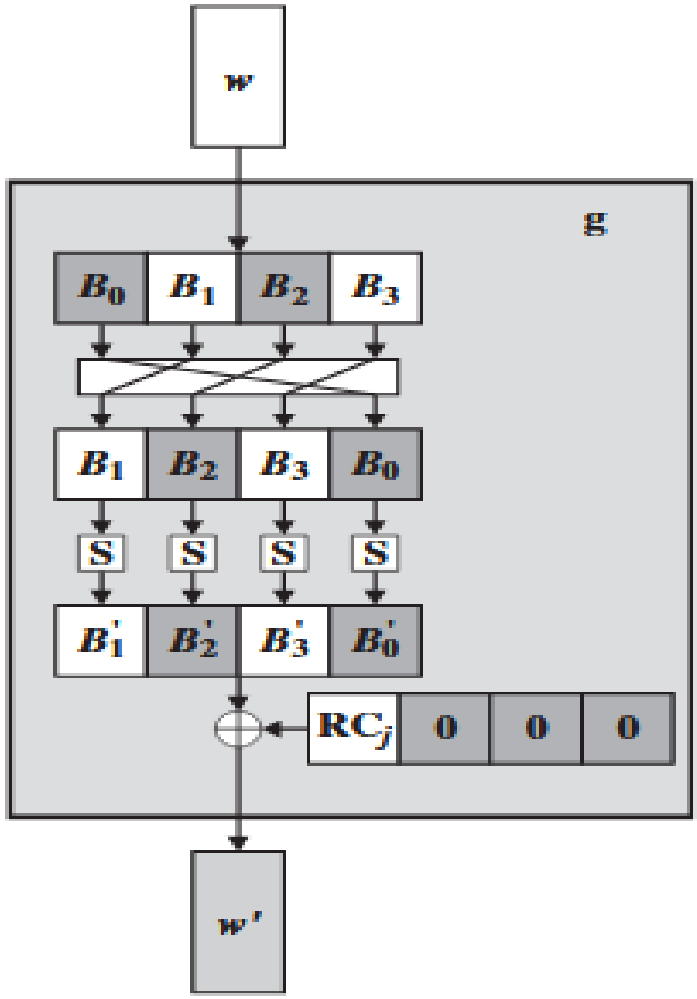
The first matrix is **State**, and the second matrix is the round key.

The **inverse add round key transformation** is identical to the forward add round key transformation, because the XOR operation is its own inverse.

# AES Key Expansion




(a) Overall algorithm




(b) Function g

# AES Key Expansion

| j | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| RC[j] | 01 | 02 | 04 | 08 | 10 | 20 | 40 | 80 | 1B | 36 |

For example, suppose that the round key for round 8 is

EA D2 73 21 B5 8D BA D2 31 2B F5 60 7F 8D 29 2F

Then the first 4 bytes (first column) of the round key for round 9 are calculated as follows:

| i (decimal) | temp | After RotWord | After SubWord | Rcon (9) | After XOR with Rcon | w[i−4] | w[i] = temp ⊕ w[i−4] |
|---|---|---|---|---|---|---|---|
| 36 | 7F8D292F | 8D292F7F | 5DA515D2 | 1B000000 | 46A515D2 | EAD27321 | AC7766F3 |

# RSA algorithm

- Named after inventors Ron Rivest, Adi Shamir and Len Adleman.
- RSA is a block cipher between 0 and n-1 for some n.
- Typical size of n is 1024 bits or 309 digits.

# RSA Algorithm

**Key Generation Alice**

| | |
|---|---|
| Select $p, q$ | $p$ and $q$ both prime, $p \neq q$ |
| *Calculate* $n = p \times q$ | |
| Calcuate $\phi(n) = (p-1)(q-1)$ | |
| Select integer $e$ | gcd $(\phi(n), e) = 1; 1 < e < \phi(n)$ |
| Calculate $d$ | $d = e^{-1} \pmod{\phi(n)}$ |
| Public key | $PU = \{e, n\}$ |
| Private key | $PR = \{d, n\}$ |

**Encryption by Bob with Alice's Public Key**

| | |
|---|---|
| Plaintext: | $M < n$ |
| Ciphertext: | $C = M^e \bmod n$ |

**Decryption by Alice with Alice's Public Key**

| | |
|---|---|
| Ciphertext: | $C$ |
| Plaintext: | $M = C^d \bmod n$ |

Figure 9.5 The RSA Algorithm

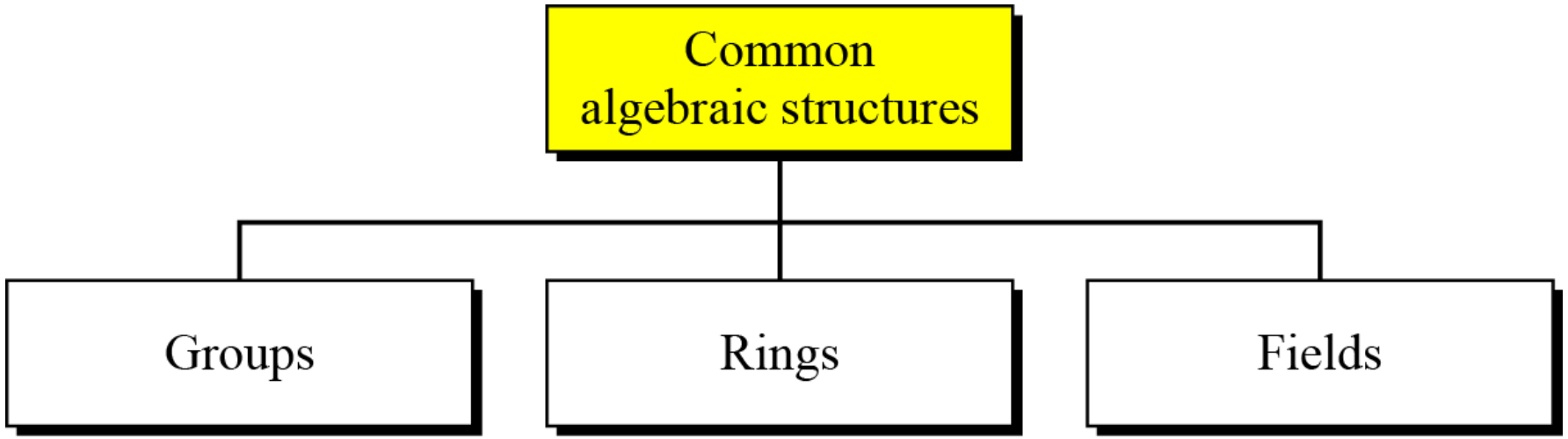
# RSA Example

- Select two prime numbers: p=5, q=7.
- n=5*7=35
- $\phi(n)=24$
- e=5;d=5; (e*d) mod $\phi(n)=1$.
- Take plaintext M=2, then after encryption 2^5 mod35= 32.
- Ciphertext 32^5 mod 35=2. (Hint: 33554432)

# 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.

***Common algebraic structure***



# Group

- A group (G) is a set of elements with a binary operation (•) that satisfies four properties. A commutative group satisfies an extra property, commutativity.

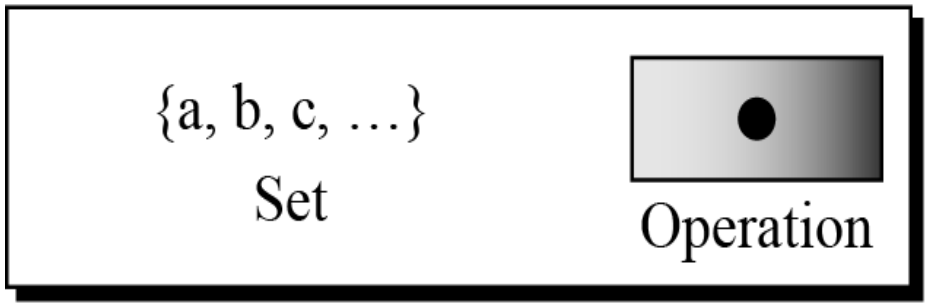
- Closure: If a and b are elements of G, then c=a•b is also an element of G.
- Associativity: If a.b & c are elements of G, then (a•b) •c=a•(b•c)
- Commutativity: For all a & b in G, a•b=b•a
- Existence of Identity: For all a in G, there exist an identity element e such that e•a=a•e=a
- Existence of Inverse : For all a in G, there exist an element a' such that a•a'=a'•a=e

# Group

Properties

1. Closure
2. Associativity
3. Commutativity (See note)
4. Existence of identity
5. Existence of inverse

Note:
The third property needs to be satisfied only for a commutative group.



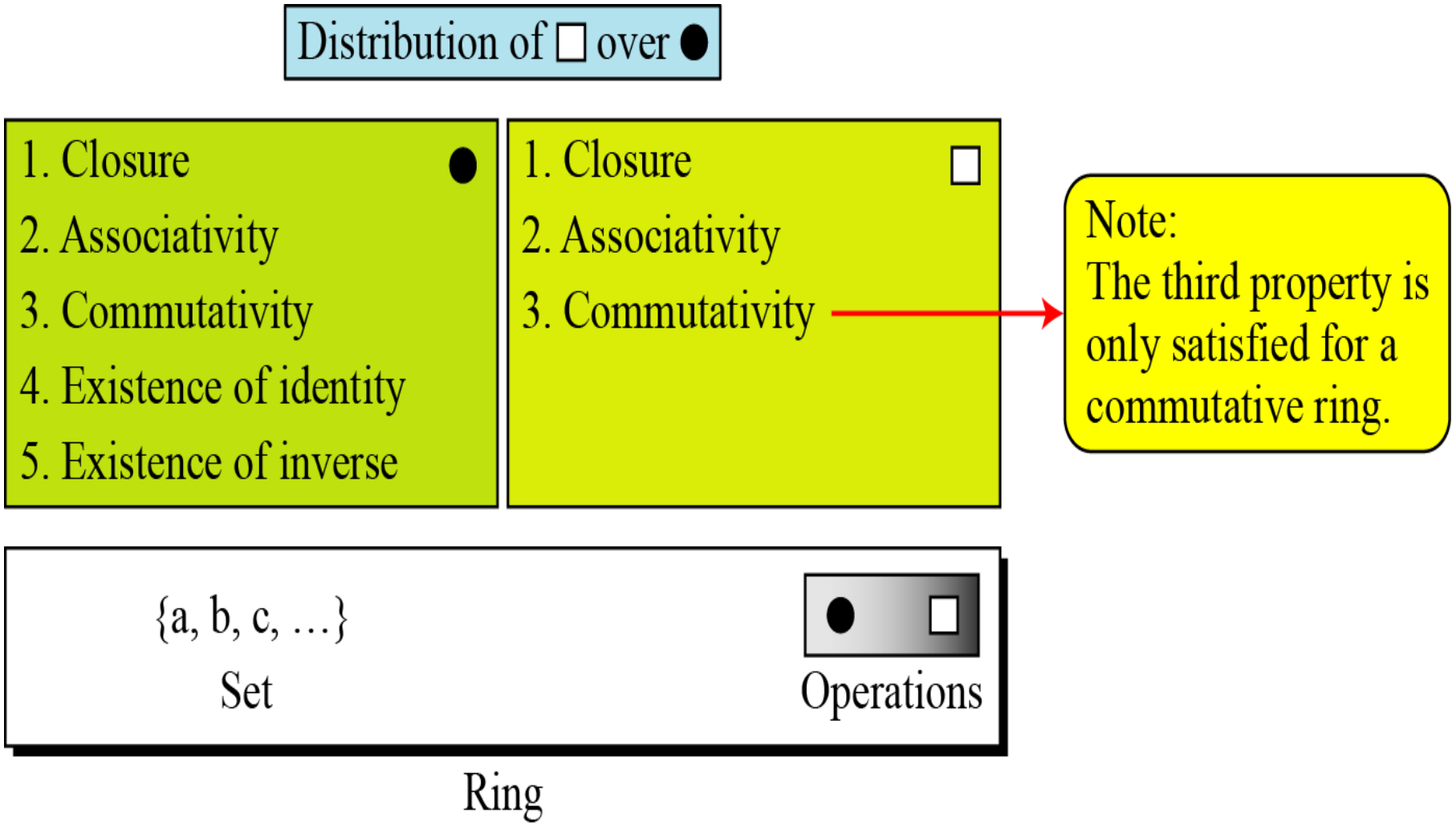

Group

# Examples of Group

- The set of residue integers with the addition operator, G = < $Z_n$ , +>, is a commutative group.
- The set $Z_{n*}$ with the multiplication operator, G = < $Z_{n*}$,x>, is also an abelian group.
- Let us define a set G = < {*a*, *b*, *c*, *d*}, •> and the operation as shown in Table below.

| • | *a* | *b* | *c* | *d* |
|---|---|---|---|---|
| *a* | *a* | *b* | *c* | *d* |
| *b* | *b* | *c* | *d* | *a* |
| *c* | *c* | *d* | *a* | *b* |
| *d* | *d* | *a* | *b* | *c* |

# Ring

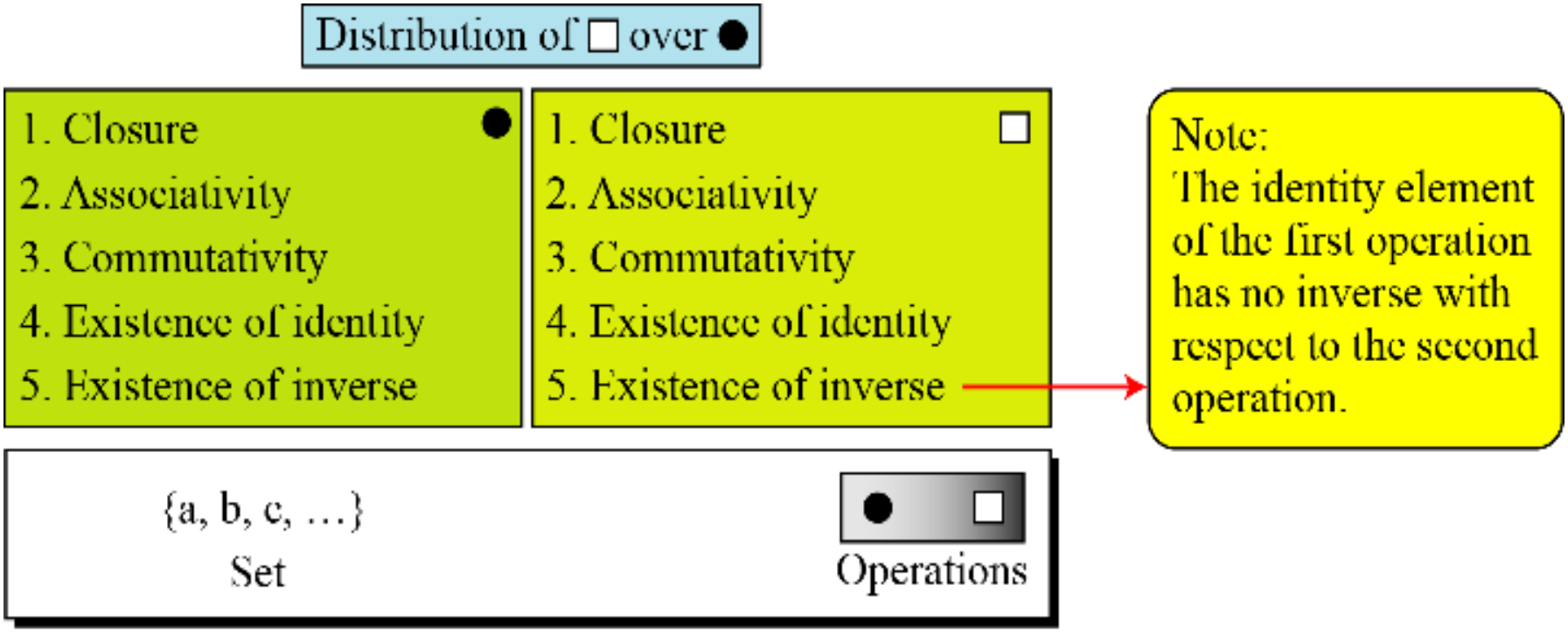


# Ring

- A ring, R = <{...}, •, □ >, is an algebraic structure with two operations.
- The first operation must satisfy all the five properties of abelian group and second operation must satisfy only two or three properties for ring & abelian ring respectively.
- In addition, second operation must be distributed over the first one.
- For all a,b, & c in R, we have
- a □ (b•c)=(a □ b) •(a □ c) &  (a•b) □ c=(a □ c) •(b □ c)
- The set Z with two operations, addition and multiplication, is a commutative ring. We show it by R = <Z, +, ×>. Addition satisfies all of the five properties; multiplication satisfies only three properties.

# Field

- A field, denoted by F = <{...}, •, □ > is a commutative ring in which the second operation satisfies all the five properties defined for the first operation except that the identity of the first operation has no inverse with respect to the second operation.

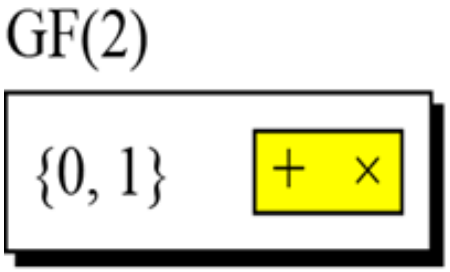


# Finite Fields

- Galois showed that for a field to be finite, the number of elements should be $p^n$, where $p$ is a prime and $n$ is a positive integer.

**A Galois field, GF($p^n$), is a finite field with $p^n$ elements.**

- A very common field in this category is GF(2) with the set {0, 1} and two operations, addition and multiplication, as shown in Figure below.




| + | 0 | 1 |
|---|---|---|
| 0 | 0 | 1 |
| 1 | 1 | 0 |

Addition

| × | 0 | 1 |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 0 | 1 |

Multiplication

| a | 0 | 1 |
|---|---|---|
| −a | 0 | 1 |

| a | 0 | 1 |
|---|---|---|
| $a^{-1}$ | — | 1 |

Inverses

# Finite Fields

- We can define GF(5) on the set $Z_5$ (5 is a prime) with addition and multiplication operators as shown in fig below.

GF(5)

{0, 1, 2, 3, 4} + ×

| + | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| 0 | 0 | 1 | 2 | 3 | 4 |
| 1 | 1 | 2 | 3 | 4 | 0 |
| 2 | 2 | 3 | 4 | 0 | 1 |
| 3 | 3 | 4 | 0 | 1 | 2 |
| 4 | 4 | 0 | 1 | 2 | 3 |

Addition

| × | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 0 | 1 | 2 | 3 | 4 |
| 2 | 0 | 2 | 4 | 1 | 3 |
| 3 | 0 | 3 | 1 | 4 | 2 |
| 4 | 0 | 4 | 3 | 2 | 1 |

Multiplication

Additive inverse

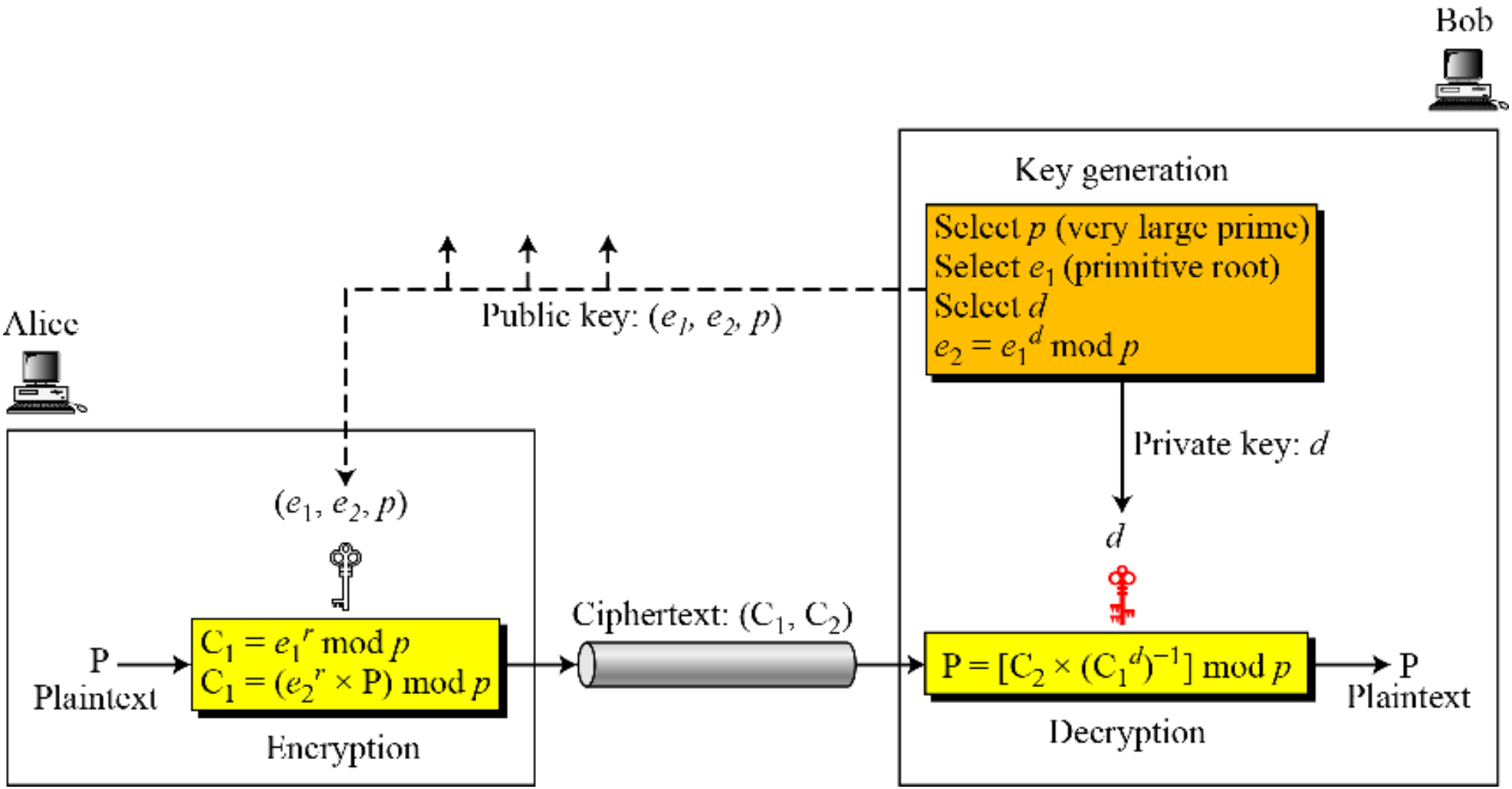
| a | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| −a | 0 | 4 | 3 | 2 | 1 |

| a | 0 | 1 | 2 | 3 | 4 |
|---|---|---|---|---|---|
| $a^{-1}$ | — | 1 | 3 | 2 | 4 |

Multiplicative inverse

# Elgamal Cryptosystem

- Figure 10.11 *Key generation, encryption, and decryption in ElGamal*



# Elgamal Cryptosystem

**Algorithm 10.9** *ElGamal key generation*

**ElGamal_Key_Generation**

{

Select a large prime $p$

Select $d$ to be a member of the group $\mathbf{G} = < \mathbf{Z}_p^*, \times >$ such that $1 \leq d \leq p - 2$

Select $e_1$ to be a primitive root in the group $\mathbf{G} = < \mathbf{Z}_p^*, \times >$

$e_2 \leftarrow e_1^{\,d} \bmod p$

Public_key $\leftarrow$ $(e_1, e_2, p)$ // To be announced publicly

Private_key $\leftarrow$ $d$ // To be kept secret

return Public_key and Private_key

}

# Elgamal Cryptosystem

**Algorithm 10.10** *ElGamal encryption*

```
ElGamal_Encryption (e1, e2, p, P)          // P is the plaintext
{
    Select a random integer r in the group G = < Zp*, × >
    C1 ← e1^r mod p
    C2 ← (P × e2^r) mod p                  // C1 and C2 are the ciphertexts
    return C1 and C2
}
```

**Algorithm 10.11** *ElGamal decryption*

```
ElGamal_Decryption (d, p, C1, C2)          // C1 and C2 are the ciphertexts
{
    P ← [C2 (C1^d)^−1] mod p               // P is the plaintext
    return P
}
```

# Example of Elgamal Cryptosystem

- *Here is a trivial example. Bob chooses p = 11 and $e_1$ = 2. and d = 3. Alice chooses r = 4 and calculates C1 and C2 for the plaintext 7.*

*Inverse calculation :* $P=[C_2 x (C_1^d)^{-1}] \bmod p$ $P=[C_2 x C_1^{p-1-d}] \bmod p$

# Example of Elgamal Cryptosystem

- *Here is a trivial example. Bob chooses p = 11 and $e_1$ = 2. and d = 3 $e_2 = e_1^d = 8$. So the public keys are (2, 8, 11) and the private key is 3. Alice chooses r = 4 and calculates C1 and C2 for the plaintext 7.*

**Plaintext: 7**

$C_1 = e_1^r \bmod 11 = 16 \bmod 11 = 5 \bmod 11$

$C_2 = (P \times e_2^r) \bmod 11 = (7 \times 4096) \bmod 11 = 6 \bmod 11$

**Ciphertext:** (5, 6)

*Bob receives the ciphertexts (5 and 6) and calculates the plaintext*

*Inverse calculation :* $P=[C_2 x (C_1^d)^{-1}] \bmod p$ $P=[C_2 x C_1^{p-1-d}] \bmod p$