UNIT II

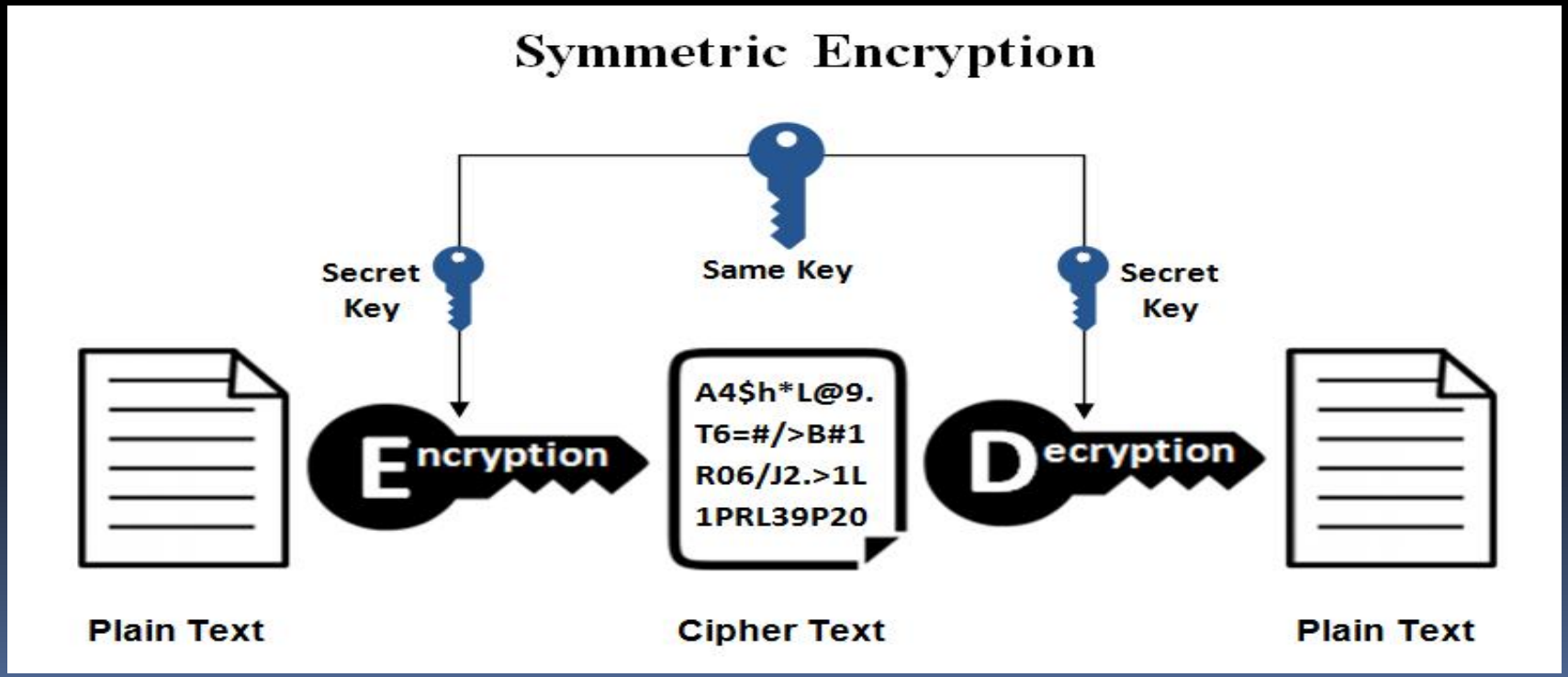
SYMMETRIC KEY CRYPTOGRAPHY

MATHEMATICS OF SYMMETRIC KEY CRYPTOGRAPHY: Algebraic structures – Modular arithmetic-Euclid‟s algorithm- Congruence and matrices – Groups, Rings, Fields- Finite fields- SYMMETRIC KEY CIPHERS: SDES – Block cipher Principles of DES – Strength of DES – Differential and linear cryptanalysis – Block cipher design principles – Block cipher mode of operation – Evaluation criteria for AES – Advanced Encryption Standard – RC4 – Key distribution.

Symmetric key cryptography

Symmetric-key algorithms are algorithms for **cryptography** that use the same **cryptographic keys** for both **encryption** of plaintext and decryption of ciphertext.

The **keys**, in practice, represent a shared secret between two or more parties that can be used to maintain a private information link.



MATHEMATICS OF SYMMETRIC KEY CRYPTOGRAPHY

- **ALGEBRIC STRUCTURES**
- In mathematics, more specifically in abstract algebra and universal algebra, an **algebraic structure** consists of a nonempty set *A* (called the **underlying set**, **carrier set** or **domain**), a collection of operations on *A* of finite arity (typically binary operations), and a finite set of identities, known as axioms,
  - **GROUPS**
  - **RINGS**
  - **FIELDS**

GROUPS

- A **group** G, sometimes denoted by {G, ·} is a set of elements with a binary operation, denoted by ·, that associates to each ordered pair (a, b) of elements in G an element (a · b) in G, such that the following axioms are obeyed:
- The operator · is generic and can refer to addition, multiplication, or some other mathematical operation.
- **(A1) Closure:**

  If a and b belong to G, then a · b is also in G.
- **(A2) Associative:**

  a · (b · c) = (a · b) · c for all a, b, c in G.
- **(A3) Identity element:**

  There is an element e in G such that a · e = e · a = a for all a in G.
- **(A4) Inverse element:**

  For each a in G there is an element a' in G such that a · a' = a' · a = e.

- **A group is said to be abelian if it satisfies the following additional condition:**

  (A5) Commutative:

  a · b = b · a for all a, b in G.

RINGS

- A **ring** R, sometimes denoted by {R, +, x}, is a set of elements with two binary operations, called addition and multiplication,such that for all a, b, c in R the following axioms are obeyed:
- Generally, we do not use the multiplication symbol, x, but denote multiplication by the concatenation of two elements.
- (A1-A5) R is an abelian group with respect to addition; that is, R satisfies axioms A1 through A5. For the case of an additive group, we denote the identity element as 0 and the inverse of a as a.
- **(M1) Closure under multiplication:**

  If a and b belong to R, then ab is also in R.
- **(M2) Associativity of multiplication:**

  a(bc) = (ab)c for all a, b, c in R.
- **(M3) Distributive laws:**

  a(b + c) = ab + ac for all a, b, c in R.
  (a + b)c = ac + bc for all a, b, c in R.
- **(M4) Commutativity of multiplication:**

  ab = ba for all a, b in R.
- **(M5) Multiplicative identity:**

  There is an element 1 in R such that a1 = 1a = a for all a in R.
- **(M6) No zero divisors:**

  If a, b in R and ab = 0, then either a = 0 or b = 0.

FIELDS

- A **field** F, sometimes denoted by {F, +, x}, is a set of elements with two binary operations, called addition and multiplication, such that for all a, b, c in F the following axioms are obeyed:

- (A1M6) F is an integral domain; that is, F satisfies axioms A1 through A5 and M1 through M6.

- (M7) Multiplicative inverse:
  For each a in F, except 0, there is an element
  a^{-1} in F such that $aa^{-1} = (a^{-1})a = 1$.

Modular Arithmetic Operations

1. $[(a \bmod n) + (b \bmod n)] \bmod n = (a + b) \bmod n$
2. $[(a \bmod n) - (b \bmod n)] \bmod n = (a - b) \bmod n$
3. $[(a \bmod n) \times (b \bmod n)] \bmod n = (a \times b) \bmod n$

Verify the above properties using suitable examples .

Modular Exponentiation

$5^{40} \bmod 7$

$5^1 \bmod 7 = 5$

$5^2 \bmod 7 = 4$

$5^4 \bmod 7 = (5^2)^2 \bmod 7$

$= 4^2 \bmod 7$

$= 16 \bmod 7 = 2$

$5^8 \bmod 7 = (5^4)^2 \bmod 7$

$= 4 \bmod 7 = 4$

$5^{16} \bmod 7 = (5^8)^2 \bmod 7$

$= 4^2 \bmod 7 = 16 \bmod 7$

$= 2$

$5^{32} \bmod 7 = (5^{16})^2 \bmod 7$

$= 4 \bmod 7$

$= 4$

$5^{32} \cdot 5^8 \bmod 7$

$4 * 4 \bmod 7$

$16 \bmod 7 = 2$

Modular Exponentiation

$7^{69} \bmod 5$

$7 \bmod 5 = 2$

$7^2 \bmod 5 = 49 \bmod 5$

$= 4$

$7^4 \bmod 5 = (7^2)^2 \bmod 5$

$= 16 \bmod 5$

$= 1$

$7^8 \bmod 5 = 1$

$7^{16} \bmod 5 = 1$

$7^{32} \bmod 5 = 1$

$7^{64} \bmod 5 = 1$

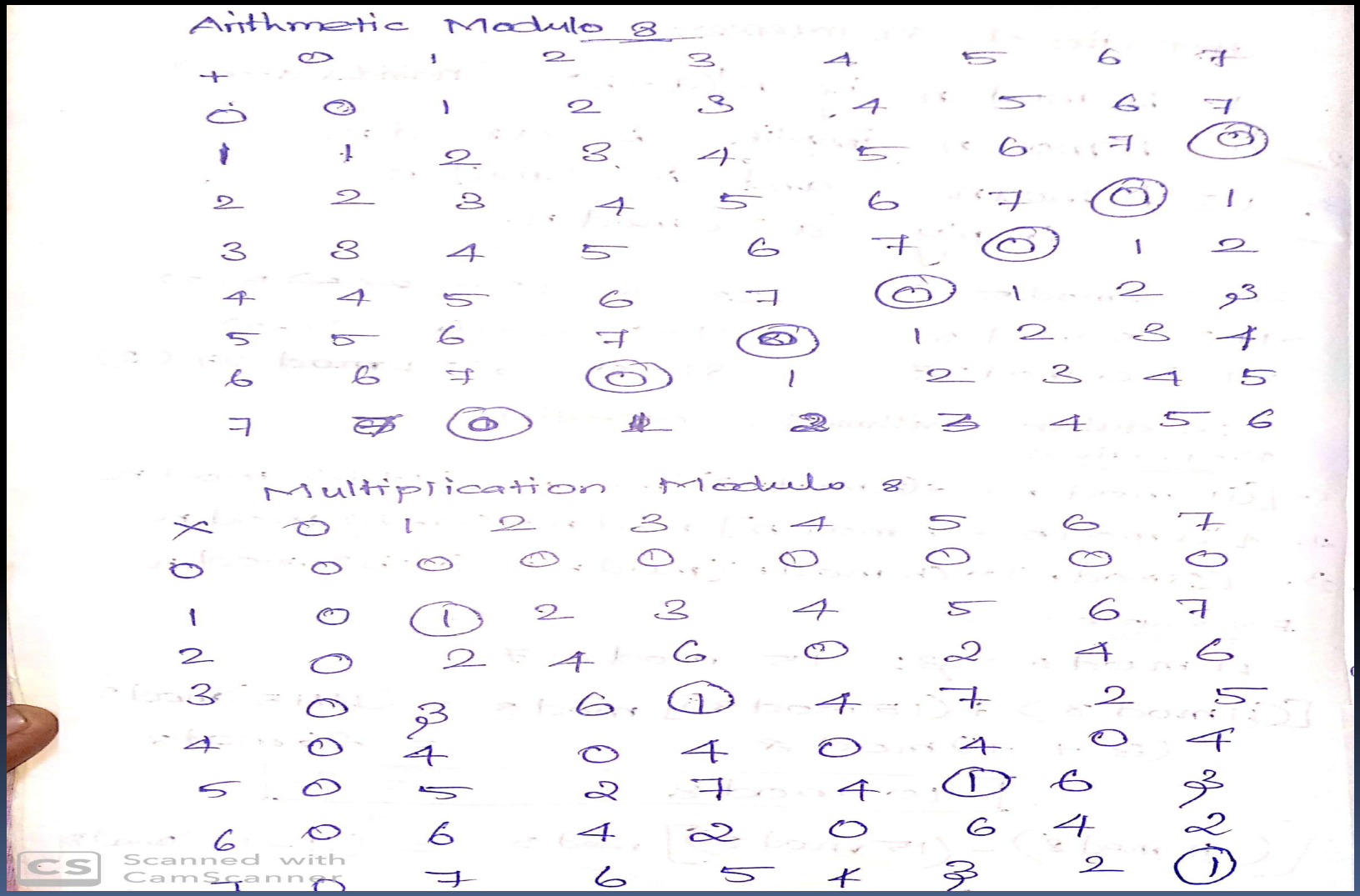
$7^{64} \cdot 7^4 \cdot 7^1 \bmod 5$

$1 \cdot 1 \cdot 2 \bmod 5$

$2 \bmod 5$

$\underline{\underline{2}}$

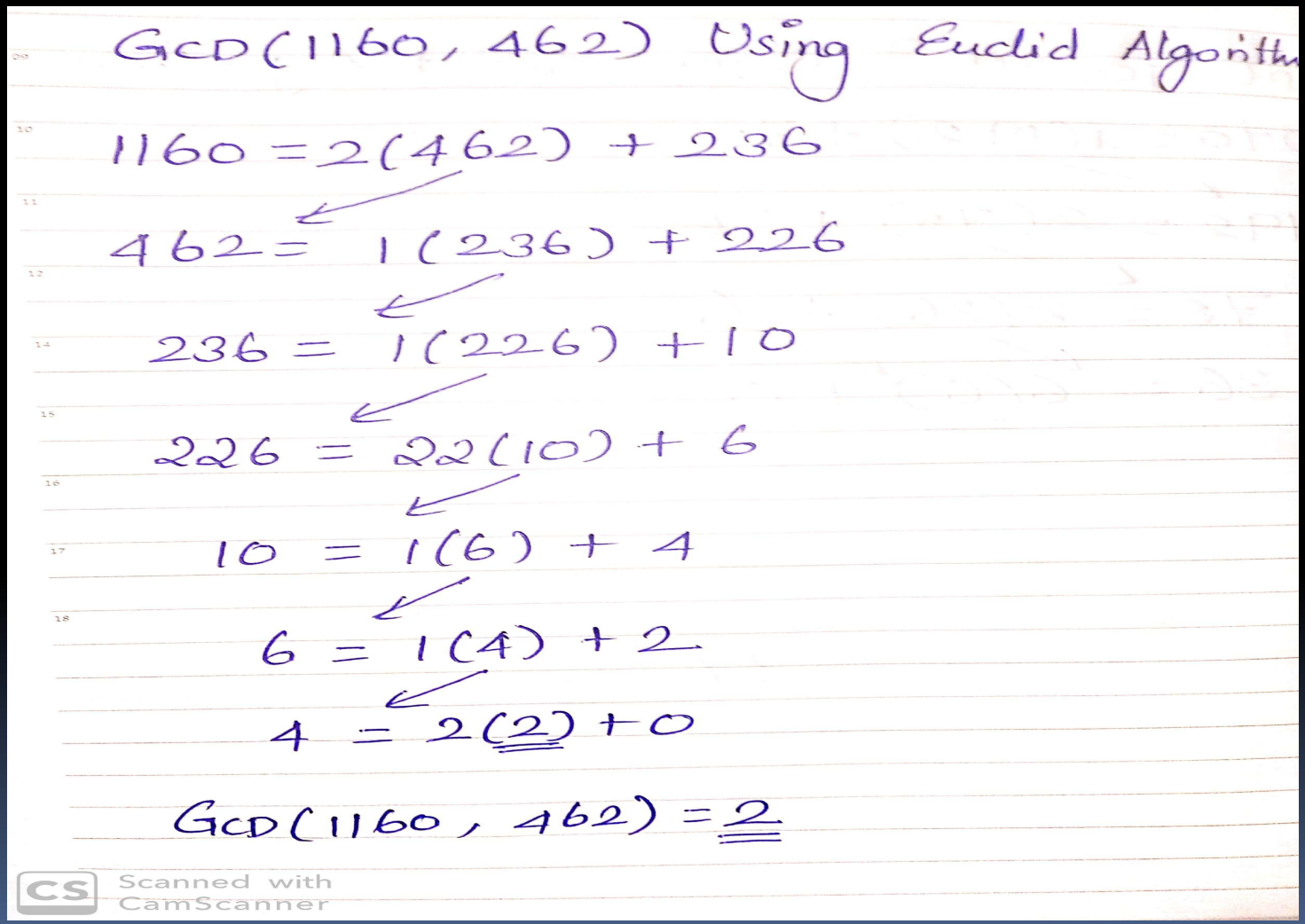
Modular Exponentiation



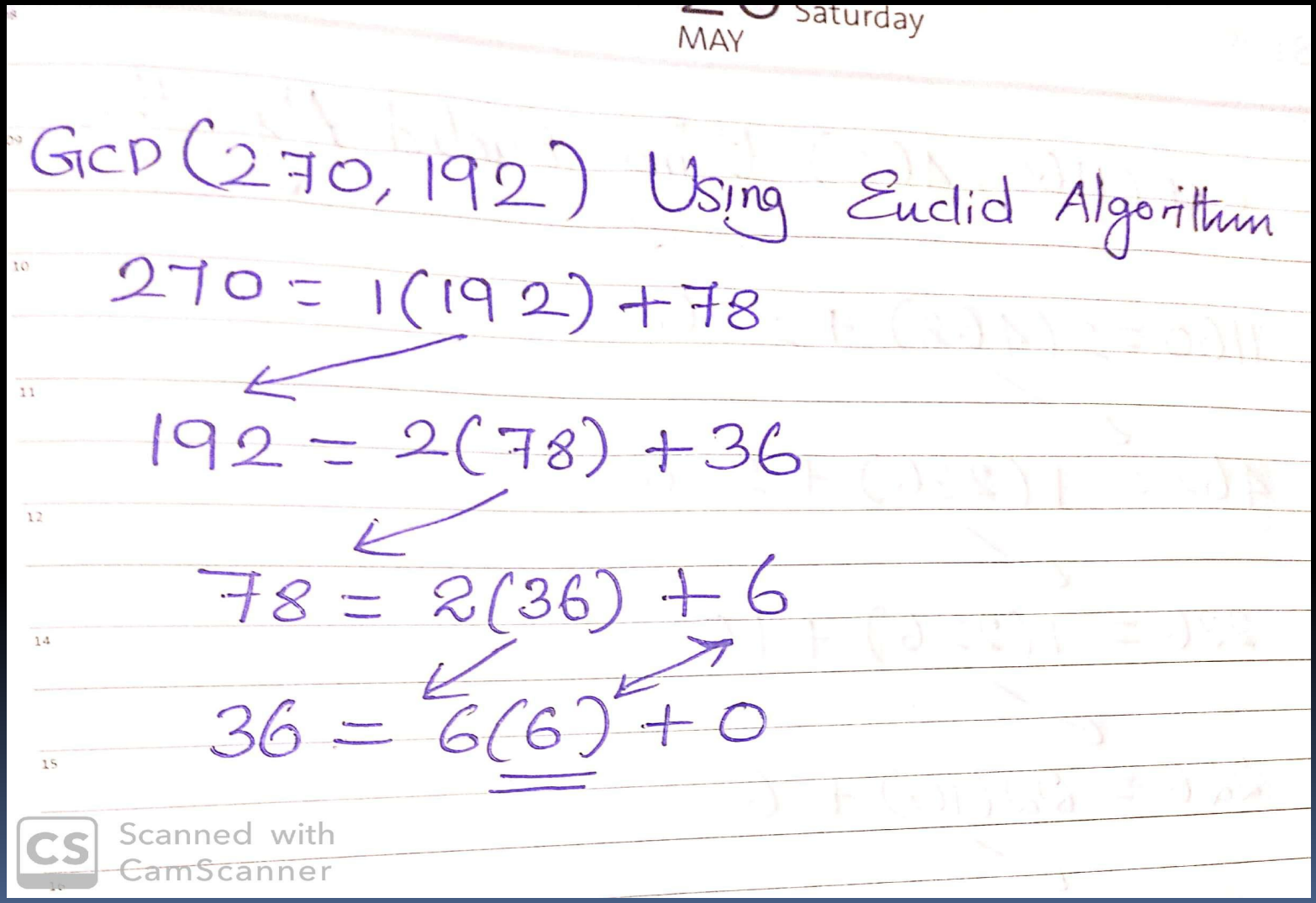
EUCLID'S ALGORITHM

Euclidean Algorithm	
Calculate	**Which satisfies**
$r_1 = a \bmod b$	$a = q_1 b + r_1$
$r_2 = b \bmod r_1$	$b = q_2 r_1 + r_2$
$r_3 = r_1 \bmod r_2$	$r_1 = q_3 r_2 + r_3$
•	•
•	•
•	•
$r_n = r_{n-2} \bmod r_{n-1}$	$r_{n-2} = q_n r_{n-1} + r_n$
$r_{n+1} = r_{n-1} \bmod r_n = 0$	$r_{n-1} = q_{n+1} r_n + 0$ $d = \gcd(a, b) = r_n$

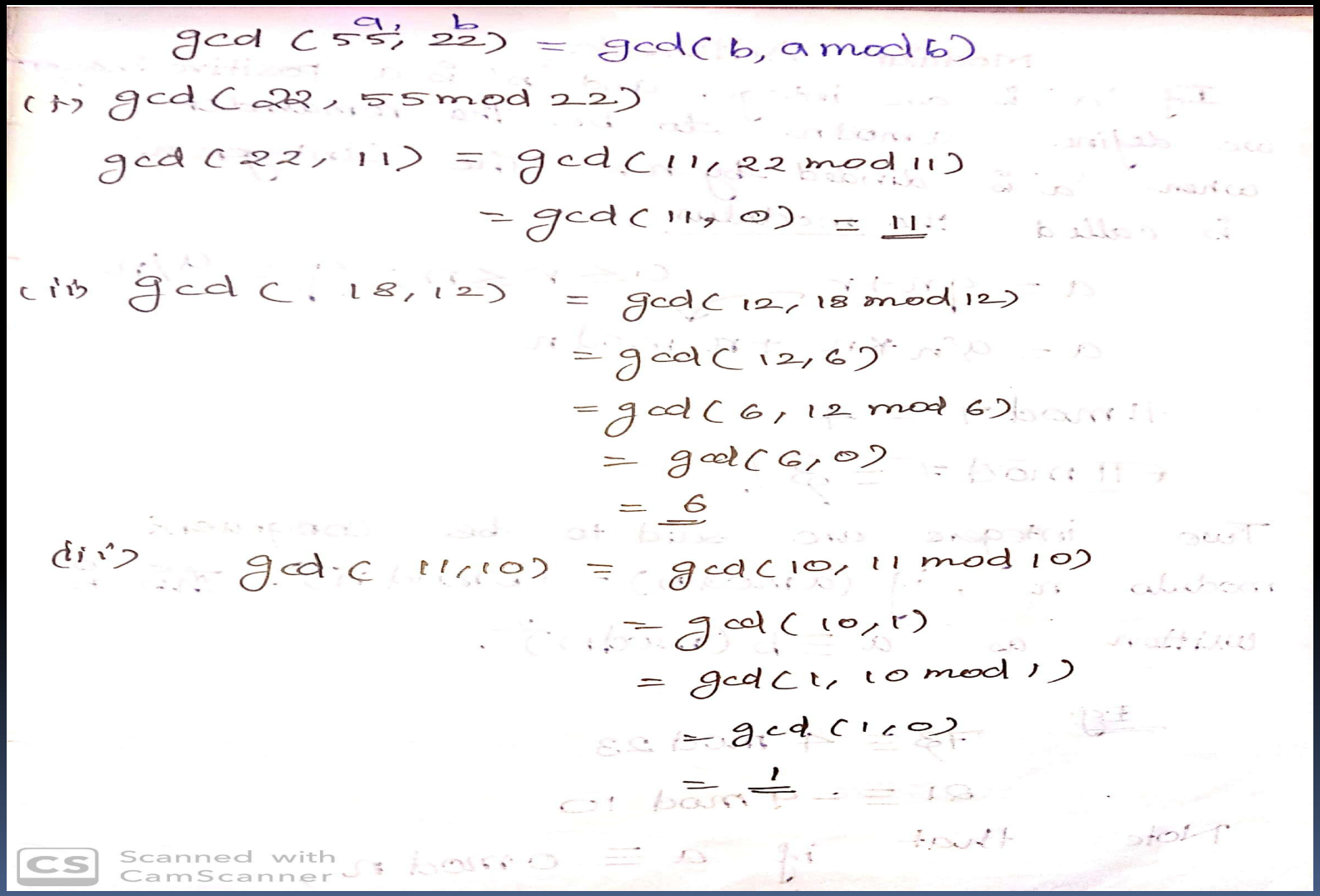
FIND GCD USING EUCLID ALG



FIND GCD USING EUCLID ALG



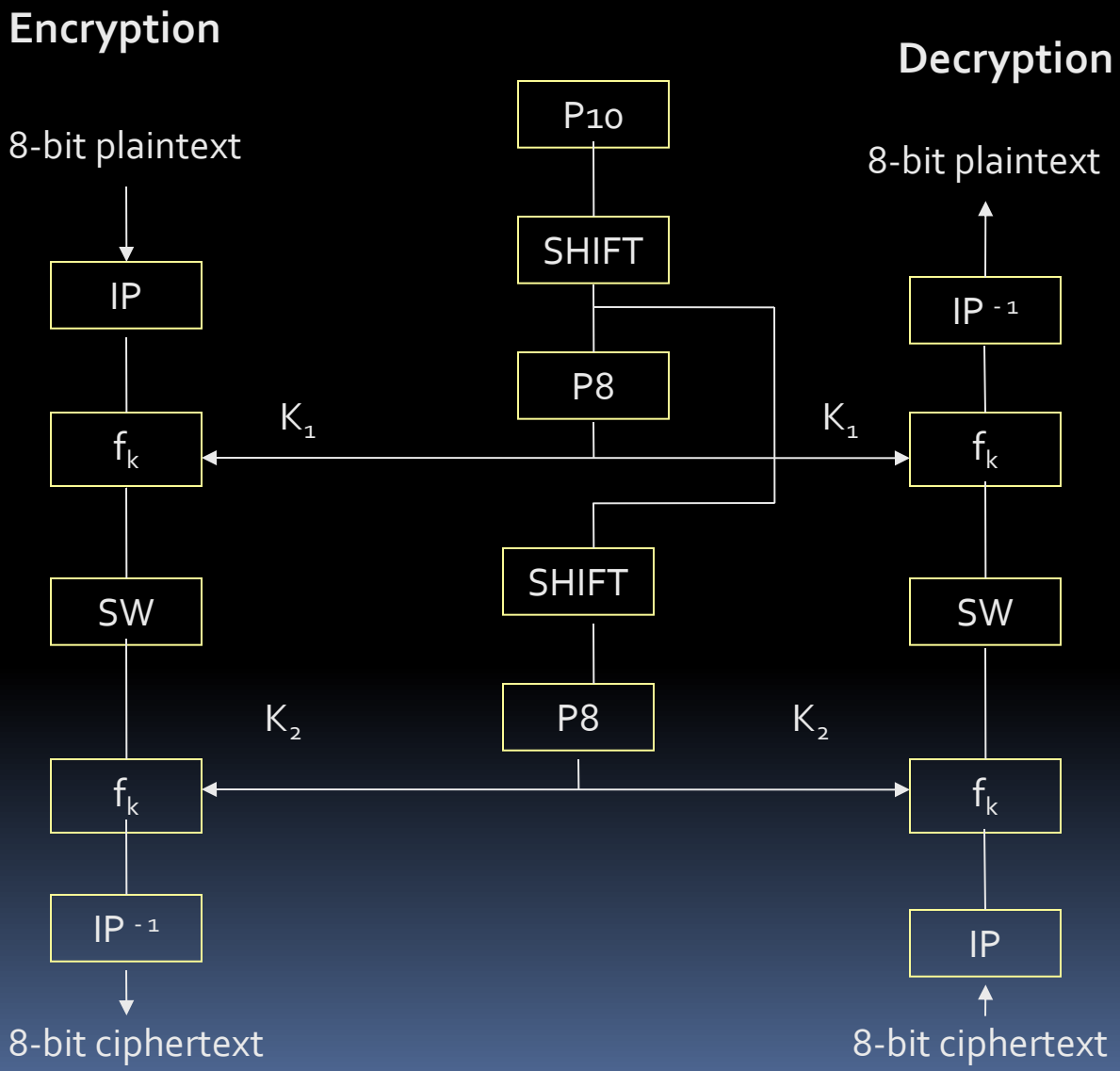
FIND GCD USING EUCLID ALG



What is Simplified DES

- **S-DES or Simplified Data Encryption Standard**
- The process of encrypting a plan text into an encrypted message with the use of S-DES has been divided into multi-steps which may help you to understand it as easily as possible.
- Points should be remembered.
- It is a block cipher.
- It has 8-bits block size of plain text or cipher text.
- It uses 10-bits key size for encryption.
- It is a symmetric cipher.
- It has Two Rounds.

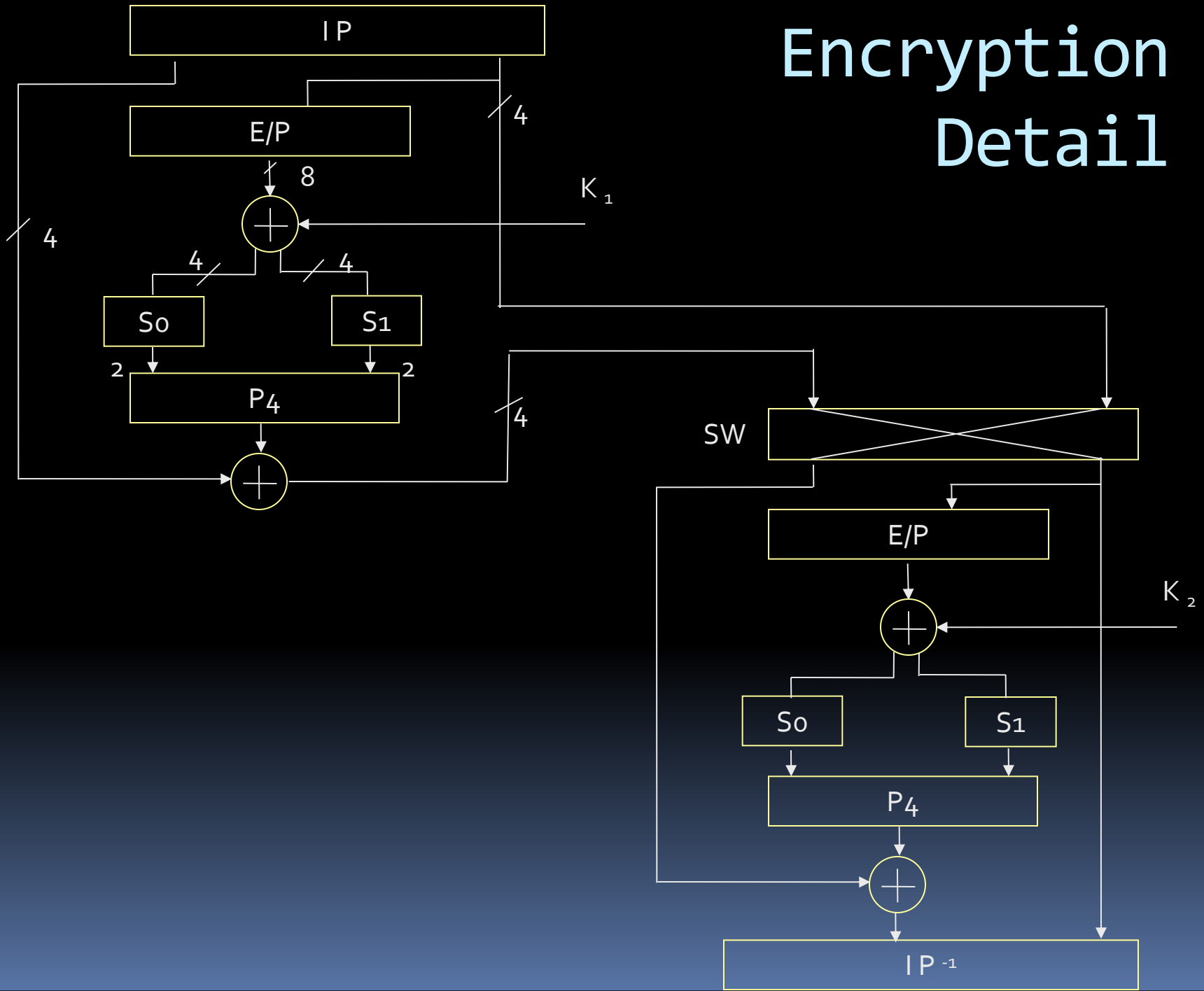
S-DES Scheme


Encryption
Decryption
P10
8-bit plaintext
8-bit plaintext
SHIFT
IP
IP -1
P8
K1
K1
fk
fk
SHIFT
SW
SW
P8
K2
K2
fk
fk
IP -1
IP
8-bit ciphertext
8-bit ciphertext

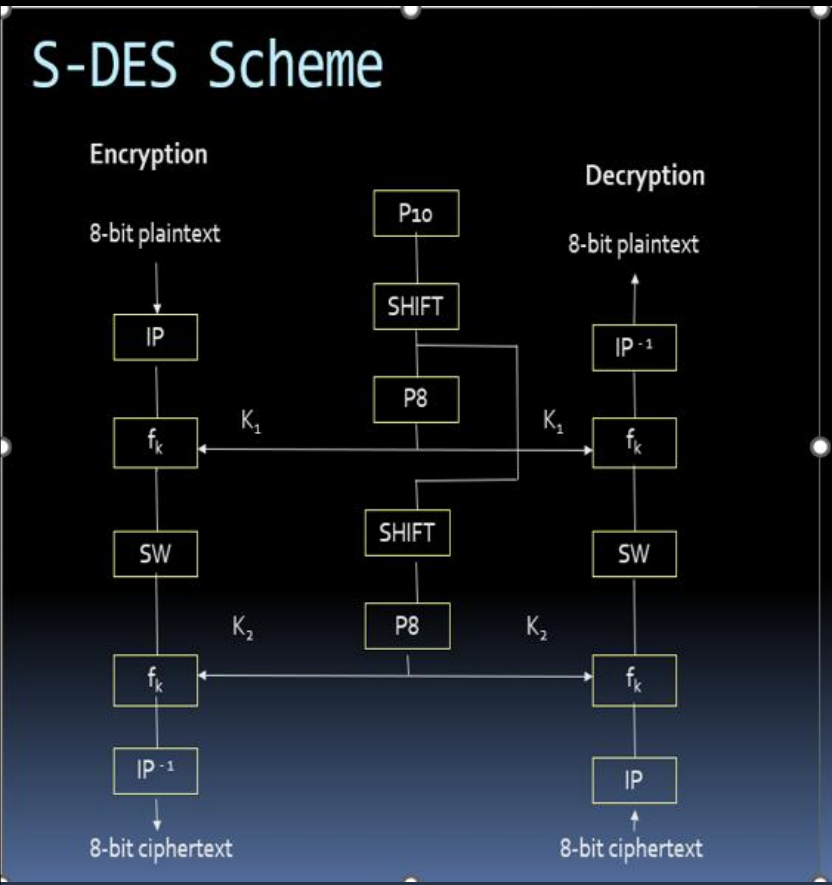
Five Functions to Encrypt

- IP – an initial permutation
- f_k - a complex, 2-input function
- SW – a simple permutation that swaps the two nybles
- f_k - a complex, 2-input function; again
- IP – inverse permutation of the initial permutation

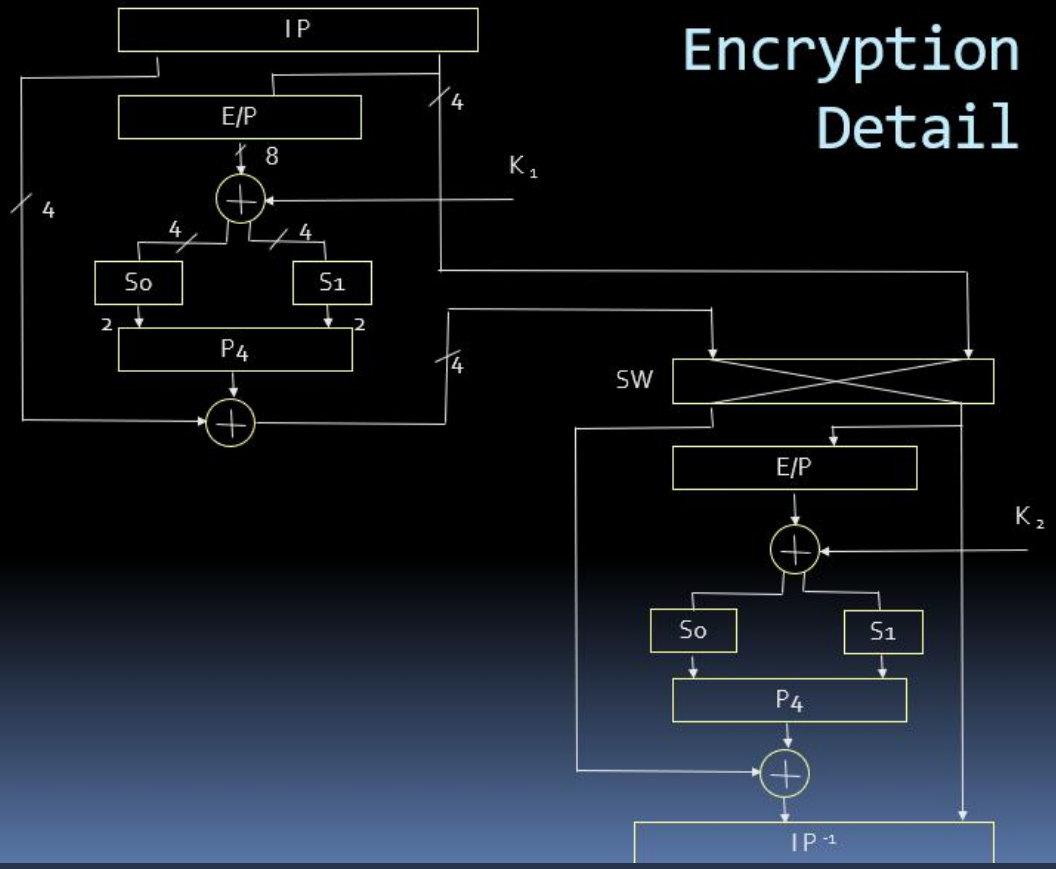
Encryption Detail



S-DES Scheme

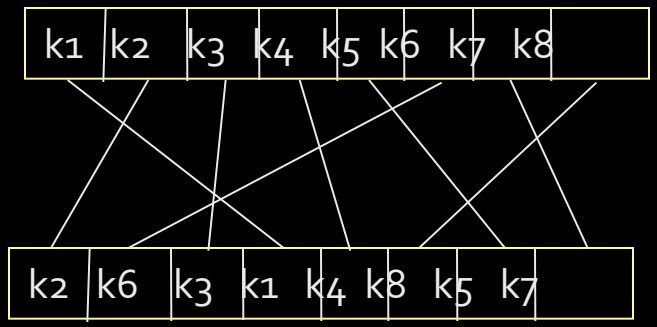


Encryption Detail



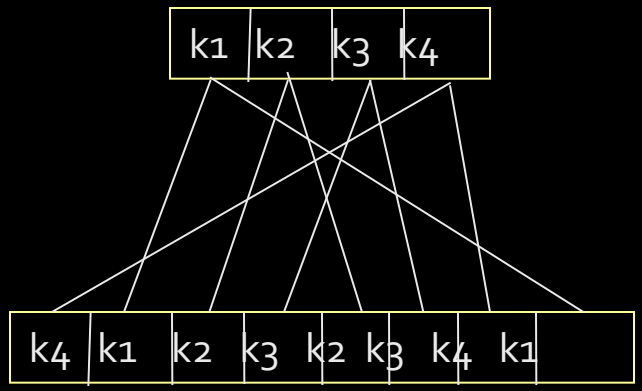
Initial Permutation (IP)

Move the bits of the original character around a little…



Expansion/Permutation (E/P)

Expand 4 bits into 8 and permutate them…



Substitution Boxes

S_0

1	0	3	2
3	2	1	0
0	2	1	3
3	1	3	2

S_1

0	1	2	3
2	0	1	3
3	0	1	0
2	1	0	3

S - Box operation

→ first and fourth input bits specifies a row of S-Box.

→ second and third input bits specifies a column of S-Box

→ 0 1 0 0 — Column (1 0) → 2nd Column; Row (0 0) → 0th Row

00 → 0th Row

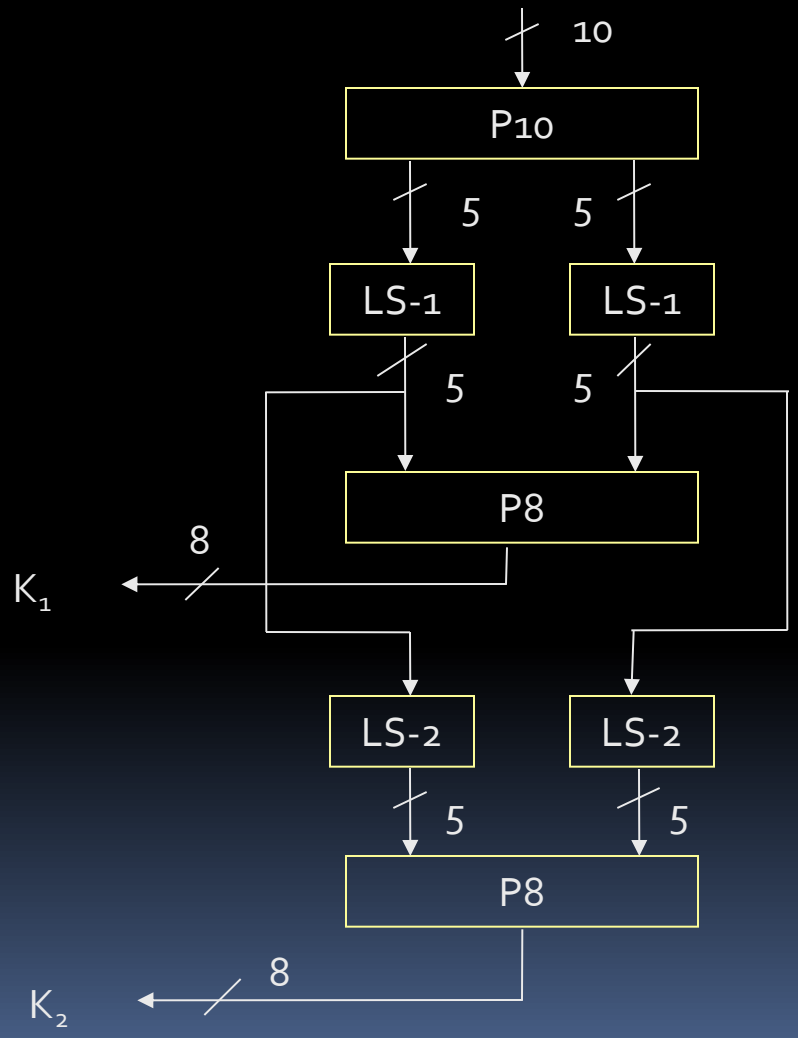
10 → 2nd Column

S_0

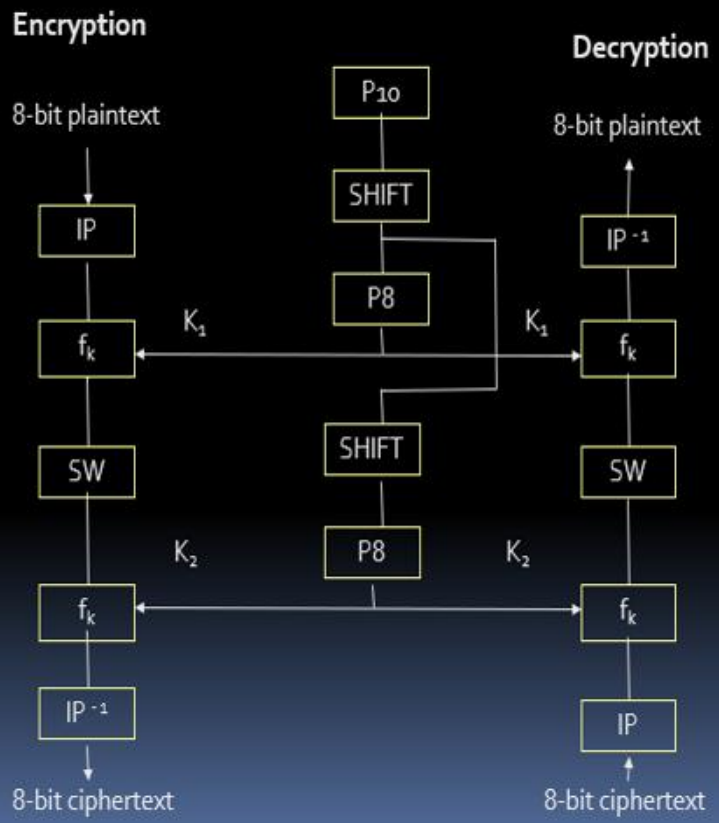
	0	1	2	3
0	1	0	(3)	2
1	3	2	1	0
2	0	2	1	3
3	3	1	3	2

3 = 11 (2 bits)

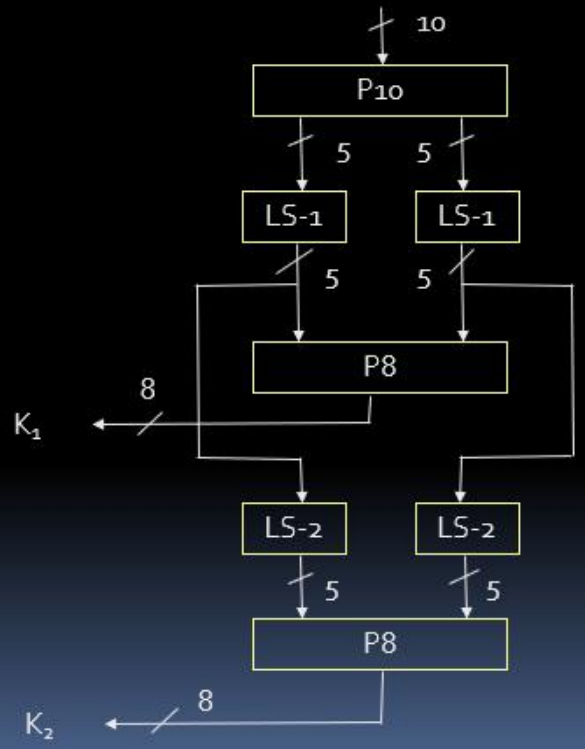
Key Generation



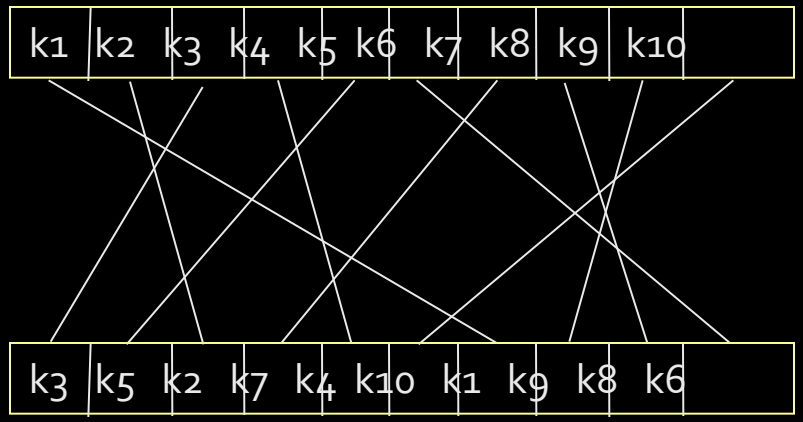
S-DES Scheme



Key Generation

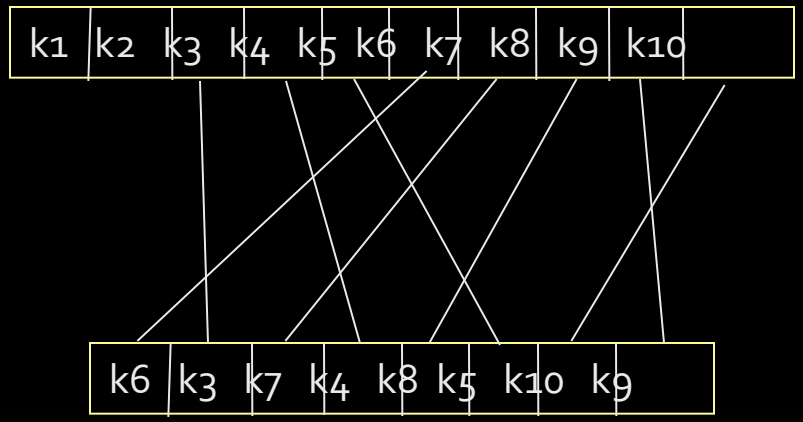


P10 Permutation



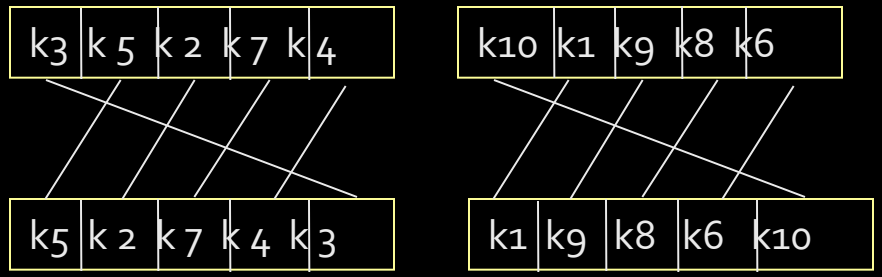
P8 Permutation

Permutate 10 into 8



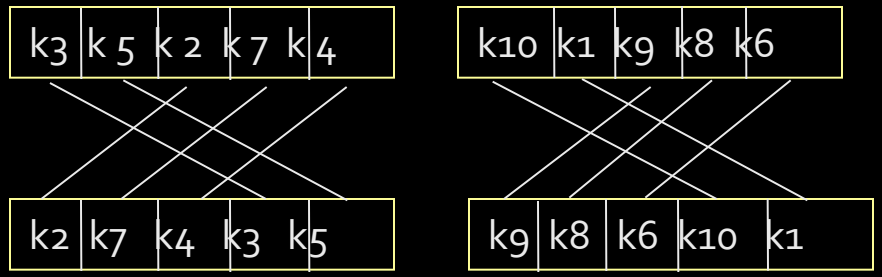
LS-1

Left circular shift 1 each 5 bit group



LS-2

Left circular shift 2 each 5 bit group



DATA ENCRYPTION STANDARD

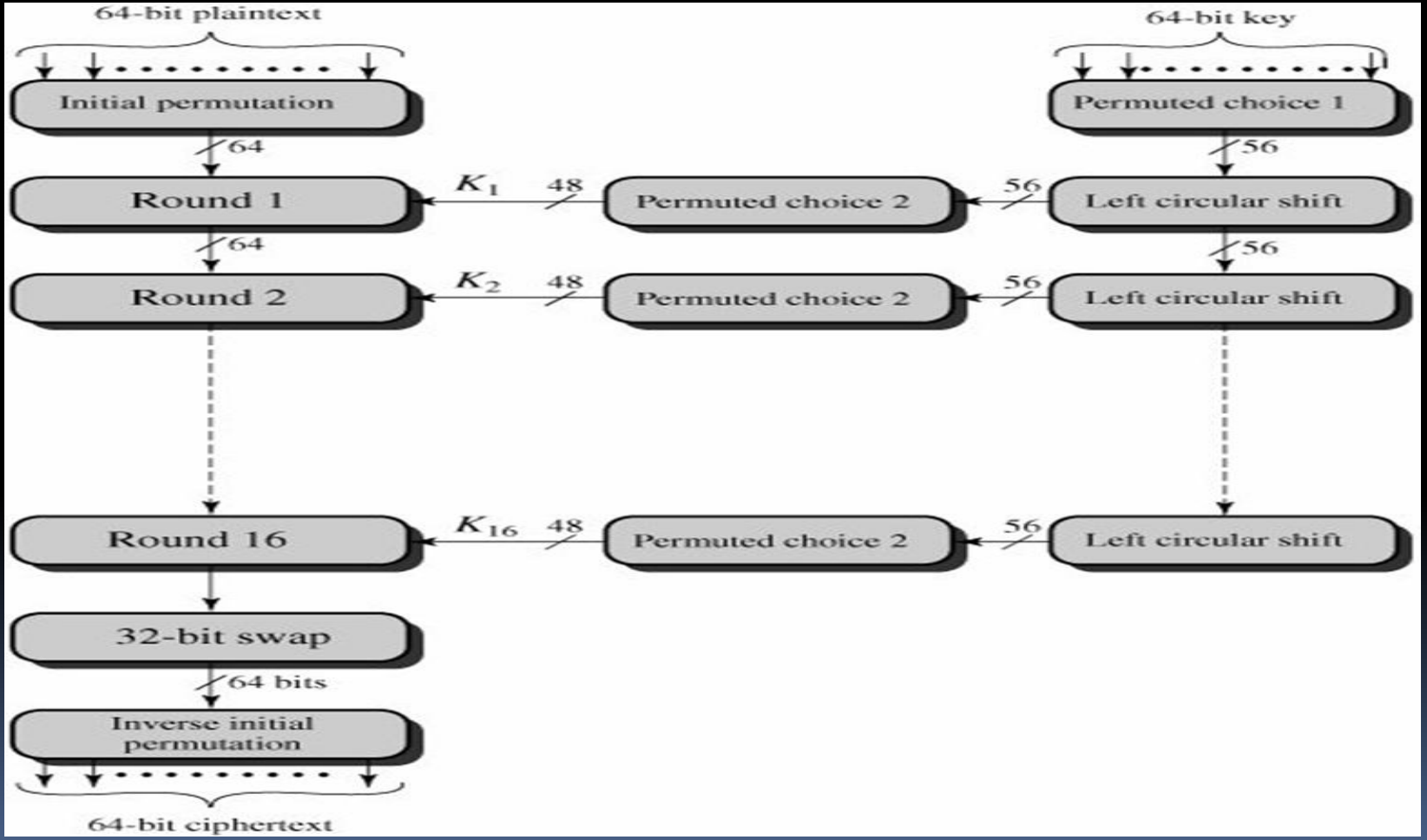
The Data Encryption Standard (DES) is a symmetric-key block cipher published by the National Institute of Standards and Technology (NIST).

DES is an implementation of a Feistel Cipher. It uses 16 round Feistel structure. The block size is 64-bit. Though, key length is 64-bit, DES has an effective key length of 56 bits, since 8 of the 64 bits of the key are not used by the encryption algorithm (function as check bits only).

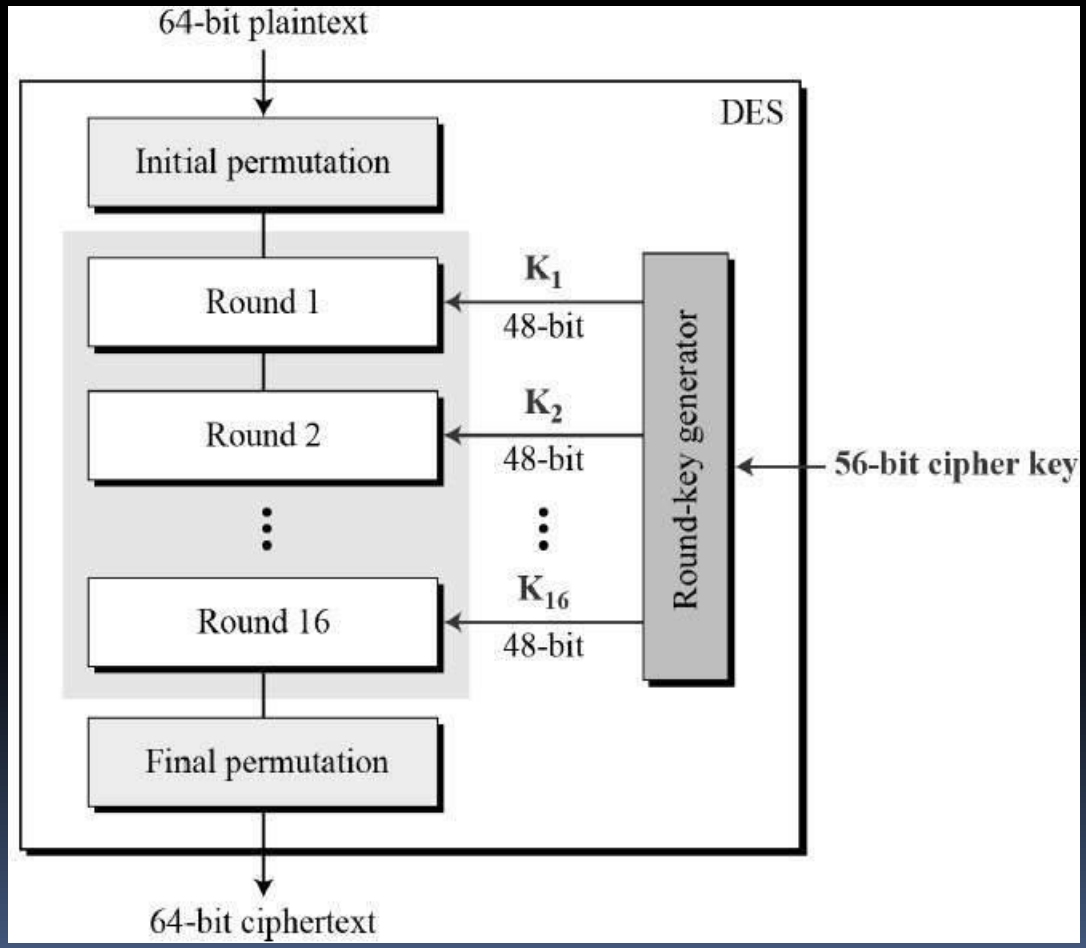
DES is based on the Feistel Cipher, all that is required to specify DES is

- Round function
- Key schedule
- Any additional processing − Initial and final permutation

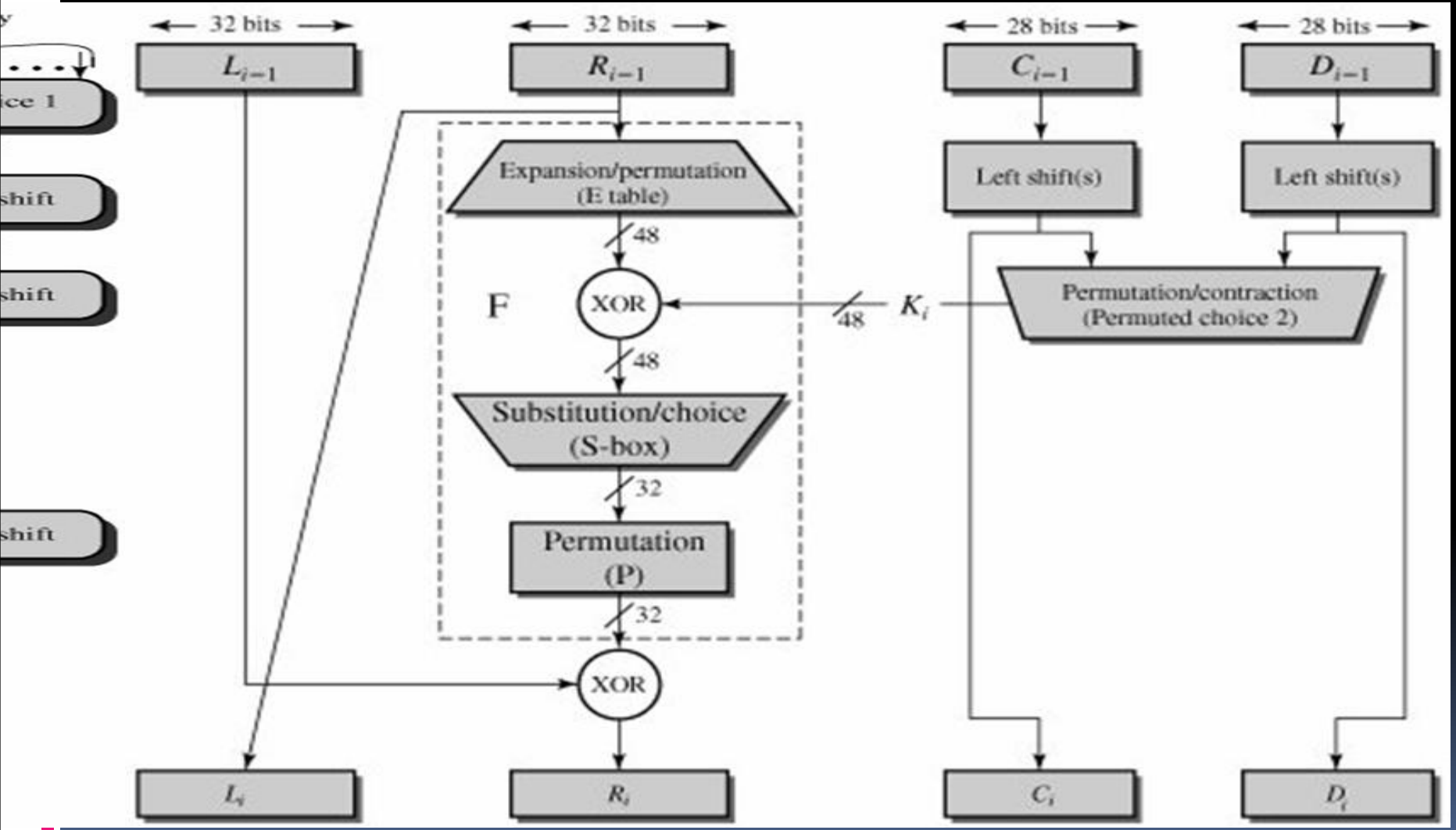
DES STRUCTURE

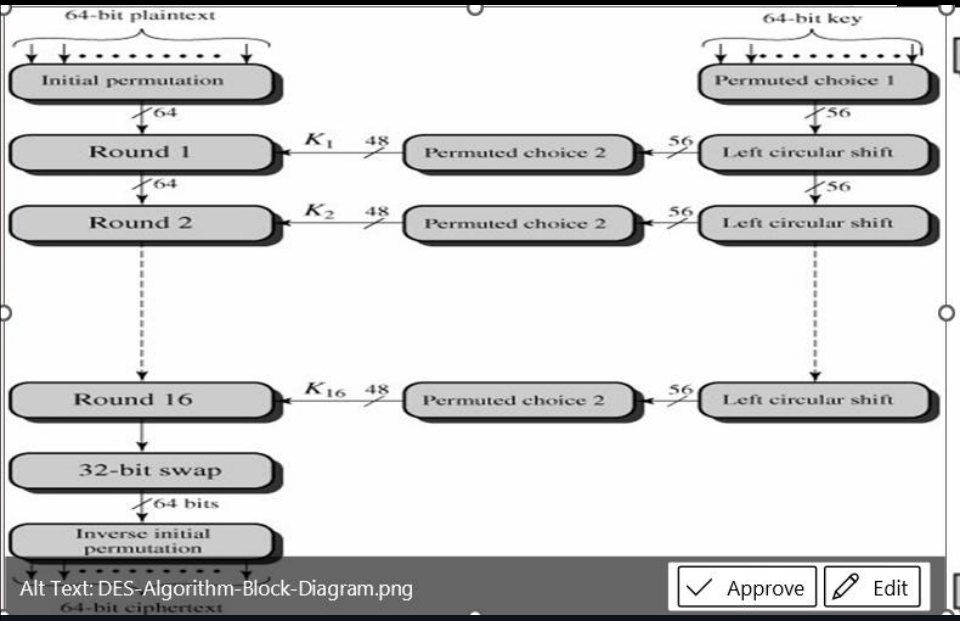


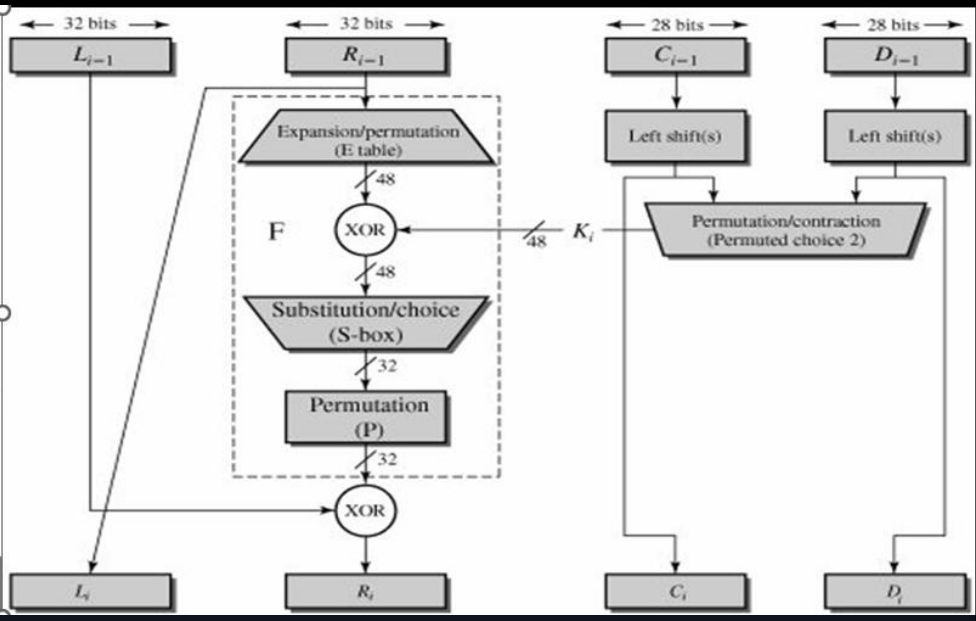
DES STRUCTURE



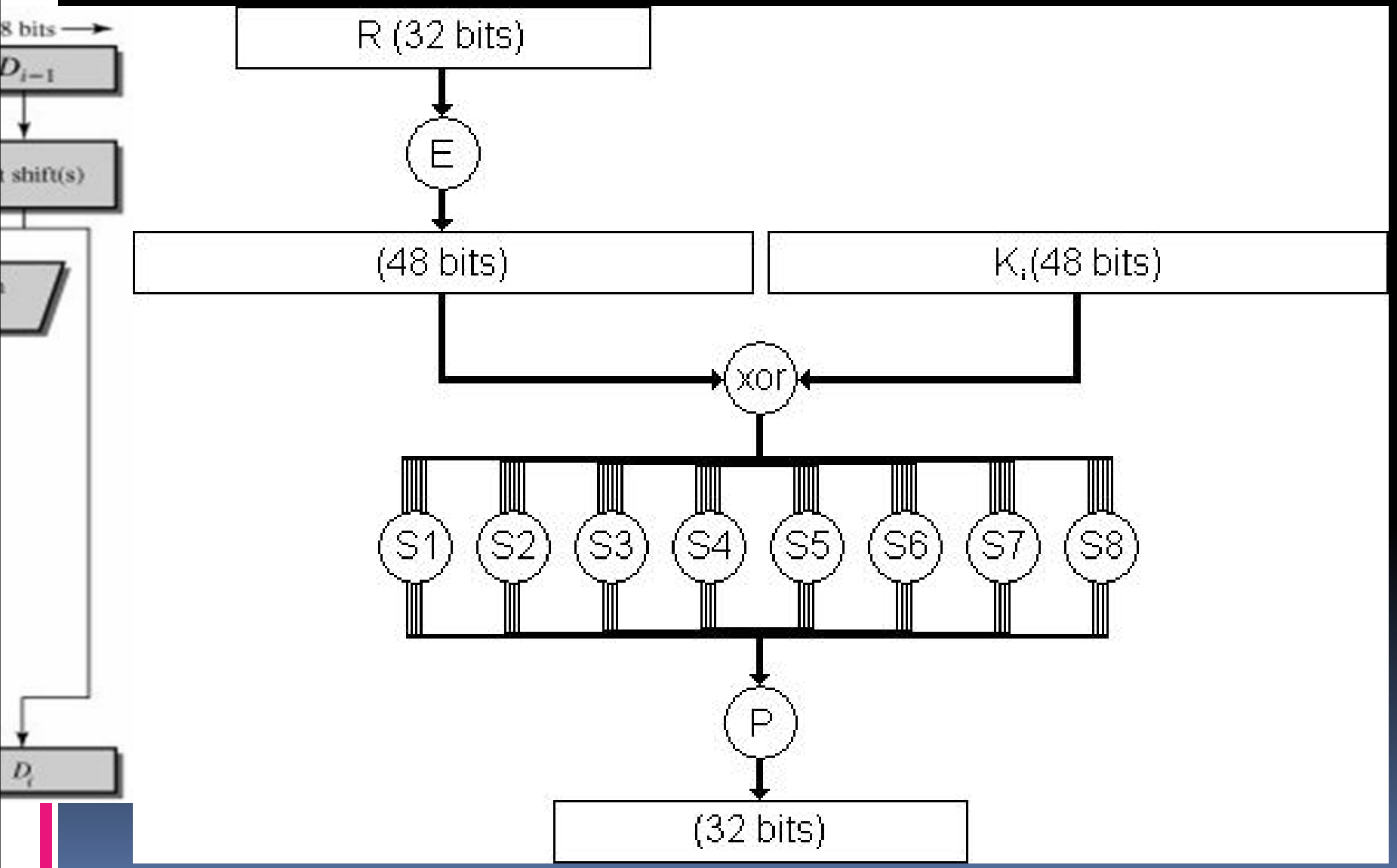
SINGLE ROUND FUNCTION

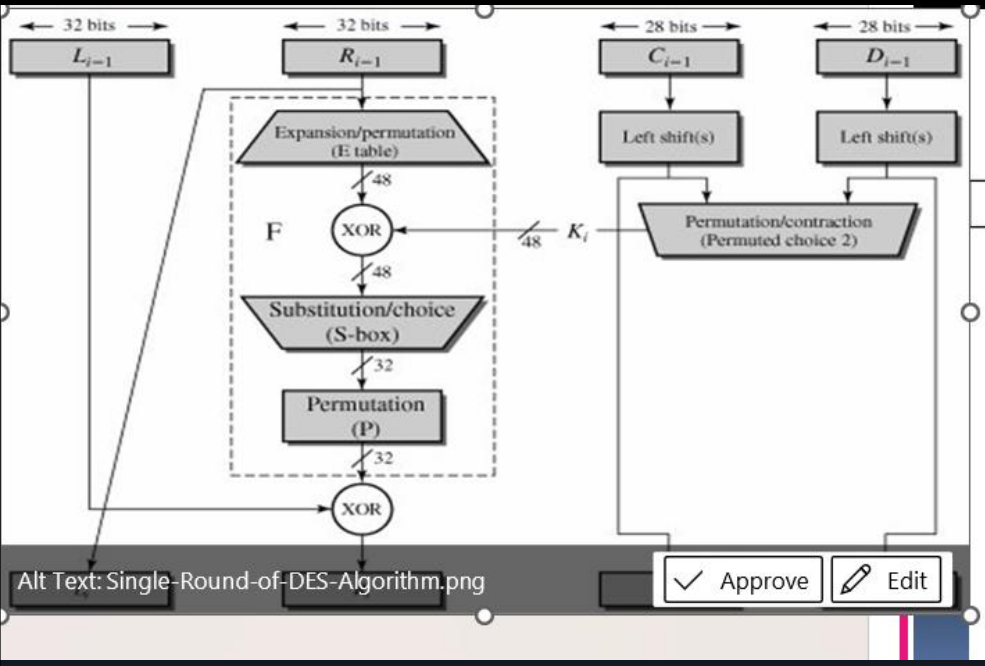


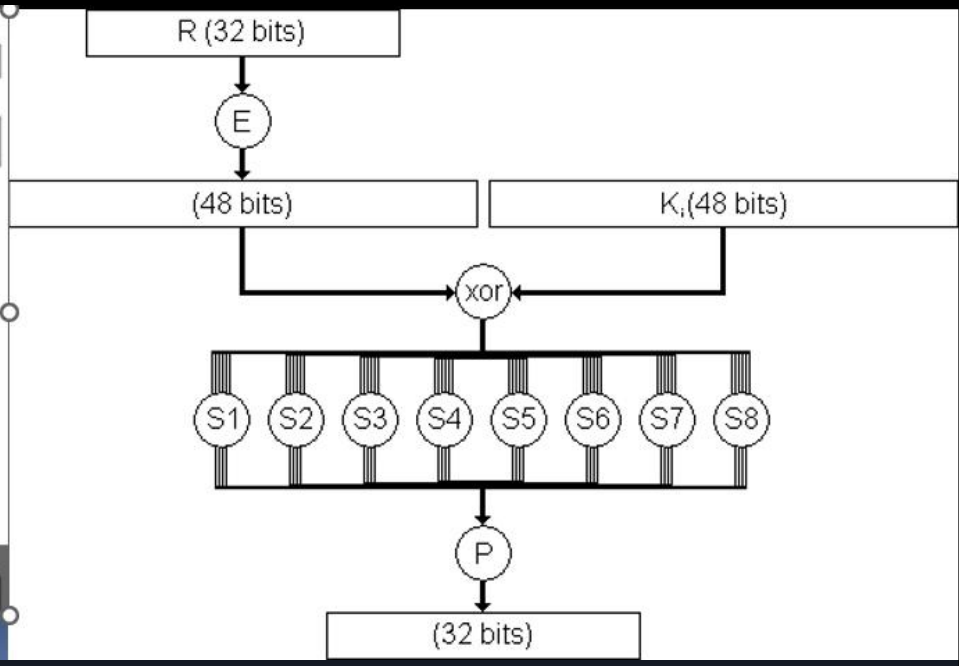

64-bit plaintext
64-bit key
Initial permutation
Permuted choice 1
64
56
Round 1
K_1
48
Permuted choice 2
56
Left circular shift
64
56
Round 2
K_2
48
Permuted choice 2
56
Left circular shift
Round 16
K_{16}
48
Permuted choice 2
56
Left circular shift
32-bit swap
64 bits
Inverse initial permutation
64-bit ciphertext
Alt Text: DES-Algorithm-Block-Diagram.png
Approve
Edit


32 bits
32 bits
28 bits
28 bits
L_{i-1}
R_{i-1}
C_{i-1}
D_{i-1}
Expansion/permutation (E table)
Left shift(s)
Left shift(s)
48
F
XOR
48
K_i
Permutation/contraction (Permuted choice 2)
48
Substitution/choice (S-box)
32
Permutation (P)
32
XOR
L_i
R_i
C_i
D_i

ROLE OF S-BOXES




32 bits
32 bits
28 bits
28 bits
L_{i-1}
R_{i-1}
C_{i-1}
D_{i-1}
Expansion/permutation
(E table)
Left shift(s)
Left shift(s)
48
F
XOR
48
K_i
Permutation/contraction
(Permuted choice 2)
48
Substitution/choice
(S-box)
32
Permutation
(P)
32
XOR
Alt Text: Single-Round-of-DES-Algorithm.png
Approve
Edit


R (32 bits)
E
(48 bits)
K_i(48 bits)
xor
S1
S2
S3
S4
S5
S6
S7
S8
P
(32 bits)

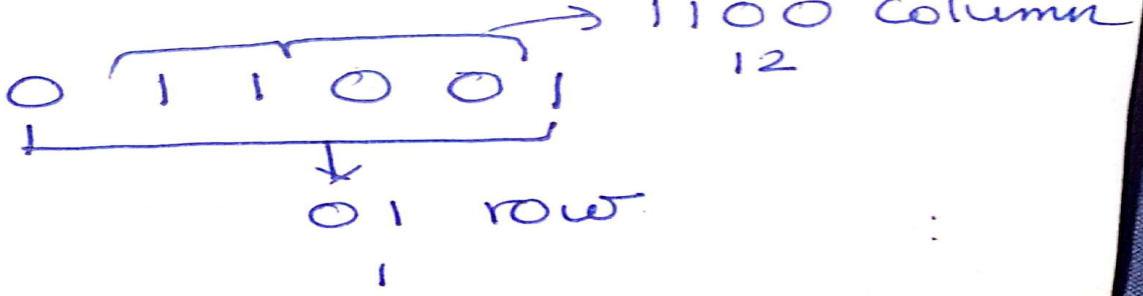
(i) 8 S-Boxes Used in DES

(ii) Each S-Box with 6 Inputs and produces 4 outputs.

(iii) The transformation defined as following

14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0

Let us Consider



1st row
12th column

AVALANCHE EFFECT

- A Small changes in either the plaintext and the key it produces a significant changes in the cipher text is called avalanche effect.
- Advantages of DES

  Brute force attack is not possible because to find 2 power 56 keys is impractical.

Double DES

- Double DES is a encryption technique which uses two instance of DES on same plain text. In both instances it uses different keys to encrypt the plain text.

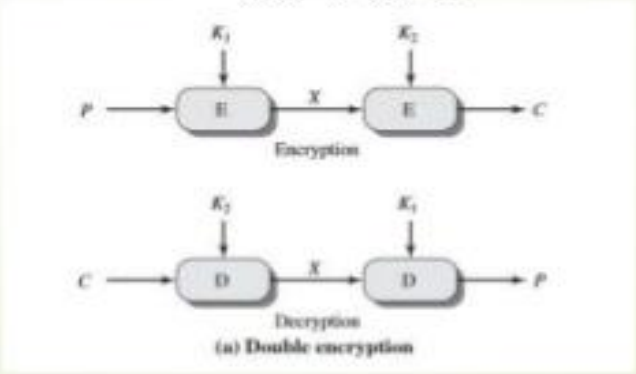
Double DES

- Given a plaintext P and two encryption keys K_1 and K_2, a cipher text can be generated as,

$$C = E(K_2, E(K_1, P))$$

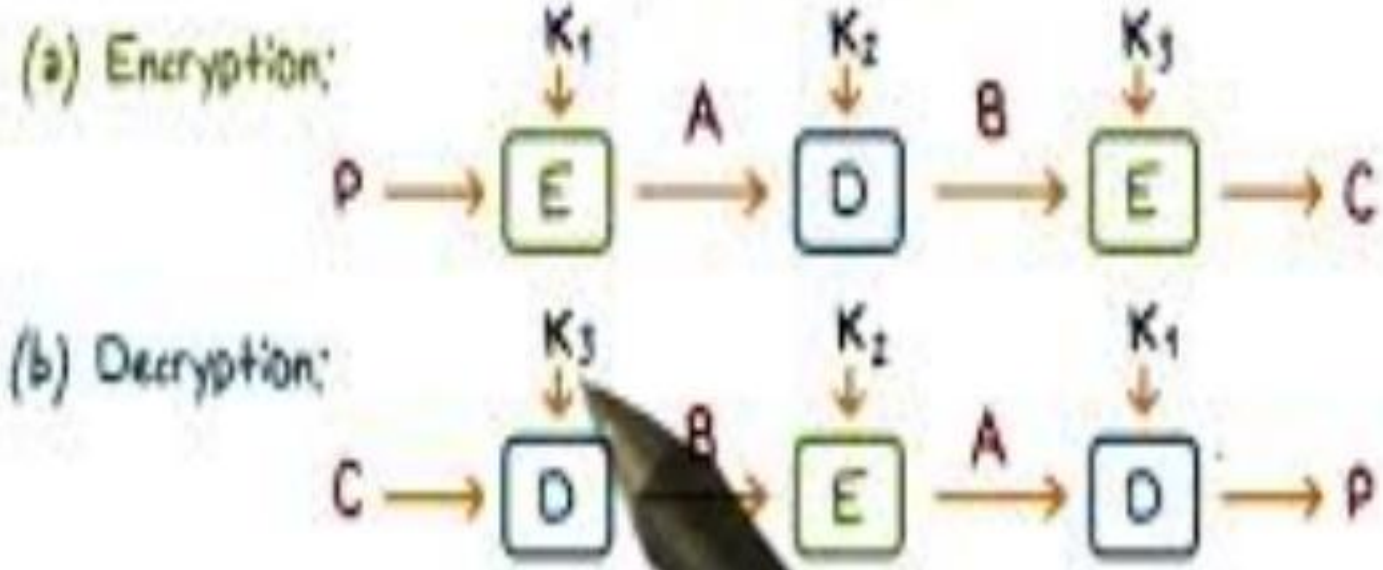
- Decryption requires that the keys be applied in reverse order,

$$P = D(K_1, D(K_2, C))$$



(a) Double encryption

Triple DES



- $K_1 = K_3$ results in an equivalent [illegible]-bit DES which provides a sufficient key space
- Distinct K_1, K_2, K_3 results in an even stronger [illegible]-bit DES
- Can run as a single DES with $K_1 = K_2$

MEET IN THE MIDDLE ATTACK

- The meet-in-the-middle attack is one of the types of known plaintext attacks.
- The intruder has to know some parts of plaintext and their ciphertexts. Using meet-in-the-middle attacks it is possible to break ciphers, which have two or more secret keys for multiple encryption using the same algorithm.
- For example, the 3DES cipher works in this way. Meet-in-the-middle attack was first presented by Diffie and Hellman for cryptanalysis of DES algorithm.

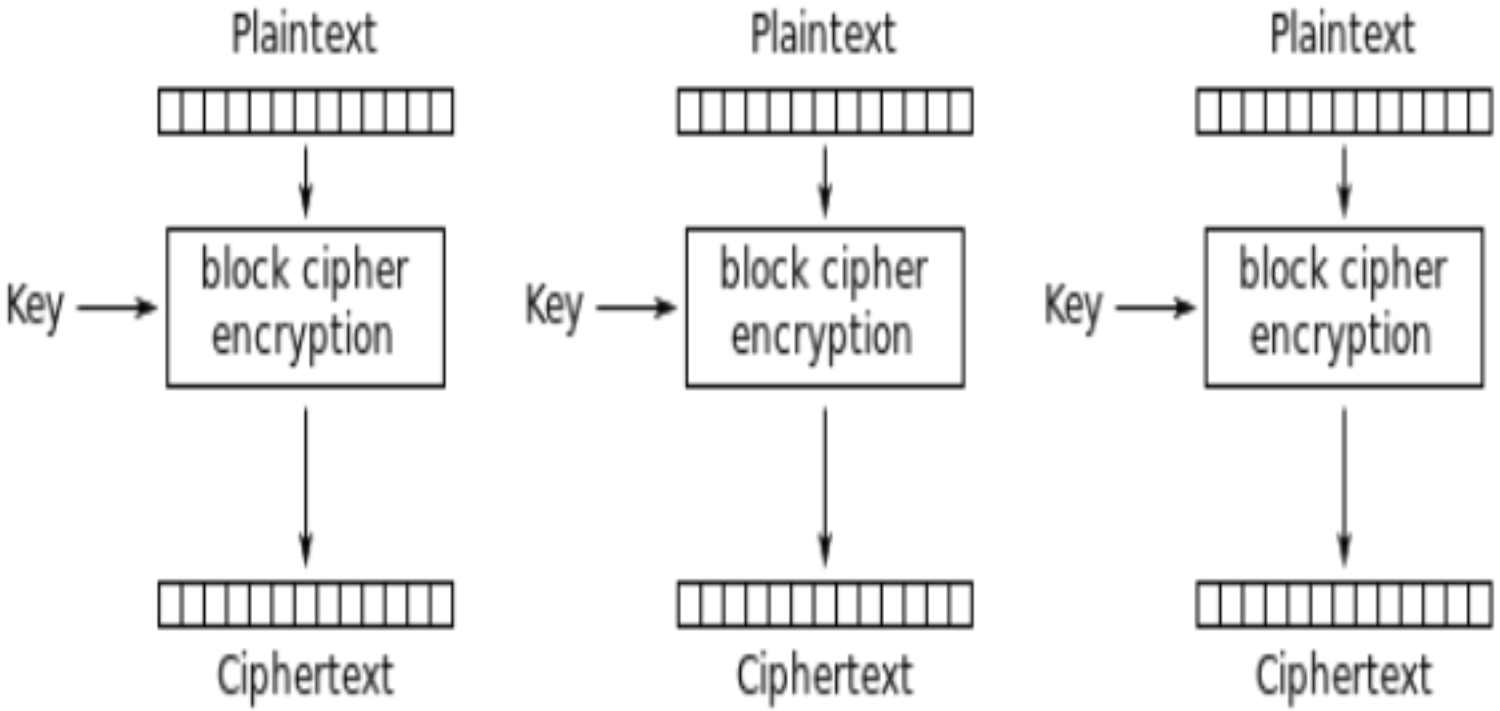
Block Cipher Modes Operation

Main block cipher modes of operation: confidentiality only

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

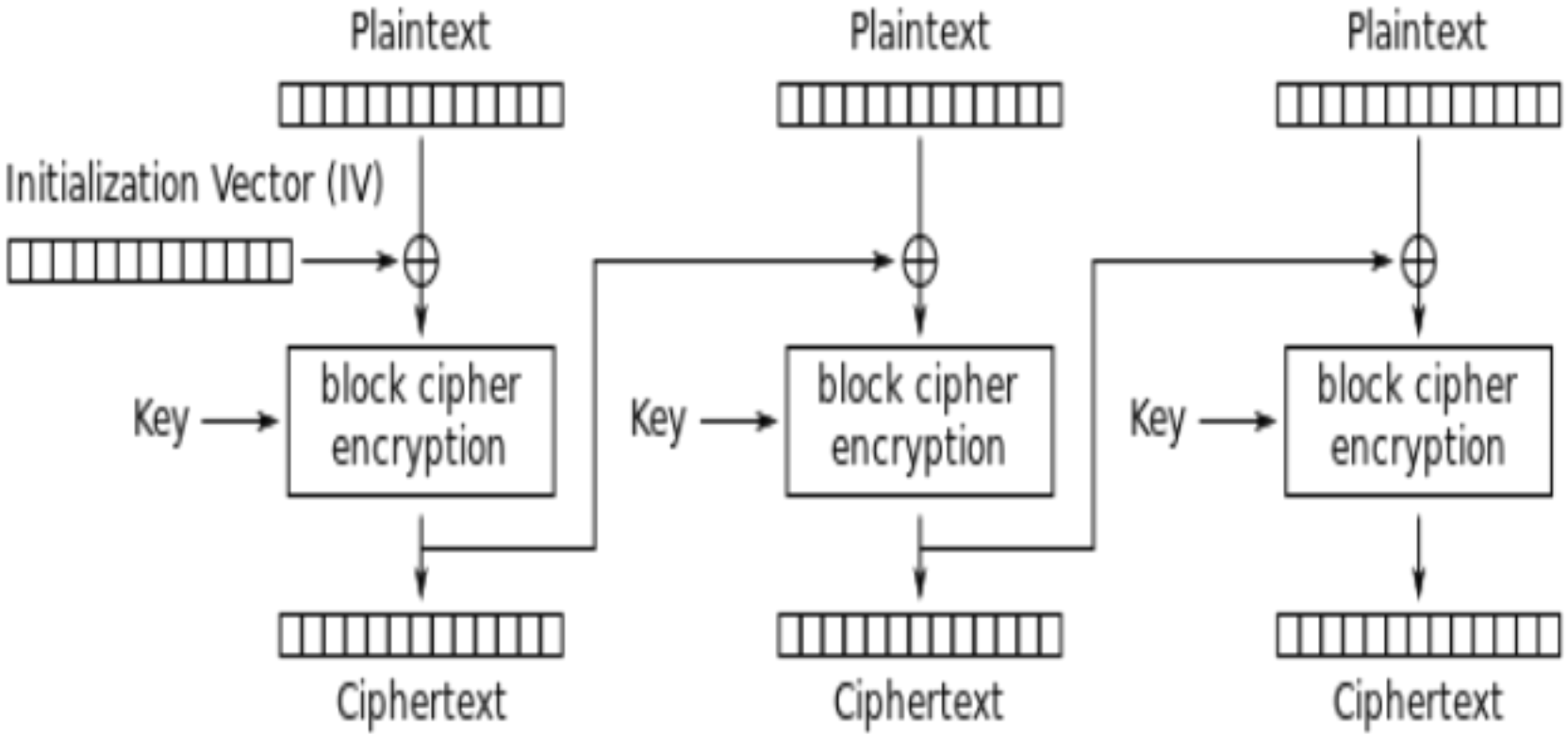
US National Institute of Standard Special Publications (NIST SP) 800-38
ISO/IEC 10116:2006
ANSI X9.52

Electronic Code Book



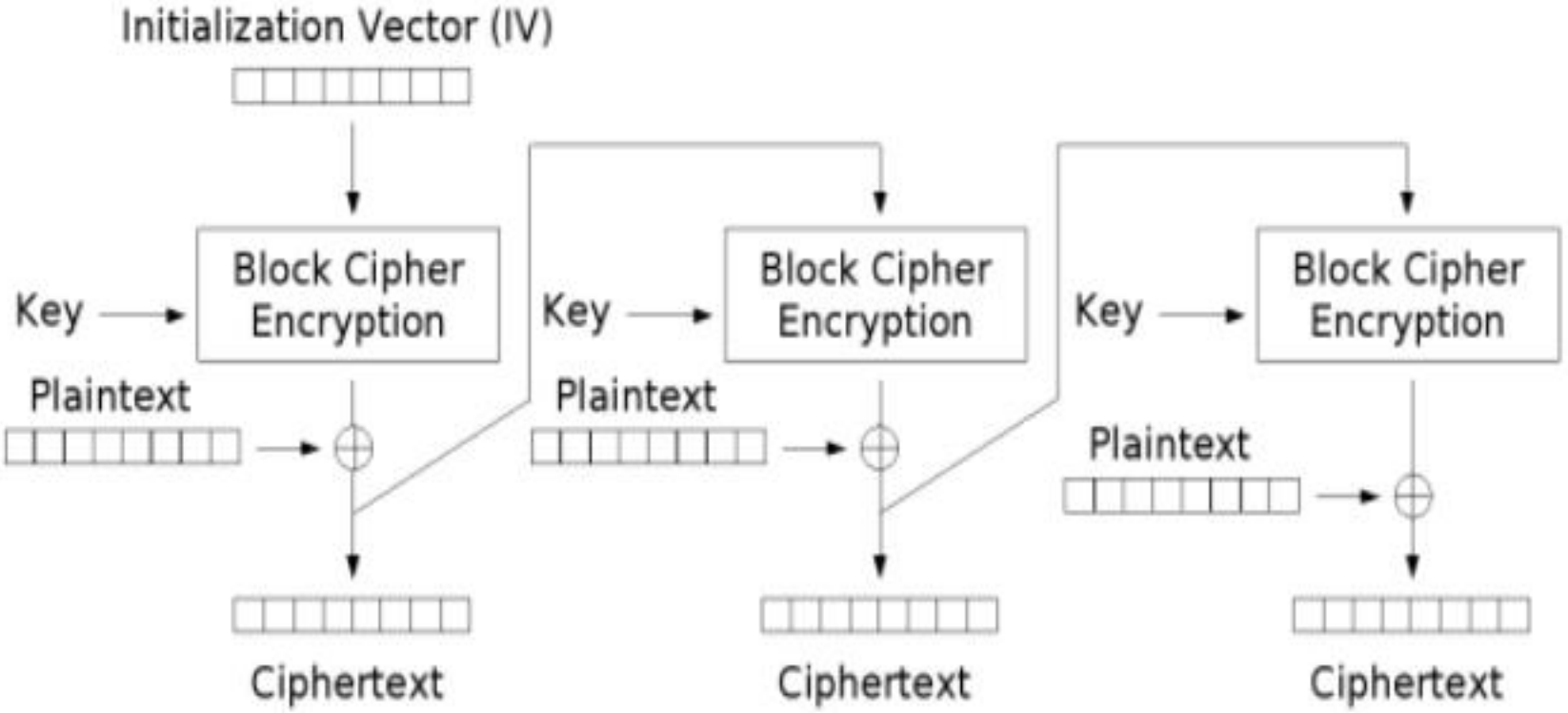
Electronic Codebook (ECB) mode encryption

Cipher Block Chaining



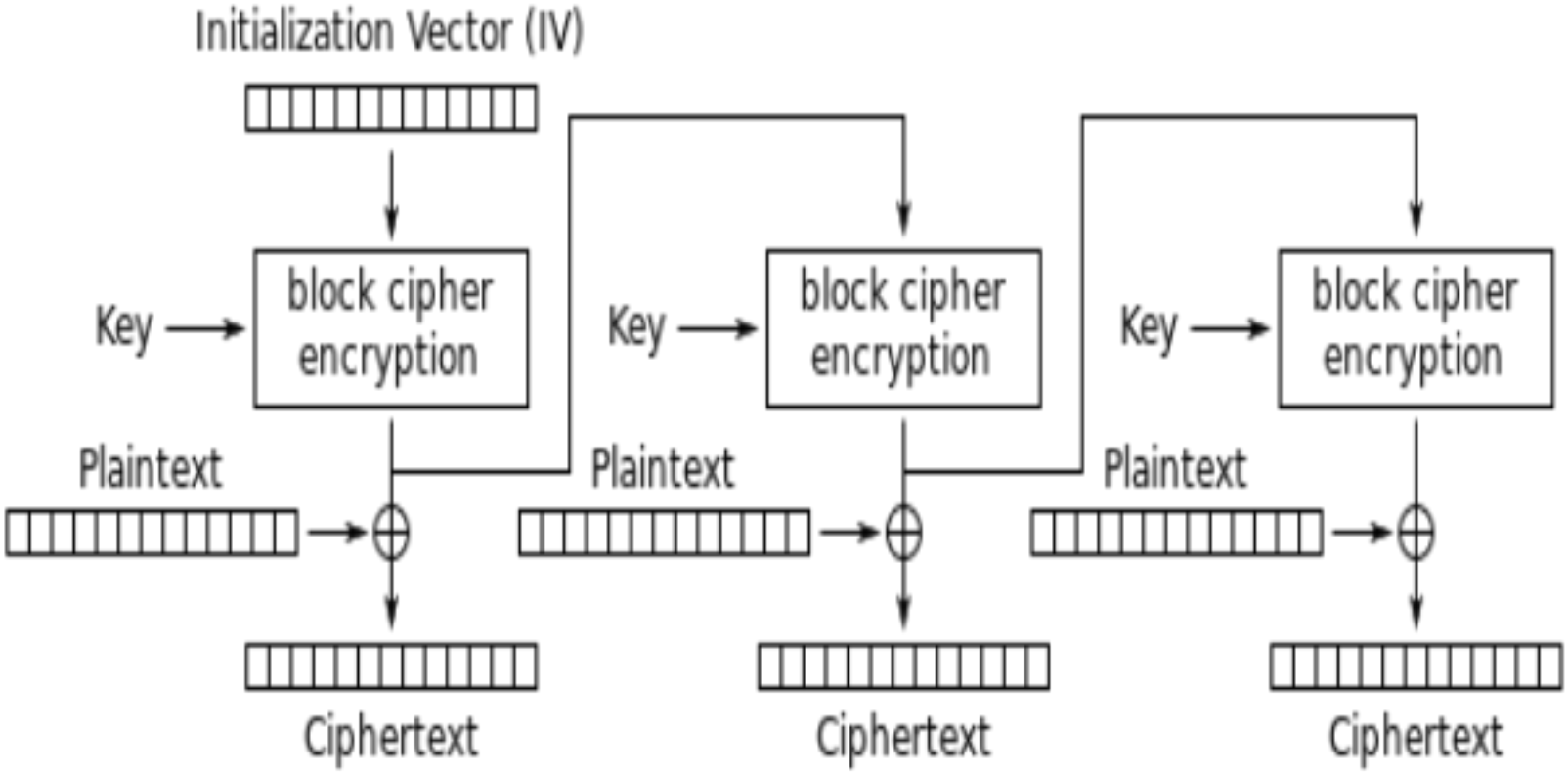
Cipher Block Chaining (CBC) mode encryption

Cipher Feedback Mode



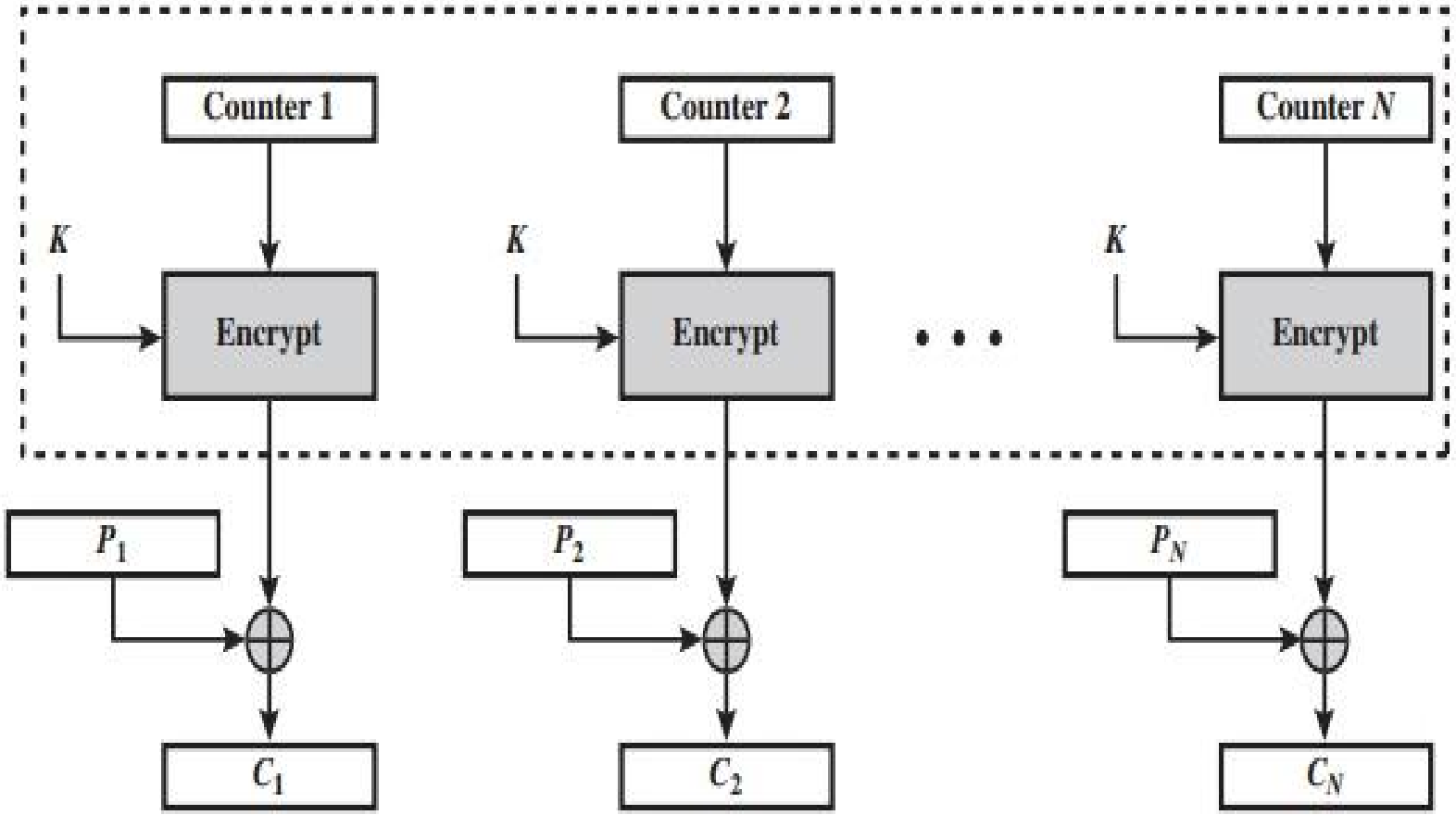
Cipher Feedback (CFB) mode encryption

Output Feedback Mode



Output Feedback (OFB) mode encryption

Counter Mode



(a) Encryption

Advanced Encryption Standard

Competition Requirements

- Private key symmetric block cipher
- 128-bit data, 128/192/256-bit keys
- Stronger & faster than Triple-DES
- Provide full specification & design details
- Both C & Java implementations

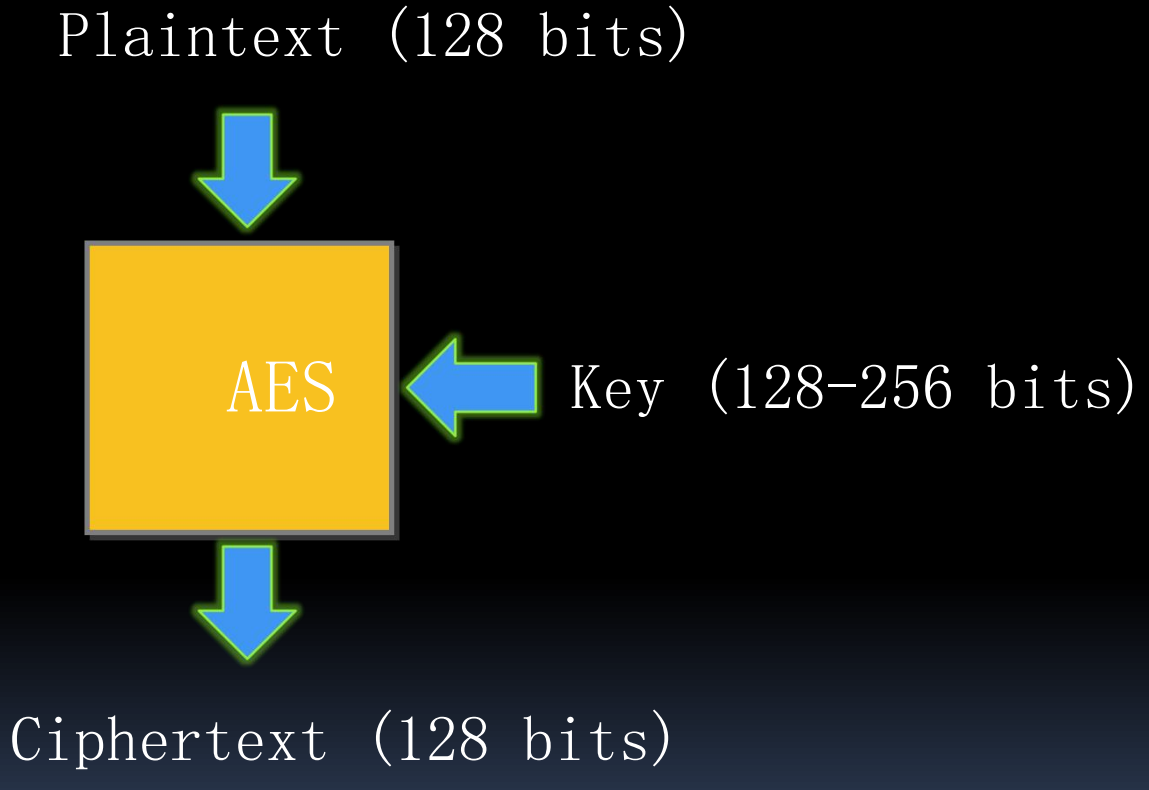
Ingredients of AES

- Block size – 128 bit plain text
- No.of Rounds(10,12,14)
- Key size-(16,24,32 bytes) or 4 words
- 1 word=32 bits
- No.of sub keys=44
- Each round using 4 words /128 bits/16 bytes

AES Evaluation Criteria

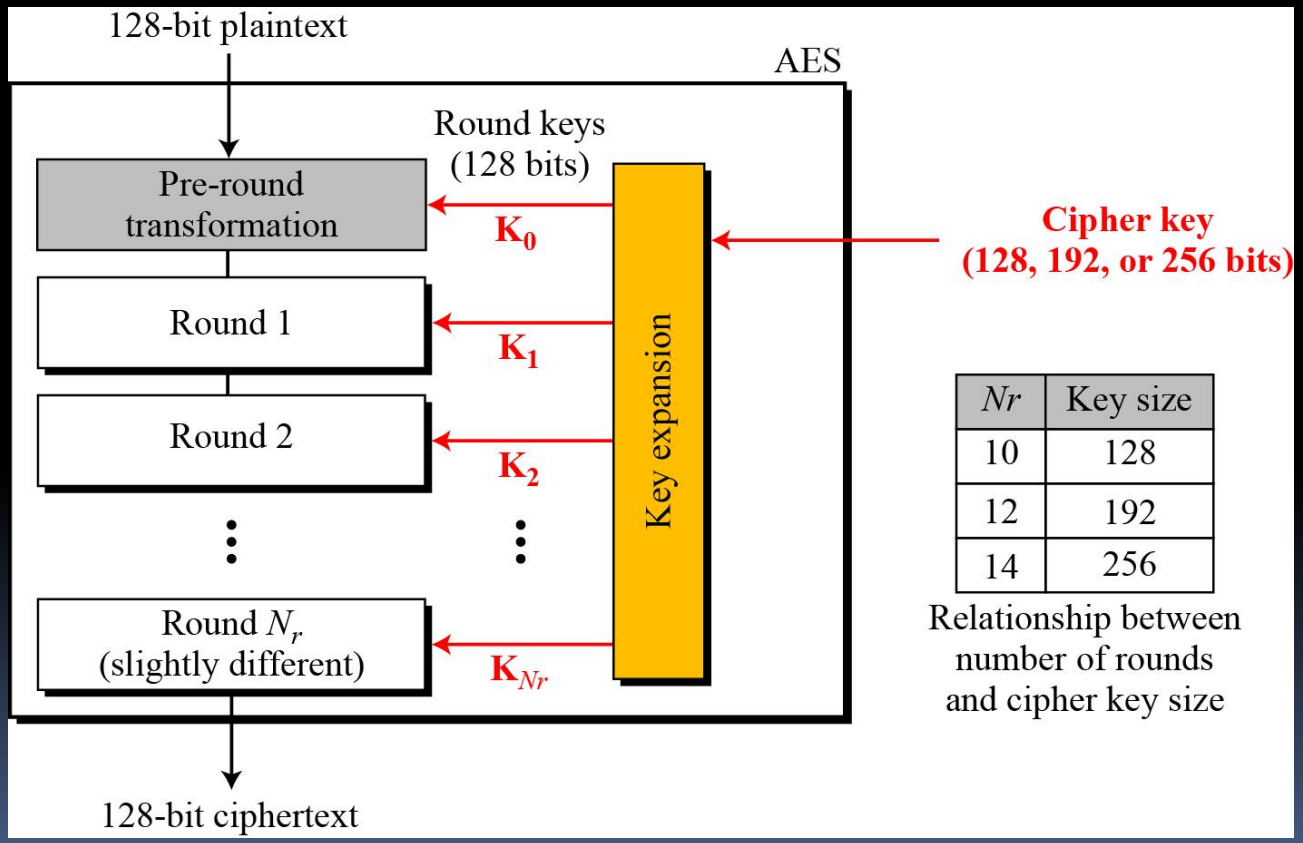
- Evaluation criteria:
  - security – effort for practical cryptanalysis
  - cost – in terms of computational efficiency
  - algorithm & implementation characteristics

AES Conceptual Scheme



Multiple rounds

- Rounds are (almost) identical
  - First and last round are a little different



Nr	Key size
10	128
12	192
14	256

Relationship between number of rounds and cipher key size

High Level Description

Key Expansion

- Round keys are derived from the cipher key using Rijndael's key schedule

Initial Round

- AddRoundKey : Each byte of the state is combined with the round key using bitwise xor

Rounds

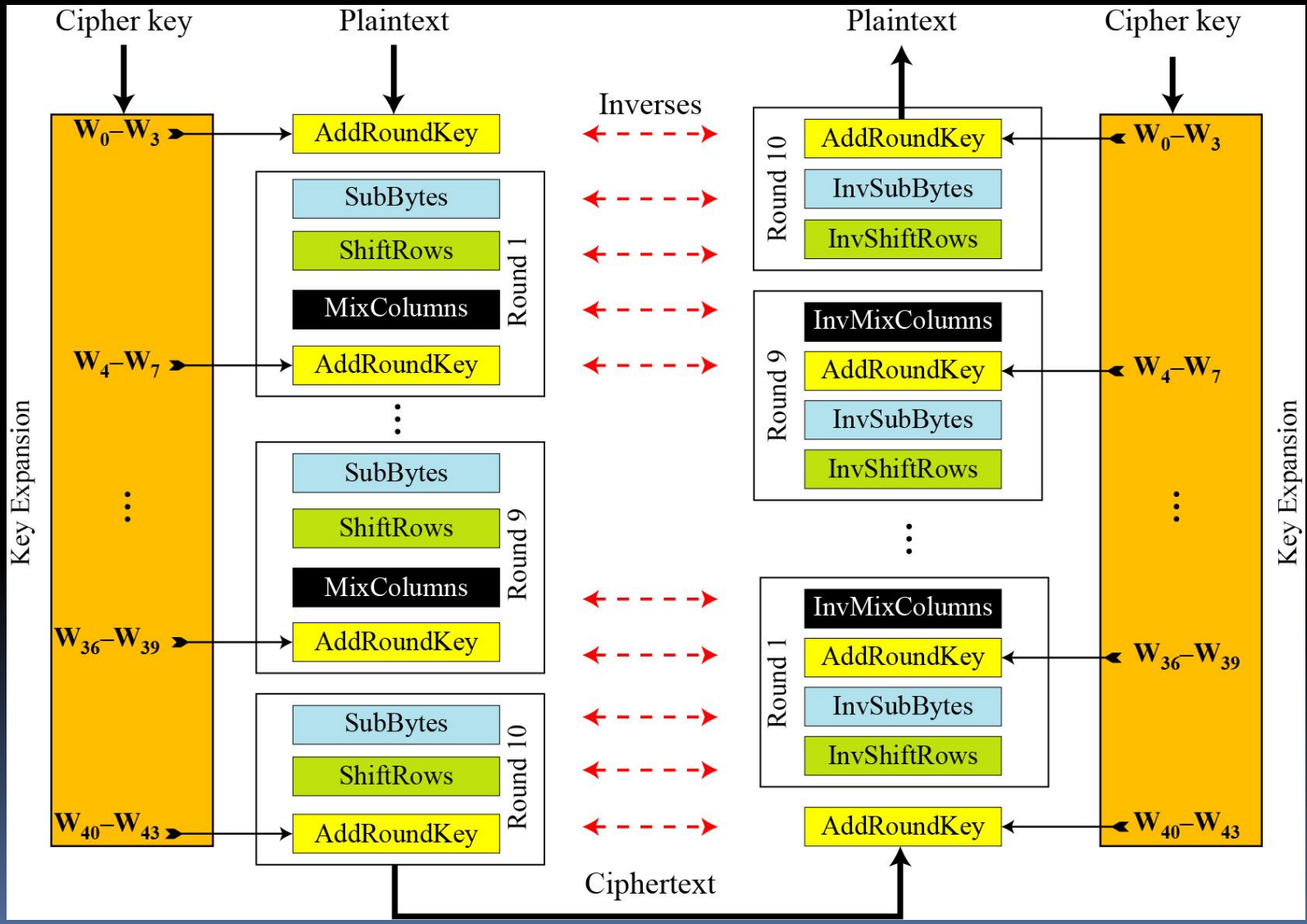
- SubBytes : non-linear substitution step
- ShiftRows : transposition step
- MixColumns : mixing operation of each column.
- AddRoundKey

Final Round

- SubBytes
- ShiftRows
- AddRoundKey

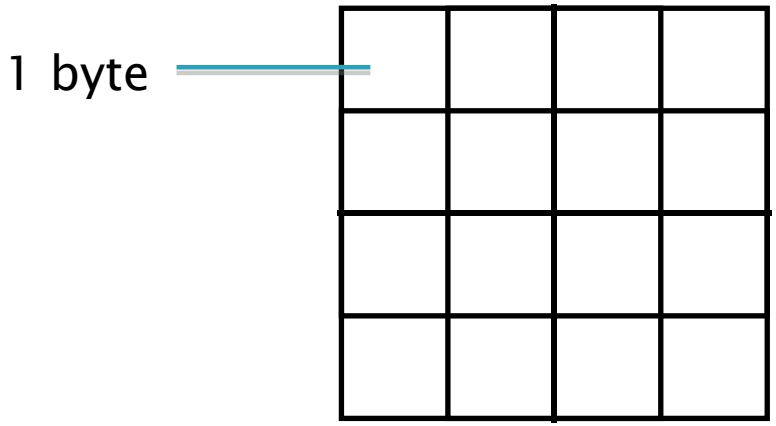
No MixColumns

Overall Structure



128-bit values

- Data block viewed as 4-by-4 table of bytes
- Represented as 4 by 4 matrix of 8-bit bytes.
- Key is expanded to array of 32 bits words



Working principles of AES

- In Rijndael the plain text blocks and keys can be arranged in variable sizes(16,24,32 bytes)
- It consist of 10,12,14 rounds
- Each round consist of four steps

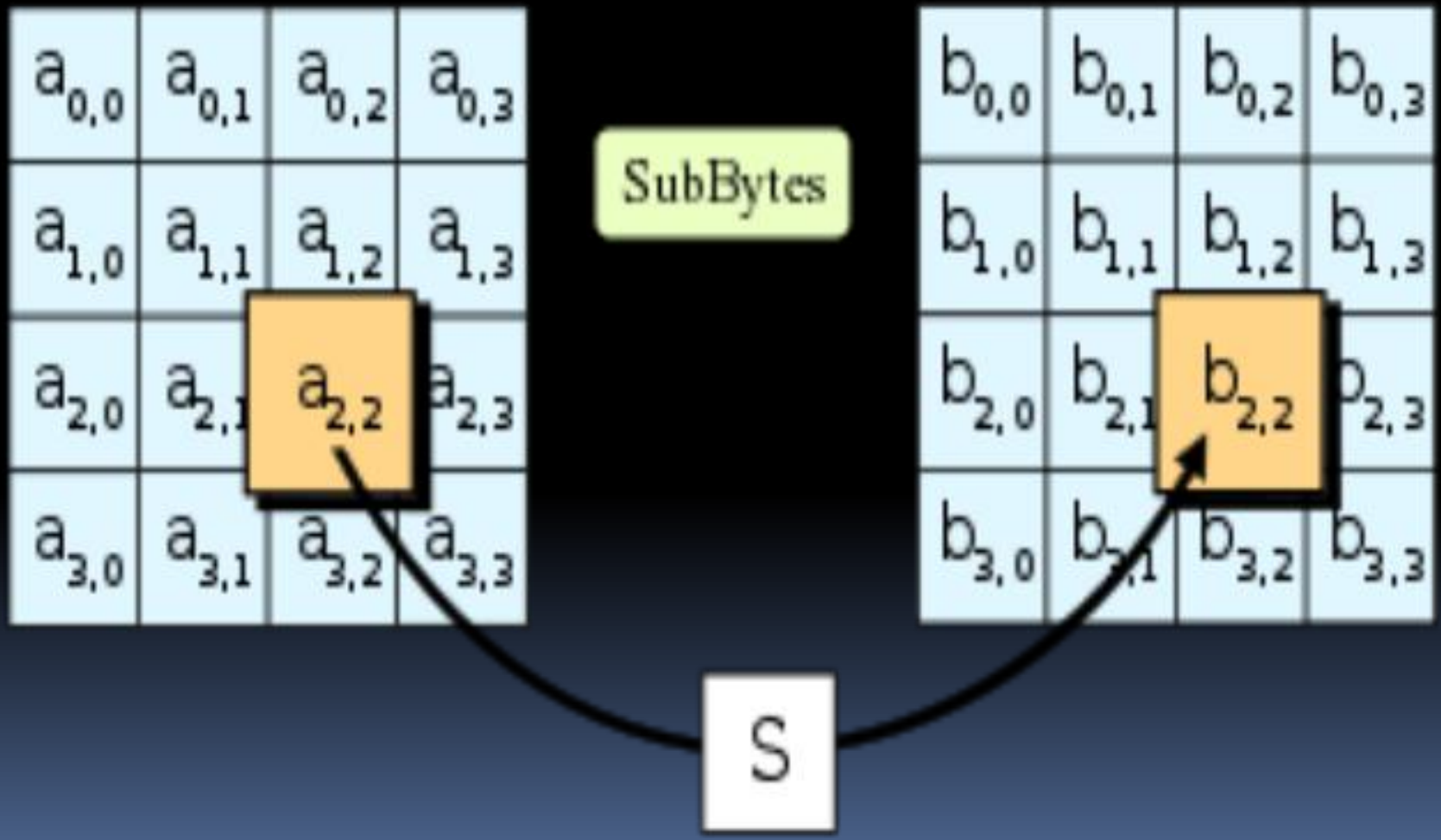
*step-1 Byte Substitution

*step-2 shift rows

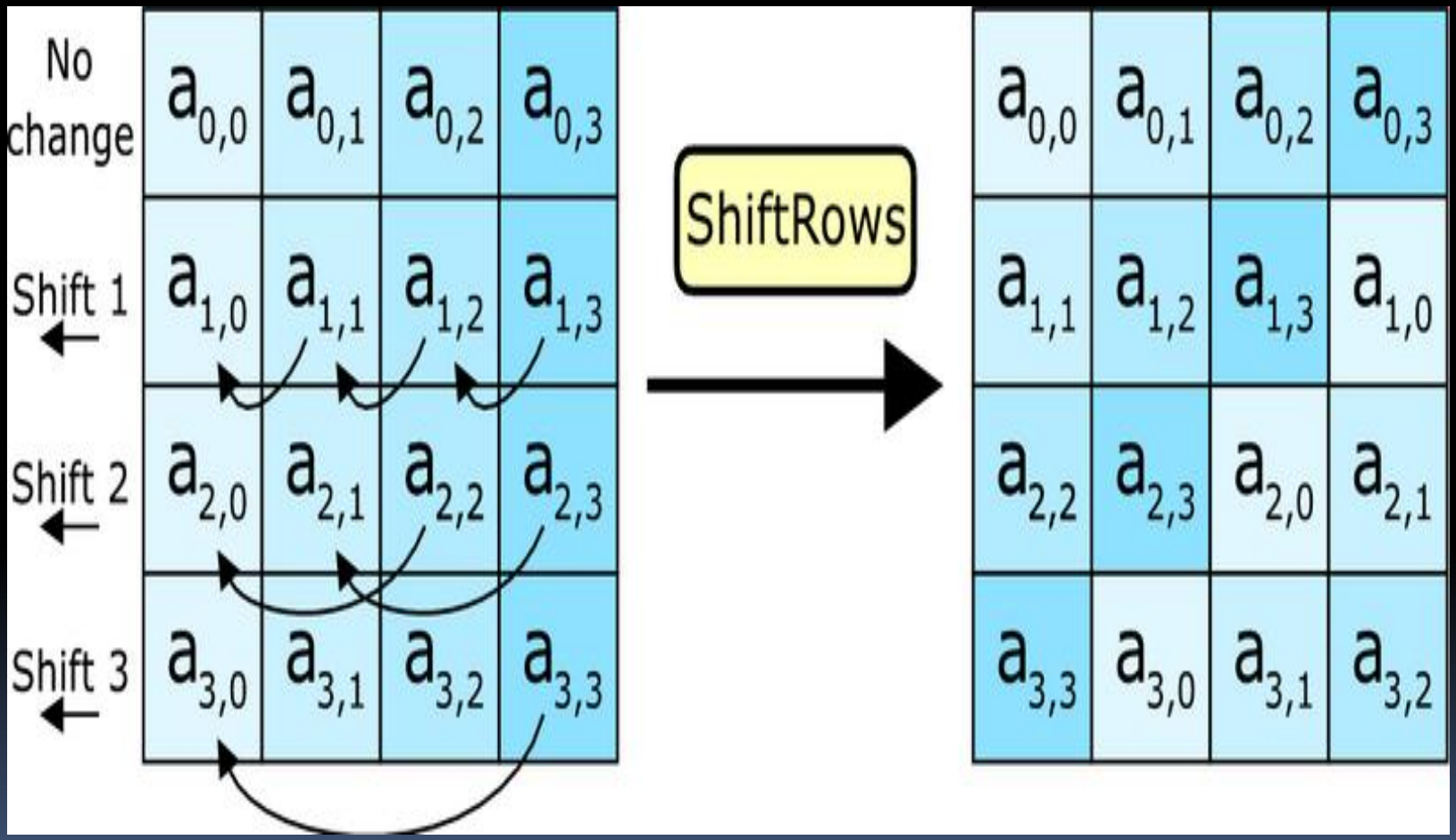
*step-3 Mix columns

*step-4 Add round key

Step-1 Byte Substitution

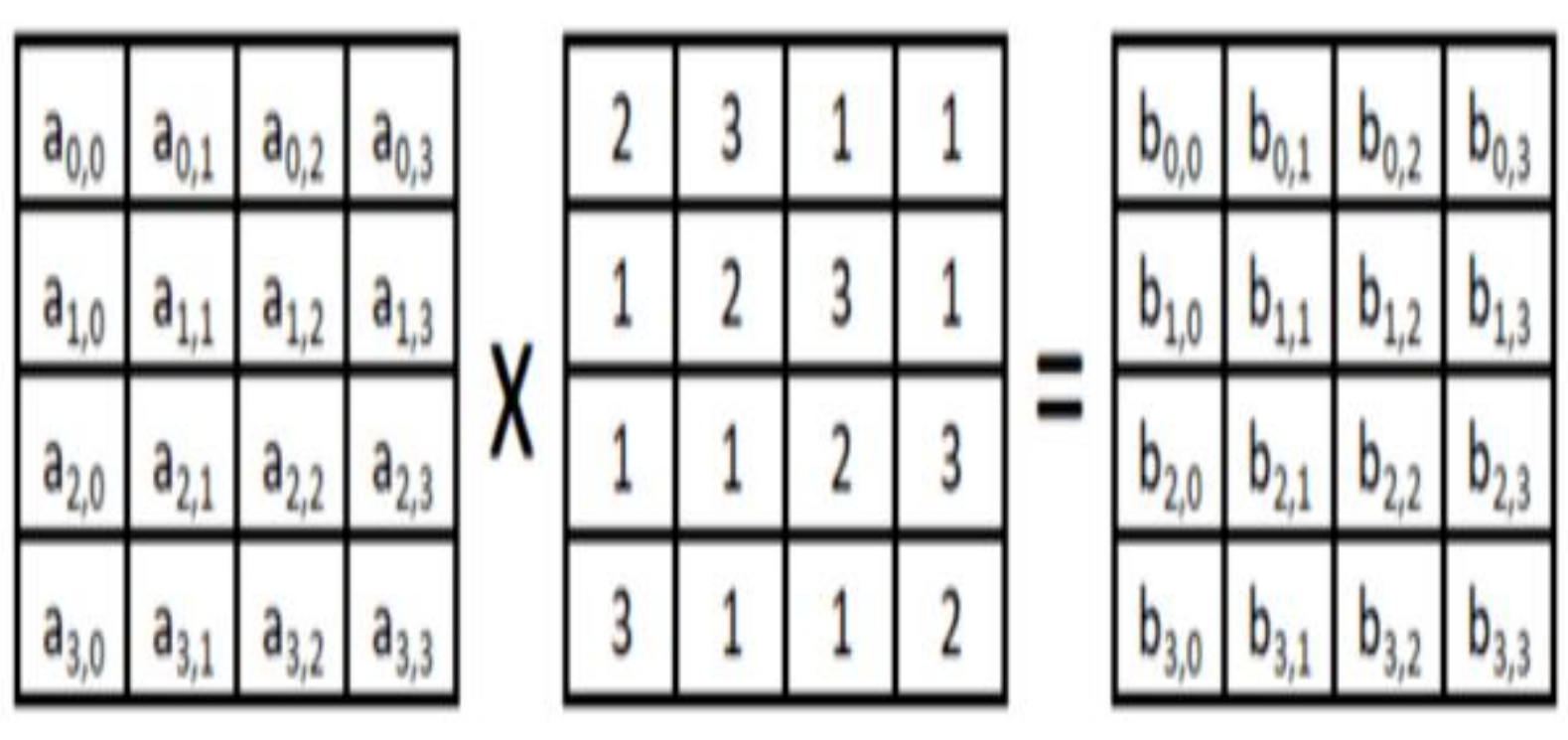


Step-2 Shift Rows

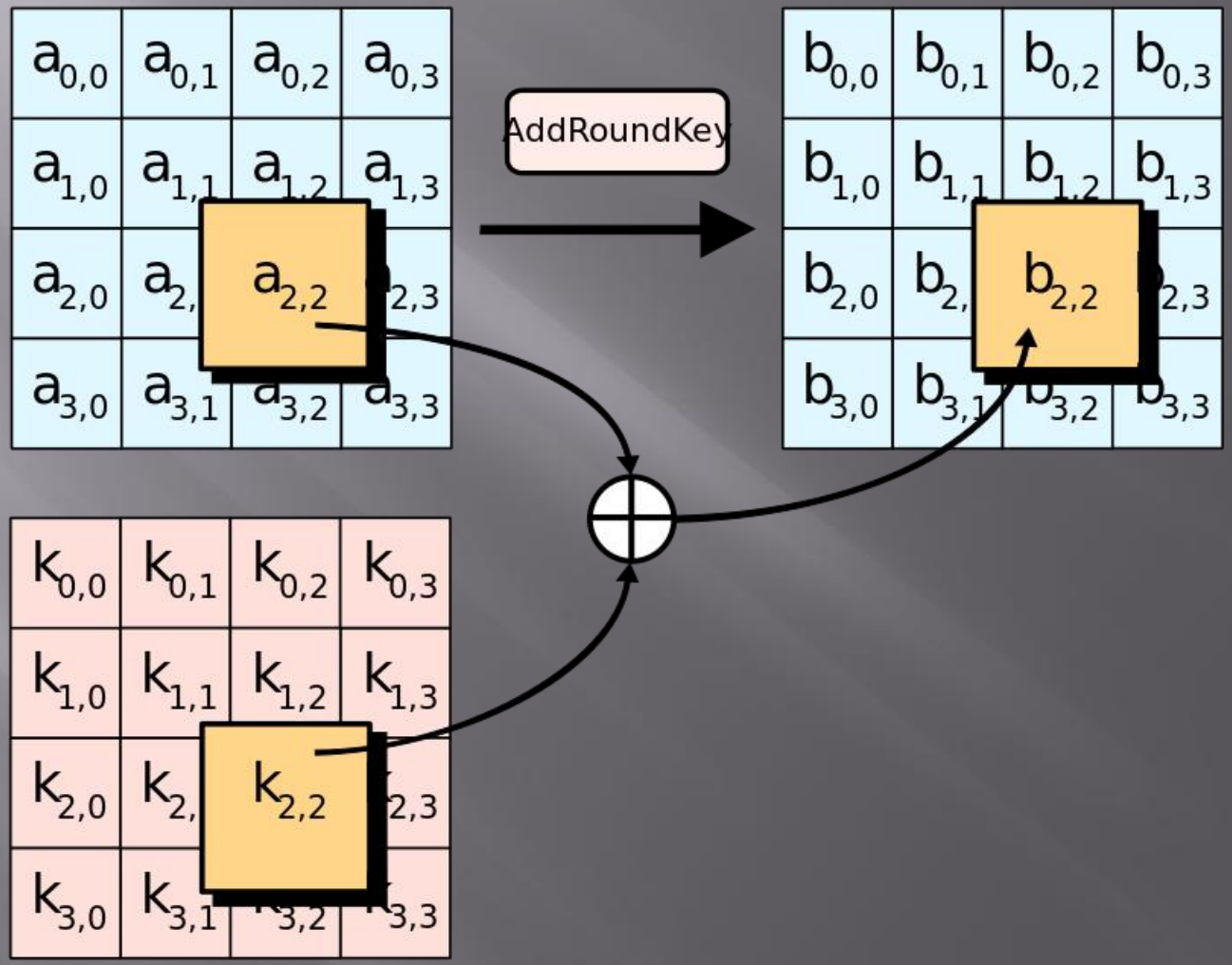


Step-3 Mix column

- The bytes of every column are mixed in a linear fashion.



Step-4 Add round key



RC4

- **RC4** was **designed** by Ron Rivest of RSA Security in 1987. While it is officially termed "Rivest Cipher 4", the RC acronym is alternatively understood to stand for "Ron's Code"
- **RC4** is a stream cipher and variable length key algorithm. This algorithm encrypts one byte at a time (or larger units on a time).

- A key input is pseudorandom bit generator that produces a stream 8-bit number that is unpredictable without knowledge of input key, The output of the generator is called key-stream, is combined one byte at a time with the plaintext stream cipher using X-OR operation.

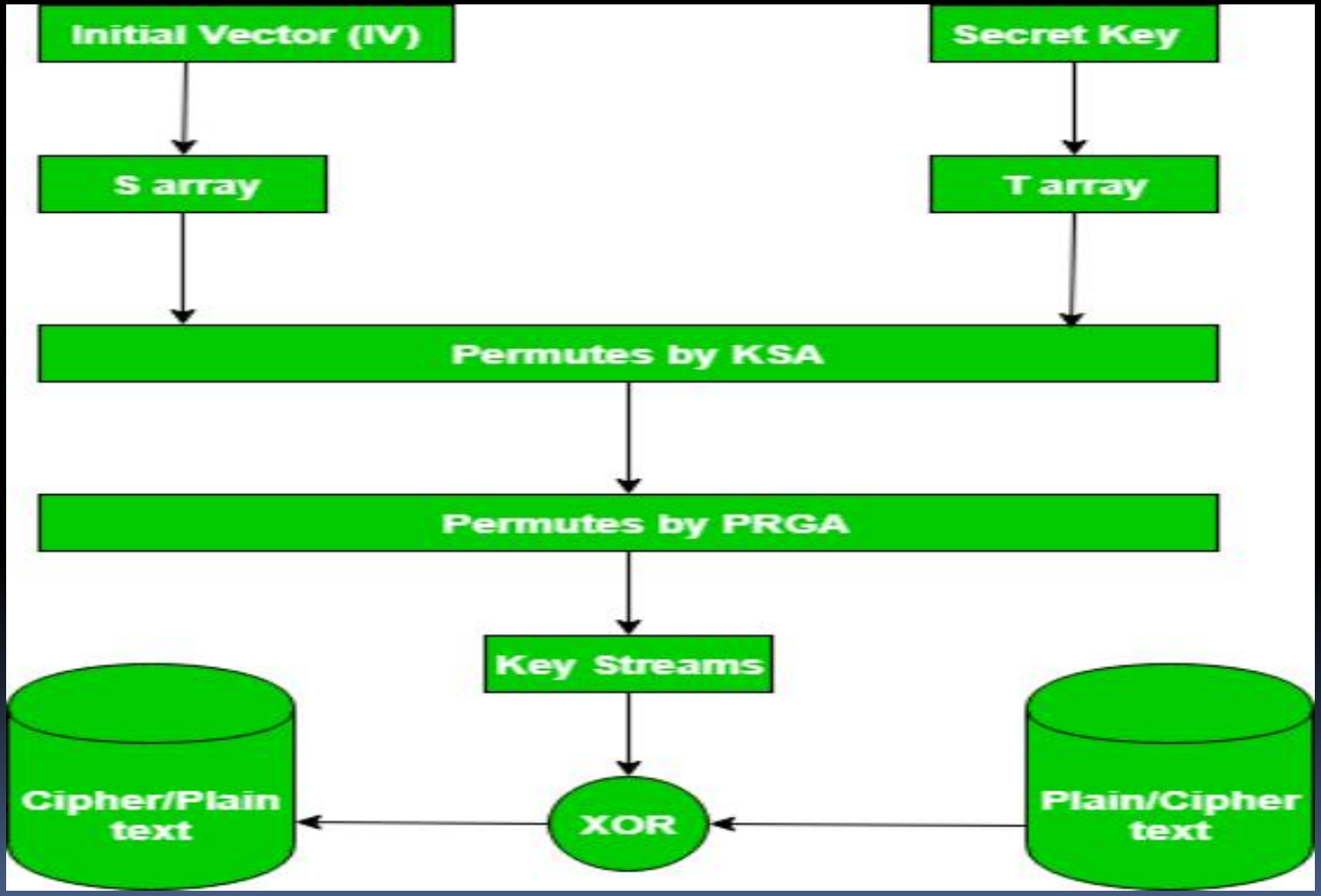
RC4 ENCRYPTION AND DECRYPTION

- RC4 Encryption 10011000 ? 01010000 = 11001000
- RC4 Decryption 11001000 ? 01010000 = 10011000

Key-Generation Algorithm

- A variable-length key from 1 to 256 byte is used to initialize a 256-byte state vector S, with elements S[0] to S[255].
- For encryption and decryption, a byte k is generated from S by selecting one of the 255 entries in a systematic fashion, then the entries in S are permuted again.
- **Key-Scheduling Algorithm**
- **Pseudo random generation algorithm (Stream Generation)**
- **Encrypt using X-Or()**

OVER ALL PROCESS



References

- https://youtu.be/vZ7YQ67Cbtc
- https://youtu.be/v1JwSaAqVNY
- https://youtu.be/SaZGjQBItBc
- https://youtu.be/QcKHfMgcnbw


Thank
you