Subject Name: Cryptography and System Security

Unit No: 02 Unit Name: Symmetric and Asymmetric key Cryptography and key Management

Faculty Name :
Dr. Dhananjay Dakhane
Dr. Sangeeta Chaudhari
Dr. Prajkta Dere

Index

Unit No: 1 Unit Name: Introduction

Lecture No: 17
The RSA algorithm

Confusion and Diffusion

- Confusion: No clue regarding the relationship between ciphertext and the key
- Diffusion: Hides relationship between plaintext and corresponding ciphertext
- Strong substitution function enhances confusion while transposition is used to enhance diffusion

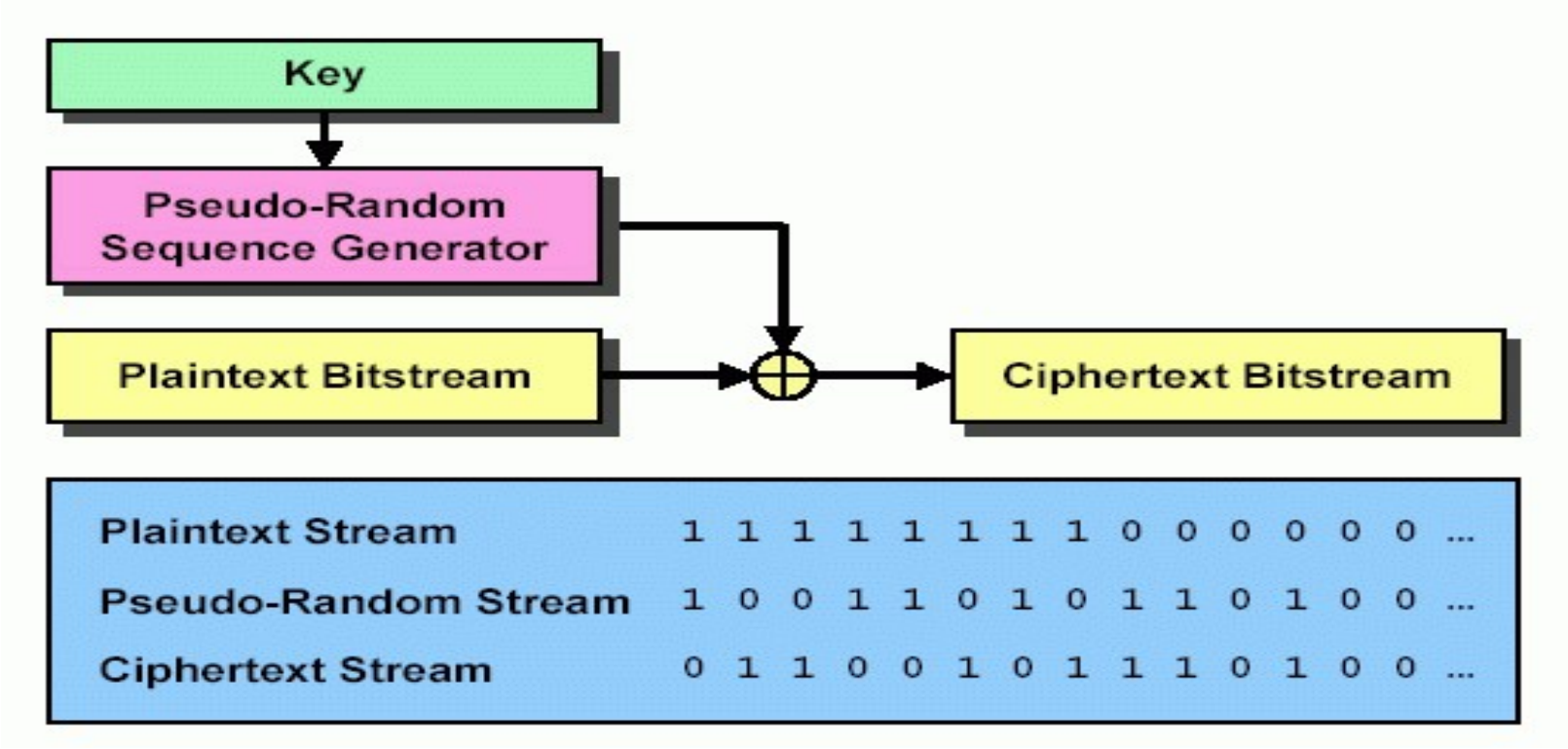
Product Ciphers

- Ciphers using substitutions or transpositions are not secure because of language characteristics

- Idea : using several ciphers in succession to make strong cipher
  - two substitutions make a more complex substitution
  - two transpositions make more complex transposition
  - but a substitution followed by a transposition makes a new much harder cipher
- Used in modern ciphers

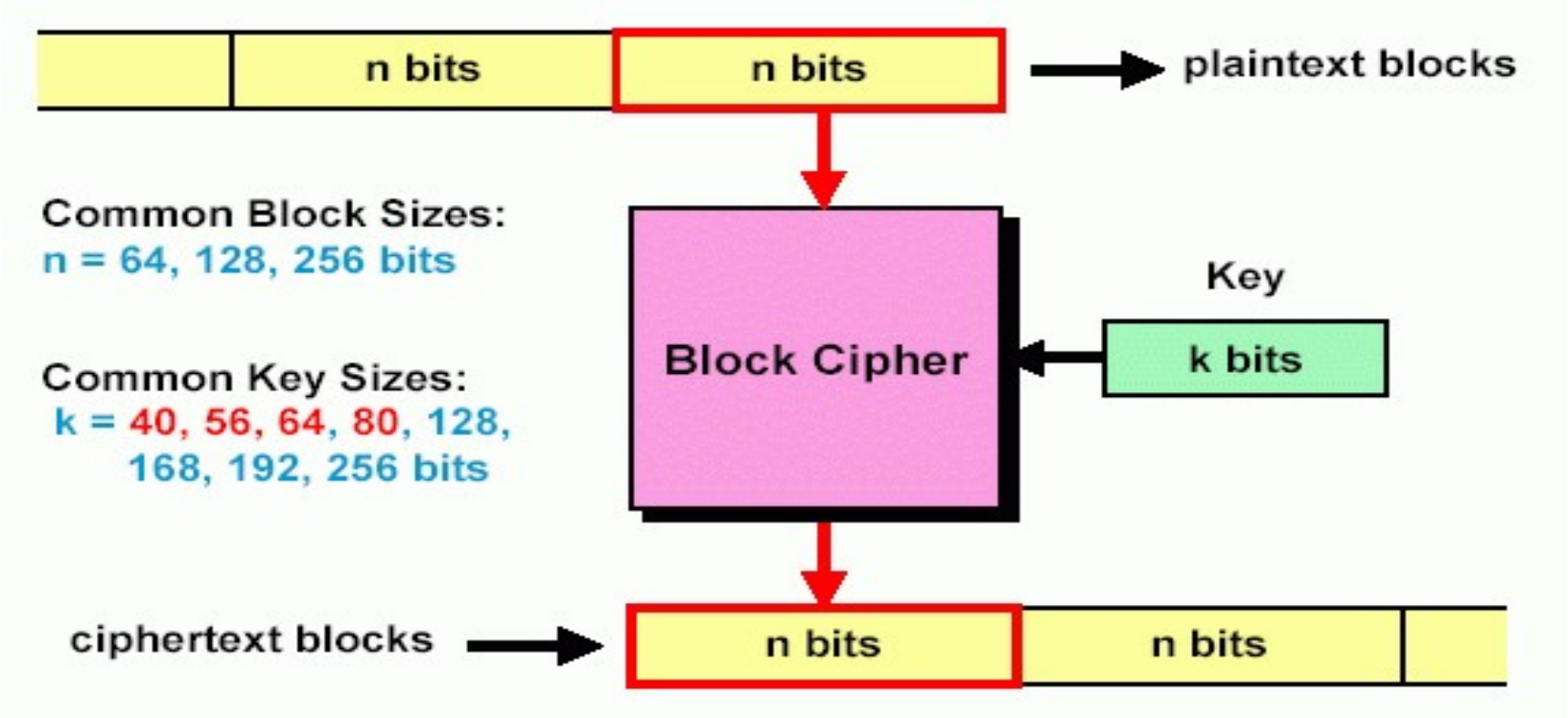
Stream and Block Ciphers

Stream Ciphers : Encryption/Decryption- on one letter/symbol at a time



Block Ciphers

- Divide input bit stream into n-bit sections,
- Encrypt only that section, no dependency/history between sections



Stream Vs Block Ciphers

Stream

- Low Diffusion all information of one symbol contained in one symbol of cipher text
- Low error propagation error will affect only one character

Block

- High Diffusion one cipher text block may depend on several plaintext letters
- Error propagation will affect transformation of all other characters in the same block

Stream Vs Block Ciphers

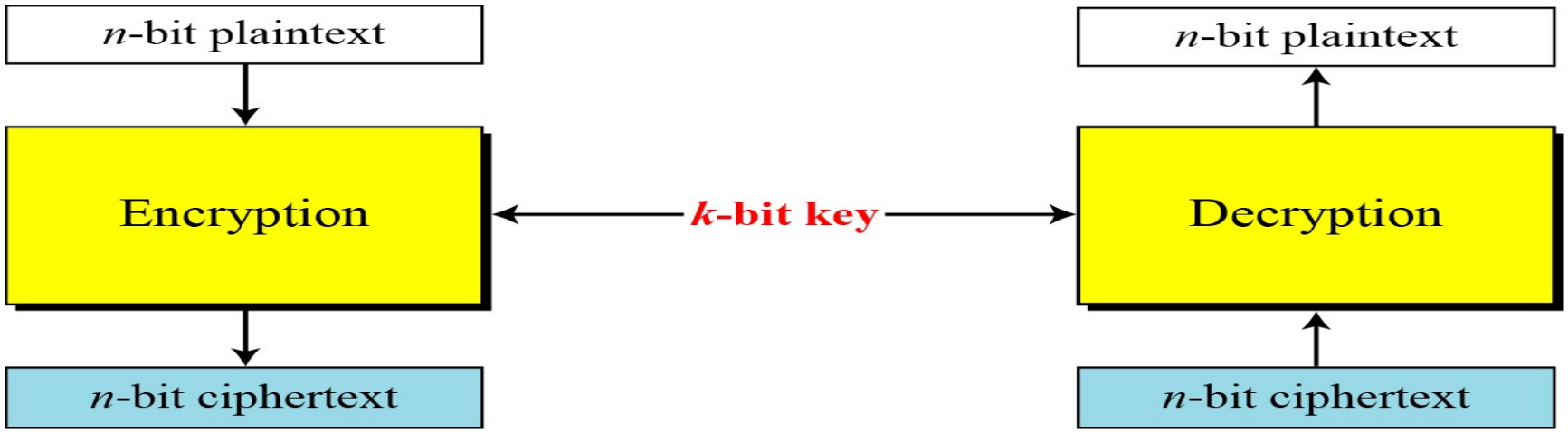
Stream

- Speed of transformation – time depends only on encryption method
- Susceptibility to insertions and modifications

Block

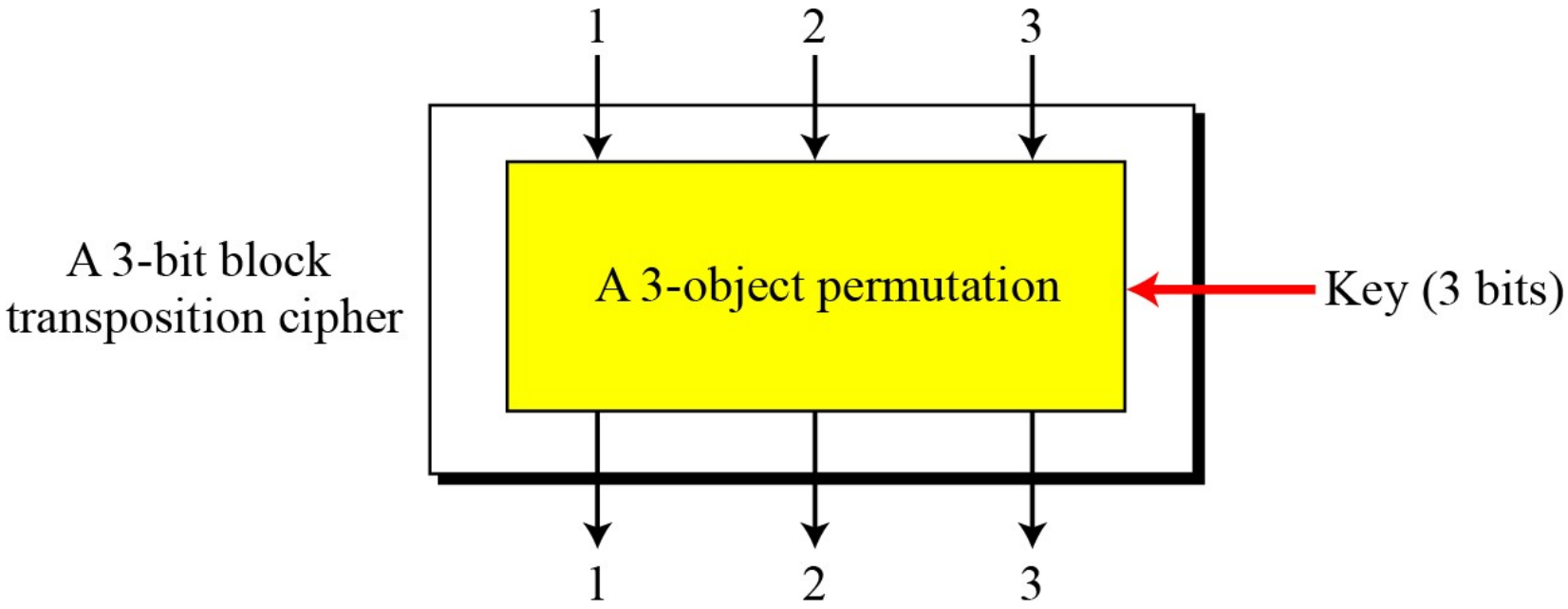
- Slowness of encryption must wait for entire block
- Immunity to insertion of symbols not possible to insert a single symbol into the block

Modern Block Cipher



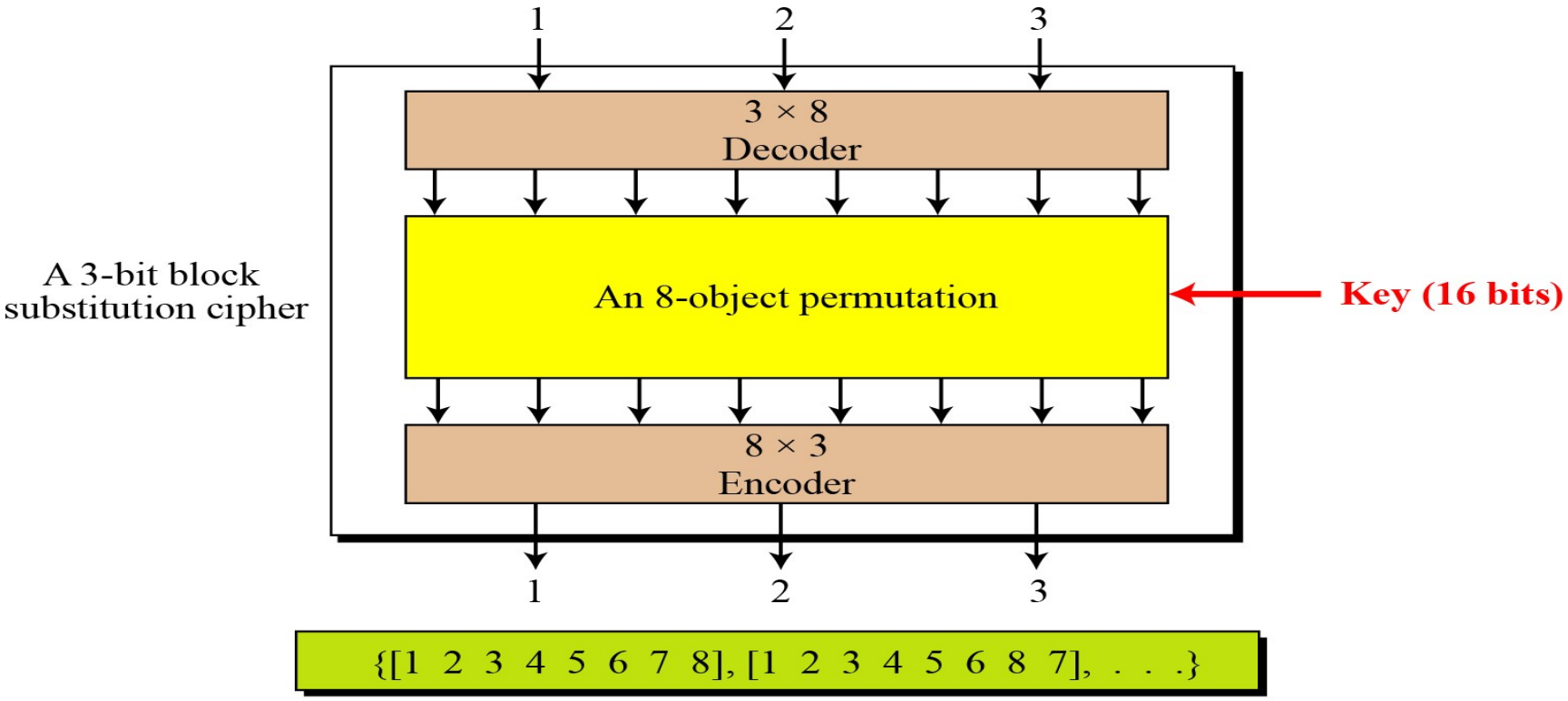
It can be Transposition or Substitution Cipher

Modern Block Cipher



A transposition block cipher modeled as a permutation

Modern Block Cipher



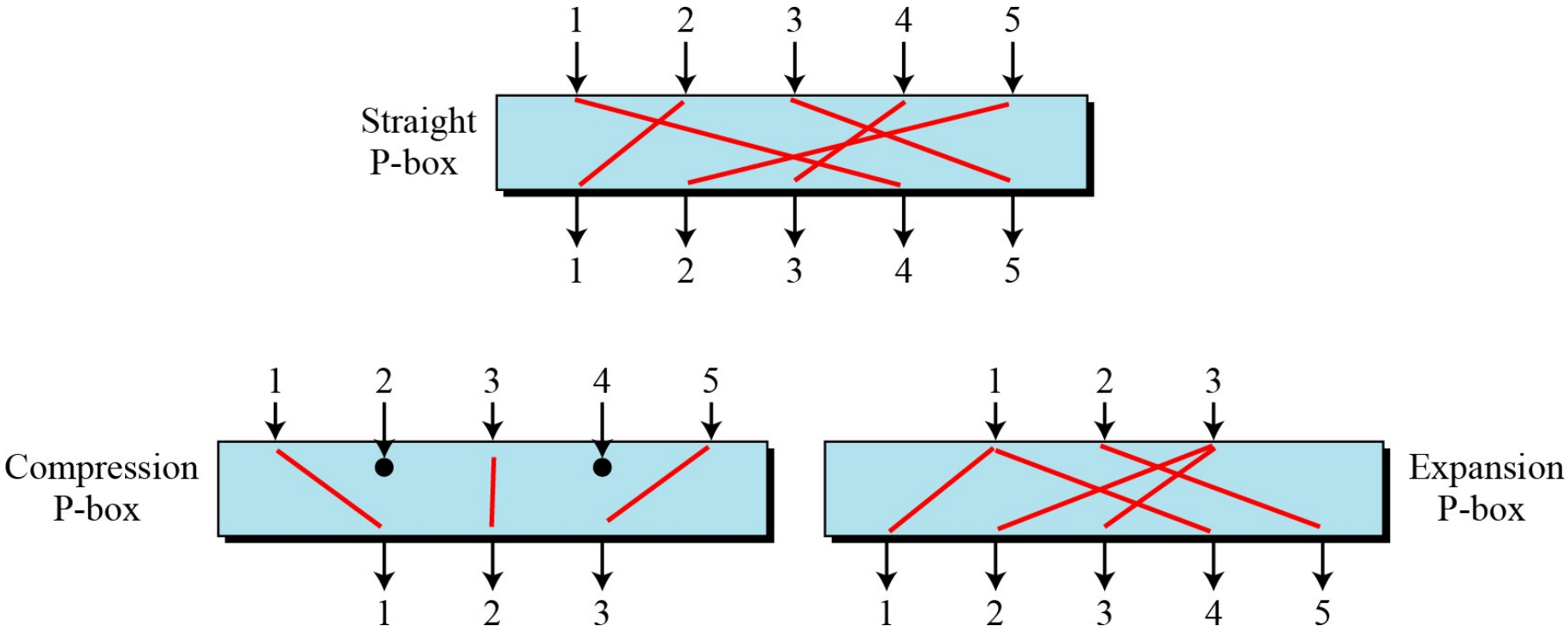
A substitution block cipher model as a permutation

Components of Modern Block Cipher

Modern block ciphers are keyed ciphers in which the key allows only partial mappings from the possible inputs to the possible outputs

- P- Box
- S-Box
- EX-OR
- Circular shift
- Swap
- Split and combine

Permutation Box(P-Box)



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

Permutation Box(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

permutation table for a straight 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

32 × 24 permutation table for Compression P-box

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

12 × 16 permutation table for Expansion P-Box

D Y PATIL
RAMRAO ADIK INSTITUTE OF TECHNOLOGY
NAVI MUMBAI

Substitution Box(S-Box)

- An S-box is a substitution cipher
- An S-box is an m × n substitution unit, where m and n are not necessarily the same
- In an S-box with three inputs and two outputs, we have

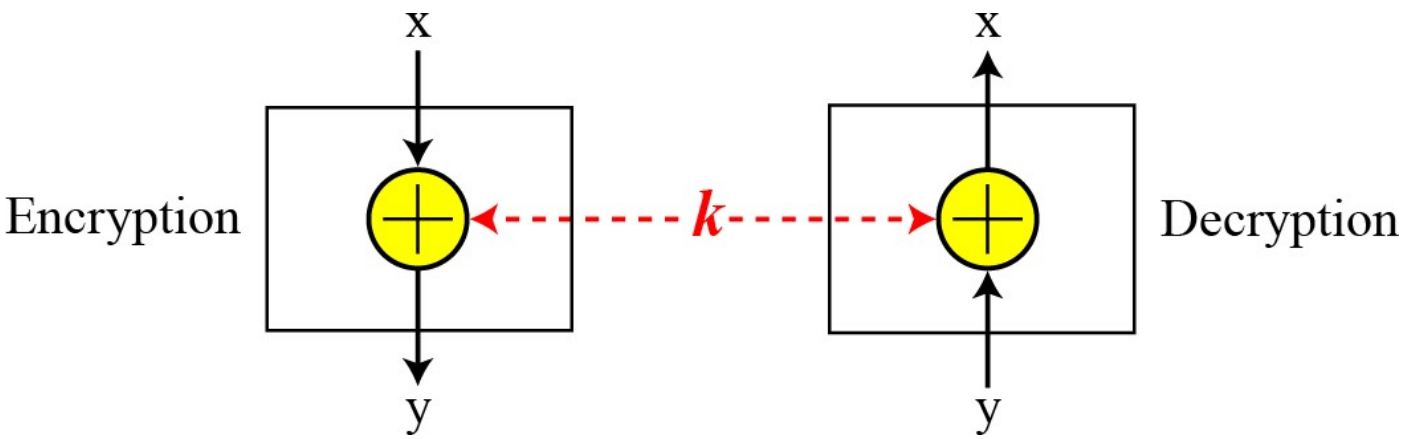
$$y_1 = x_1 \oplus x_2 \oplus x_3 \qquad y_2 = x_1$$

or

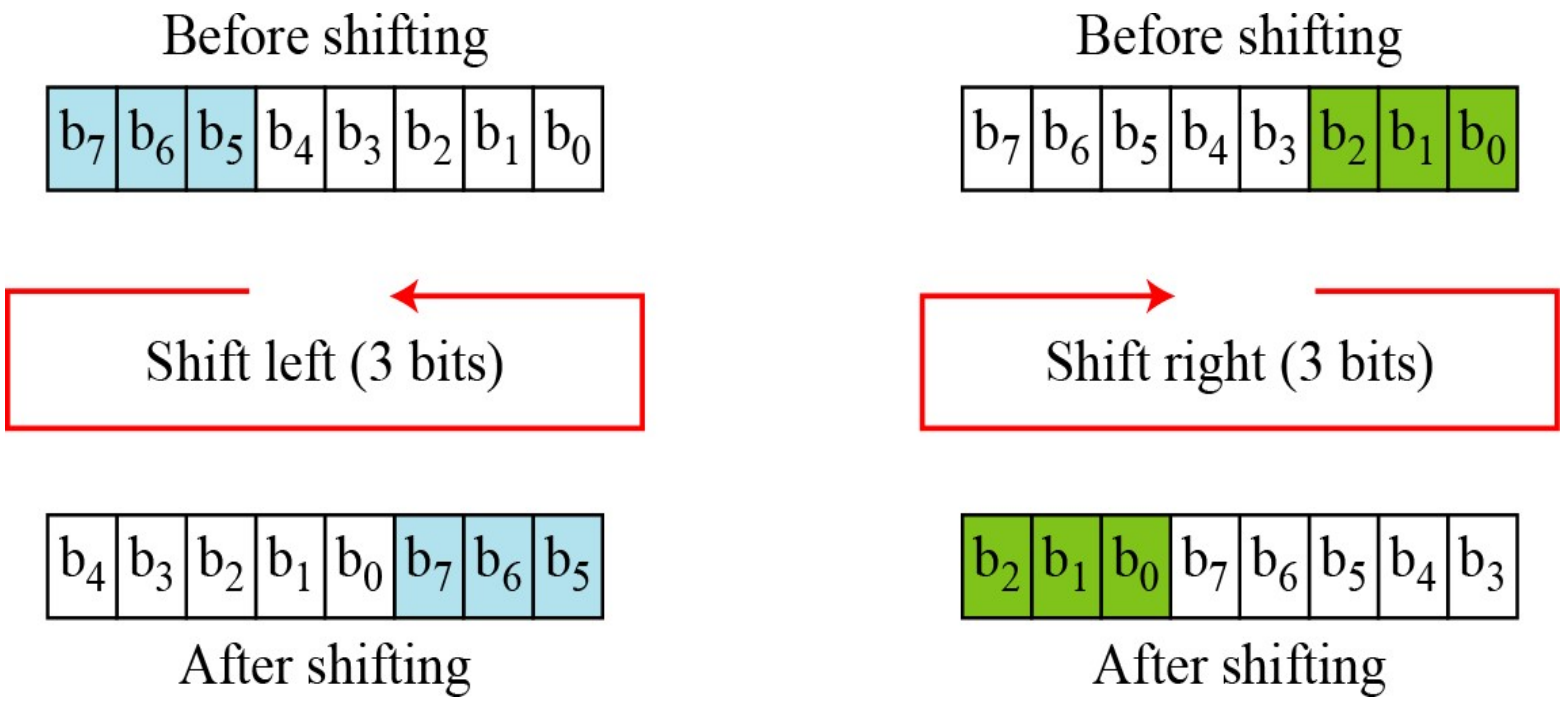
$$y_1 = (x_1)^3 + x_2 \qquad y_2 = (x_1)^2 + x_1x_2 + x_3$$

Exclusive OR (EX-OR)

An EX-OR is an invertible operation

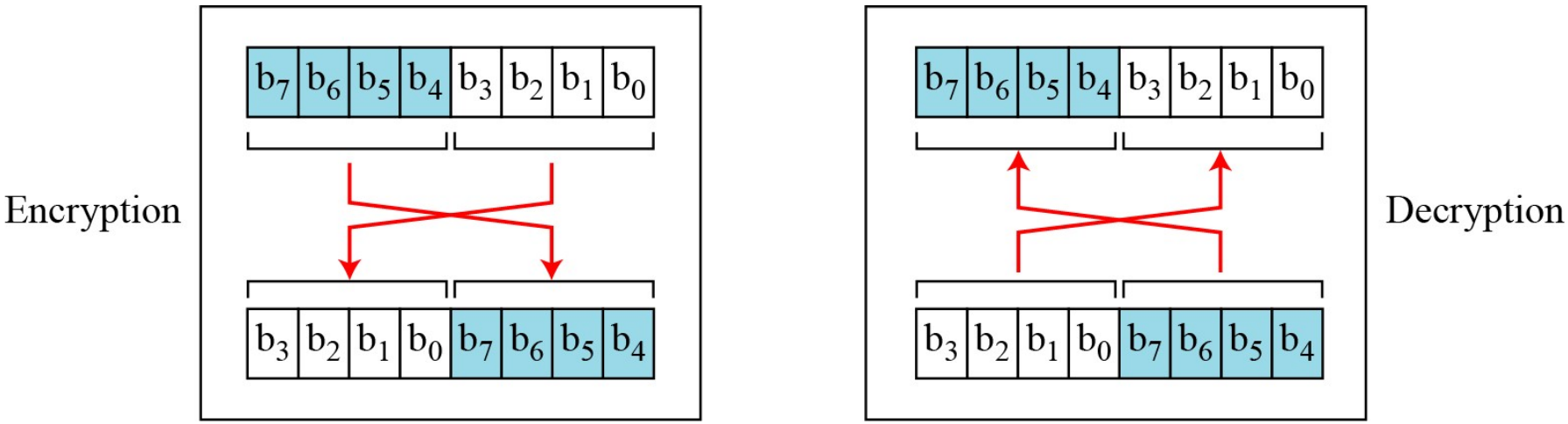


Circular Shift Operation

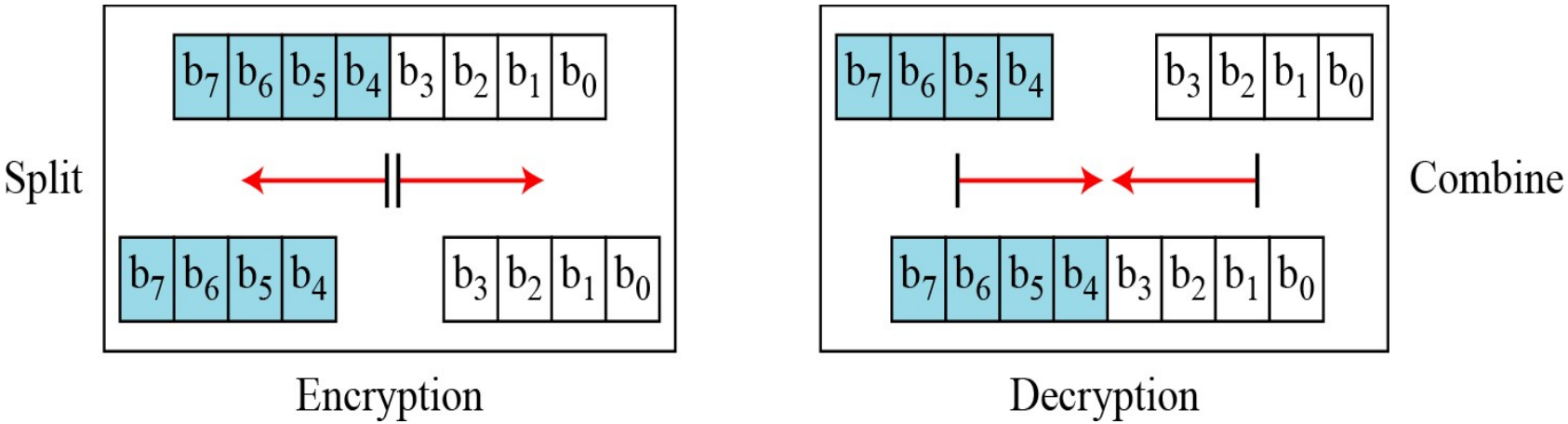


Swap Operation

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



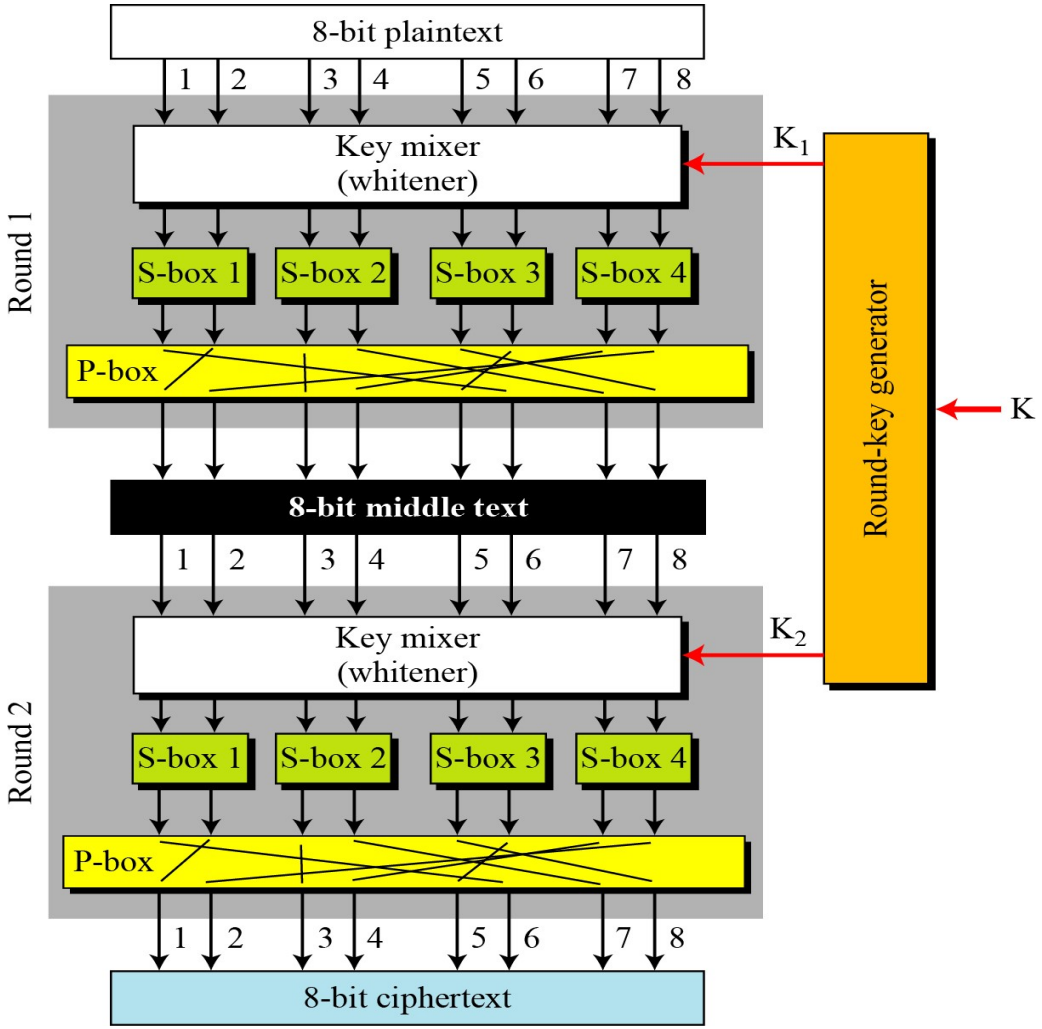
Split and Combine Operation



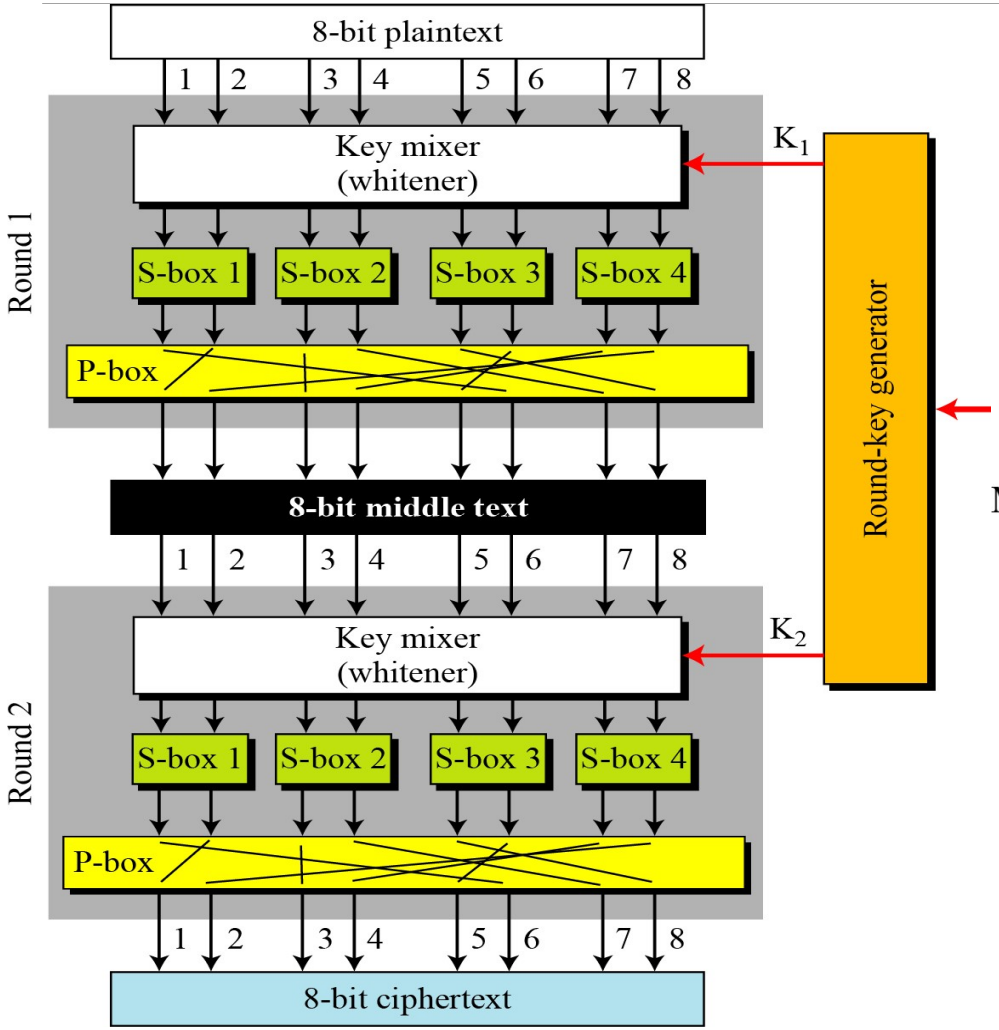
Product Cipher

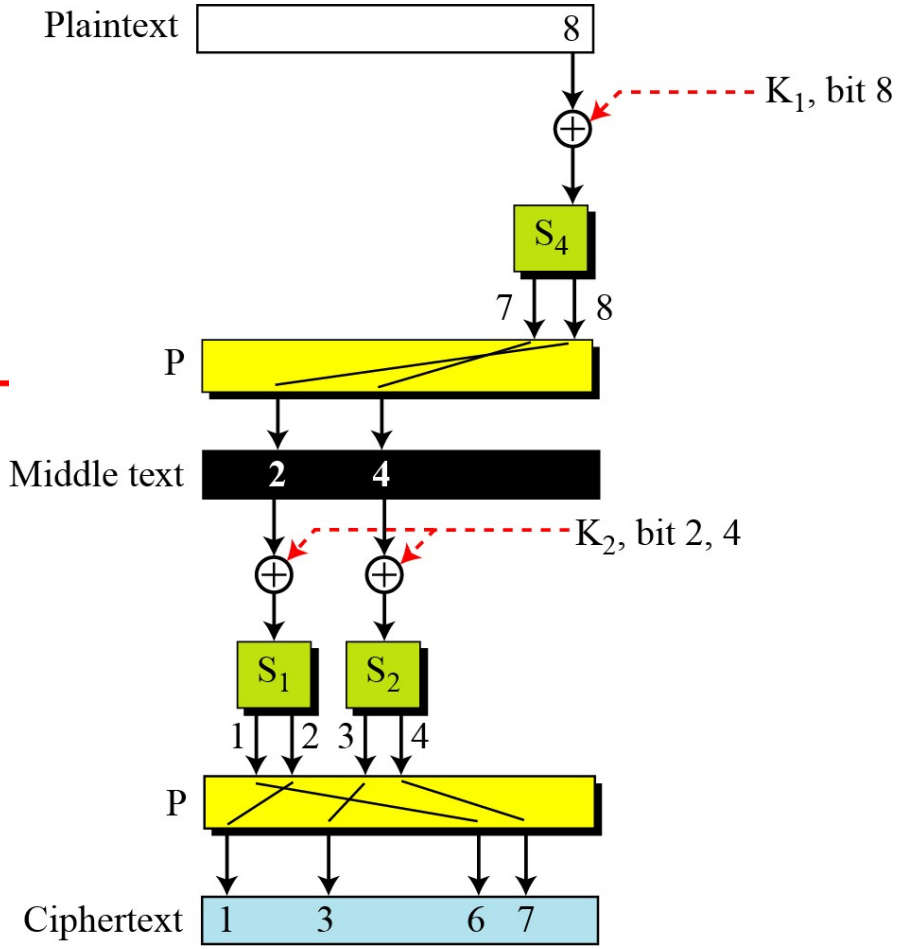
- A product cipher is a complex cipher combining substitution, permutation and other components
- Diffusion and confusion can be achieved using iterated product ciphers where each iteration is a combination of S-boxes, P-boxes, and other components
- Two types of product cipher:
  - Feistel ciphers
  - Non-Feistel ciphers

Product Cipher



Product Cipher(Confusion and Diffusion)



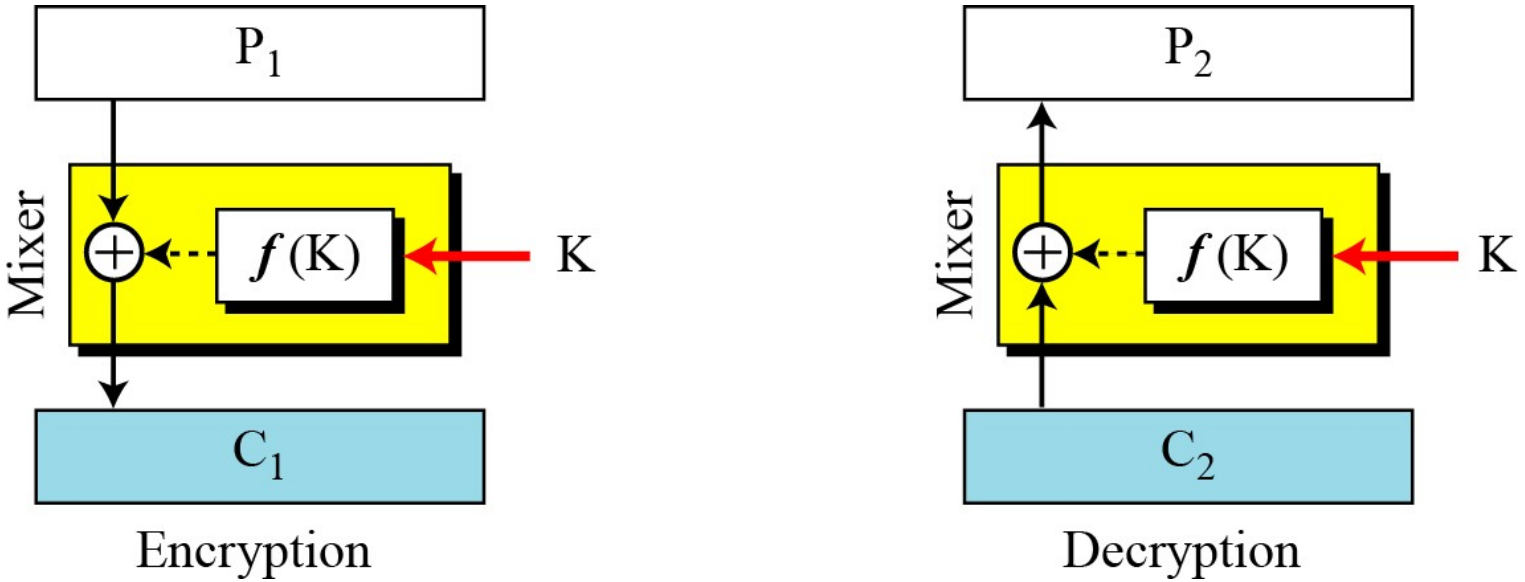


D Y PATIL
RAMRAO ADIK INSTITUTE OF TECHNOLOGY
NAVI MUMBAI

Feistel Cipher

- A Feistel cipher can have three types of components:
  - self-invertible
  - invertible
  - noninvertible

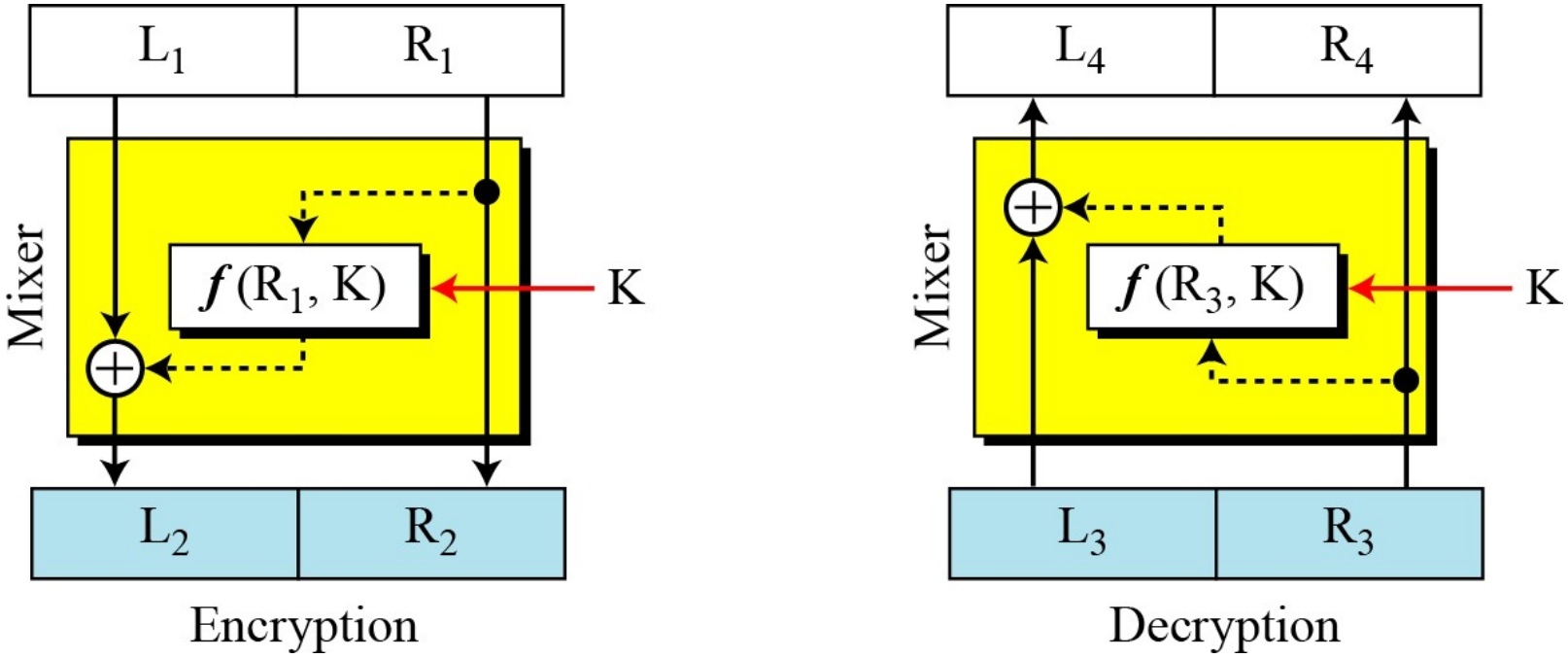
Feistel Cipher



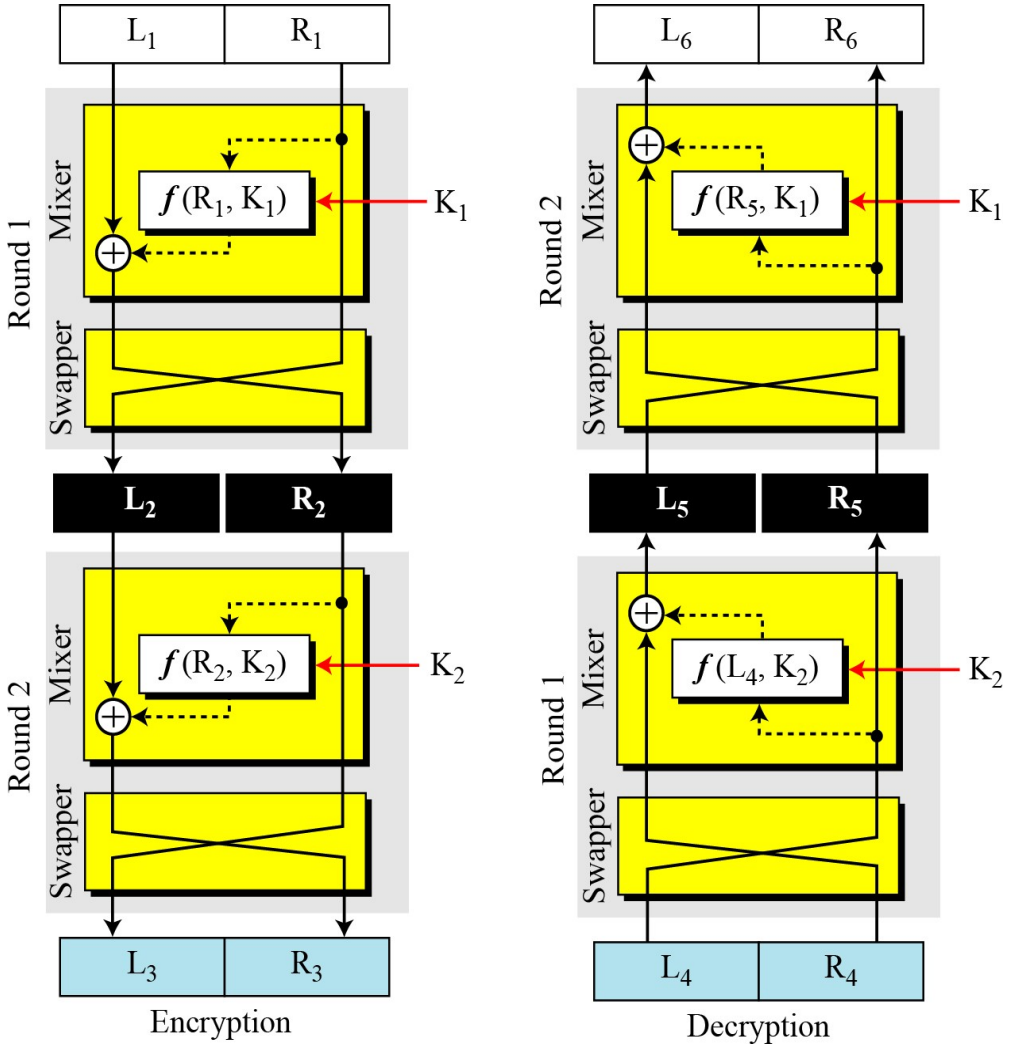
Encryption: $C = P \oplus f(K) = 0111 \oplus 1001 = 1110$

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

Improvement in Feistel Cipher

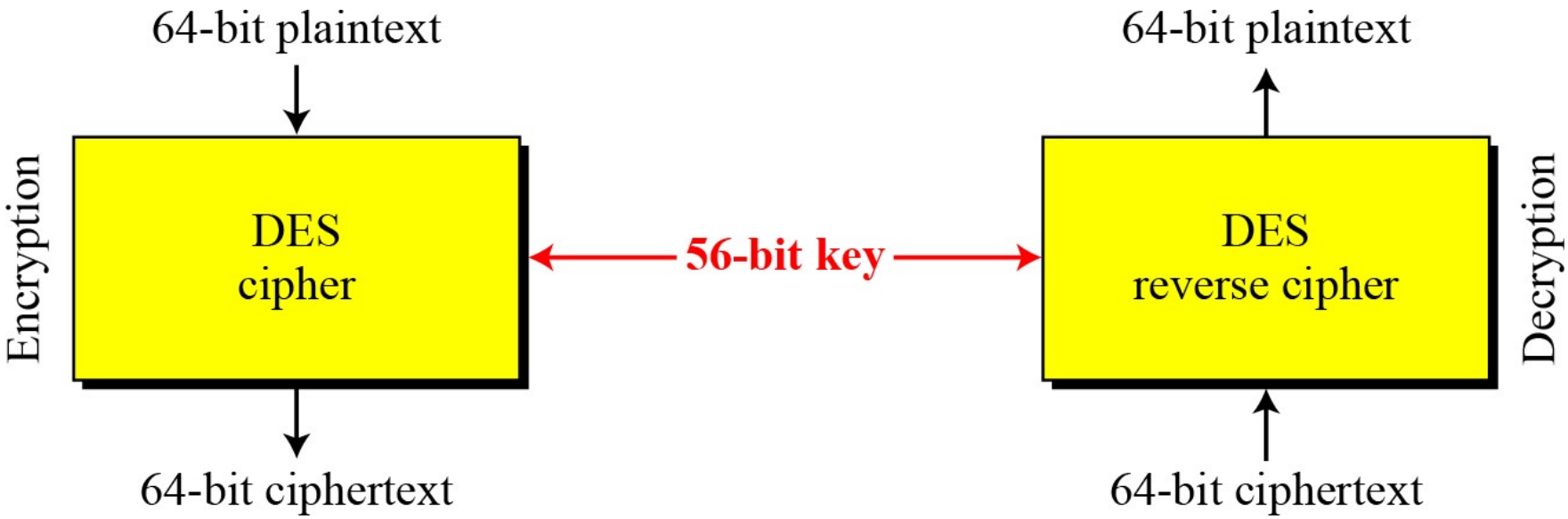


Final Design of Feistel Cipher

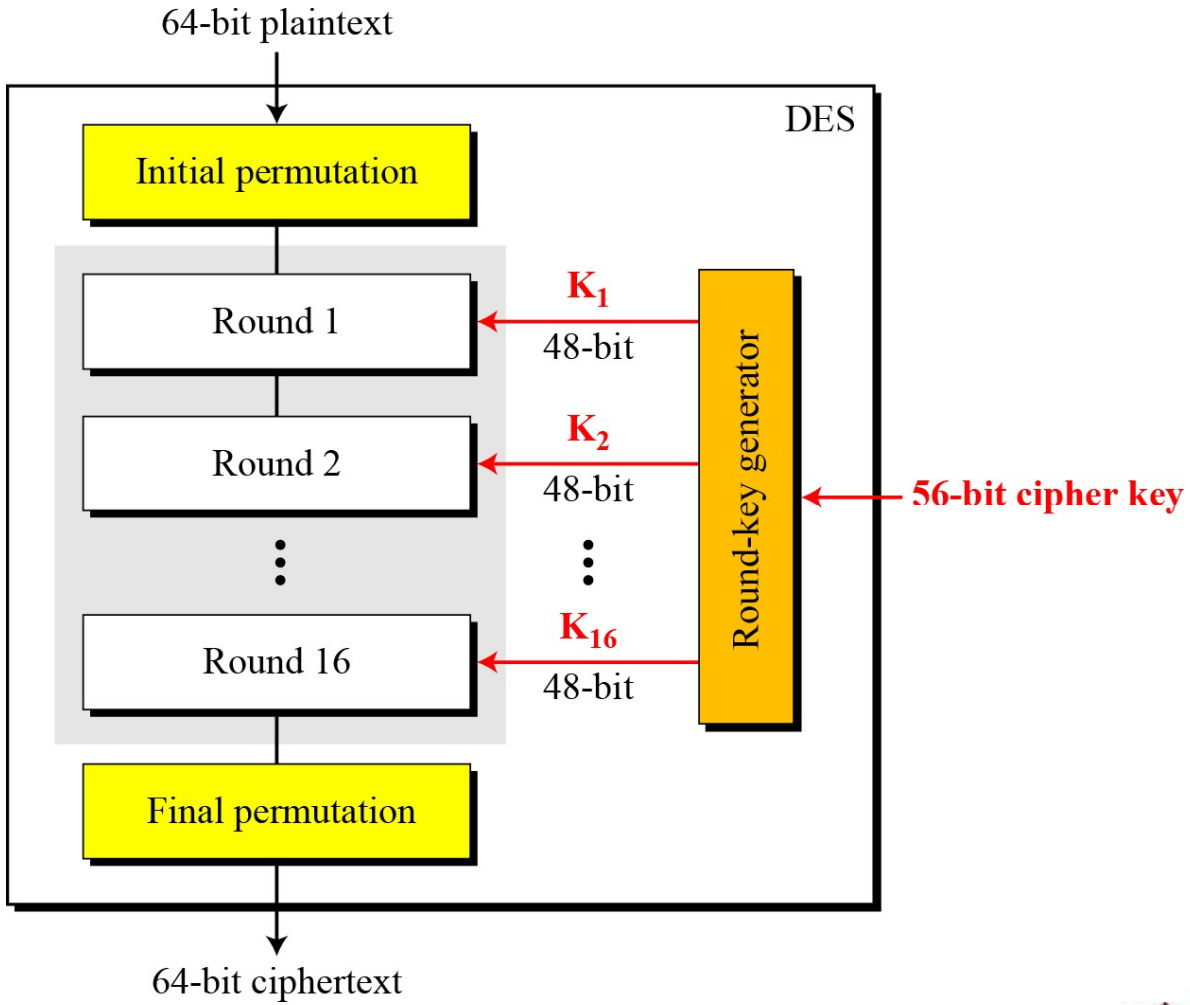


Data Encryption Standard

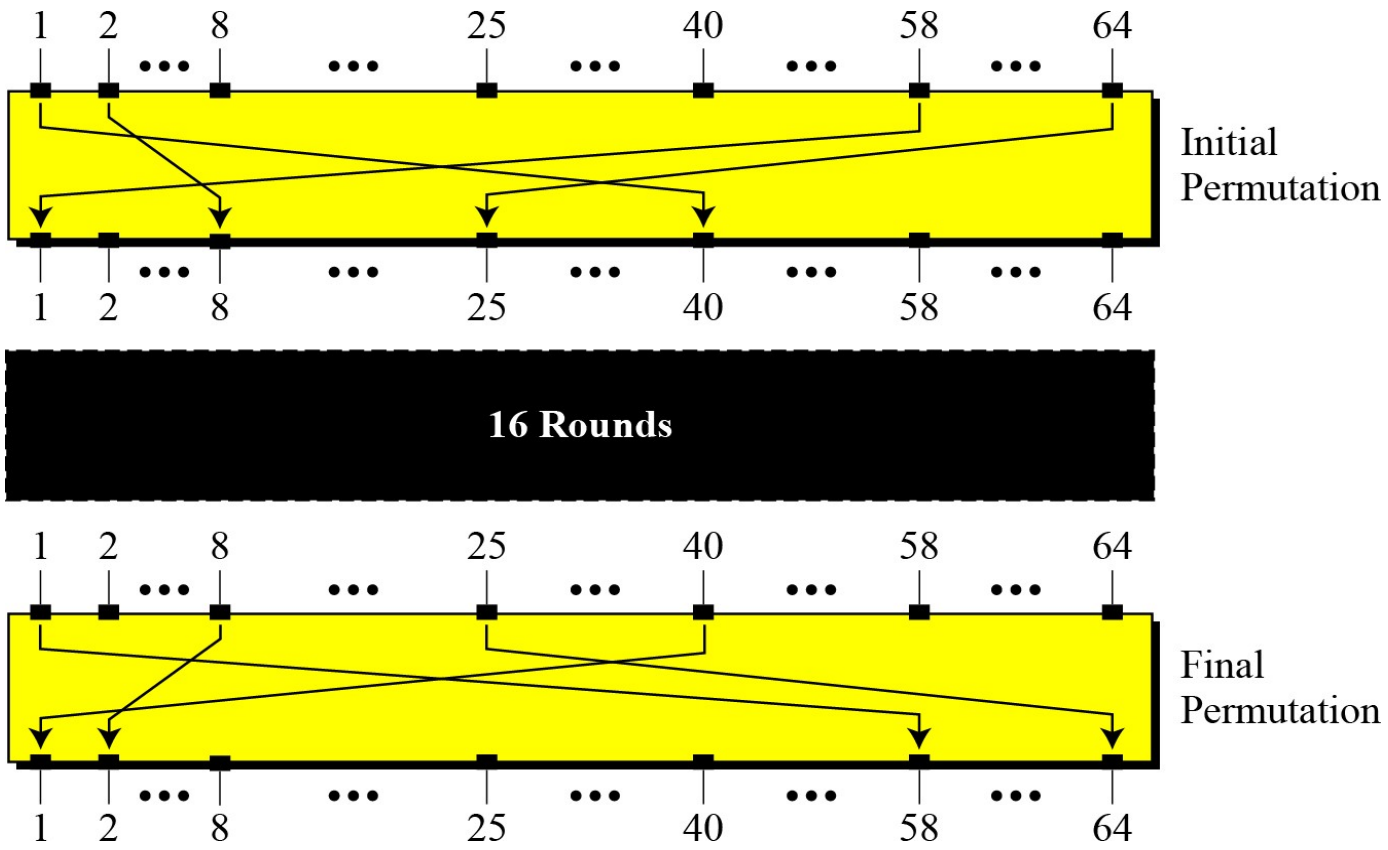
- The Data Encryption Standard (DES) is a symmetric-key block cipher published by the National Institute of Standards and Technology (NIST).



General Structure of DES



Initial and Final Permutations



Initial and Final Permutation Tables

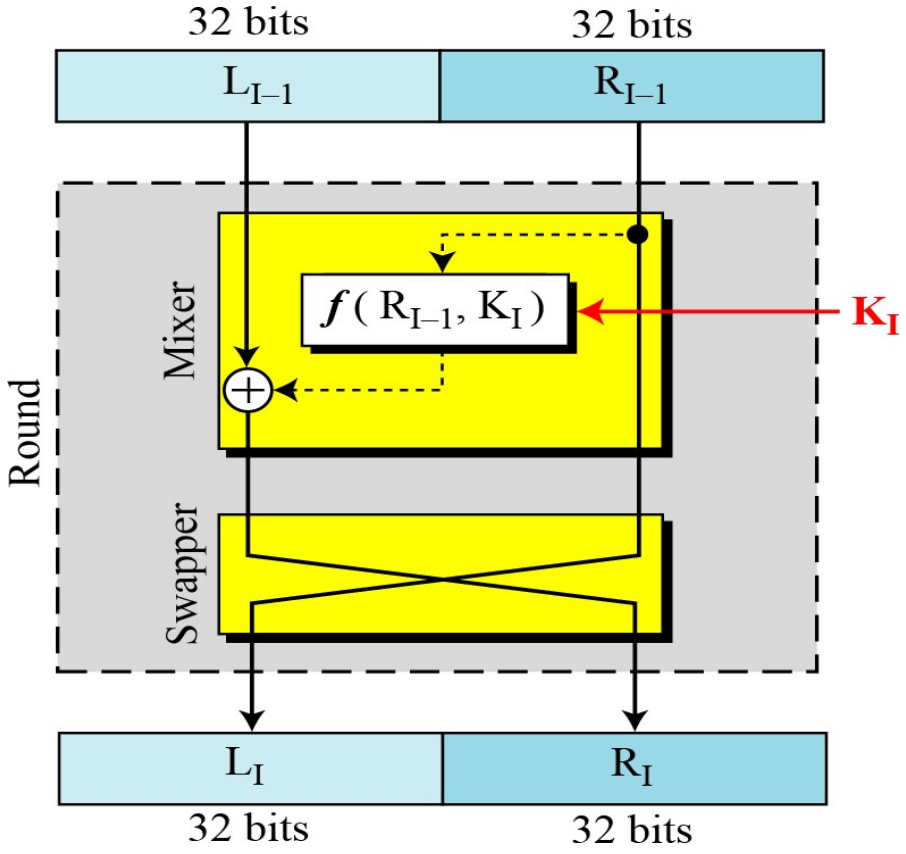
Initial Permutation							
58	50	42	34	26	18	10	02
60	52	44	36	28	20	12	04
62	54	46	38	30	22	14	06
64	56	48	40	32	24	16	08
57	49	41	33	25	17	09	01
59	51	43	35	27	19	11	03
61	53	45	37	29	21	13	05
63	55	47	39	31	23	15	07

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

The initial and final permutations are straight P-boxes that are inverses of each other. They have no cryptography significance in DES.

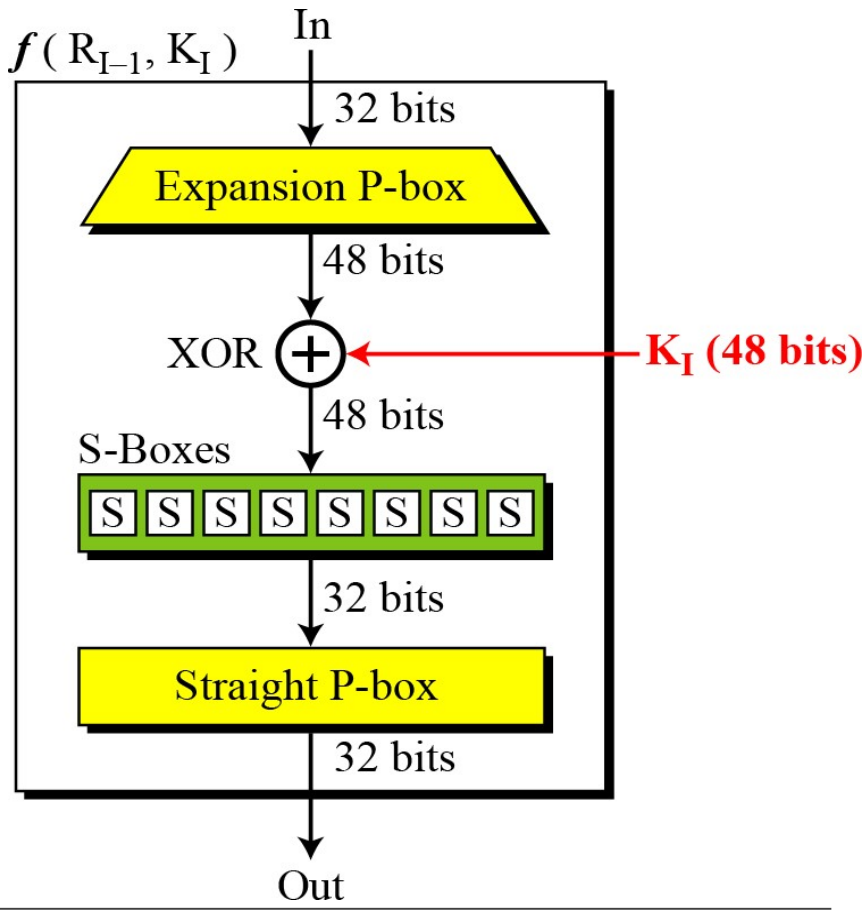
DES Rounds

DES uses 16 rounds. Each round of DES is a Feistel cipher.



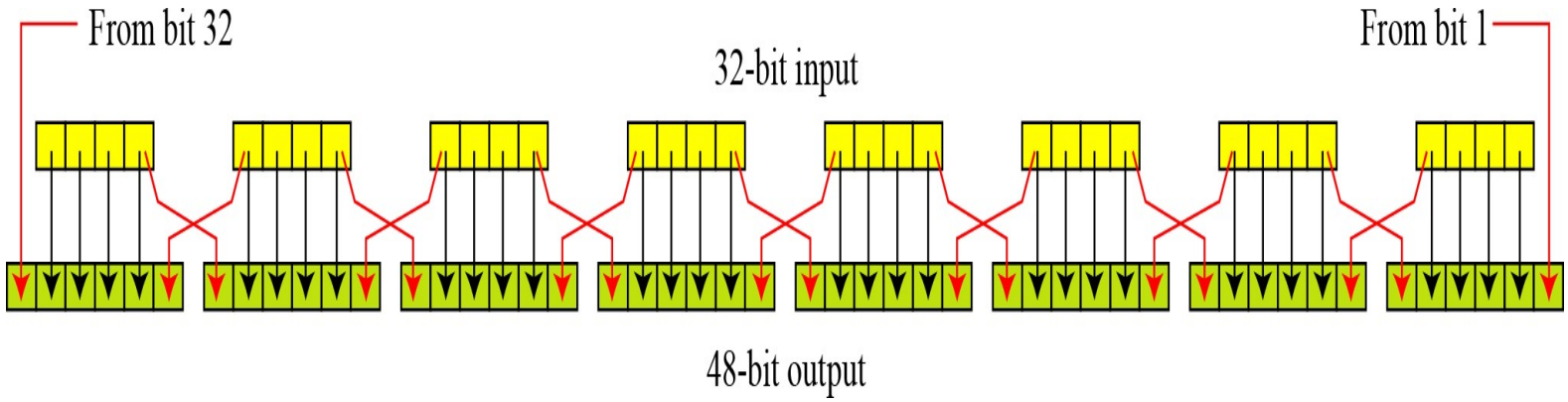
DES Function

The heart of DES is the DES function. The DES function applies a 48-bit key to the rightmost 32 bits to produce a 32-bit output.



Expansion P-Box

Since R_{I-1} is a 32-bit input and K_I is a 48-bit key, we first need to expand R_{I-1} to 48 bits.



Expansion permutation

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

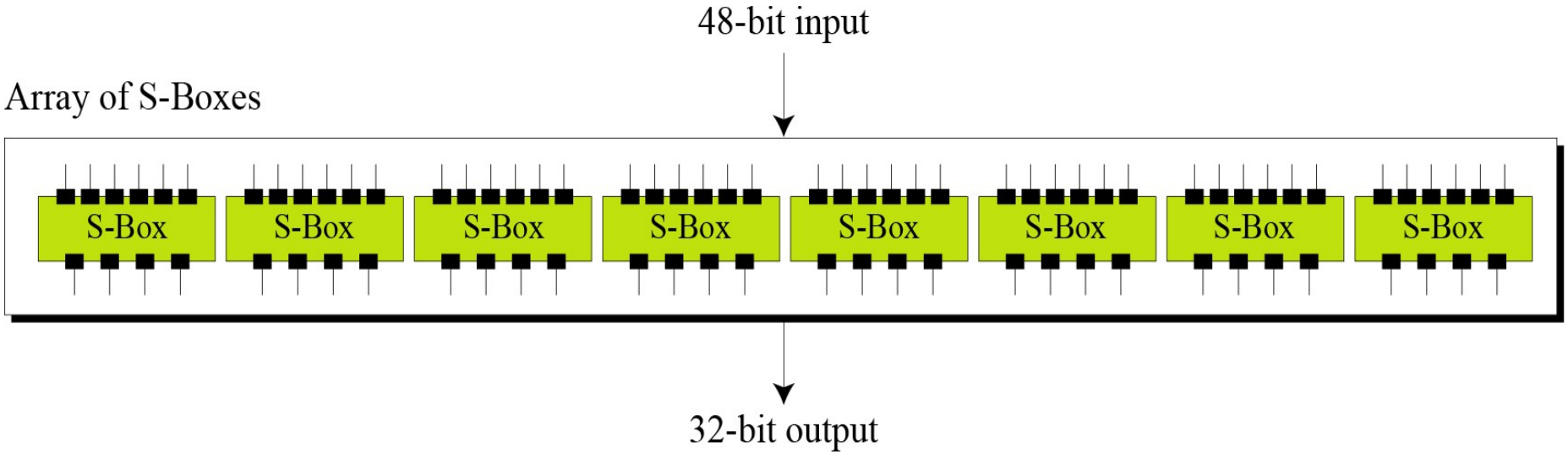
Whitener(XOR)

- After the expansion permutation, DES uses the XOR operation on the expanded right section and the round key.

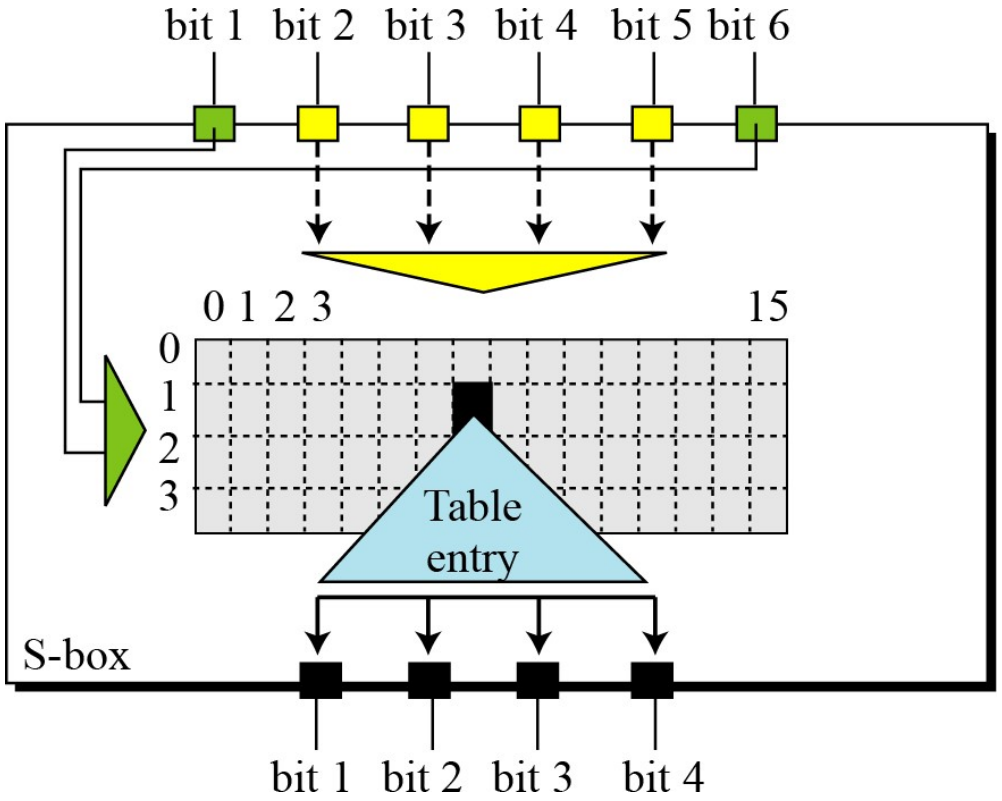
- Both the right section and the key are 48-bits in length. Also note that the round key is used only in this operation.

S-Boxes

The S-boxes do the real mixing (confusion). DES uses 8 S-boxes, each with a 6-bit input and a 4-bit output.



S-box Rule


bit 1 bit 2 bit 3 bit 4 bit 5 bit 6
0 1 2 3
15
0
1
2
3
Table entry
S-box
bit 1 bit 2 bit 3 bit 4

Straight Permutation

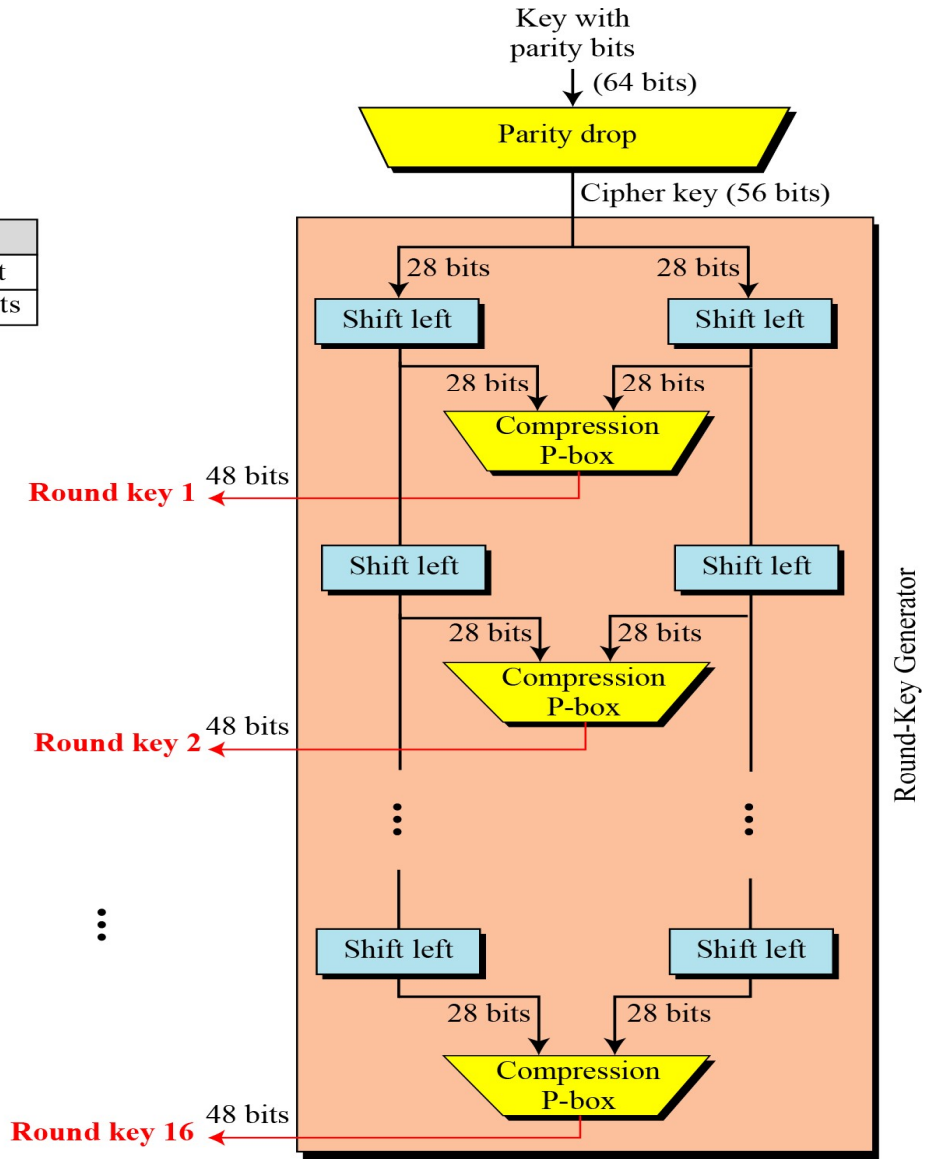
16	07	20	21	29	12	28	17
01	15	23	26	05	18	31	10
02	08	24	14	32	27	03	09
19	13	30	06	22	11	04	25

Straight permutation table

Key Generation

Shifting

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



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

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

Cipher and Reverse Cipher

Using mixers and swappers, we can create the cipher and reverse cipher, each having 16 rounds.

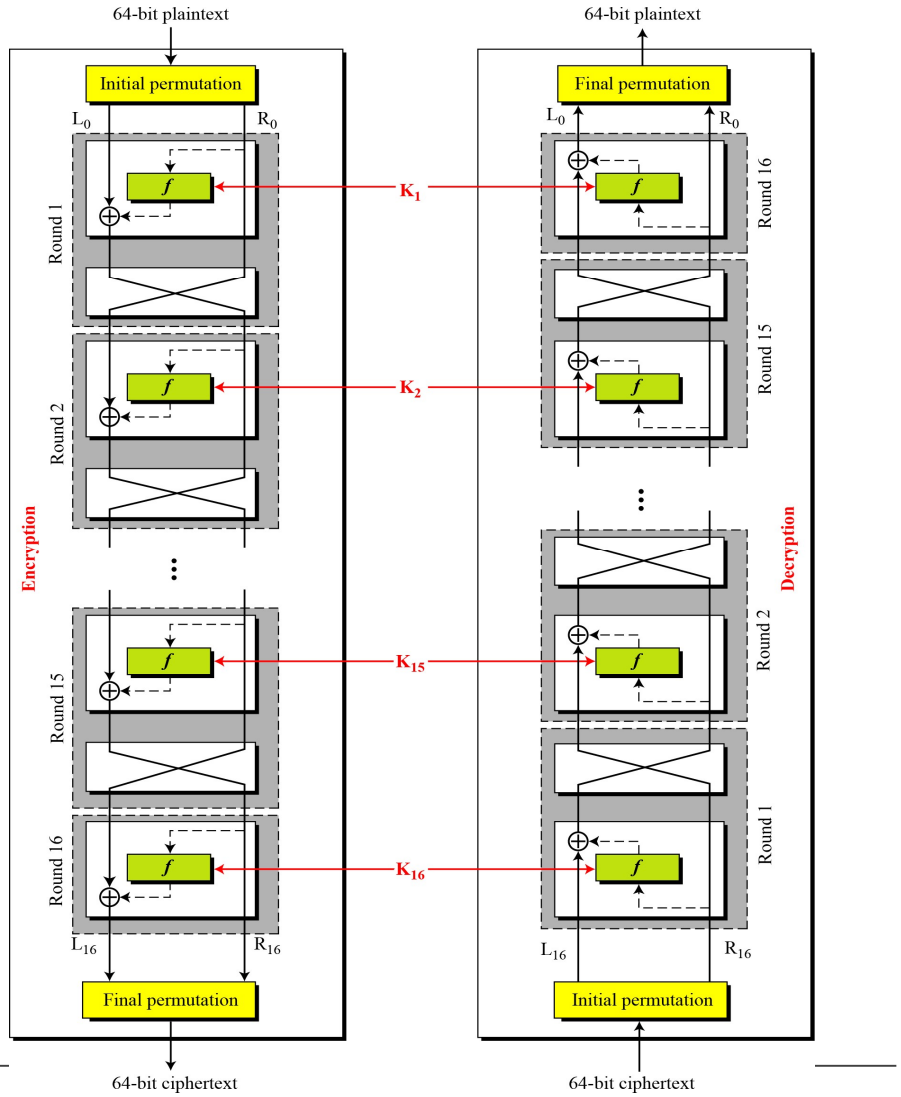
First Approach :

To achieve this goal, one approach is to make the last round (round 16) different from the others; it has only a mixer and no swapper.

Alternative Approach :

We can make all 16 rounds the same by including one swapper to the 16th round and add an extra swapper after that (two swappers cancel the effect of each other).

DES Cipher and Reverse Cipher for the first Approach



DES Example

Plaintext: 123456ABCD132536 Key: AABB09182736CCDD
CipherText: C0B7A8D05F3A829C

Plaintext: 123456ABCD132536			
*After initial permutation:*14A7D67818CA18AD After splitting: L_0=14A7D678 R_0=18CA18AD			
Round	*Left*	*Right*	*Round Key*
Round 1	18CA18AD	5A78E394	194CD072DE8C
Round 2	5A78E394	4A1210F6	4568581ABCCE
Round 3	4A1210F6	B8089591	06EDA4ACF5B5
Round 4	B8089591	236779C2	DA2D032B6EE3

DES Example(Cont.…)

Round 5	236779C2	A15A4B87	69A629FEC913
Round 6	A15A4B87	2E8F9C65	C1948E87475E
Round 7	2E8F9C65	A9FC20A3	708AD2DDB3C0
Round 8	A9FC20A3	308BEE97	34F822F0C66D
Round 9	308BEE97	10AF9D37	84BB4473DCCC
Round 10	10AF9D37	6CA6CB20	02765708B5BF
Round 11	6CA6CB20	FF3C485F	6D5560AF7CA5
Round 12	FF3C485F	22A5963B	C2C1E96A4BF3
Round 13	22A5963B	387CCDAA	99C31397C91F
Round 14	387CCDAA	BD2DD2AB	251B8BC717D0
Round 15	BD2DD2AB	CF26B472	3330C5D9A36D
Round 16	19BA9212	CF26B472	181C5D75C66D
After combination: 19BA9212CF26B472			
Ciphertext: C0B7A8D05F3A829C			*(after final permutation)*

DES Properties

Avalanche *effect*

- *Avalanche effect means a small change in the plaintext (or key)*
  should create a significant change in the ciphertext
- *DES strongly support this property*

Plaintext	0000000000000000
Key	22234512987ABB23
Ciphertext	4789FD476E82A5F1
Plaintext	0000000000000001
Key	4789FD476E82A5F1
Ciphertext	0A4ED5C15A63FEA3

DES Properties

Completeness

- Completeness effect means that each bit of the ciphertext needs to depend on many bits on the plaintext.

- The confusion and diffusion produced by P-boxes and S-boxes in DES show a very strong completeness effect.

DES Design Criteria

S-Box:

- S-boxes are non-linear.
- If we change single bit in the input, two or more bits will be changed in the output.
- The design provides confusion and diffusion of bits from
- each round to the next.

P-Box:

- Provide diffusion of bits

Number of Rounds:

- DES uses sixteen rounds of Feistel ciphers.
- The ciphertext is thoroughly a random function of plaintext.

DES Weaknesses

1. Weaknesses in S-boxes
    - In S-box 4, the last 3 output bits can be derived in the same way as the first output bit by complementing some of the input bits.
    - Two specifically chosen inputs to an S-box array can create the same output
    - It is possible to obtain the same output in a single round by changing bits in only three neighboring S-boxes.

DES Weaknesses

2. Weaknesses in P-boxes

- It is not clear why the designers of DES used initial and final permutations; these have no security benefits.

- In the expansion permutation(inside the function), the first and fourth bits of every 4-bit series are repeated.

DES Weaknesses

3. Weaknesses in Key

Table 6.18 *Weak keys*

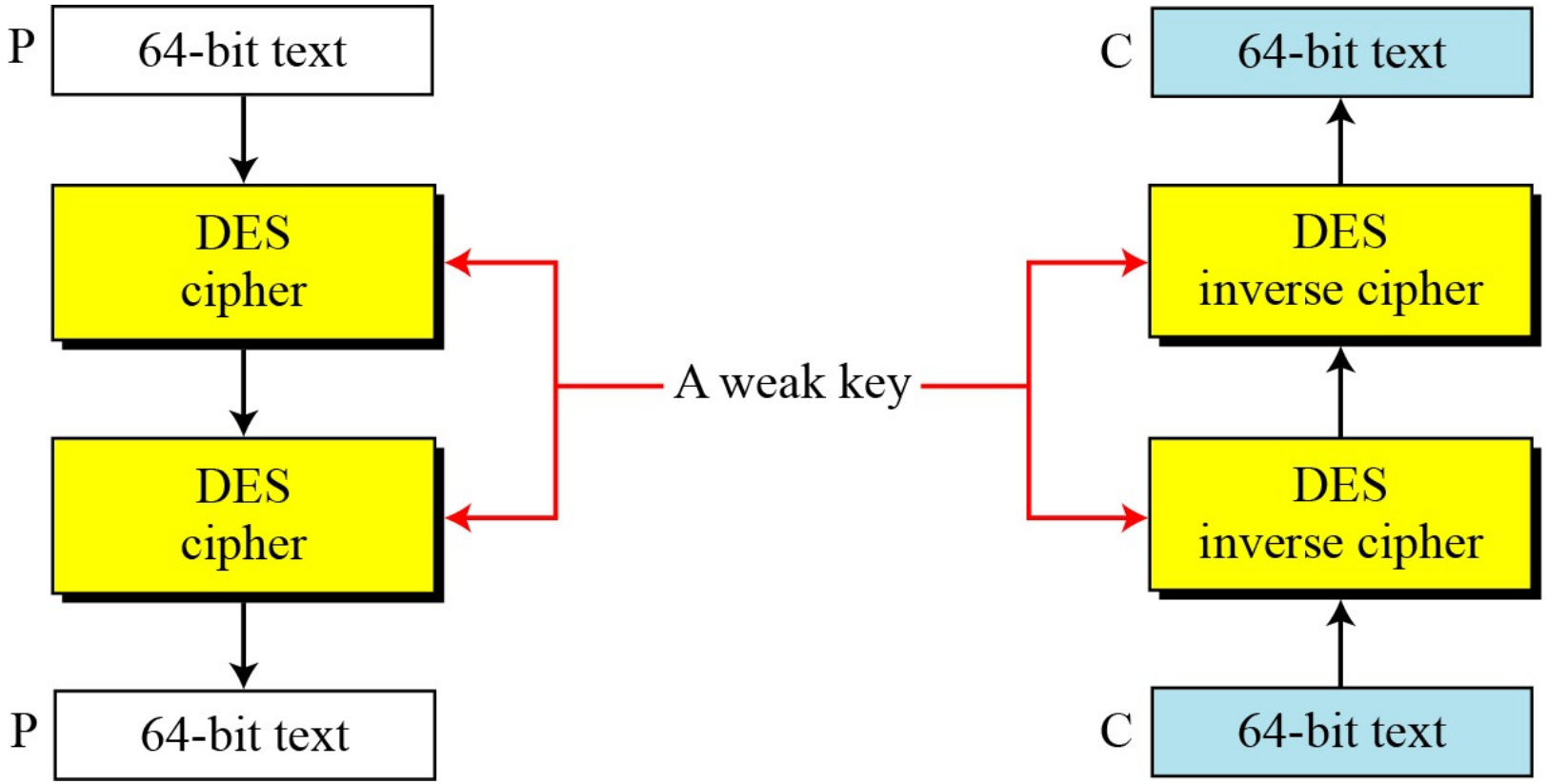
Keys before parities drop (64 bits)	*Actual key (56 bits)*
0101 0101 0101 0101	0000000 0000000
1F1F 1F1F 0E0E 0E0E	0000000 FFFFFFF
E0E0 E0E0 F1F1 F1F1	FFFFFFF 0000000
FEFE FEFE FEFE FEFE	FFFFFFF FFFFFFF

DES Weaknesses

3. Weaknesses in Key

- Let us try the first weak key to encrypt a block two times. After two encryptions with the same key the original plaintext block is created
- The encryption algorithm is used two times, not one encryption followed by another decryption

Double Encryption and Decryption with Weak Key



DES Weaknesses

Weaknesses in Key

Table 6.19 *Semi-weak keys*

First key in the pair	*Second key in the pair*
01FE 01FE 01FE 01FE	FE01 FE01 FE01 FE01
1FE0 1FE0 0EF1 0EF1	E01F E01F F10E F10E
01E0 01E1 01F1 01F1	E001 E001 F101 F101
1FFE 1FFE 0EFE 0EFE	FE1F FE1F FE0E FE0E
011F 011F 010E 010E	1F01 1F01 0E01 0E01
E0FE E0FE F1FE F1FE	FEE0 FEE0 FEF1 FEF1

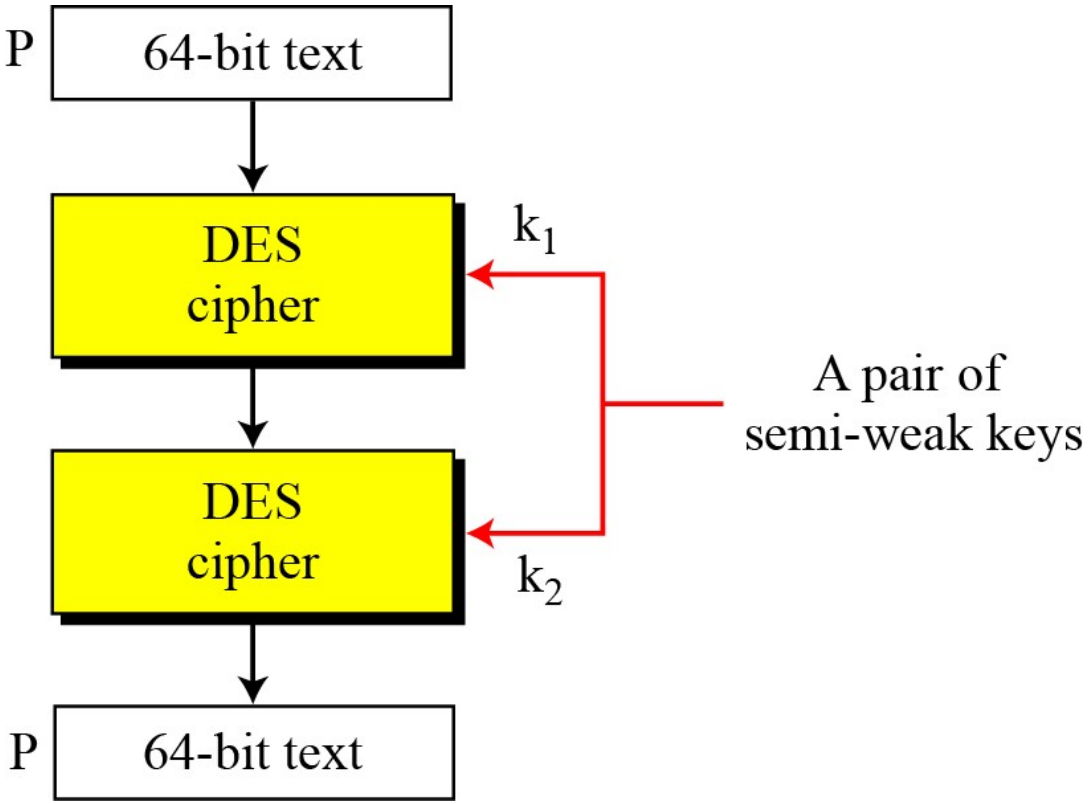
DES Weaknesses

Weaknesses in Key

Round key 1	9153E54319BD	6EAC1ABCE642
Round key 2	6EAC1ABCE642	9153E54319BD
Round key 3	6EAC1ABCE642	9153E54319BD
Round key 4	6EAC1ABCE642	9153E54319BD
Round key 5	6EAC1ABCE642	9153E54319BD
Round key 6	6EAC1ABCE642	9153E54319BD
Round key 7	6EAC1ABCE642	9153E54319BD
Round key 8	6EAC1ABCE642	9153E54319BD
Round key 9	9153E54319BD	6EAC1ABCE642
Round key 10	9153E54319BD	6EAC1ABCE642
Round key 11	9153E54319BD	6EAC1ABCE642
Round key 12	9153E54319BD	6EAC1ABCE642
Round key 13	9153E54319BD	6EAC1ABCE642
Round key 14	9153E54319BD	6EAC1ABCE642
Round key 15	9153E54319BD	6EAC1ABCE642
Round key 16	6EAC1ABCE642	9153E54319BD

DES Weaknesses

Weaknesses in Key



Double DES

- The first approach is to use double DES (2DES).

Meet-in-the-Middle Attack

- However, using a known-plaintext attack called meet-in-the-middle attack proves that double DES improves this vulnerability slightly (to 2^{57} tests), but not tremendously (to 2^{112})

Security of DES

Brute-Force Attack

- DES can be broken using 2^{55} encryptions.

Differential Cryptanalysis

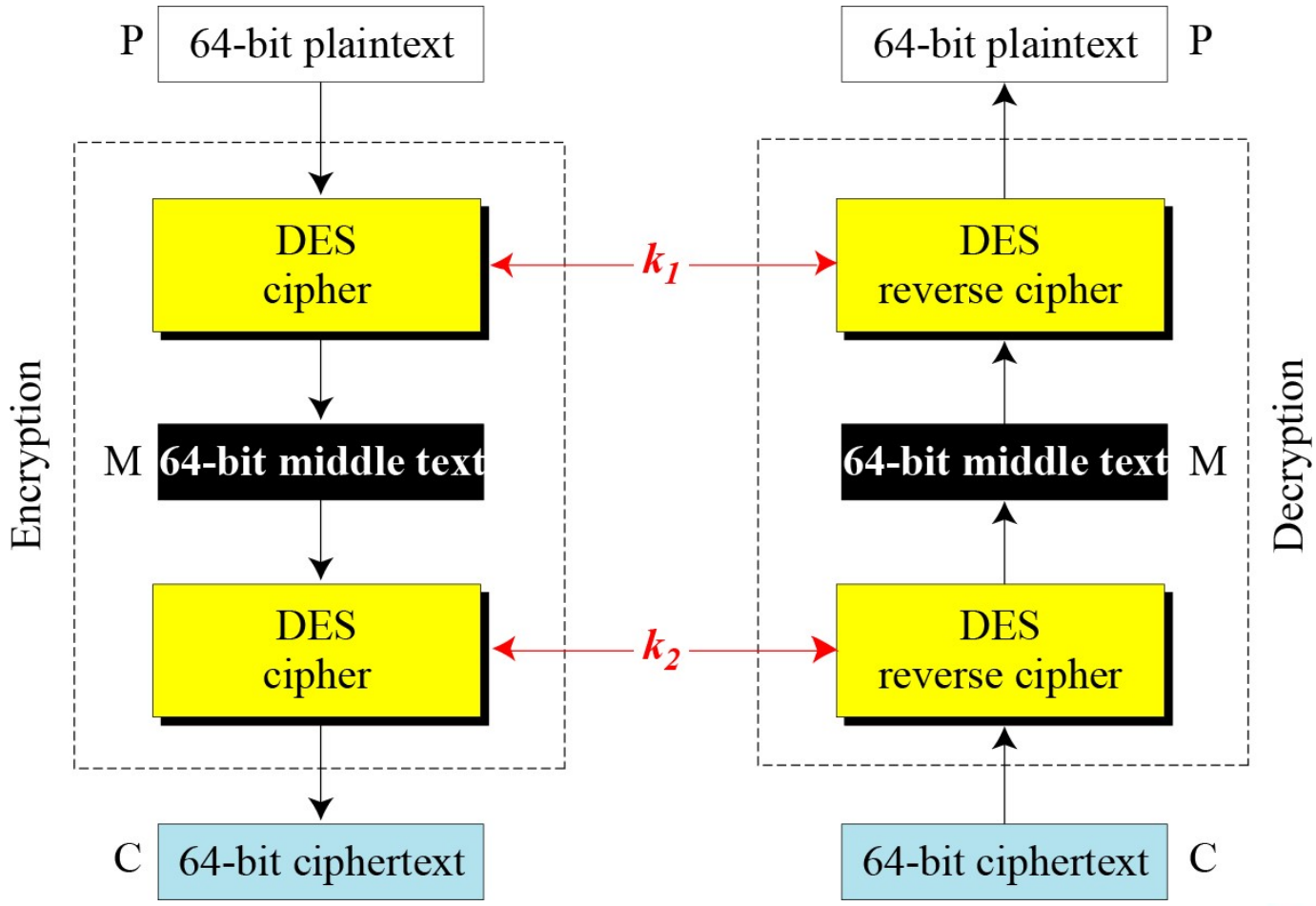
- Chosen Plaintext Attack has been revealed that the designers of DES already knew about this type of attack and designed S-boxes and chose 16 as the number of rounds to make DES specifically resistant to this type of attack

Security of DES

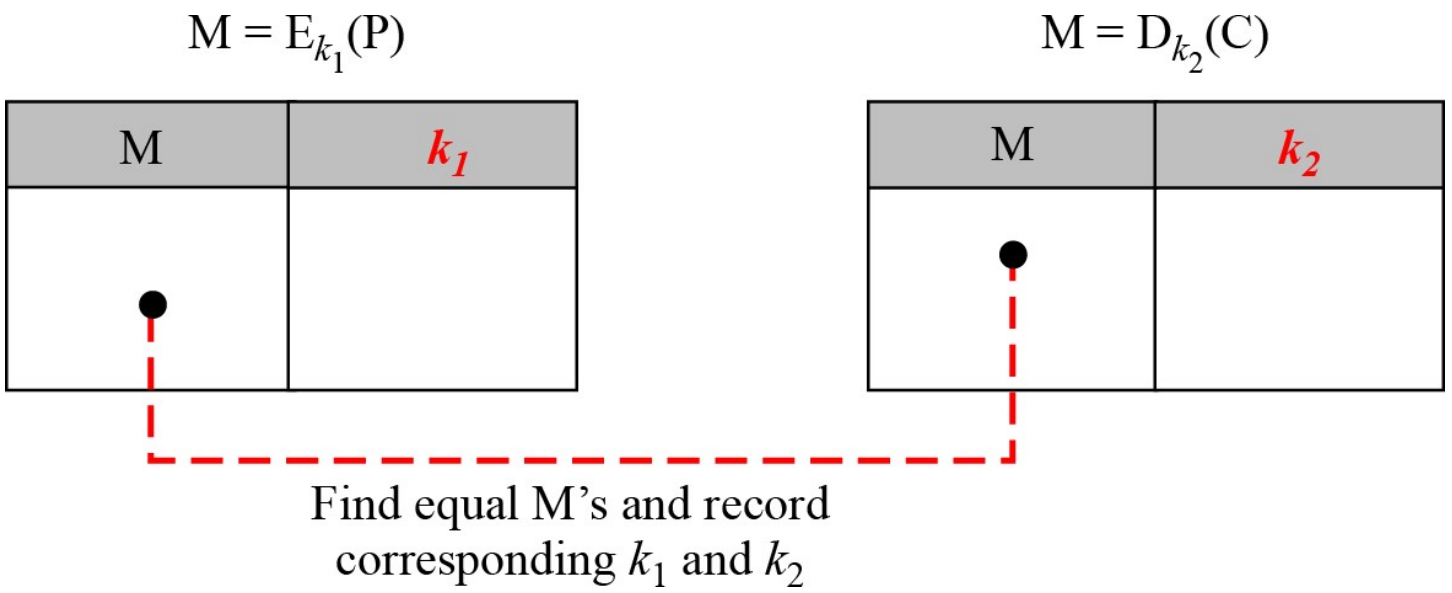
Linear Cryptanalysis

- known Plaintext attack
- Linear cryptanalysis is newer than differential cryptanalysis.
- DES is more vulnerable to linear cryptanalysis than to differential cryptanalysis.
- S-boxes are not very resistant to linear cryptanalysis.
- It has been shown that DES can be broken using 2^{43} pairs of known plaintexts. However, from the practical point of view, finding so many pairs is very unlikely.

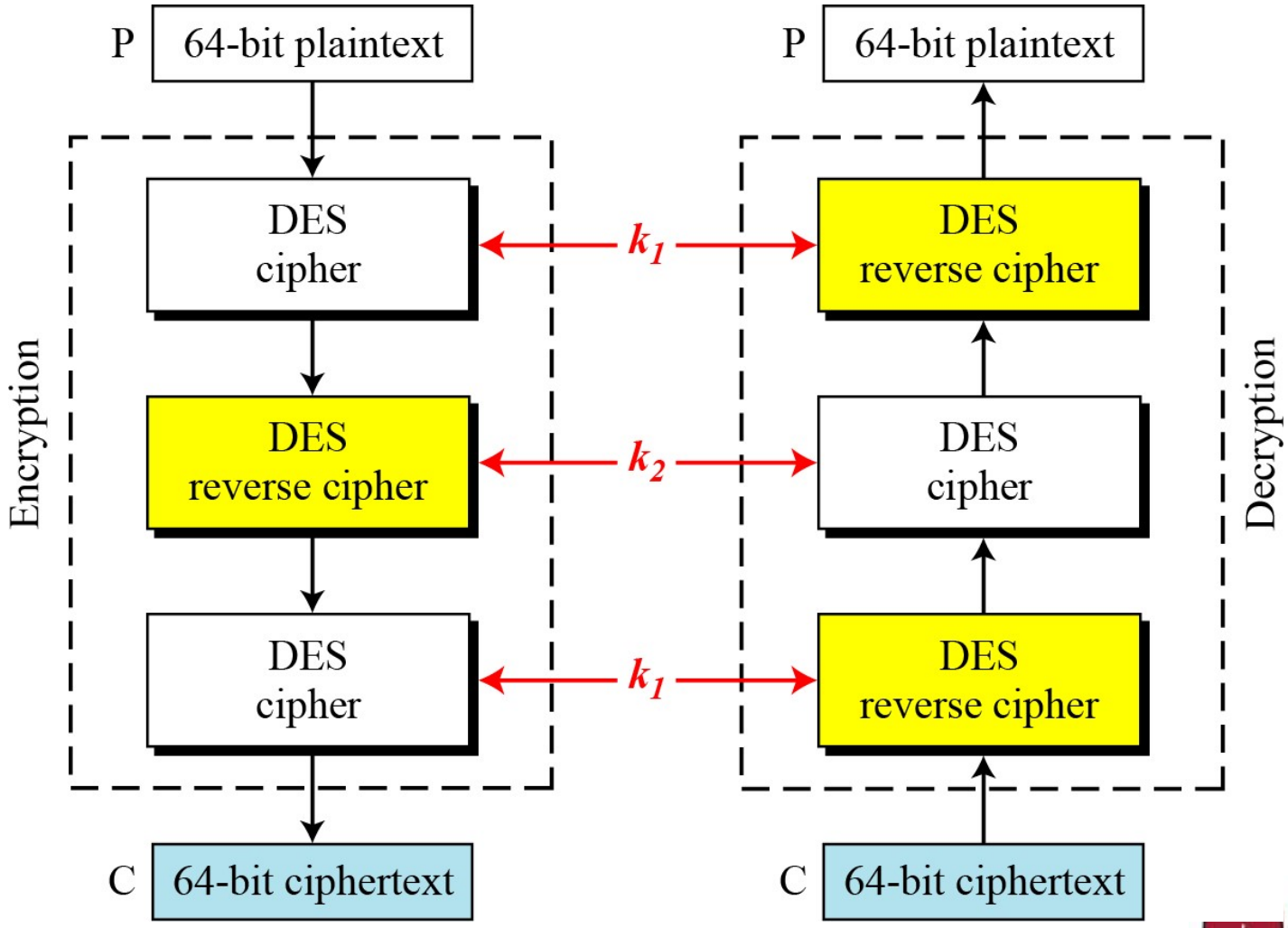
Double DES



Double DES


M = E_{k_1}(P)
M = D_{k_2}(C)
M
k_1
M
k_2
Find equal M's and record
corresponding k_1 and k_2

Triple DES



Triple DES

Triple DES with Three Keys

- The possibility of known-plaintext attacks on triple DES with two keys has enticed some applications to use triple DES with three keys

- Triple DES with three keys is used by many applications such as PGP

Security of DES

- DES, as the first important block cipher, has gone through much scrutiny.

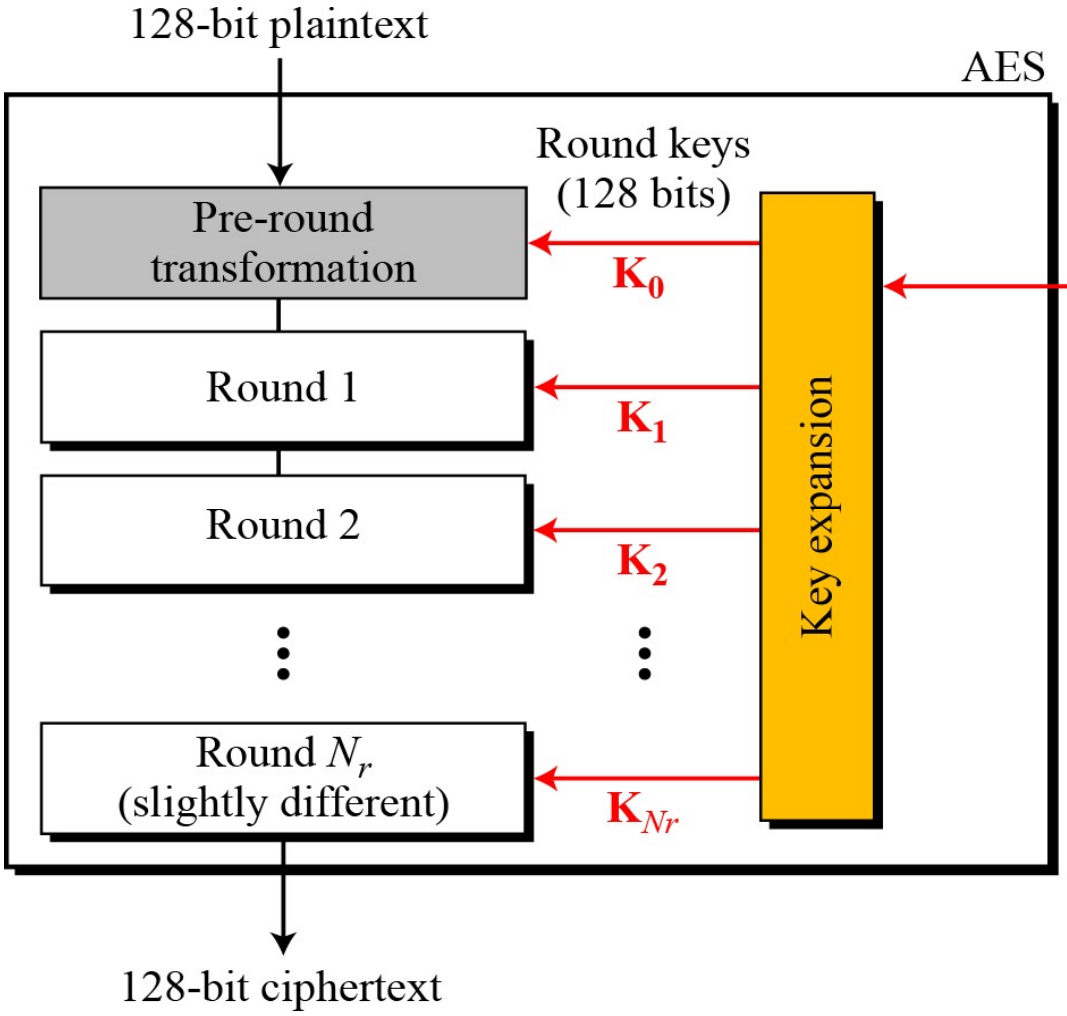
- Among the attempted attacks, three are of interest:

  - Brute-Force Attack
  - Differential Cryptanalysis
  - Linear Cryptanalysis

Advanced Encryption Standard(AES)

- Developed at NIST in 1997
- AES is a non-Feistel cipher(only inverible comonents) that encrypts and decrypts a data block of 128 bits
- It uses 10, 12, or 14 rounds
- The key size, which can be 128, 192, or 256 bits, depends on the number of rounds

Advanced Encryption Standard(AES)

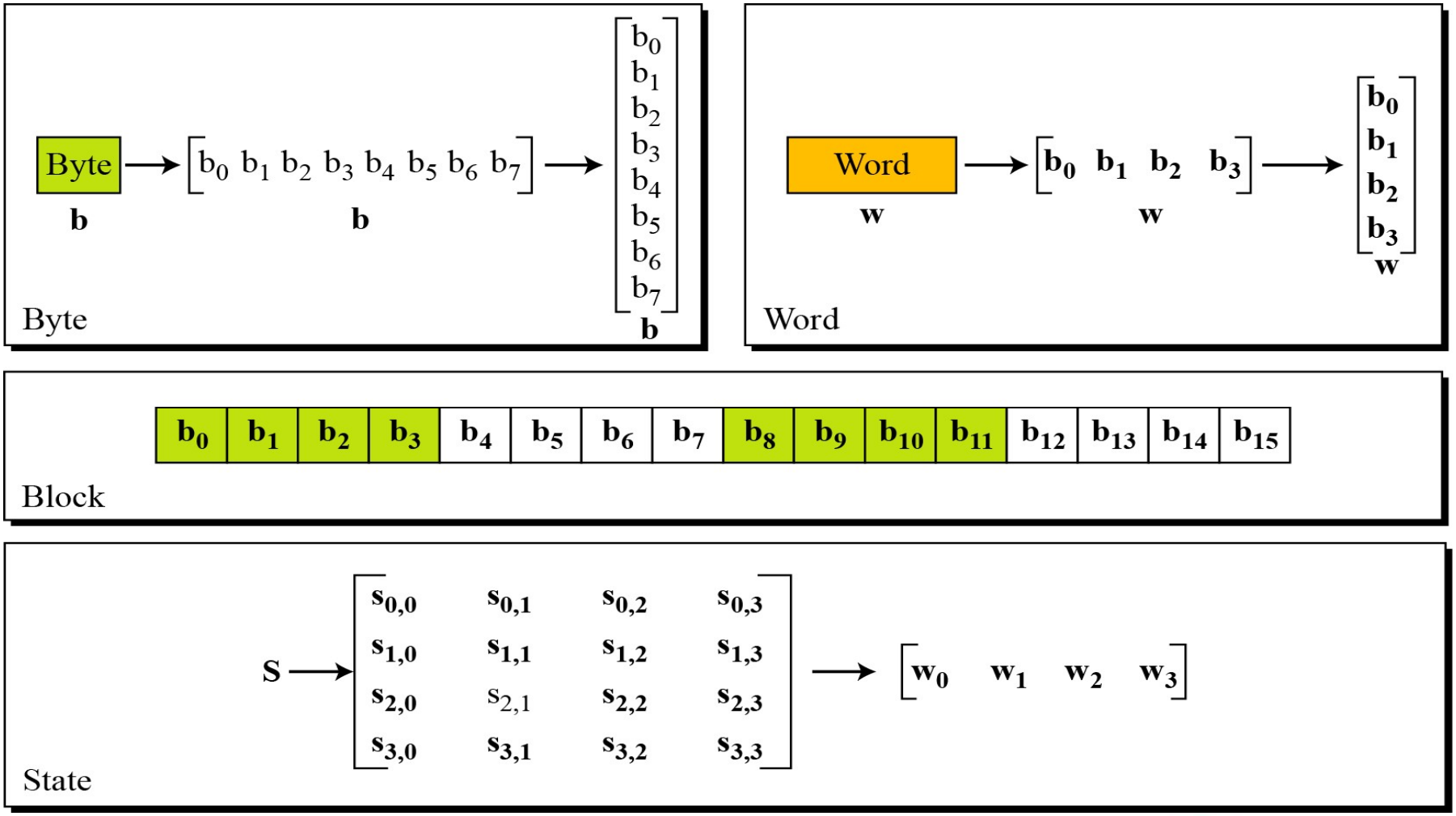


Cipher key
(128, 192, or 256 bits)

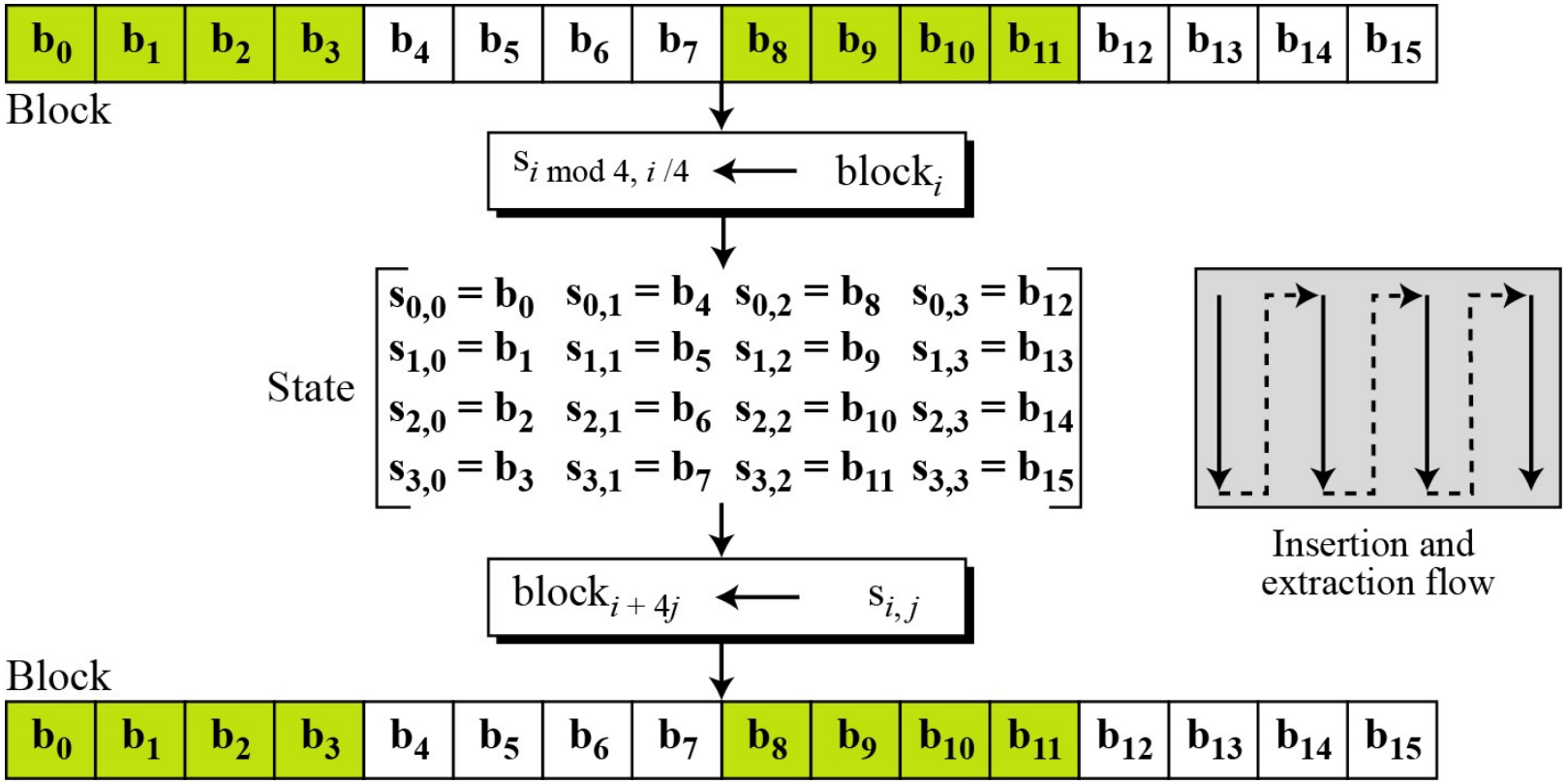
Nr	Key size
10	128
12	192
14	256

Relationship between number of rounds and cipher key size

Data Units in AES



Block-to-state and state-to-block Transformation

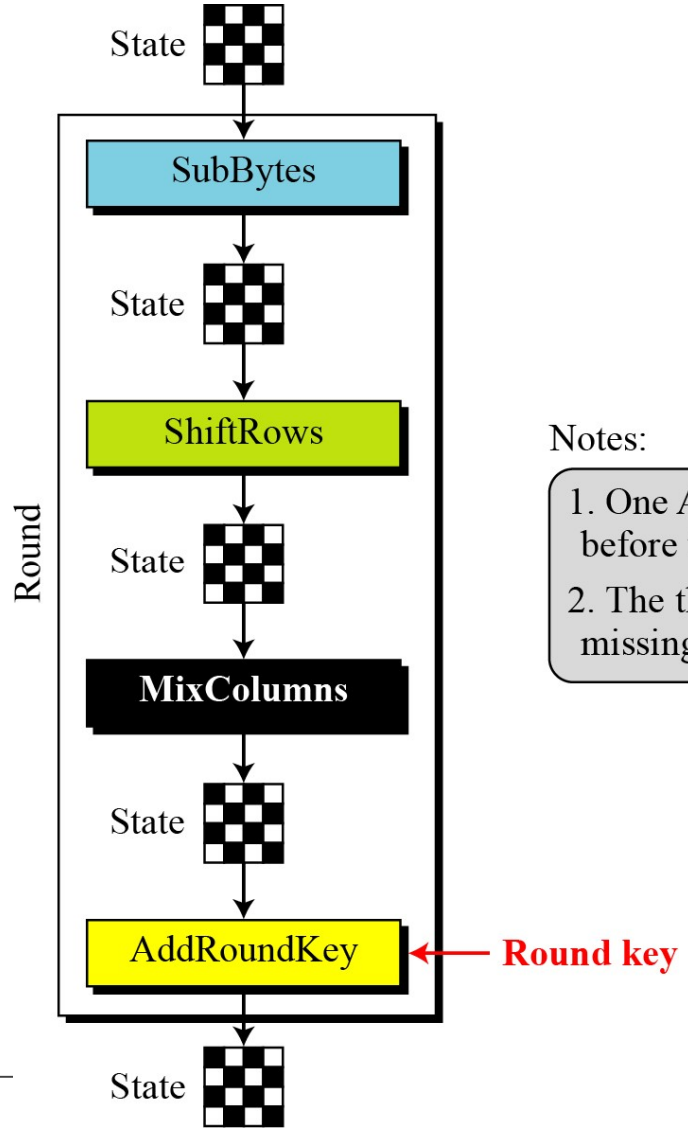


Changing Plaintext to State

Text	A	E	S	U	S	E	S	A	M	A	T	R	I	X	Z	Z
Hexadecimal	00	04	12	14	12	04	12	00	0C	00	13	11	08	23	19	19

$$\begin{bmatrix} 00 & 12 & 0C & 08 \\ 04 & 04 & 00 & 23 \\ 12 & 12 & 13 & 19 \\ 14 & 00 & 11 & 19 \end{bmatrix}$$ State

Structure of Each Round at the Encryption Side



Notes:

1. One AddRoundKey is applied before the first round.
2. The third transformation is missing in the last round.

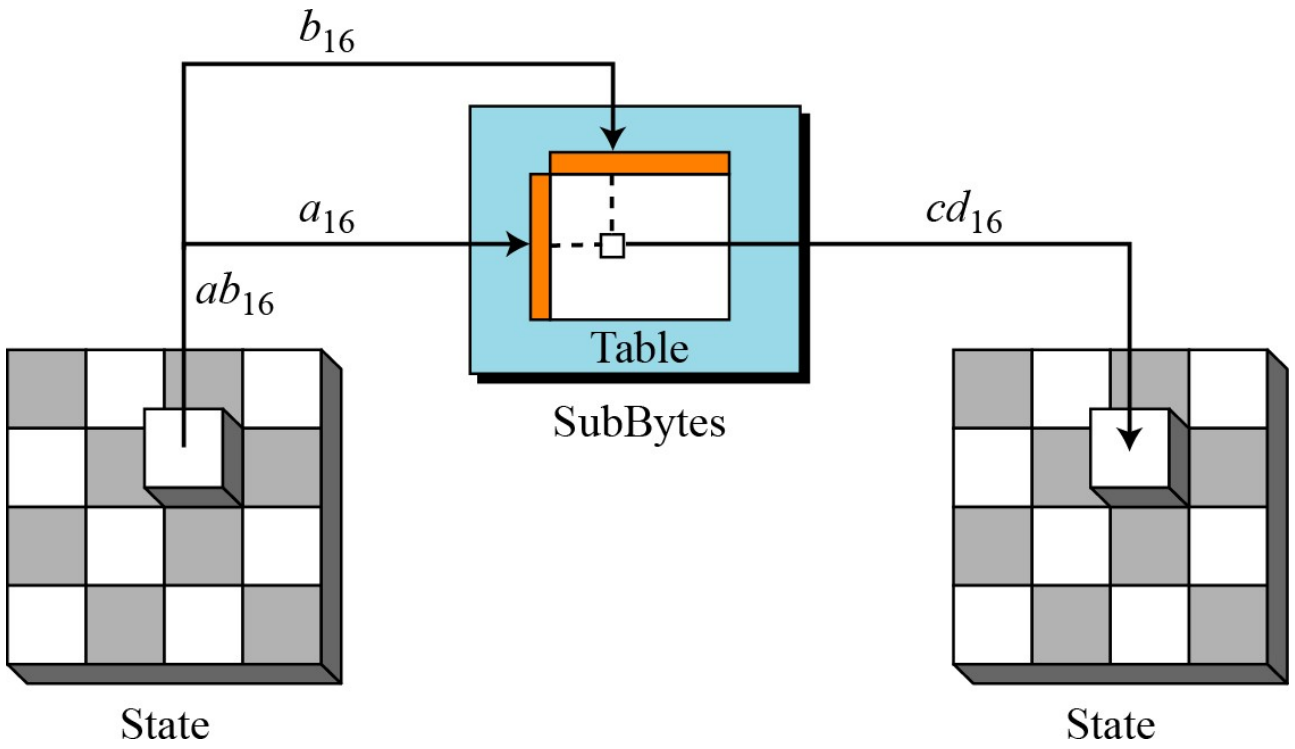
AES: Transformations

- AES uses four types of transformations:
  - ✓ Substitution
  - ✓ Permutation
  - ✓ Mixing
  - ✓ key-adding

AES: Transformations

- AES uses two invertible substitution transformations :
  - ✓ SubBytes :Byte is substituted as two hexadecimal digits.
  - ✓ InvSubBytes

SubBytes Transformations



SubBytes Transformations

Table 7.1 *SubBytes transformation table*

	0	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*A*	*B*	*C*	*D*	*E*	*F*
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8

SubBytes Transformations

Table 7.1 *SubBytes transformation table (continued)*

	0	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*A*	*B*	*C*	*D*	*E*	*F*
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	CB	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

InvSubBytes Transformations

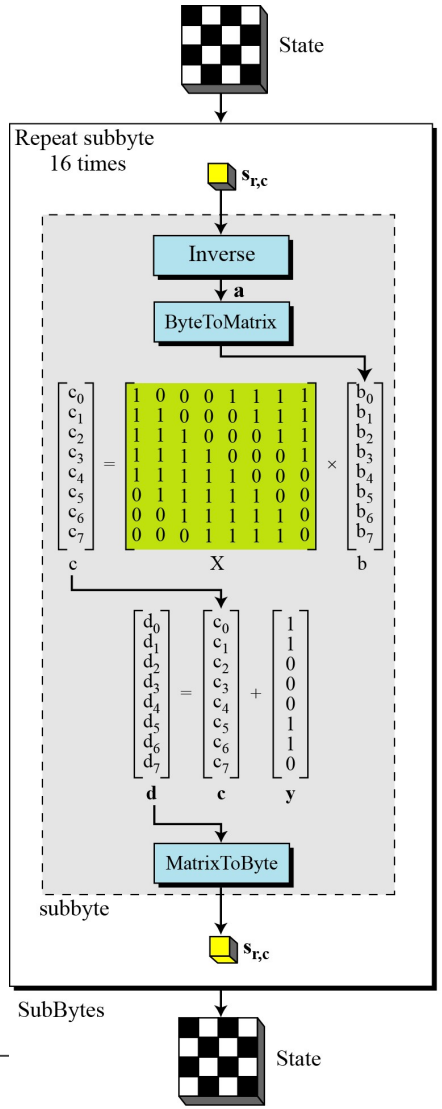
Table 7.2 *InvSubBytes transformation table*

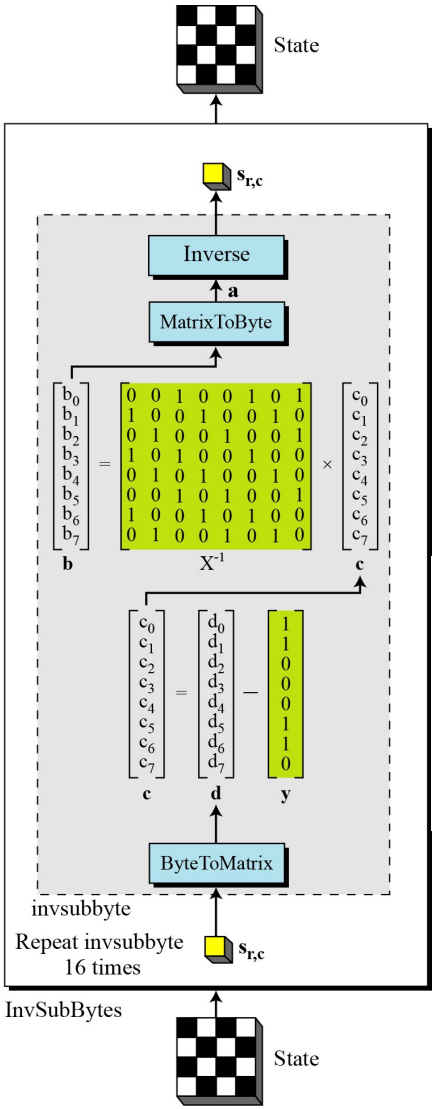
	0	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*A*	*B*	*C*	*D*	*E*	*F*
0	52	09	6A	D5	30	36	A5	38	BF	40	A3	9E	81	F3	D7	FB
1	7C	E3	39	82	9B	2F	FF	87	34	8E	43	44	C4	DE	E9	CB
2	54	7B	94	32	A6	C2	23	3D	EE	4C	95	0B	42	FA	C3	4E
3	08	2E	A1	66	28	D9	24	B2	76	5B	A2	49	6D	8B	D1	25
4	72	F8	F6	64	86	68	98	16	D4	A4	5C	CC	5D	65	B6	92
5	6C	70	48	50	FD	ED	B9	DA	5E	15	46	57	A7	8D	9D	84
6	90	D8	AB	00	8C	BC	D3	0A	F7	E4	58	05	B8	B3	45	06
7	D0	2C	1E	8F	CA	3F	0F	02	C1	AF	BD	03	01	13	8A	6B

InvSubBytes Transformations

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

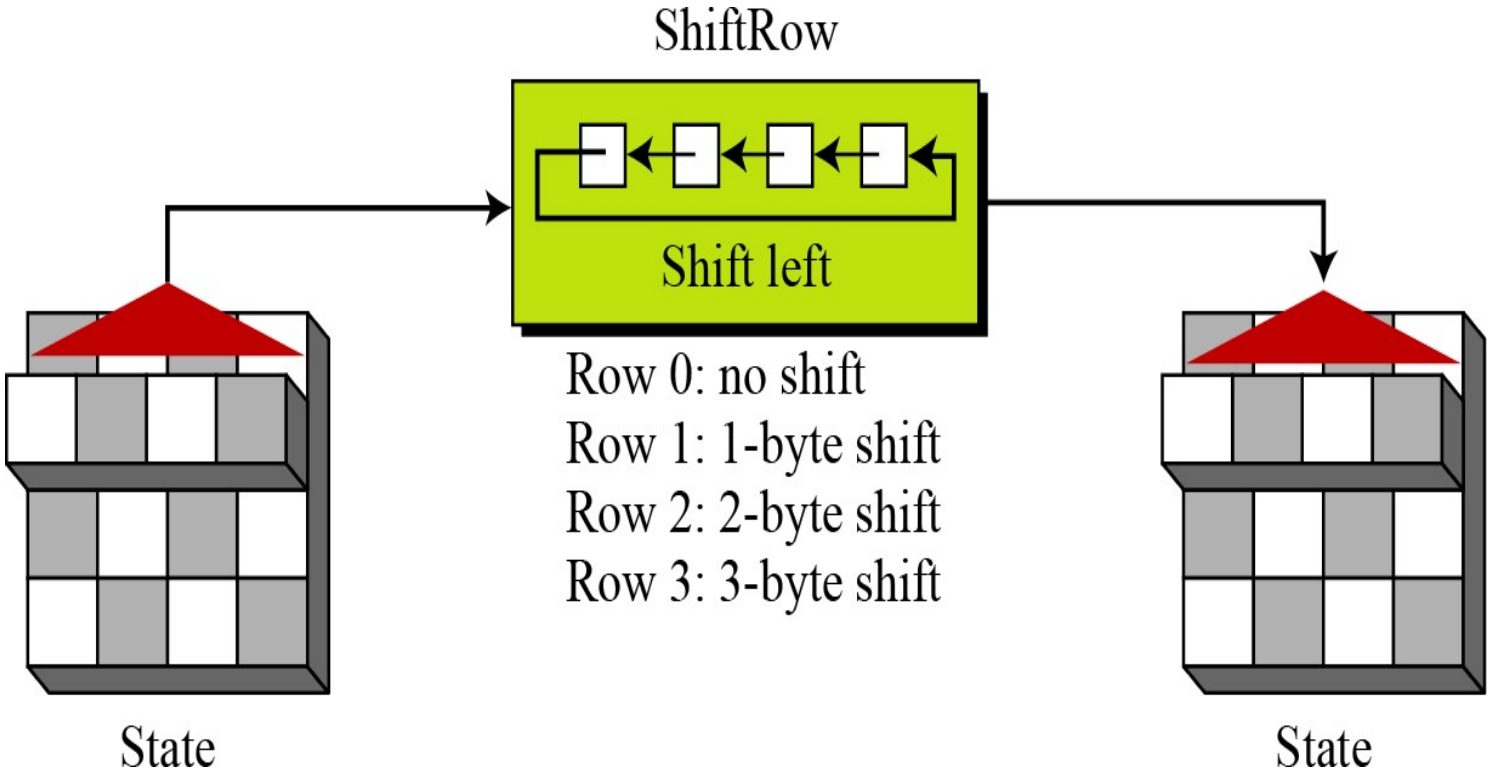
SubBytes and InvSubBytes Process



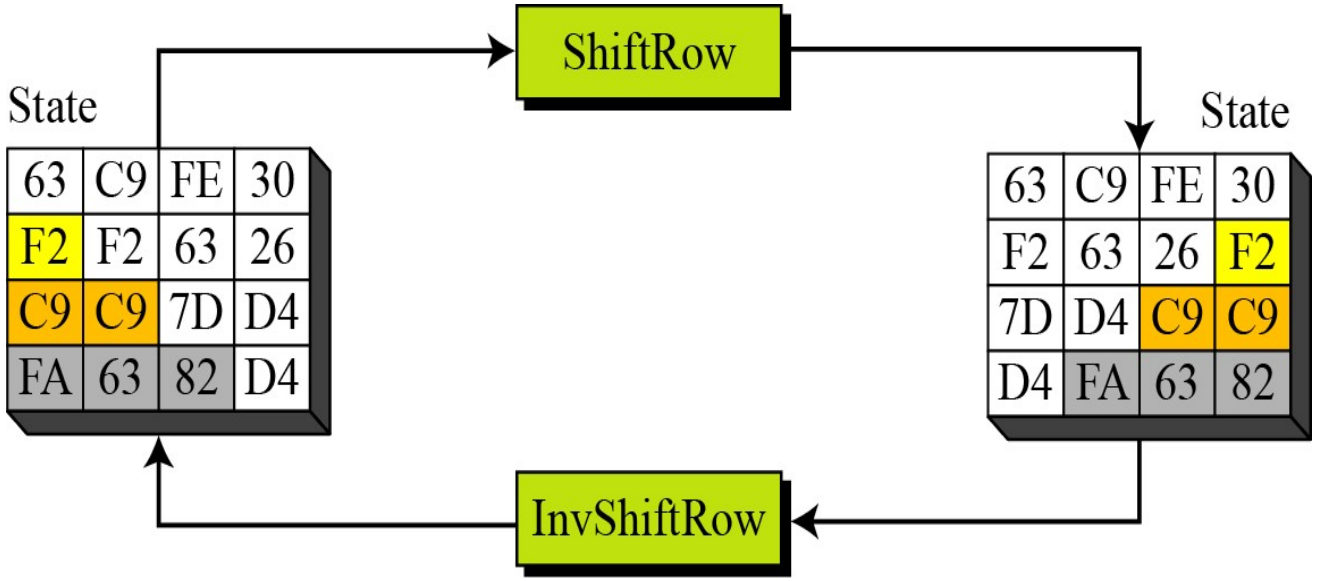


D Y PATIL
RAMRAO ADIK INSTITUTE OF TECHNOLOGY
NAVI MUMBAI

ShiftRows


ShiftRow
Shift left
Row 0: no shift
Row 1: 1-byte shift
Row 2: 2-byte shift
Row 3: 3-byte shift
State
State

ShiftRows/InvShiftRows



Mixing

- It is interbyte transformation which changes the bits inside a byte, based on the bits inside the neighboring bytes.
- Mixing bytes provide diffusion at the bit level.

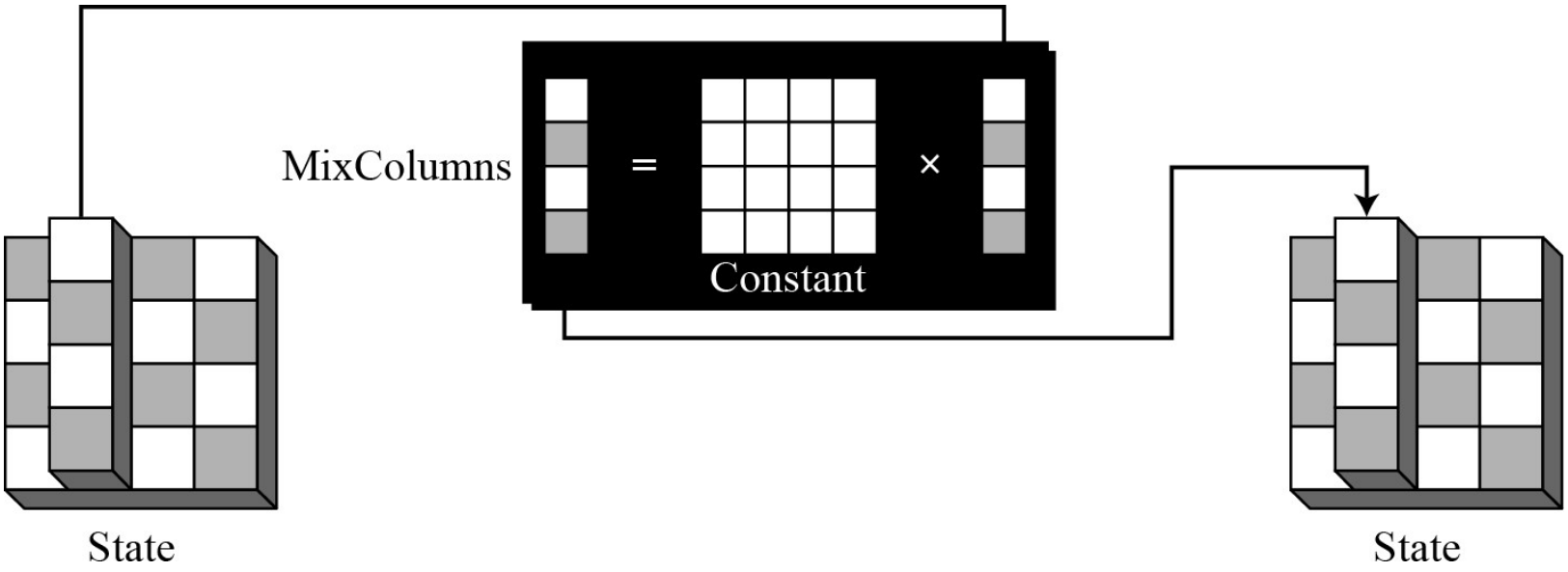
$$\begin{bmatrix} ax + by + cz + dt \\ ex + fy + gz + ht \\ ix + jy + kz + lt \\ mx + ny + oz + pt \end{bmatrix} = \begin{bmatrix} a & b & c & d \\ e & f & g & h \\ i & j & k & l \\ m & n & o & p \end{bmatrix} \times \begin{bmatrix} x \\ y \\ z \\ t \end{bmatrix}$$

New matrix | **Constant matrix** | Old matrix

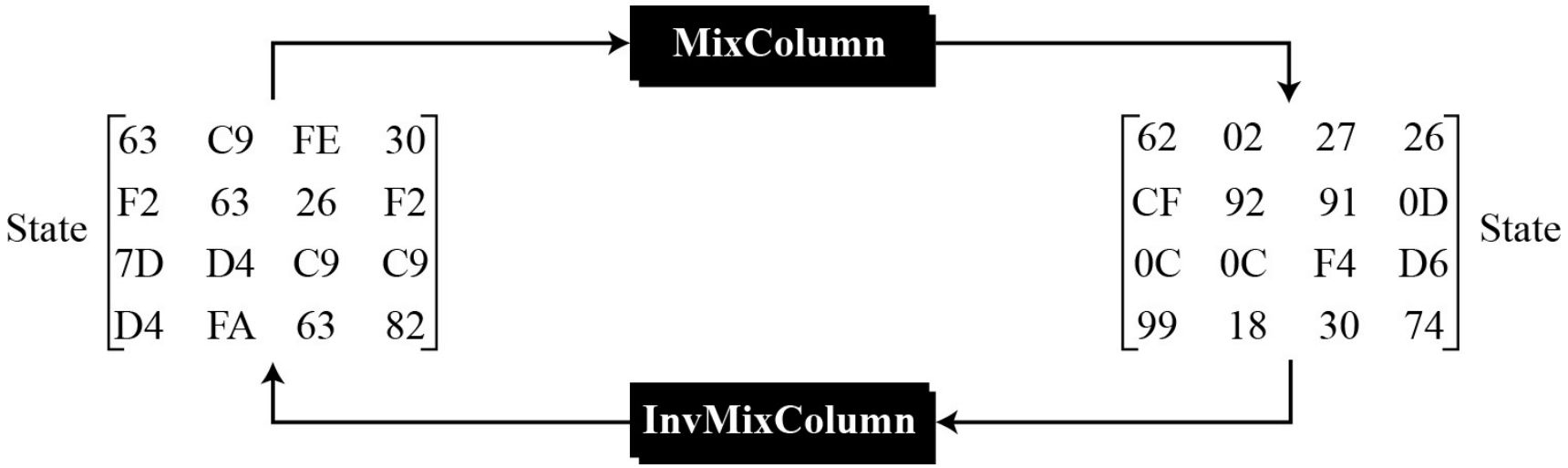
Mixing bytes using matrix multiplication

MixColumns

- It operates at the column level
- It transforms each columnof the state to a new column.

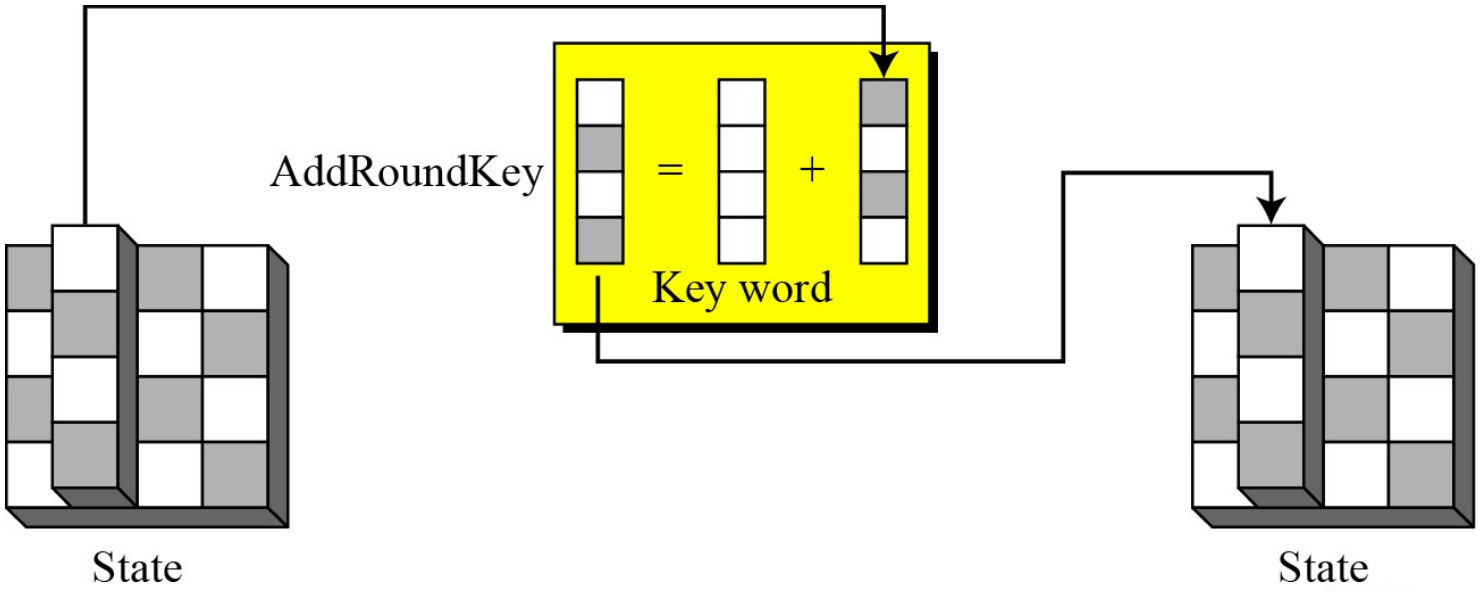


MixColumns/InvMixColumns

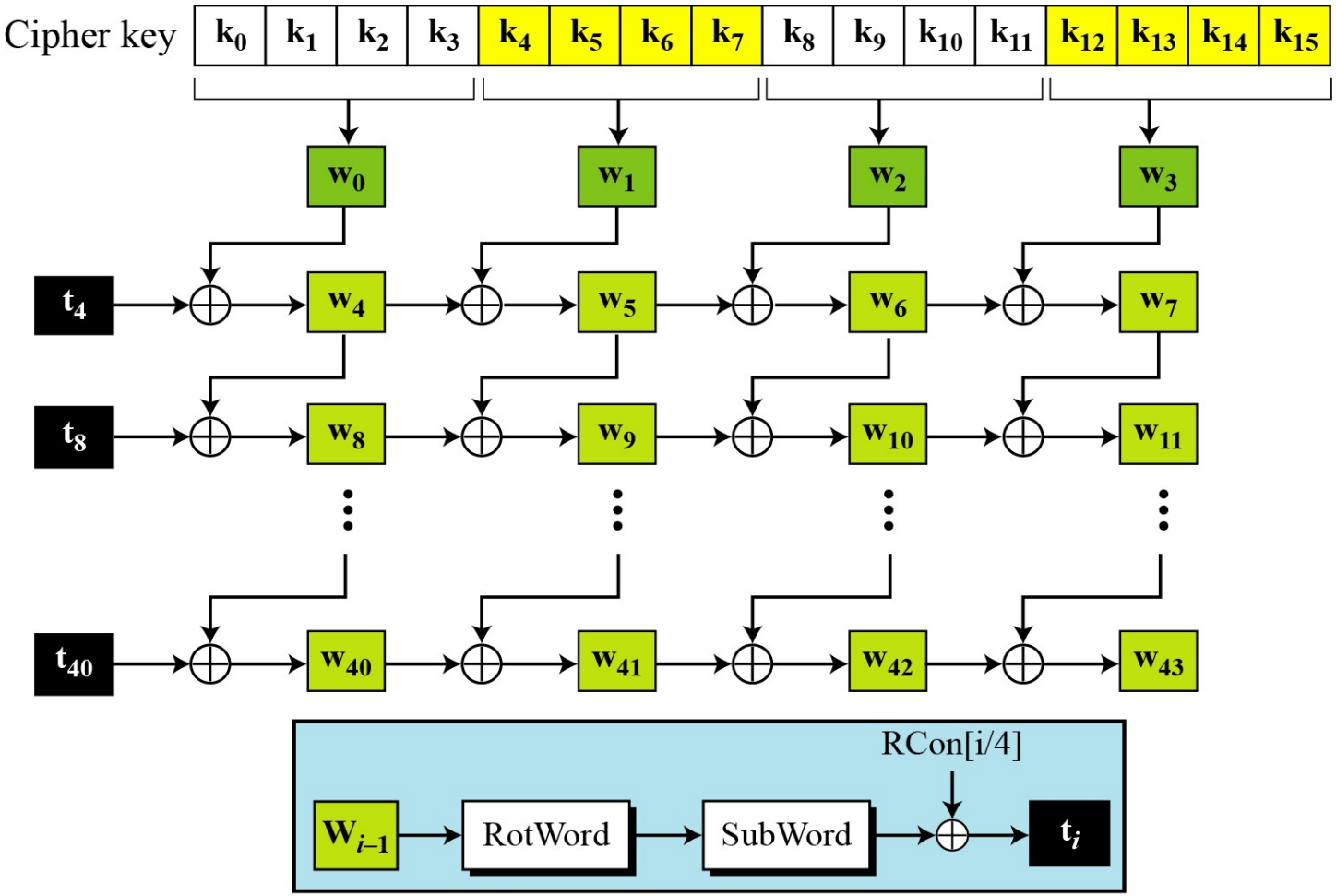


AddRoundKey

- It proceeds one column at a time.
- It adds a round key word with each state column matrix
- The operation in AddRoundKey is matrix addition.



Key Expansion in AES-128



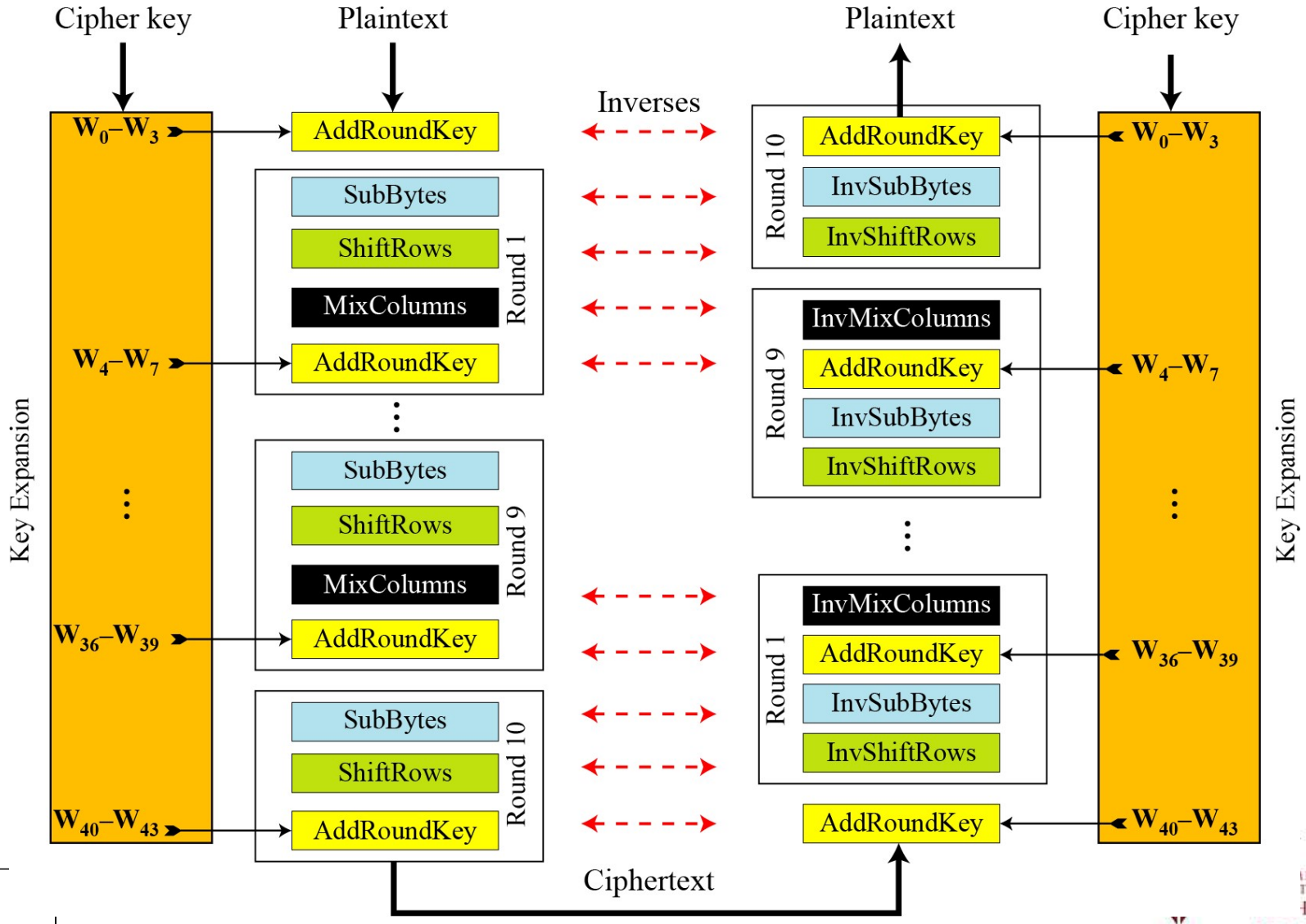
Making of t_i (temporary) words $i = 4\ N_r$

Key Expansion in AES-128

Table 7.4 *RCon constants*

Round	*Constant (RCon)*	*Round*	*Constant (RCon)*
1	$(\underline{\mathbf{01}}\ 00\ 00\ 00)_{16}$	6	$(\underline{\mathbf{20}}\ 00\ 00\ 00)_{16}$
2	$(\underline{\mathbf{02}}\ 00\ 00\ 00)_{16}$	7	$(\underline{\mathbf{40}}\ 00\ 00\ 00)_{16}$
3	$(\underline{\mathbf{04}}\ 00\ 00\ 00)_{16}$	8	$(\underline{\mathbf{80}}\ 00\ 00\ 00)_{16}$
4	$(\underline{\mathbf{08}}\ 00\ 00\ 00)_{16}$	9	$(\underline{\mathbf{1B}}\ 00\ 00\ 00)_{16}$
5	$(\underline{\mathbf{10}}\ 00\ 00\ 00)_{16}$	10	$(\underline{\mathbf{36}}\ 00\ 00\ 00)_{16}$

AES Cipher and Inverse Cipher



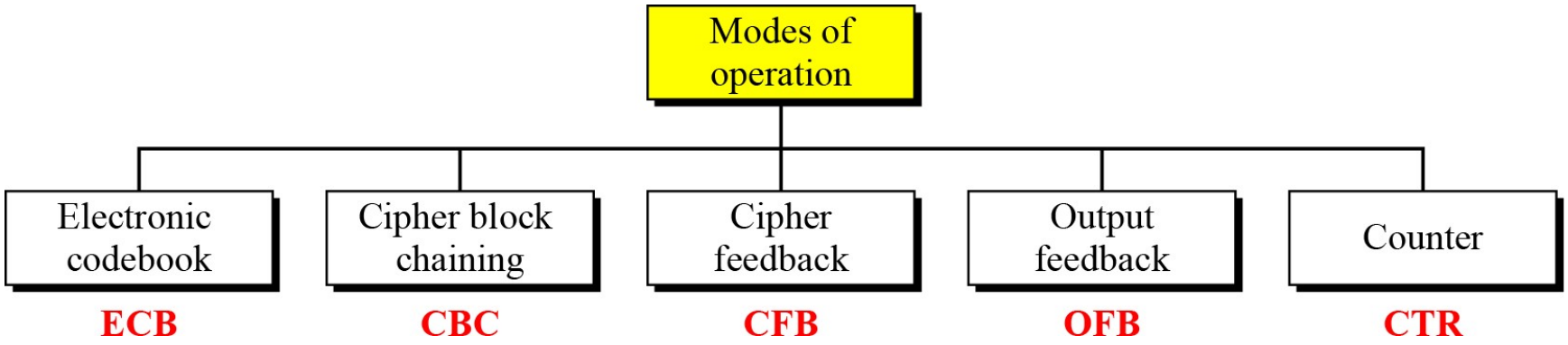
DES Vs AES

	DES	AES
Length	Key length is of 56 bits.	Key length varies from 128 bits, 192 bits to 256 bits.
Rounds of Operations	16 rounds of identical operations.	Rounds per key length: 128 bits - 10; 192 bits - 12; 256 bits - 14.
Network	DES structure is based on feistal network.	AES structure is based on substitution-permutation network.
Security	DES is weak and 3DES(Triple DES) is more secure than DES.	AES is de-facto world standard and is more secure than DES.
Rounds	Expansion, XOR operation with round key, Substitution and Permutation.	Byte substitution, Shift Row, Mix Column and Key Addition.
Size	DES can encrypt 64 bits of plain text.	AES can encrypt 128 bits of plain text.
Designed By	IBM	NIST
Known attacks	Brute-force, Linear crypt-analysis and Differential crypt-analysis.	No known attack

Modern Block Ciphers

- Symmetric-key encipherment can be done using modern block ciphers.
- Modes of operation have been devised to encipher text of any size employing either DES or AES.
  - ✓ Electronic Codebook (ECB) Mode
  - ✓ Cipher Block Chaining (CBC) Mode
  - ✓ Cipher Feedback (CFB) Mode
  - ✓ Output Feedback (OFB) Mode
  - ✓ Counter (CTR) Mode

Modern Block Ciphers

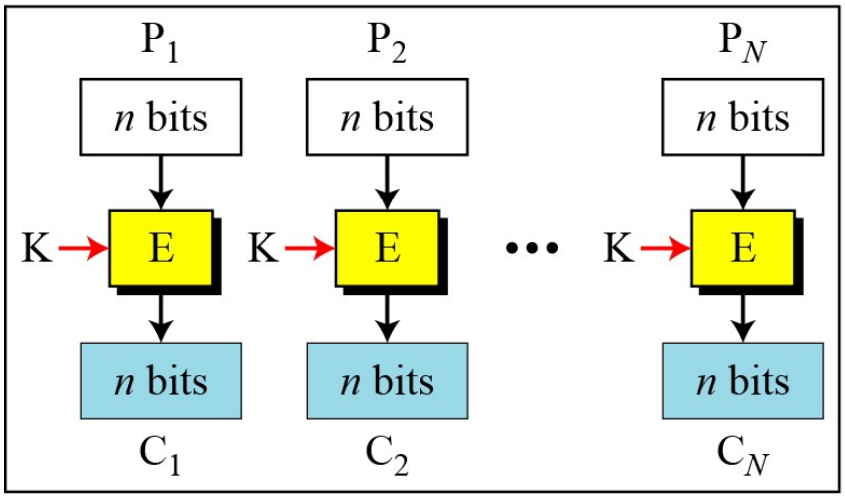


Electronic Codebook (ECB) Mode

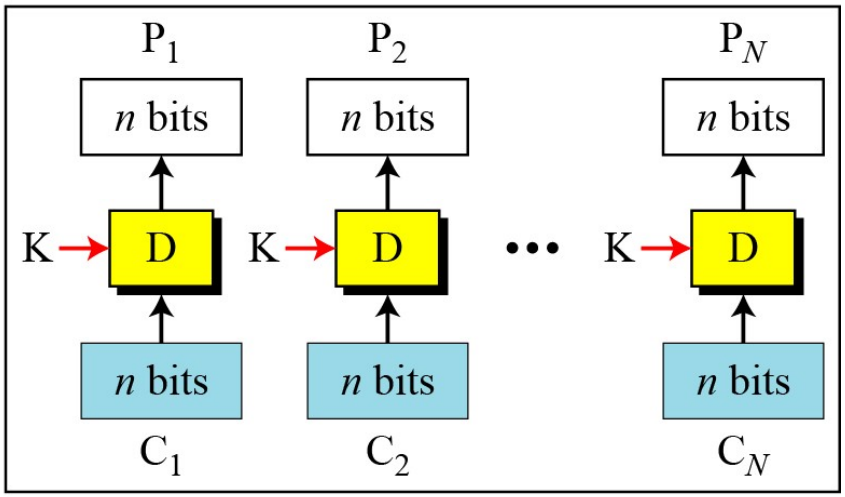
The simplest mode of operation is called the electronic codebook (ECB) mode.

Encryption: $C_i = E_K(P_i)$ Decryption: $P_i = D_K(C_i)$

E: Encryption D: Decryption
P_i: Plaintext block i C_i: Ciphertext block i
K: Secret key



Encryption



Decryption

Electronic Codebook (ECB) Mode

Error Propagation

A single bit error in transmission can create errors in several bits in the corresponding block. However, the error does not have any effect on the other blocks.

Algorithm 8.1 *Encryption for ECB mode*

```
ECB_Encryption (K, Plaintext blocks)
{
    for (i = 1 to N)
    {
        C_i ← E_K (P_i)
    }
    return Ciphertext blocks
}
```

Electronic Codebook (ECB) Mode

Ciphertext Stealing

- A technique called ciphertext stealing(CTS) can make it possible to use ECB mode without padding. In this technique the last two plaintext blocks, P_{N-1} and P_N , are encrypted differently and out of order, as shown below, assuming that P_{N-1} has n bits and P_N has m bits, where m ≤ n .

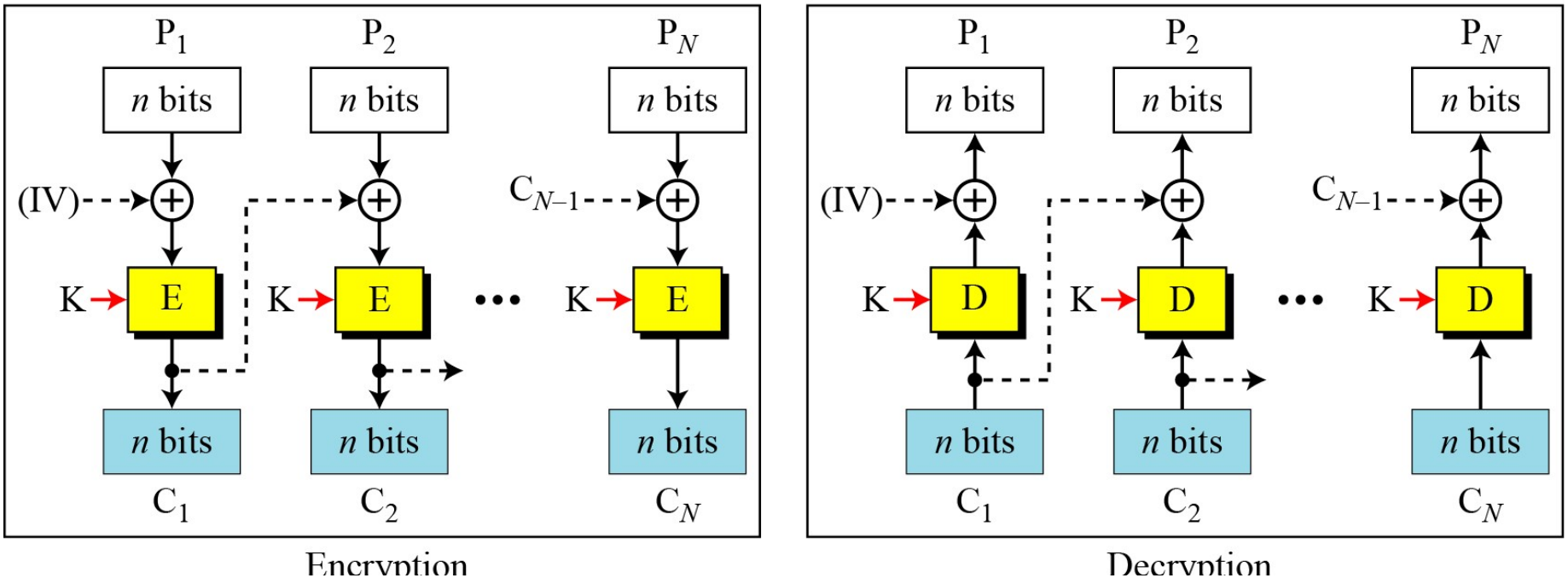
$$X = E_K(P_{N-1}) \rightarrow C_N = head_m(X)$$

$$Y = P_N \mid tail_{n-m}(X) \rightarrow C_{N-1} = E_K(Y)$$

Cipher Block Chaining (CBC) Mode

E: Encryption
P_i: Plaintext block i
K: Secret key

D : Decryption
C_i : Ciphertext block i
IV: Initial vector (C_0)



Encryption:

$C_0 = IV$

$C_i = E_K (P_i \oplus C_{i-1})$

Decryption:

$C_0 = IV$

$P_i = D_K (C_i) \oplus C_{i-1}$

PATIL RAMRAO ADIK INSTITUTE OF TECHNOLOGY NAVI MUMBAI

Cipher Block Chaining (CBC) Mode

Error Propagation

In CBC mode, a single bit error in ciphertext block C_j during transmission may create error in most bits in plaintext block P_j during decryption.

Algorithm 8.2 *Encryption algorithm for ECB mode*

```
CBC_Encryption (IV, K, Plaintext blocks)
{
    C0 ← IV
    for (i = 1 to N)
    {
        Temp ← Pi ⊕ Ci−1
        Ci ← EK (Temp)
    }
    return Ciphertext blocks
}
```

Cipher Block Chaining (CBC) Mode

Ciphertext Stealing

The ciphertext stealing technique described for ECB mode can also be applied to CBC mode, as shown below.

$$\begin{array}{lcllcl} U = P_{N-1} \oplus C_{N-2} & \rightarrow & X = E_K(U) & \rightarrow & C_N & = head_m(X) \\ V = P_N \mid pad_{n-m}(0) & \rightarrow & Y = X \oplus V & \rightarrow & C_{N-1} & = E_K(Y) \end{array}$$

The head function is the same as described in ECB mode; the pad function inserts 0's.

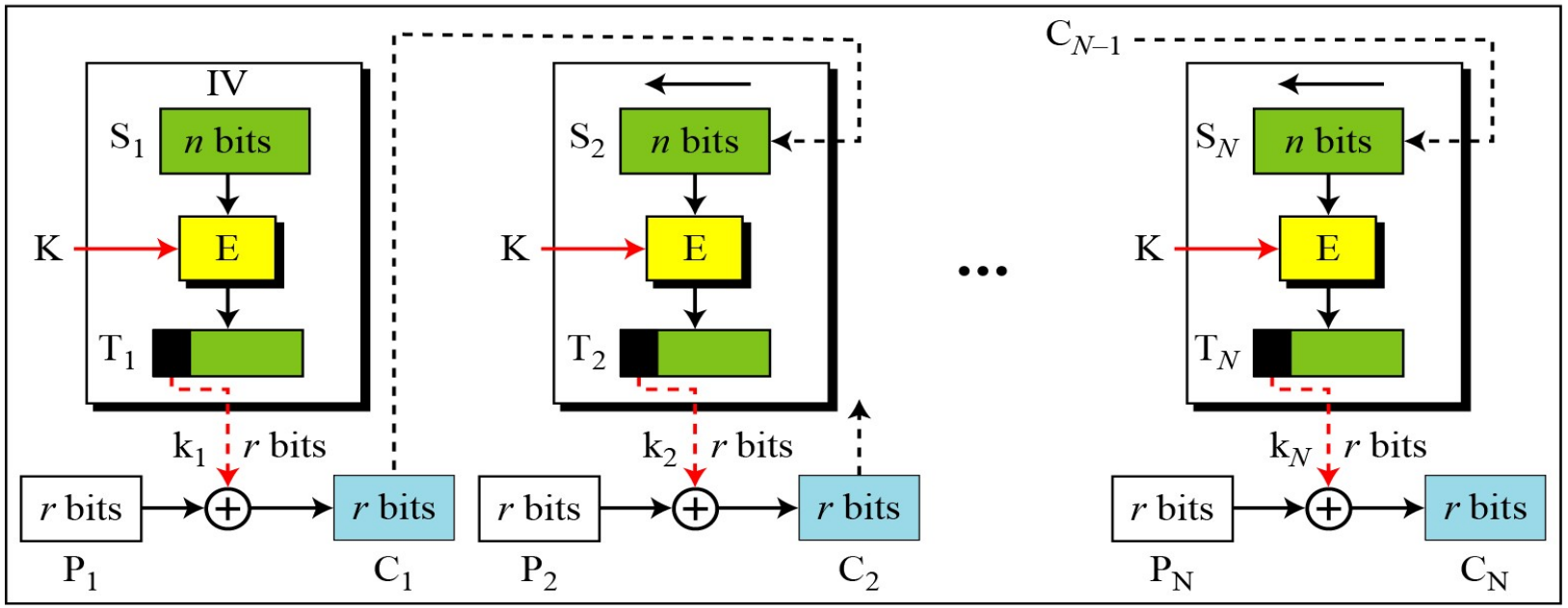
Cipher Feedback(CFB) Mode

In some situations, we need to use DES or AES as secure ciphers, but the plaintext or ciphertext block sizes are to be smaller.

E : Encryption
P_i: Plaintext block i
K: Secret key

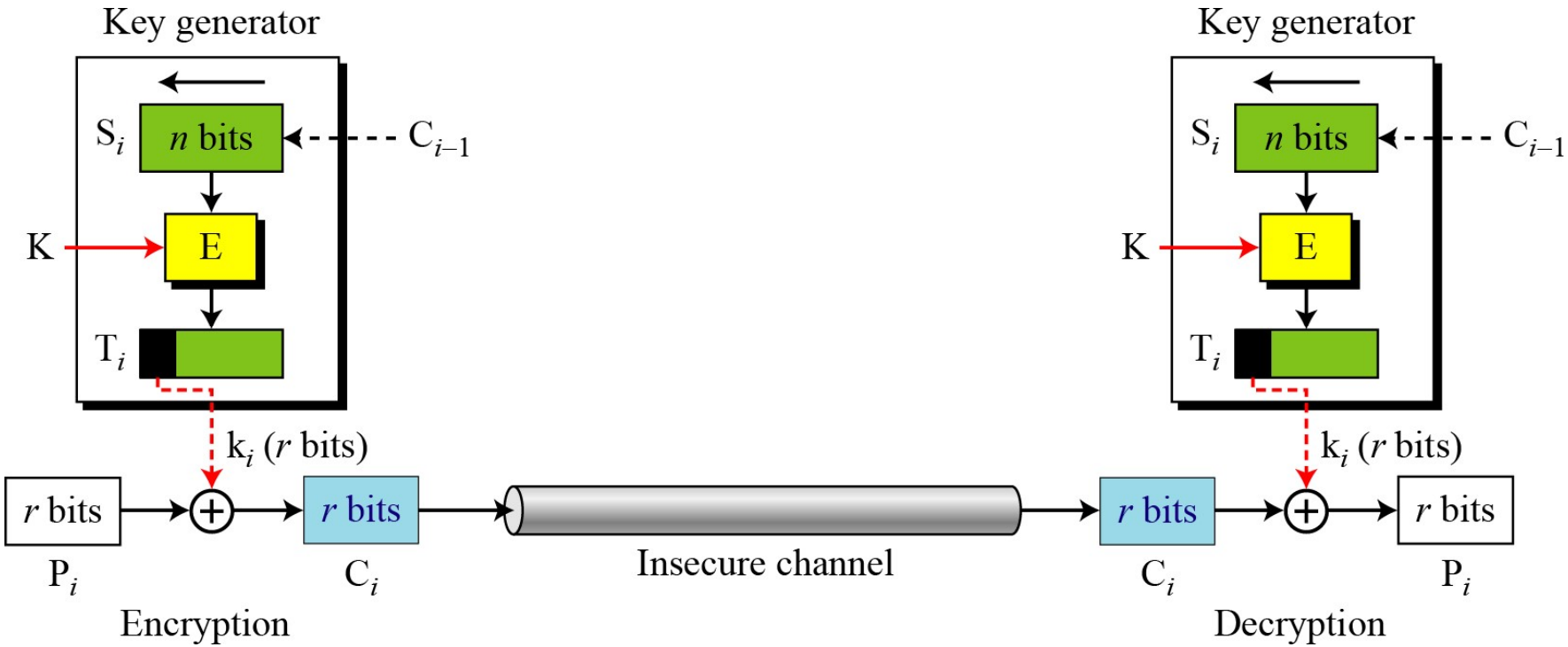
D : Decryption
C_i: Ciphertext block i
IV: Initial vector (S_1)

S_i: Shift register
T_i: Temporary register



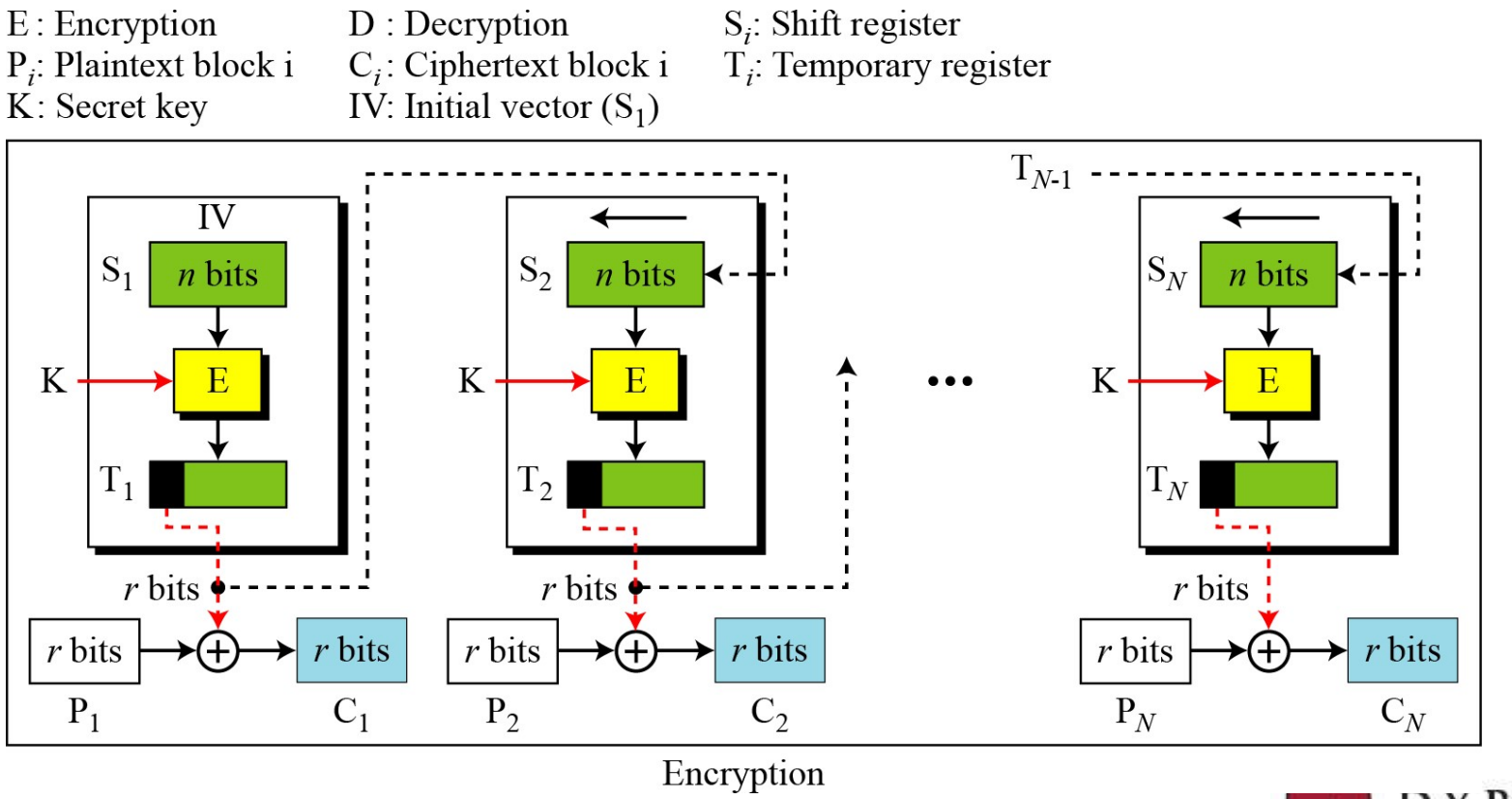
Encryption

CFB as a Stream Cipher

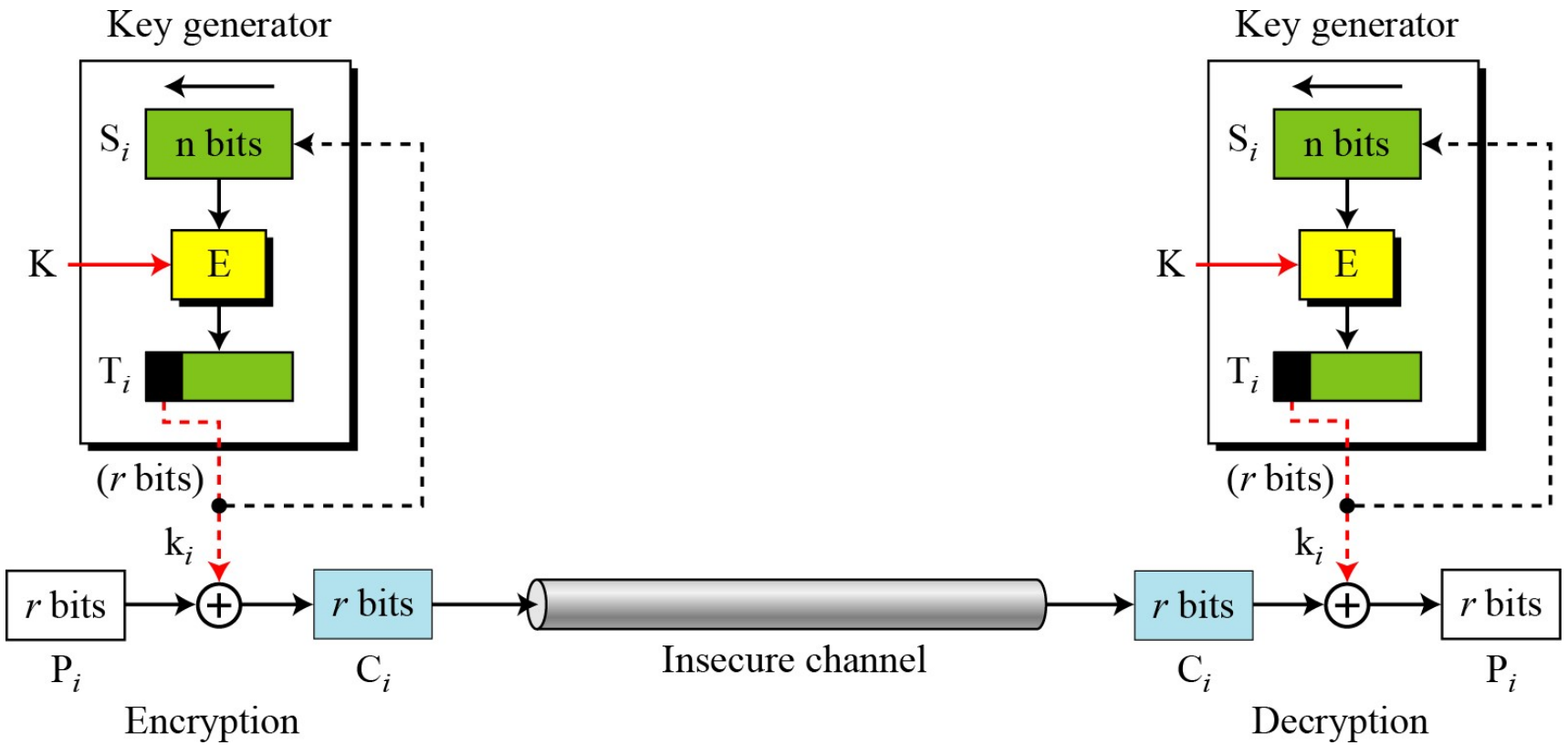


Output Feedback(OFB) Mode

In this mode each bit in the ciphertext is independent of the previous bit or bits. This avoids error propagation.



OFB as a Stream Cipher



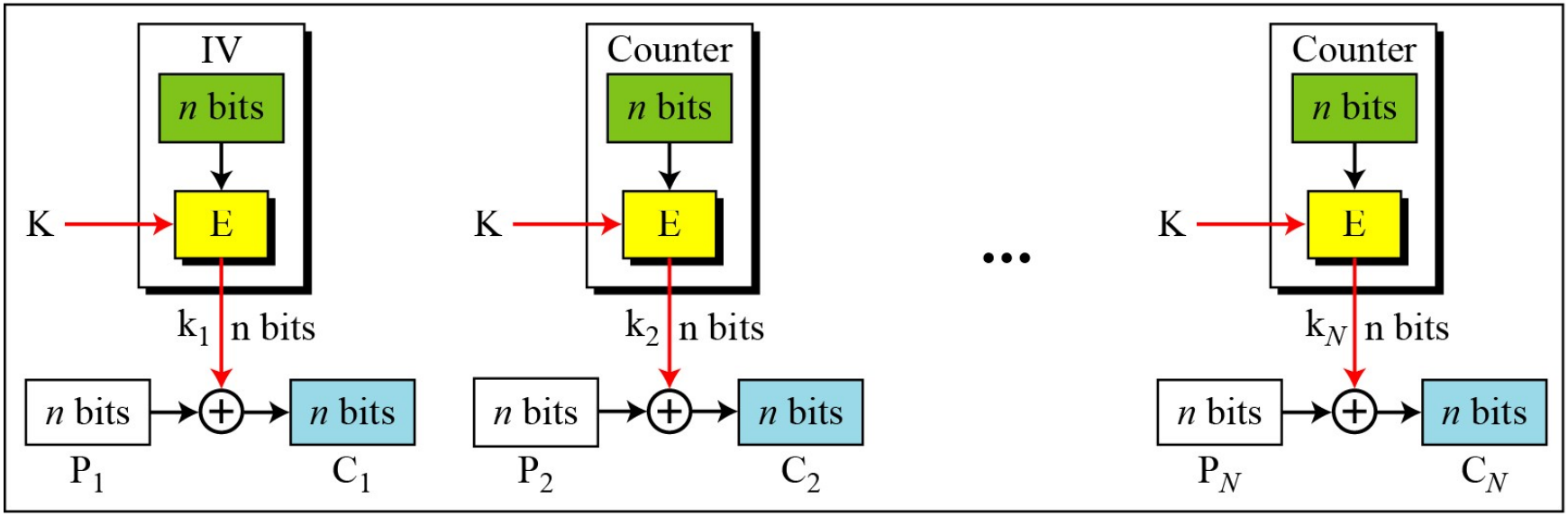
Counter(CTR) Mode

In the counter (CTR) mode, there is no feedback. The pseudorandomness in the key stream is achieved using a counter.

E : Encryption
P_i : Plaintext block *i*
K : Secret key

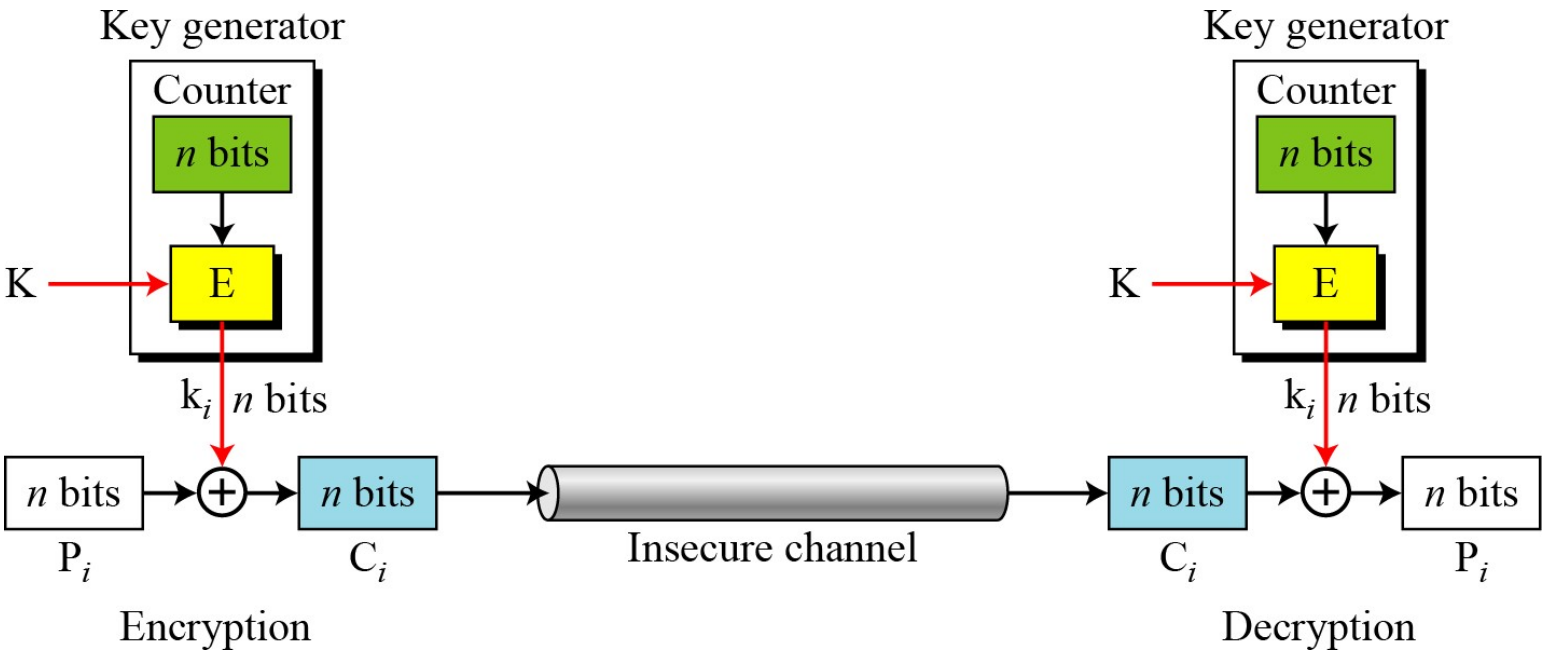
IV: Initialization vector
C_i: Ciphertext block *i*
k_i : Encryption key *i*

The counter is incremented for each block.



Encryption

Counter(CTR) Mode



Comparison of Modes of Cipher

Table 8.1 *Summary of operation modes*

Operation Mode	*Description*	*Type of Result*	*Data Unit Size*
ECB	Each n-bit block is encrypted independently with the same cipher key.	Block cipher	n
CBC	Same as ECB, but each block is first exclusive-ored with the previous ciphertext.	Block cipher	n
CFB	Each r-bit block is exclusive-ored with an r-bit key, which is part of previous cipher text	Stream cipher	$r \leq n$
OFB	Same as CFB, but the shift register is updated by the previous r-bit key.	Stream cipher	$r \leq n$
CTR	Same as OFB, but a counter is used instead of a shift register.	Stream cipher	n

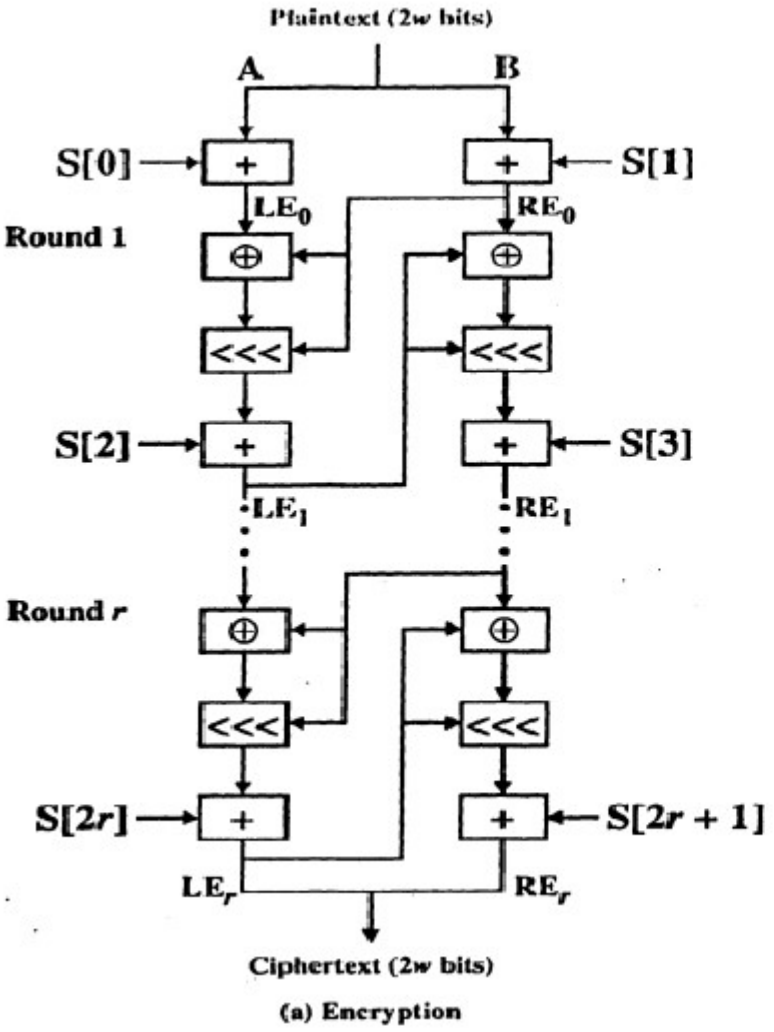
RC5

- RC-Rivest Cipher - Stream Cipher
- It is a proprietary cipher owned by RSADSI
- RC5 can vary key size / data size / no. of rounds
- RC5 is a family of ciphers RC5-w/r/b
- w = word size in bits (16/32/64)
- r = number of rounds (0..255)
- b = number of bytes in key (0..255)

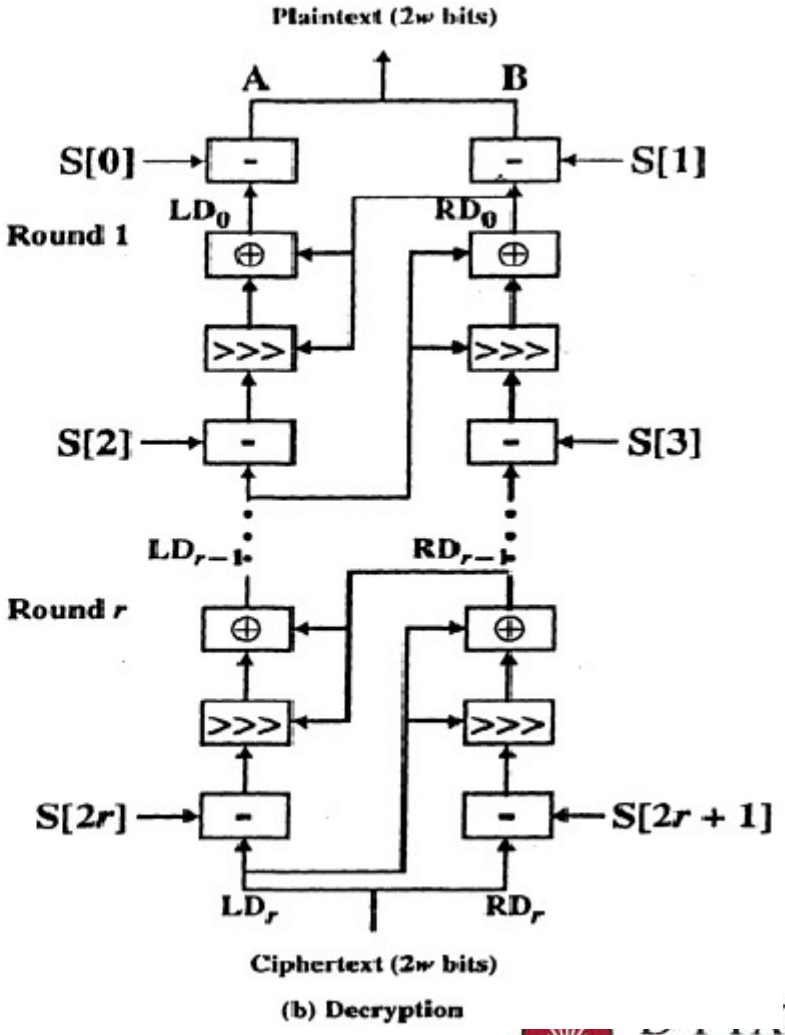
RC5 Key Schedule

- RC5 uses 2r subkey words (w-bits)
- subkeys are stored in array S[i], i=0..t-1
- key schedule consists of
  - initializing S to a fixed pseudorandom value, based on constants e and phi
  - the byte key is copied (little-endian) into a c-word array L
  - a mixing operation then combines L and S to form the final S array

RC5



(a) Encryption



(b) Decryption

RAMRAO ADIK INSTITUTE OF TECHNOLOGY
NAVI MUMBAI

RC5 Encryption

- Split input into two halves A & B

$LE_0 = A + S[0];$
$RE_0 = B + S[1];$
for i = 1 to r do
$LE_i = ((LE_{i-1} \text{ XOR } RE_{i-1}) <<< RE_{i-1}) + S[2 \times i];$
$RE_i = ((RE_{i-1} \text{ XOR } LE_i) <<< LE_i) + S[2 \times i + 1];$

- Each round is like 2 DES rounds
- It need reasonable number of rounds (e.g. 12-16)

RC5 Modes

- RFC2040 defines 4 modes used by RC5
- RC5 Block Cipher, is ECB mode
- RC5-CBC, is CBC mode
- RC5-CBC-PAD, is CBC with padding by bytes with value being the number of padding bytes
- RC5-CTS, a variant of CBC which is the same size as the original message, uses ciphertext stealing to keep size same as original

Thank You

D Y PATIL
RAMRAO ADIK
INSTITUTE OF
TECHNOLOGY
NAVI MUMBAI