# UNIT-3

- Stream ciphers and block ciphers,
- Block Cipher structure,
- Data Encryption standard (DES) with example,
- Strength of DES,
- Design principles of block cipher,
- Introduction to AES and its operations with structure,
- Its transformation functions,
- key expansion,
- example and implementation

Stream ciphers

- A **stream cipher** is one that encrypts a digital data stream one bit or one byte at a time.

- Examples of classical stream ciphers are Autokeyed Vigenère cipher ,A5/1, RC4 and Vernam cipher.
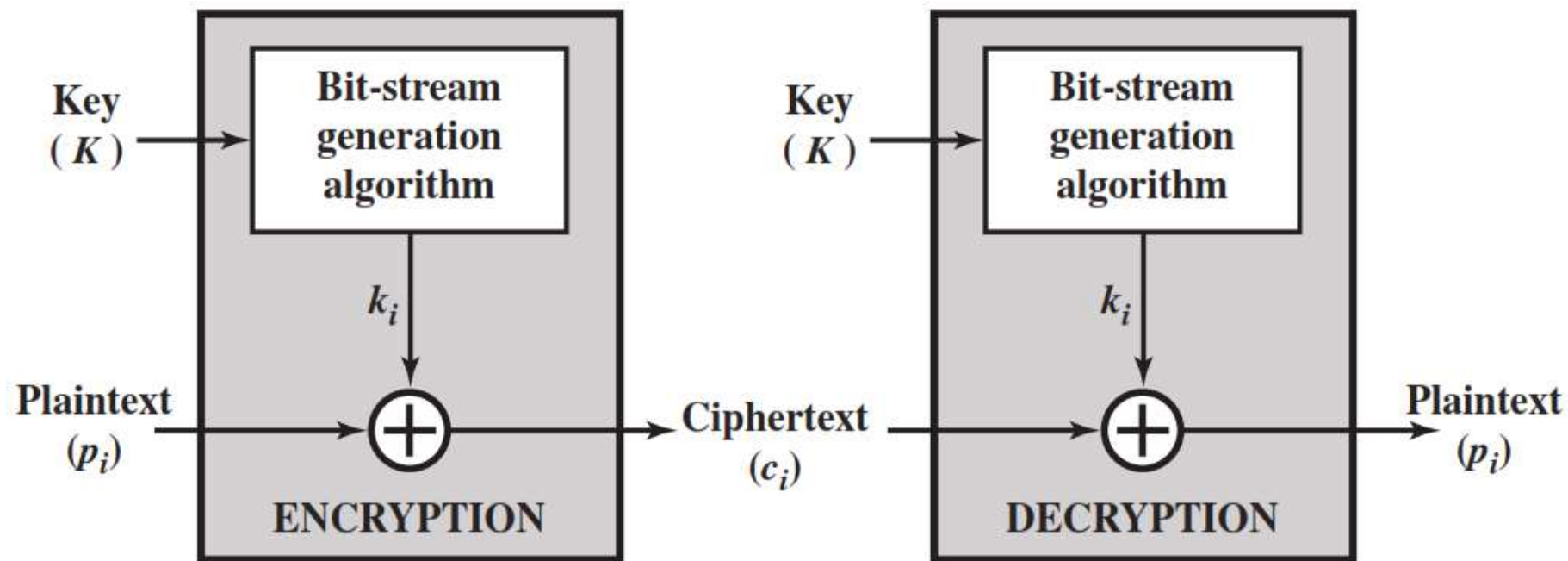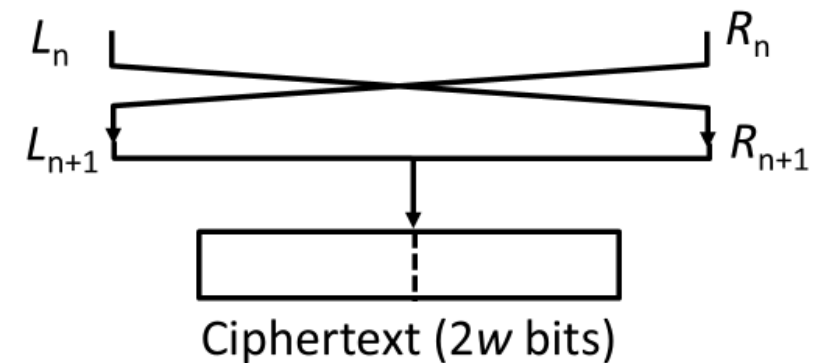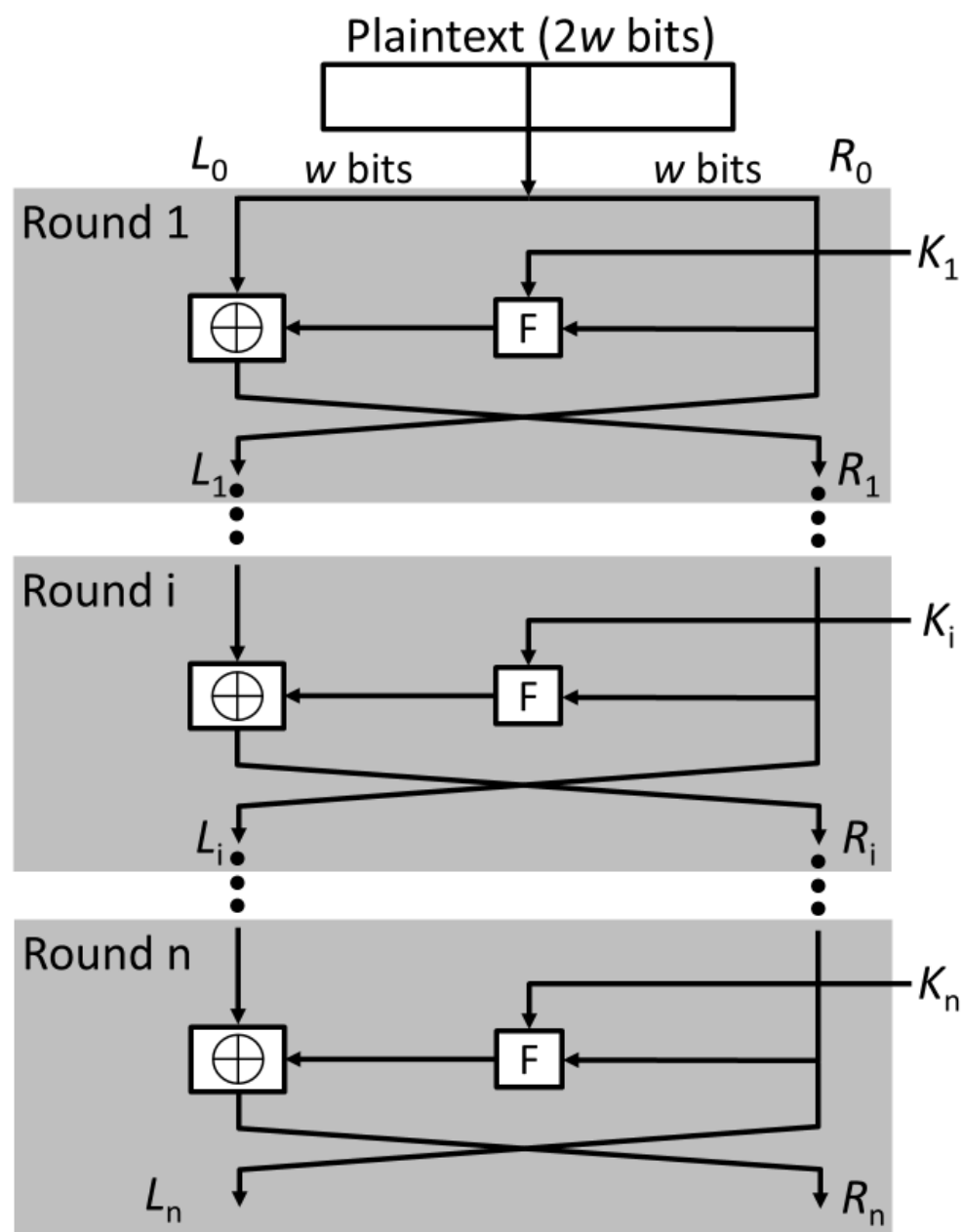
Block Cipher

- A **block cipher** is one in which a block of plaintext is treated as a whole and used to produce a ciphertext block of equal length.

- Typically, a block size of **64 or 128** bits is used.

- Examples are Feistel Cipher, DES, Triple DES and AES
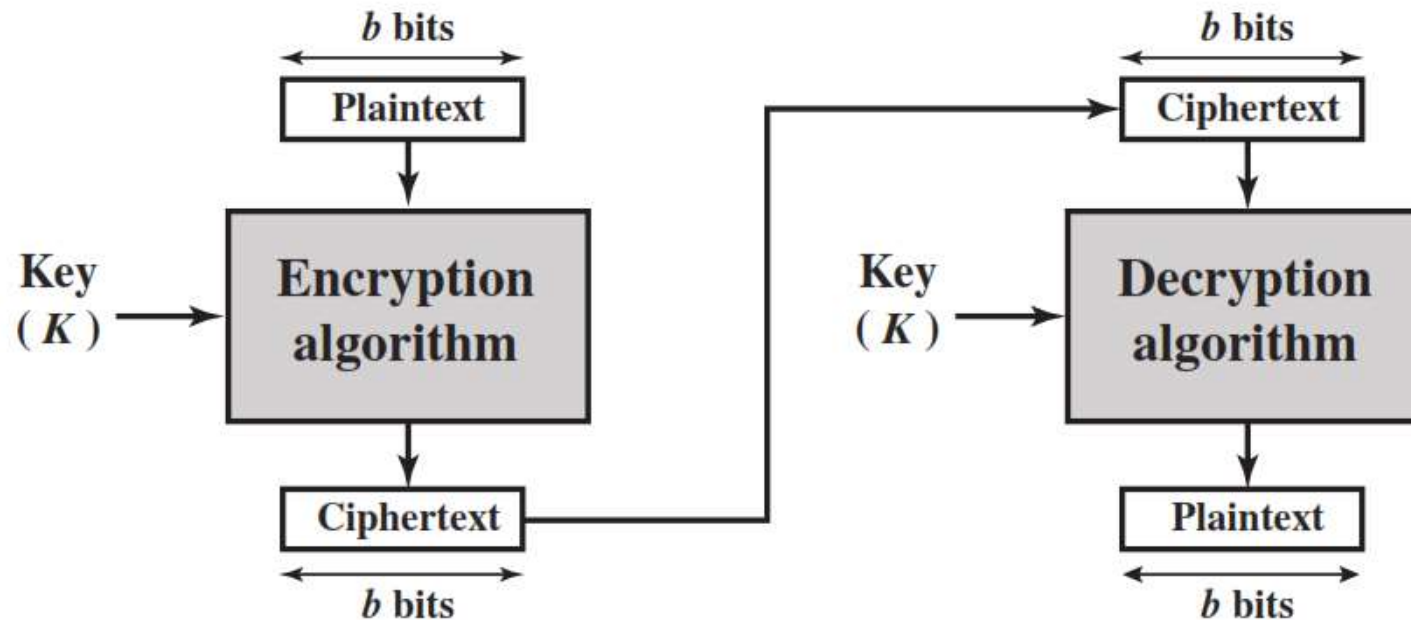
Diffusion and Confusion

- **Diffusion** hides the relationship between the ciphertext and the plaintext.

- This is achieved by having each plaintext digit affect the value of many ciphertext digits.

- **Confusion** hides the relationship between the ciphertext and the key.

- This is achieved by the use of a complex substitution algorithm.

Feistel Cipher Structure
Or Block Cipher Structure

Plaintext ($2w$ bits)

L_0 w bits w bits R_0

Round 1 K_1

L_1 R_1

Round i K_i

L_i R_i

Round n K_n

L_n R_n

L_n R_n

L_{n+1} R_{n+1}

Ciphertext ($2w$ bits)

Feistel Cipher Structure

- Input plaintext block of length 2w bits

- key K = n bits , Sub-keys: K_1, K_2, …, K_n (Derived from K)

- All rounds have the same structure.

- A **substitution** is performed by taking exclusive-OR on left half(Li) of the data and the output of round function F which has inputs right half(Ri) and sub key ki.

- A **permutation** is performed that consists of interchange of two halves of data.

- This structure is called **Substitution-Permutation Network** (SPN)

Feistel Network Factors

- **Block size:** Common block size of 64-bit. However, the new algorithms uses a 128-bit, 256-bit block size.

- **Key size:** Key sizes of 64 bits or less are now widely considered to be insufficient, These days at least 128 bit, more better, e.g. 192 or 256 bit

- **Number of rounds:** A typical size is 16 rounds.

- **Round function F:** Again, greater complexity generally means greater resistance to cryptanalysis.

- **Subkey generation algorithm:** Greater complexity in this algorithm should lead to greater difficulty of cryptanalysis.

Feistel Encryption & Decryption
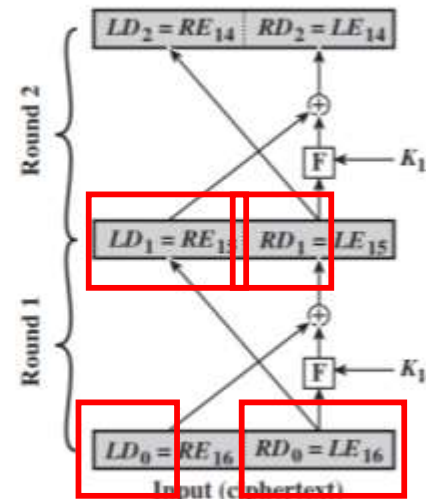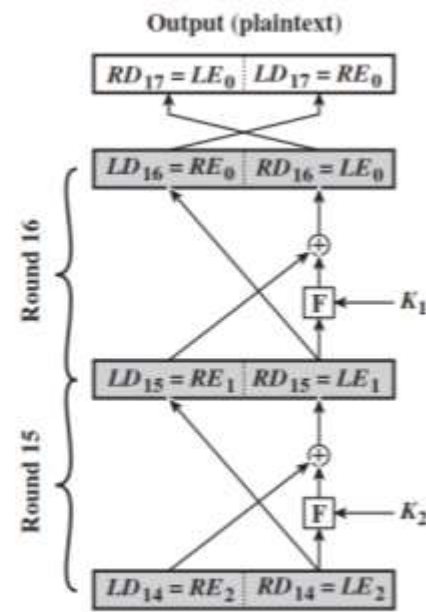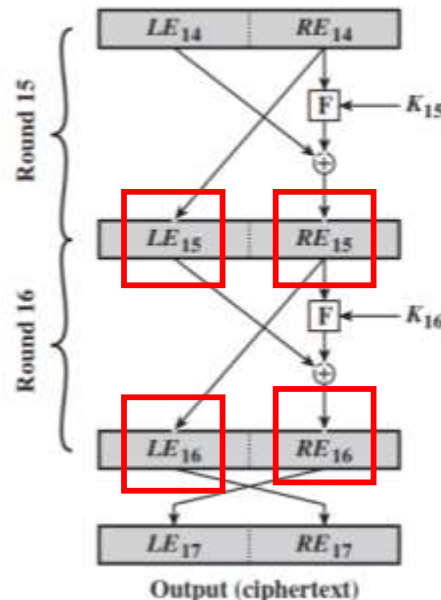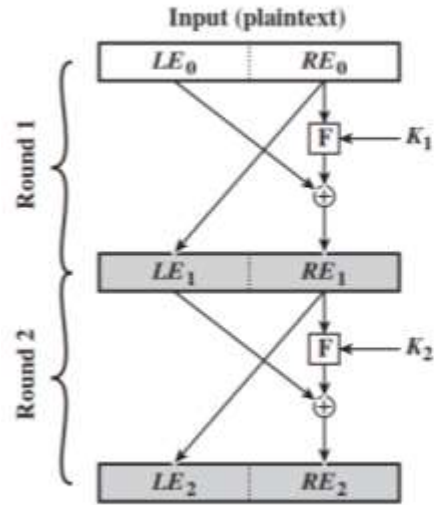
- Prove that o/p of first round of Decryption is equal to 32-bit swap of i/p of 16th round of Encryption
- $LD_1 = RE_{15}$ & $RD_1 = LE_{15}$
- On Encryption Side:

  $$LE_{16} = RE_{15}$$

  $$RE_{16} = LE_{15} \oplus F(RE_{15}, K_{16})$$

- On Decryption Side:

  $$LD_1 = RD_0 = LE_{16} = RE_{15}$$

  $$RD_1 = LD_0 \oplus F(RD_0, K_{16})$$

  $$= RE_{16} \oplus F(RE_{15}, K_{16})$$

  $$= [LE_{15} \oplus F(RE_{15}, K_{16})] \oplus F(RE_{15}, K_{16})$$

  Thus,
  $$LD_1 = RE_{15} \ \& \ RD_1 = LE_{15}$$

Data Encryption Standard (DES)

- Type: Block Cipher
- Block Size : 64-bit
- Key Size: 64-bit, with only 56-bit effective
- Number of Rounds: 16

DES Encryption Algorithm
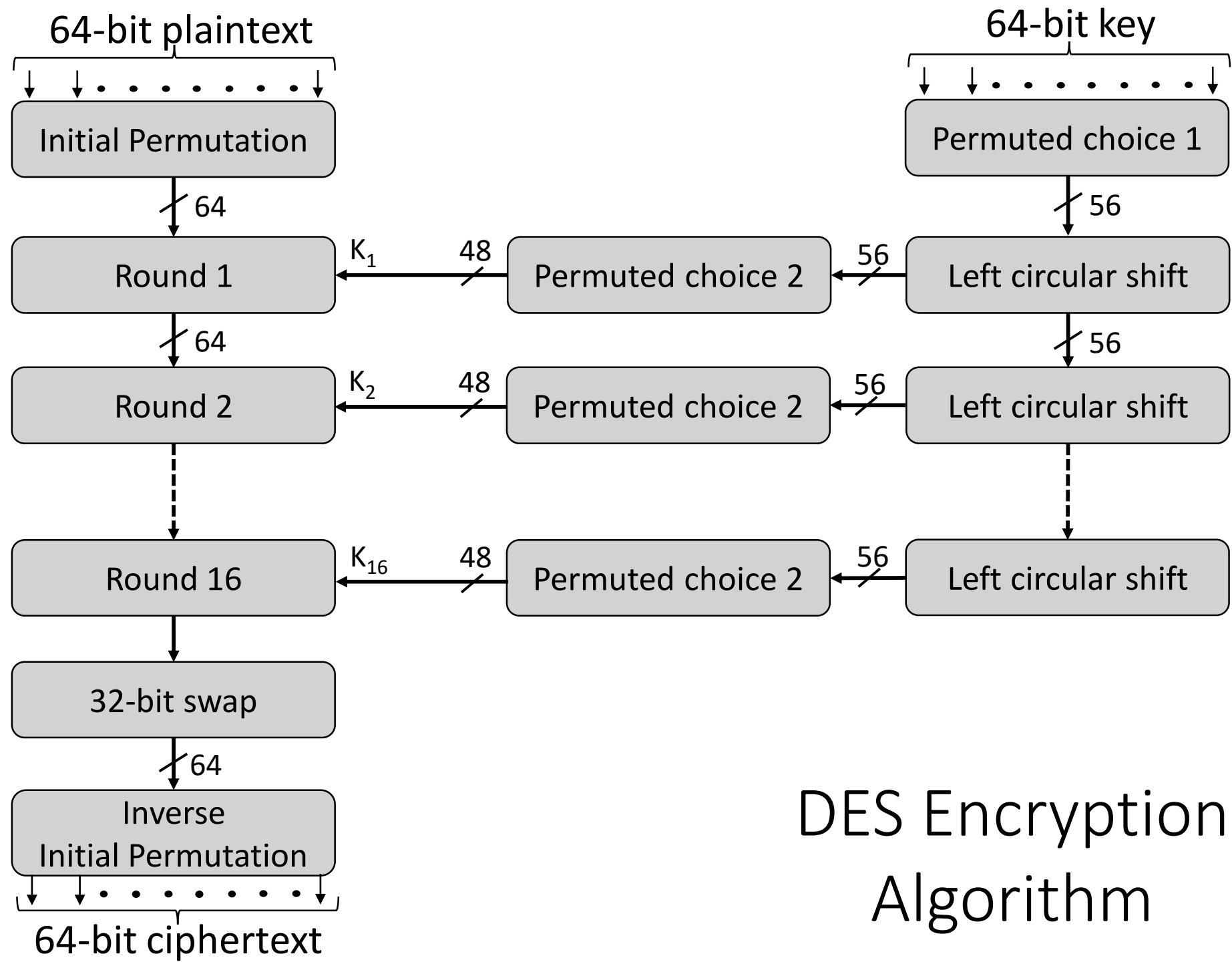
DES Encryption Algorithm (Cont…)

- First, the 64-bit plaintext passes through an **initial permutation** (IP) that rearranges the bits to produce the permuted input.

- This is followed by a phase consisting of sixteen rounds of the same function, which involves both **permutation** and **substitution** functions.

- Finally, the preoutput is passed through a permutation that is the **inverse of the initial permutation** function, to produce the 64-bit ciphertext.

- The 56-bit key is passed through a **permutation function**.

- For each of the sixteen rounds, a subkey (K_i) is produced by the combination of a **left circular shift** and a **permutation**.

DES Single Round

32-bits	32-bits	28-bits	28-bits
L_{i-1}	R_{i-1}	C_{i-1}	D_{i-1}

Expansion/ permutation (E table)

48

XOR

K_i

48

48

Left Shift (S)

Left Shift (S)

Permutation/ compression (Permuted choice 2)

Substitution/choice (S-box)

32

Permutation (P)

32

XOR

L_i

R_i

C_i

D_i
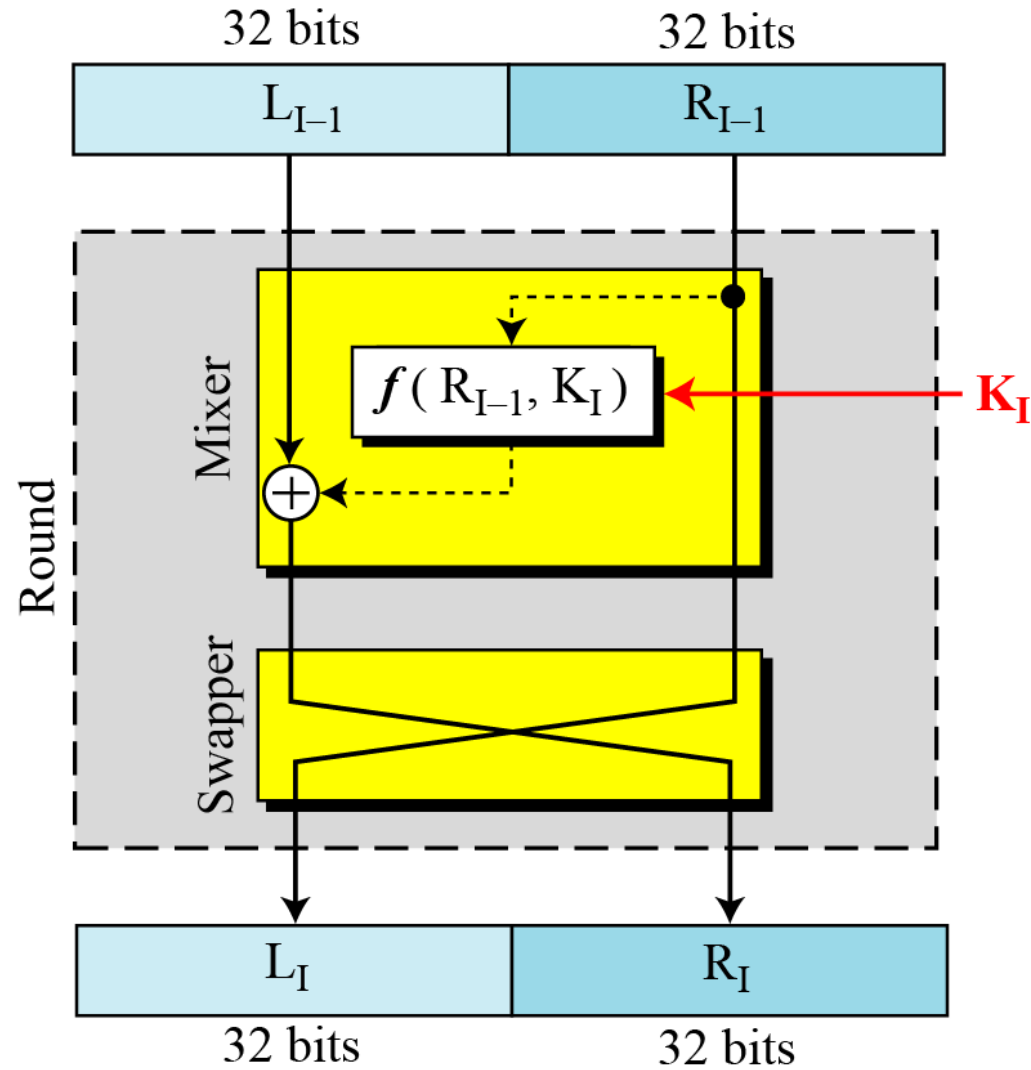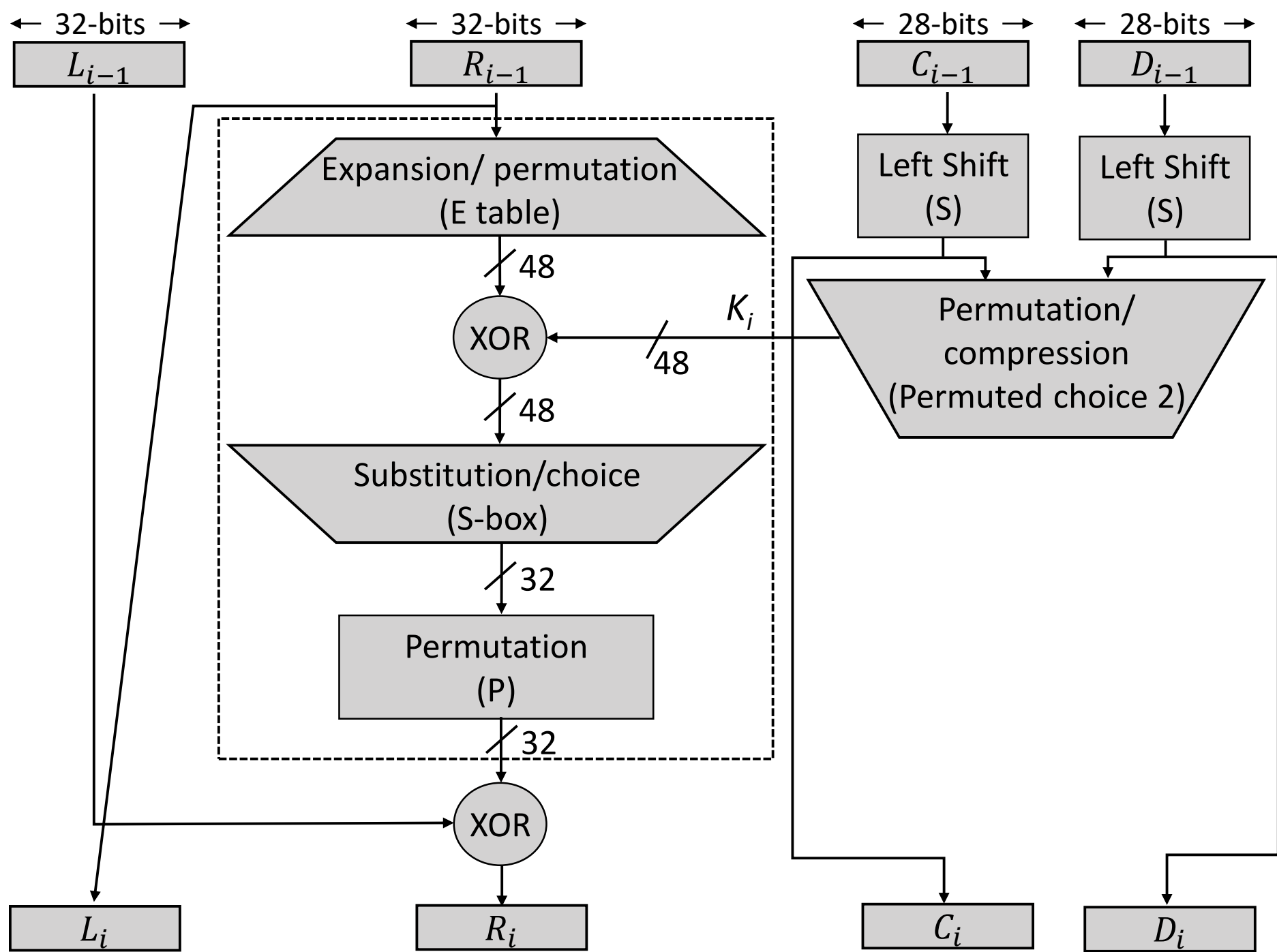
DES Single Round (Cont…)

1. Key Transformation
   - Permutation of selection of sub-key from original key

2. Expansion Permutation (E-table)
   - Right half is expanded from 32-bits to 48-bits

3. S-box Substitution
   - Accepts 48-bits from XOR operation and produce 32-bits using 8 substitution boxes (each S-boxes has a 6-bit i/p and 4-bit o/p).

4. P-Box Permutation

5. XOR and Swap

Role of S-box
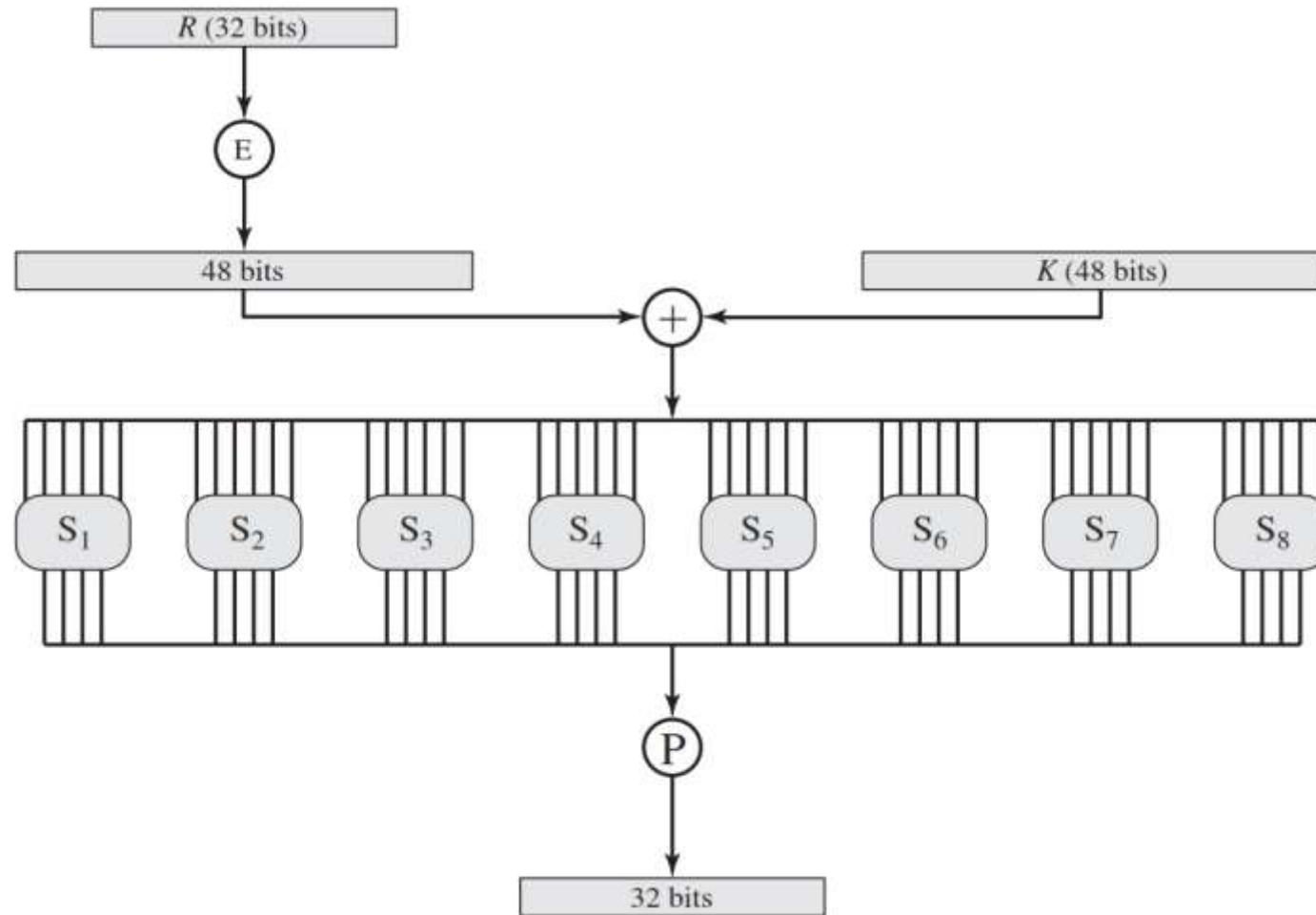
Role of S-box (Cont…)

- The outer two bits of each group select one row of an S-box.
- Inner four bits selects one column of an S-box.

	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

S-box 1

- Example:

Input | 0 | 1 | 1 | 0 | 0 | 1 |

Output | 1 | 0 | 0 | 1 |

Row

Column
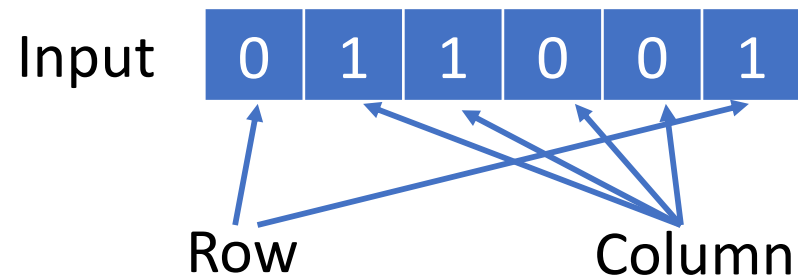
Avalanche Effect

- Desirable property of any encryption algorithm is that a change in one bit of the plaintext or of the key should produce a change in many bits of cipher text.

- DES performs strong **avalanche effect**.

Plaintext: 0000000000000000   Key: 22234512987ABB23
Ciphertext: 4789FD476E82A5F1

Plaintext: 000000000000000**1**   Key: 22234512987ABB23
Ciphertext: 0A4ED5C15A63FEA3

- Although the two plaintext blocks differ only in the rightmost bit, the ciphertext blocks differ in 29 bits.

- This means that changing approximately 1.5 % of the plaintext creates a change of approximately 45 % in the ciphertext.

AES (Advanced Encryption Standard)

- The Rijndael proposal for AES defined a cipher in which the block length and the key length can be independently specified to be 128, 192, or 256 bits.

Key size (words/ bytes/ bits)	4/16/128	6/24/192	8/32/256
Block size (words/ bytes/ bits)	4/16/128	4/16/128	4/16/128
Round key size (words/ bytes/ bits)	4/16/128	4/16/128	4/16/128
Number of Rounds	10	12	14

- AES designed to have characteristics
    1. Resistance against all known attacks
    2. Speed and code compactness on a wide range of platforms
    3. Design simplicity

AES (Advanced Encryption Standard)

Plaintext (128 bits)

AES

Key (128-256 bits)

Ciphertext (128 bits)

AES Structure

Plaintext—16 bytes (128 bits)

Input state (16 bytes)

Initial transformation

State after initial transformation (16 bytes)

Round 1 (4 transformations)

Round 1 output state (16 bytes)

Round $N - 1$ (4 transformations)

Round $N - 1$ output state (16 bytes)

Round N (3 transformations)

Final state (16 bytes)

Ciphertext—16 bytes (128 bits)

Key—M bytes

Key (M bytes)

Round 0 key (16 bytes)

Round 1 key (16 bytes)

Round $N - 1$ key (16 bytes)

Round N key (16 bytes)

Key expansion

Initialization
1. Expand 16-byte key to get the actual **key block** to be used.
2. Initialize 16-byte plaintext block called as **state**.
3. XOR the **state** with the **key block**.

For each round
1. Apply S-box
2. Rotate rows of state
3. Mix columns
4. Add Round key: XOR the state with key block.

Data Units in AES

$$\text{Byte} \rightarrow \begin{bmatrix} b_0 & b_1 & b_2 & b_3 & b_4 & b_5 & b_6 & b_7 \end{bmatrix} \rightarrow \begin{bmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \\ b_6 \\ b_7 \end{bmatrix}$$

b   **b**

b

Byte
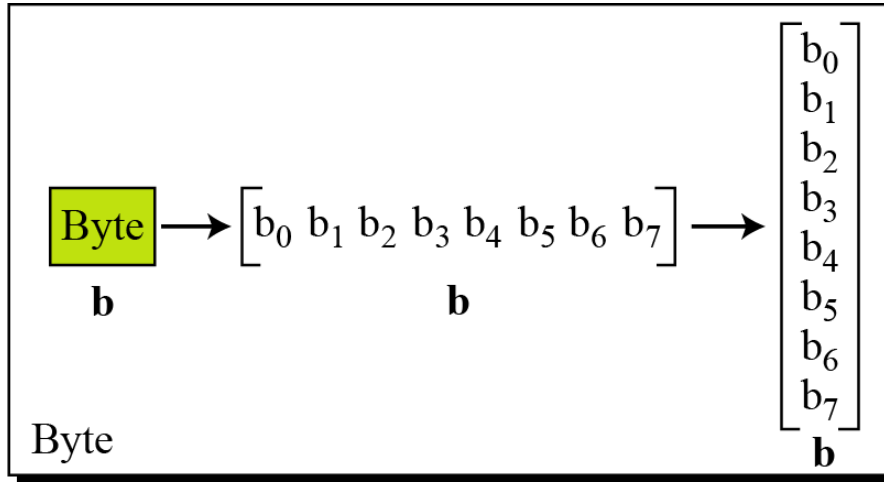
Block to State & State to Block

| b_0 | b_1 | b_2 | b_3 | b_4 | b_5 | b_6 | b_7 | b_8 | b_9 | b_{10} | b_{11} | b_{12} | b_{13} | b_{14} | b_{15} |

Block

$$s_{i \bmod 4,\ i/4} \longleftarrow block_i$$

State
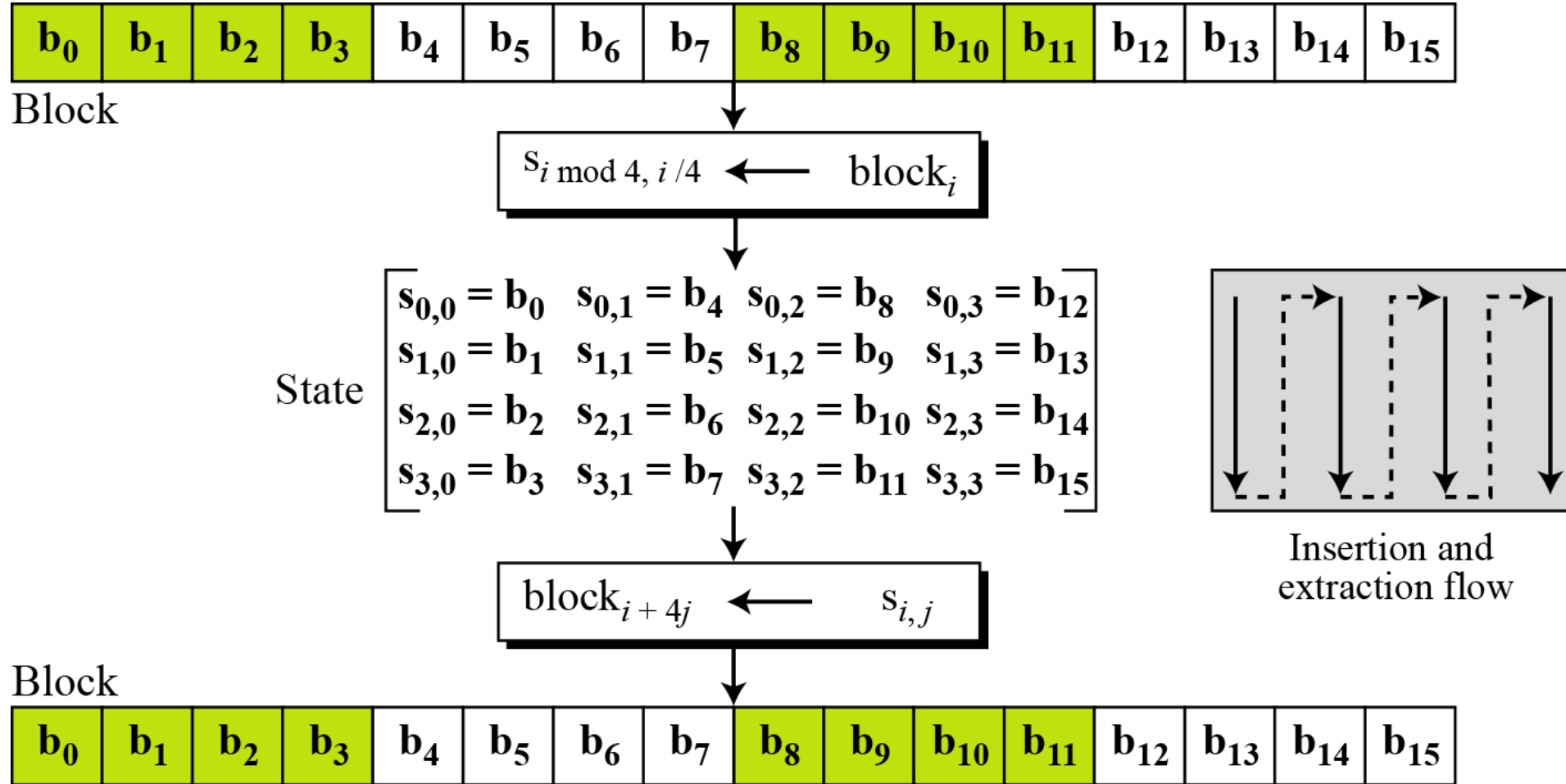$$\begin{bmatrix} s_{0,0} = b_0 & s_{0,1} = b_4 & s_{0,2} = b_8 & s_{0,3} = b_{12} \\ s_{1,0} = b_1 & s_{1,1} = b_5 & s_{1,2} = b_9 & s_{1,3} = b_{13} \\ s_{2,0} = b_2 & s_{2,1} = b_6 & s_{2,2} = b_{10} & s_{2,3} = b_{14} \\ s_{3,0} = b_3 & s_{3,1} = b_7 & s_{3,2} = b_{11} & s_{3,3} = b_{15} \end{bmatrix}$$

Insertion and extraction flow

$$block_{i+4j} \longleftarrow s_{i,j}$$

Block

| b_0 | b_1 | b_2 | b_3 | b_4 | b_5 | b_6 | b_7 | b_8 | b_9 | b_{10} | b_{11} | b_{12} | b_{13} | b_{14} | b_{15} |

Plain Text 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

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

AES Structure

- The first N-1 rounds consist of four distinct transformation functions.

| **SubBytes** | • The 16 input bytes are substituted using an **S-box** |
ShiftRows	• Each of the four rows of the matrix is shifted to the left
MixColumns	• Each column of four bytes is now transformed using a special mathematical function.
AddRoundKey	• The 16 bytes of the matrix are now considered as 128 bits and are XORed to the 128 bits of the round key.

SubByte Transformation

- The forward substitute byte transformation, called **SubBytes**, is a simple table lookup
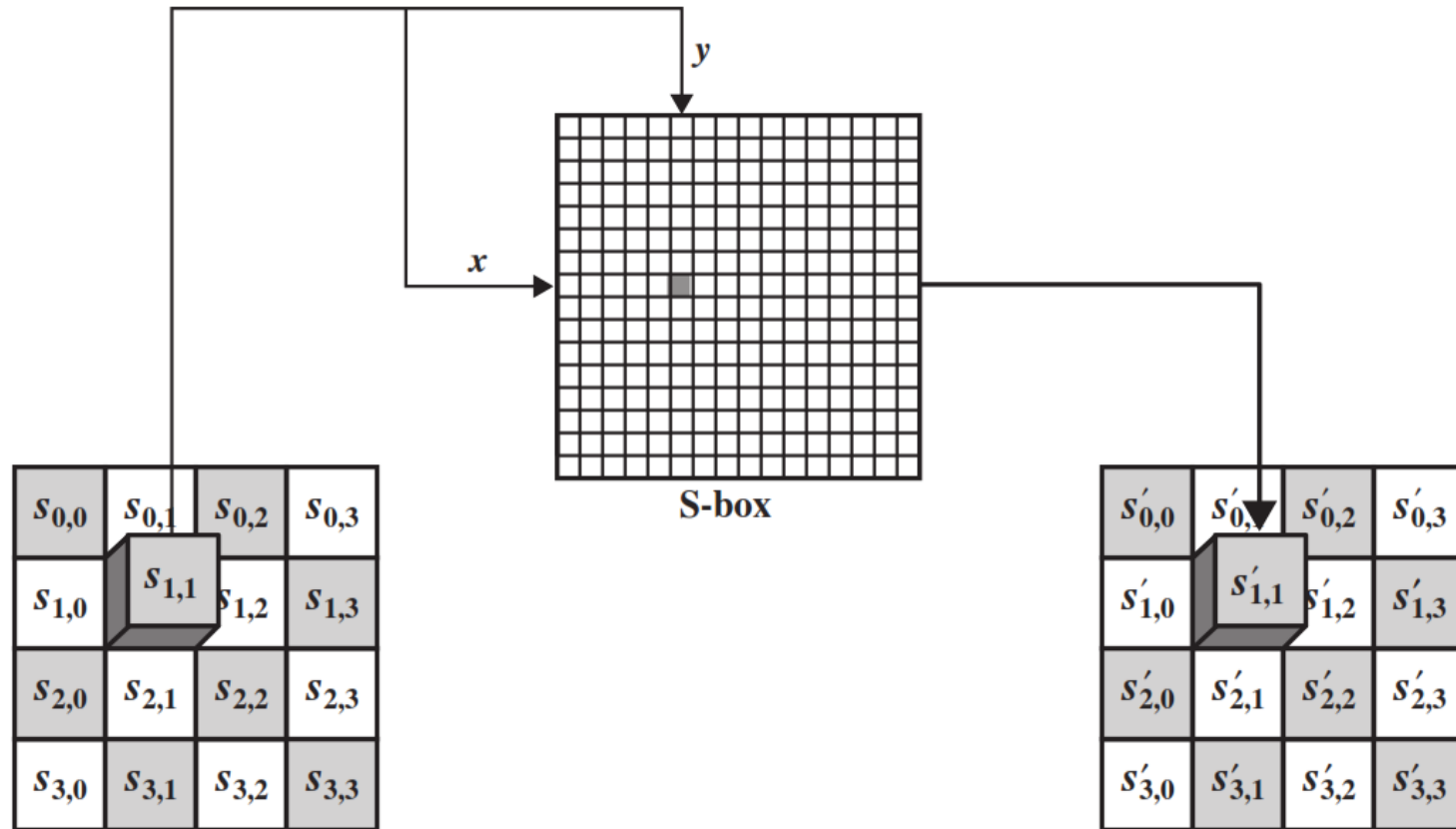
ShiftRows

- 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.

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

MixColumns

- Each byte of a column is mapped into a new value that is a function of all four bytes in that column.

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

$\rightarrow$

47	40	A3	4C
37	D4	70	9F
94	E4	3A	42
ED	A5	A6	BC

AddRoundKey

- In the forward add round key transformation, the 128 bits of State are bitwise XORed with the 128 bits of the round key.

47	40	A3	4C
37	D4	70	9F
94	E4	3A	42
ED	A5	A6	BC

$\oplus$

AC	19	28	57
77	FA	D1	5C
66	DC	29	00
F3	21	41	6A

$=$

EB	59	8B	1B
40	2E	A1	C3
F2	38	13	42
1E	84	E7	D6

State   Round Key

AES Overall Structure
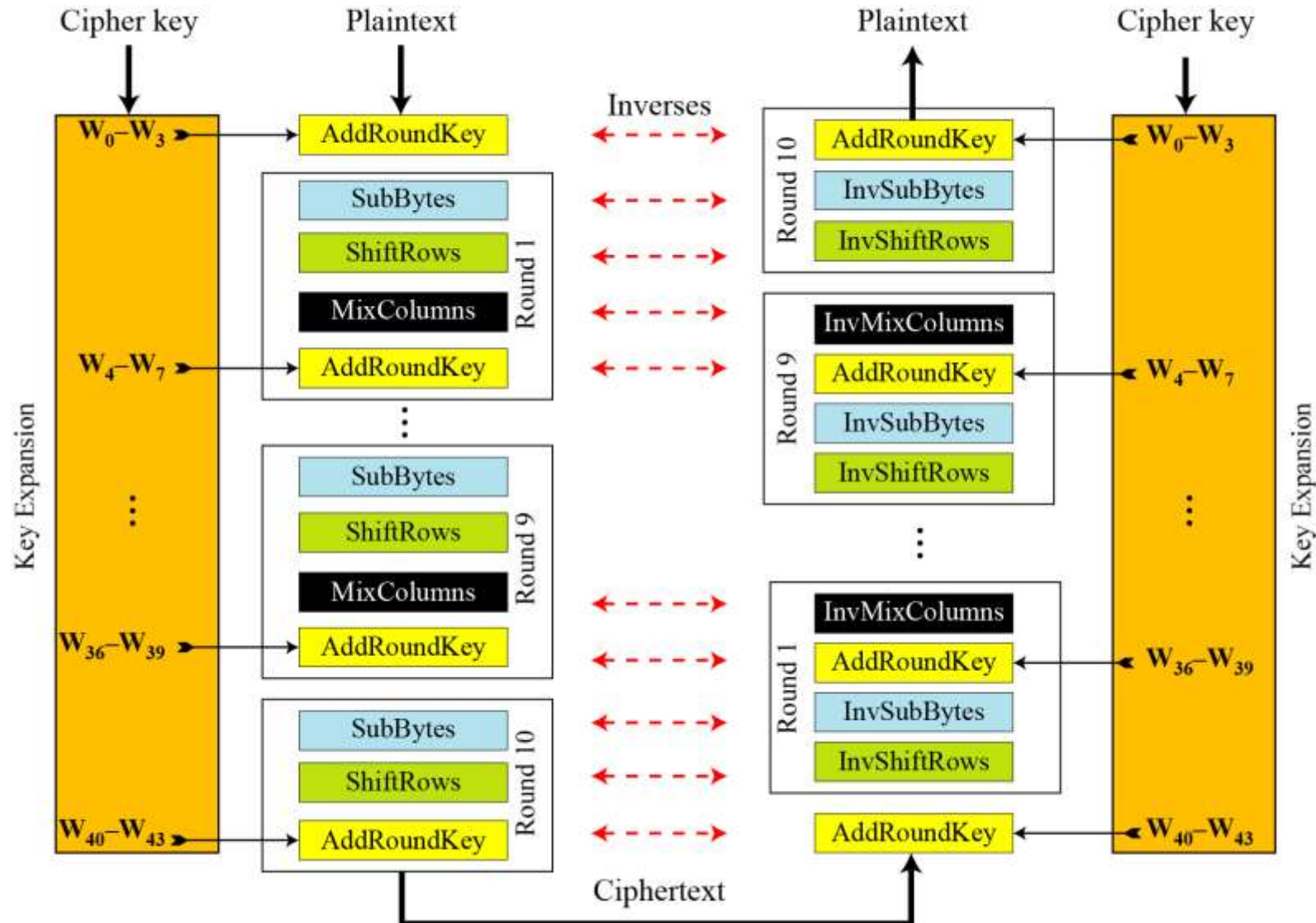
AES Key Expansion
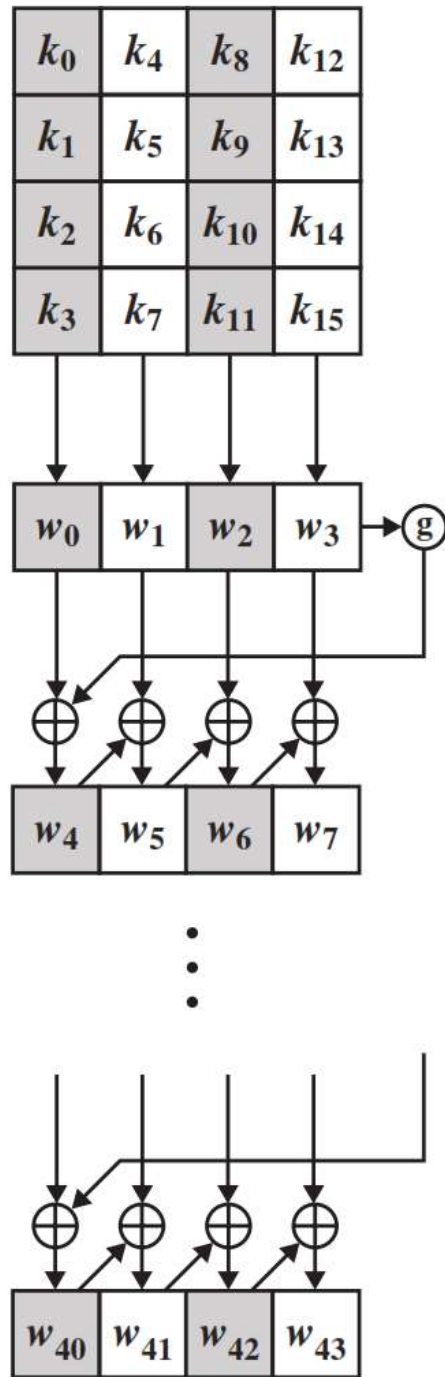
- The AES key expansion algorithm takes as input a four-word (16-byte) key and produces a linear array of **44 words** (176 bytes).

- Each added word **w[i]** depends on the immediately preceding word, w[i - 1].

- In three out of four cases, a simple XOR is used.

Key Expansion Example

Plaintext:	0123456789abcdeffedcba9876543210
Key:	0f1571c947d9e8590cb7add6af7f6798
Ciphertext:	ff0b844a0853bf7c6934ab4364148fb9

Key Words	Auxiliary Function
w0 = 0f 15 71 c9 w1 = 47 d9 e8 59 w2 = 0c b7 ad d6 w3 = af 7f 67 98	RotWord(w3) = 7f 67 98 af = x1 SubWord(x1) = d2 85 46 79 = y1 Rcon(1) = 01 00 00 00 y1 ⊕ Rcon(1) = d3 85 46 79 = z1
w4 = w0 ⊕ z1 = dc 90 37 b0 w5 = w4 ⊕ w1 = 9b 49 df e9 w6 = w5 ⊕ w2 = 97 fe 72 3f w7 = w6 ⊕ w3 = 38 81 15 a7	RotWord(w7) = 81 15 a7 38 = x2 SubWord(x2) = 0c 59 5c 07 = y2 Rcon(2) = 02 00 00 00 y2 ⊕ Rcon(2) = 0e 59 5c 07 = z2
w8 = w4 ⊕ z2 = d2 c9 6b b7 w9 = w8 ⊕ w5 = 49 80 b4 5e w10 = w9 ⊕ w6 = de 7e c6 61 w11 = w10 ⊕ w7 = e6 ff d3 c6	RotWord(w11) = ff d3 c6 e6 = x3 SubWord(x3) = 16 66 b4 83 = y3 Rcon(3) = 04 00 00 00 y3 ⊕ Rcon(3) = 12 66 b4 8e = z3
w12 = w8 ⊕ z3 = c0 af df 39 w13 = w12 ⊕ w9 = 89 2f 6b 67 w14 = w13 ⊕ w10 = 57 51 ad 06 w15 = w14 ⊕ w11 = b1 ae 7e c0	RotWord(w15) = ae 7e c0 b1 = x4 SubWord(x4) = e4 f3 ba c8 = y4 Rcon(4) = 08 00 00 00 y4 ⊕ Rcon(4) = ec f3 ba c8 = 4

AES Mix Column

2	3	1	1
1	2	3	1
1	1	2	3
3	1	1	2

*

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

=

?			

$\{02\} * \{87\} \oplus \{03\} * \{6E\} \oplus \{01\} * \{46\} \oplus \{01\} * \{A6\}$

$$X^7 + X^6 + X^5 + X^4 + X^3 + X^2 + \boxed{X^1} + 1$$

0 0 0 0 0 0 1 0

$02 = 0000\ 0010 = X$

$87 = 1000\ 0111 = X^7 + X^2 + X + 1$

$\boxed{X^7} + X^6 + X^5 + X^4 + X^3 + \boxed{X^2} + \boxed{X^1} + \boxed{1}$

·1   0   0   0   0   1   1   1

$\{02\} * \{87\} = X * (X^7 + X^2 + X + 1)$

$= \boxed{X^8} + X^3 + X^2 + X$

$= X^4 + \cancel{X^3} + \cancel{X} + 1 + \cancel{X^3} + X^2 + \cancel{X}$

$= X^4 + X^2 + 1$

Use irreducible Polynomial Theorem, GF (2³)

$X^8 = X^4 + X^3 + X + 1$

$$X^7 + X^6 + X^5 + X^4 + X^3 + X^2 + X^1 + 1$$

0   0   0   1   0   1   0   1