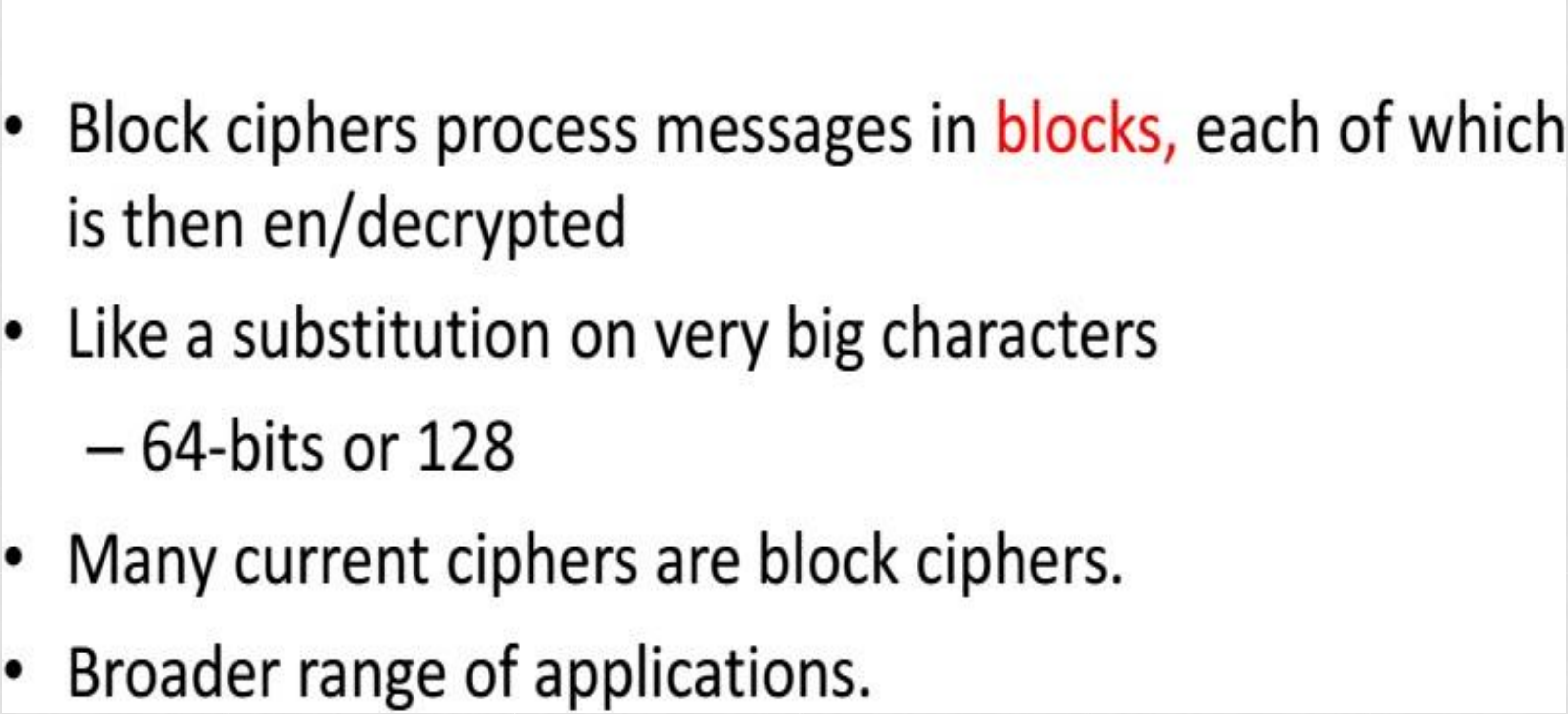
Information Security: Cryptography

Block Ciphers



- Block ciphers process messages in blocks, each of which is then en/decrypted
- Like a substitution on very big characters
  - 64-bits or 128
- Many current ciphers are block ciphers.
- Broader range of applications.

SYMMETRIC BLOCK ENCRYPTION ALGORITHMS

A block cipher processes the plaintext input in fixed-size blocks and produces a block of cipher text of equal size for each plaintext block.

- DES
- Triple DES
- AES etc.,

DATA ENCRYPTION STANDARD

- The **Data Encryption Standard (DES)** was designed by IBM.
- Block Cipher
- DES , adopted in 1977 by National Bureau of Standards(NBS), now National institute of Standards and Technology(NIST) as Federal information processing standard 46.
- It was unbroken for more than 10 years since its publication and some aspects of its design were kept secret by IBM at the request of the US National Security Agency (NSA); some people believed that IBM and NSA had hidden a trapdoor in DES that only they knew about (that they could use to crack DES)

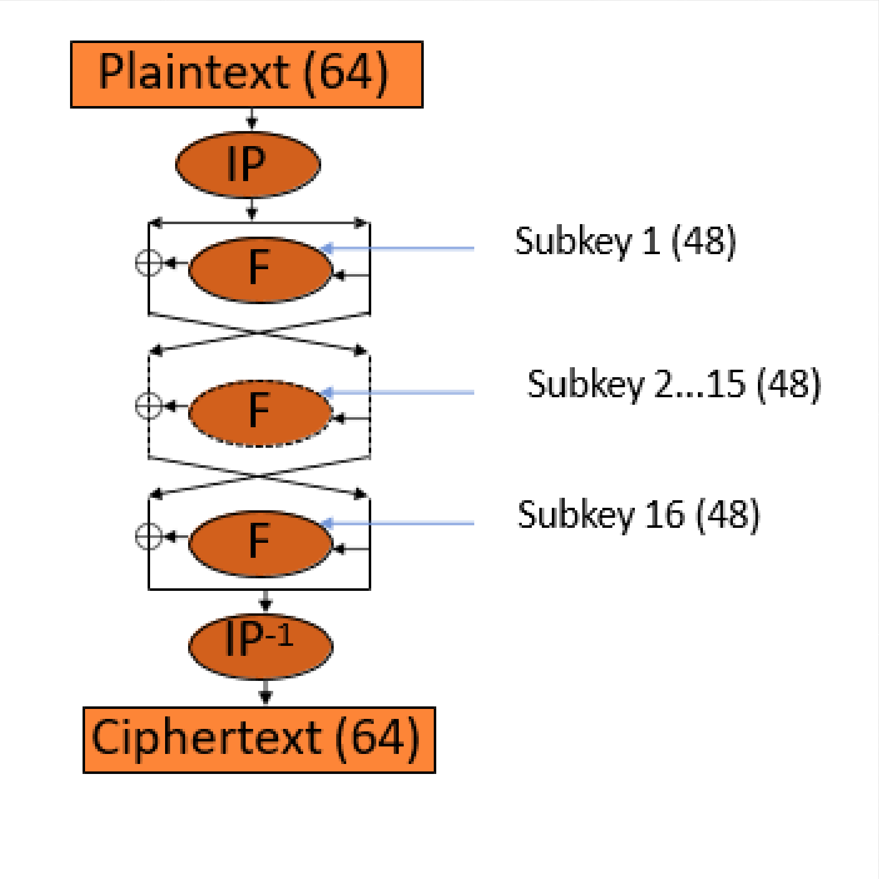
DES CRITICISM

- Before adopting as standard ,intense criticism.
  - First, Enormous reduction in the key.
- Too short to withstand brute force attack.
  - Second , internal structure S-boxes was classified.

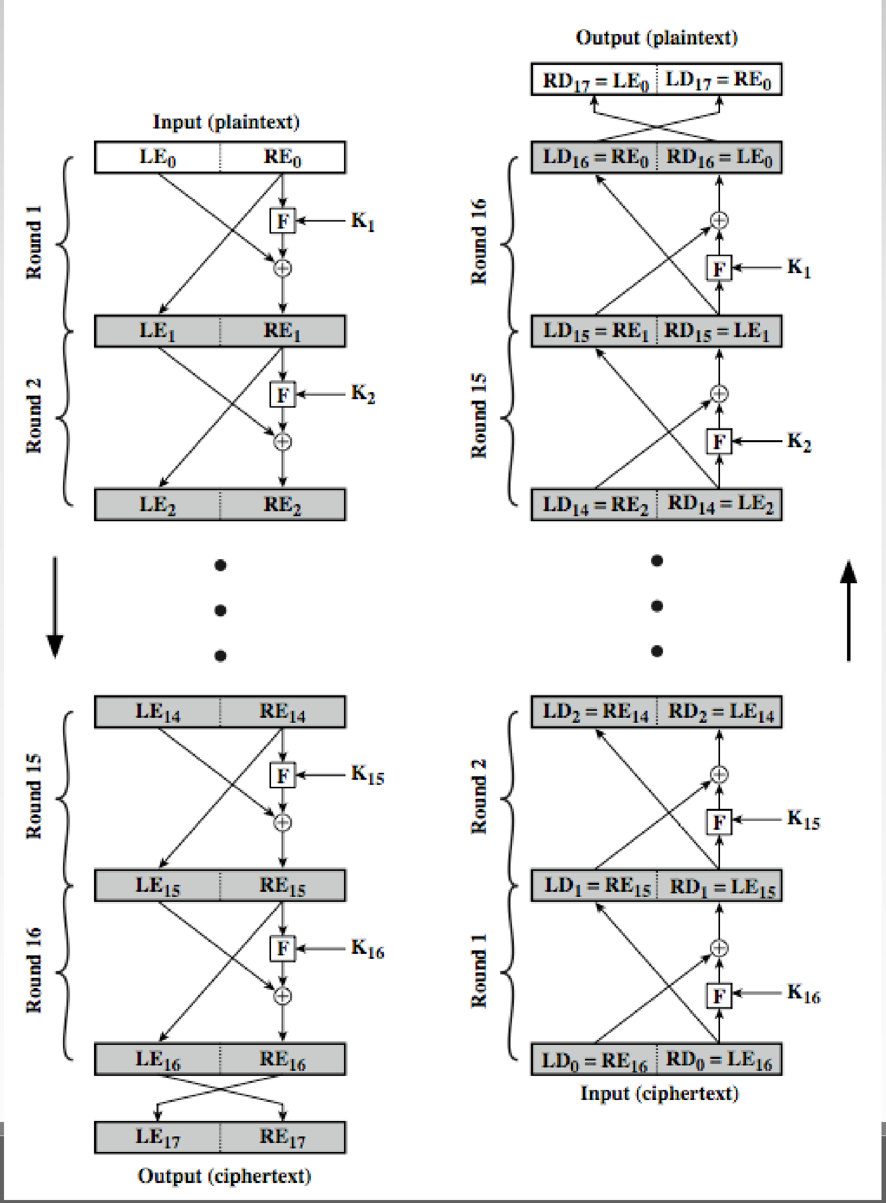
DES WIDE USE

- DES flourish in financial applications.
- In 1994, NIST reaffirm to use it for more 5 years.
- In 1999, NIST issue a new version of its standard triple DES.

DES Block Cipher



Feistel Cipher Structure



DATA ENCRYPTION STANDARD

- Block Cipher
- Block size – 64 bits
- Follows Feistel Structure
- Total Rounds – 16
- Key Size – 64 bits
- Sub keys – 16
- Sub key size – 48 bits
- Cipher text – 64 bits

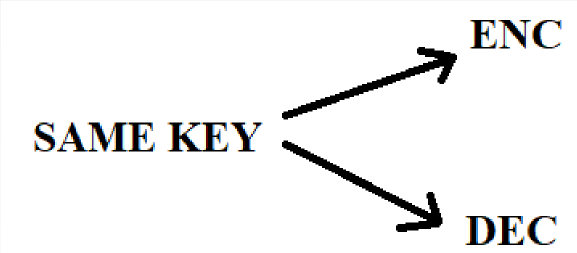
DES Introduction

DATA ENCRYPTION STANDARD (DES):

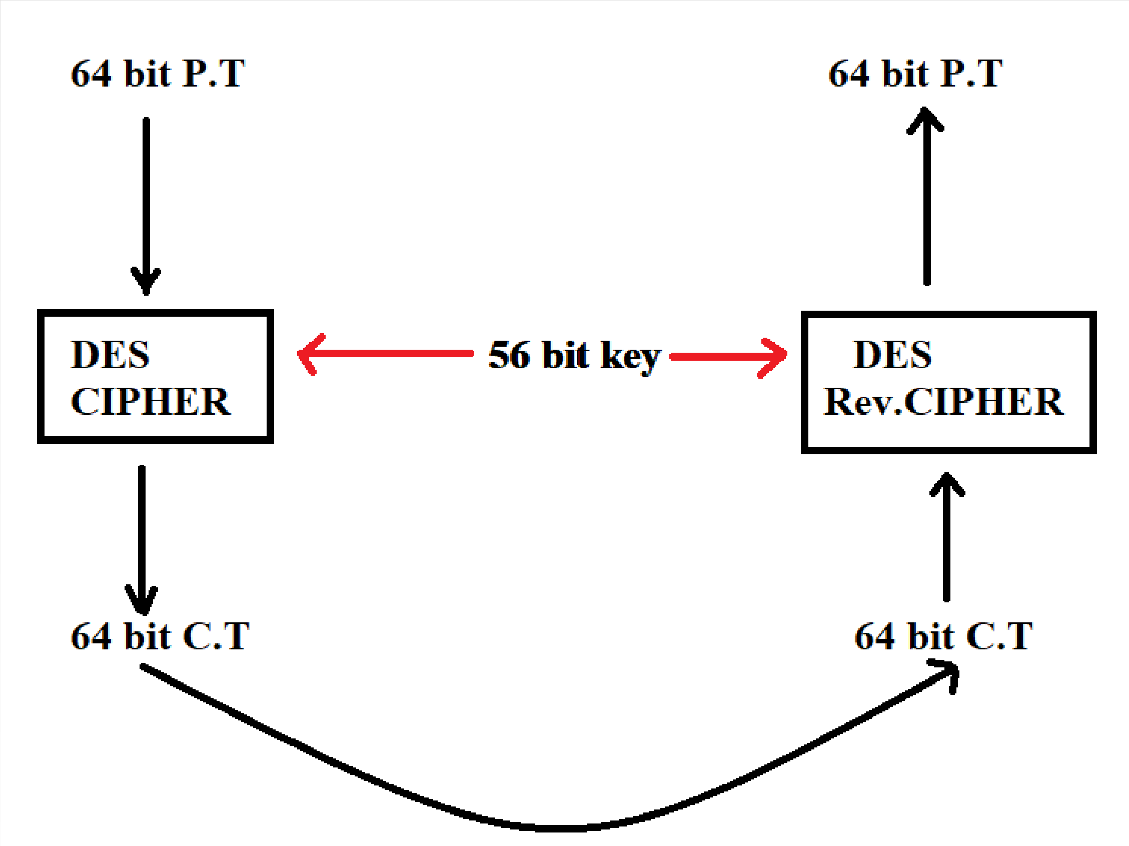
→ Symmetric Key-block cipher

→ Published by (NIST) National Institute of Standards and Technology.

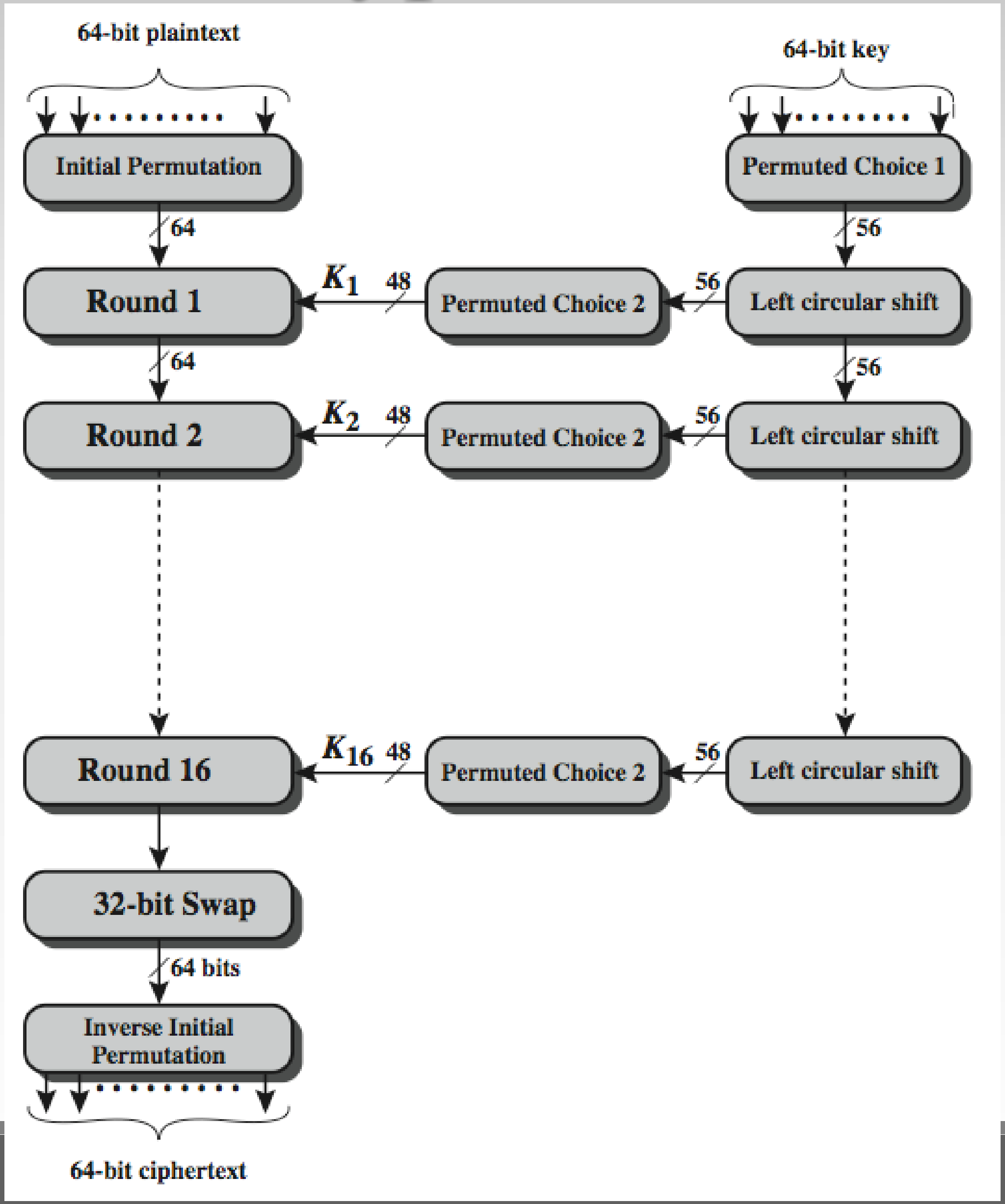
→ DES is block cipher.



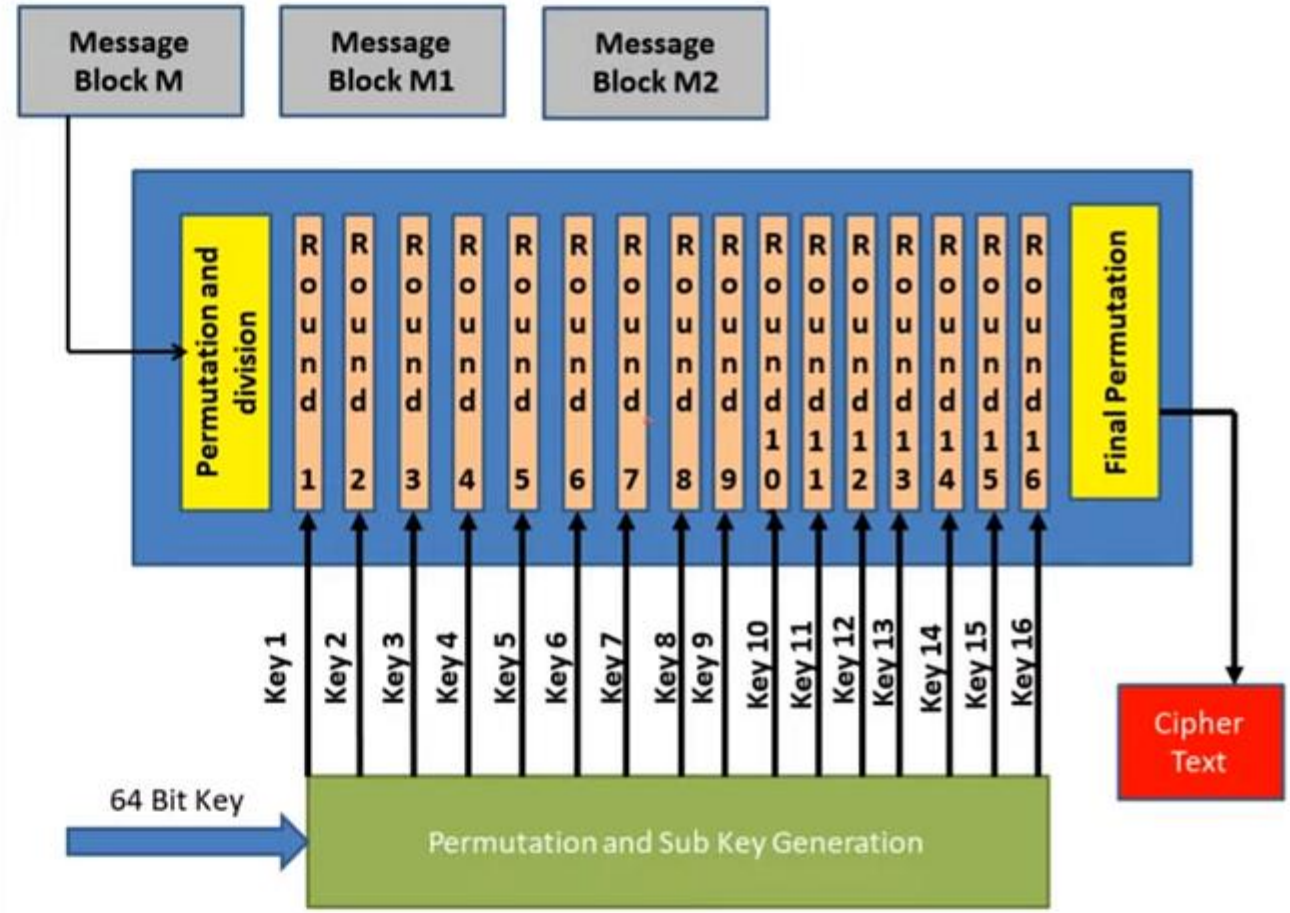
Overview of DES



DES Encryption Overview



DES Encryption Overview



DES Encryption (1/2)

- There are two inputs to the encryption function: plaintext and key.
- The plaintext must be 64 bit in length and the key is 56 bits in length*.
- Left hand side of the figure, the processing of the plaintext proceeds in following phases.
- First , the 64-bit plaintext passes through an initial permutation(IP) that rearrange the bits to produce permuted input.

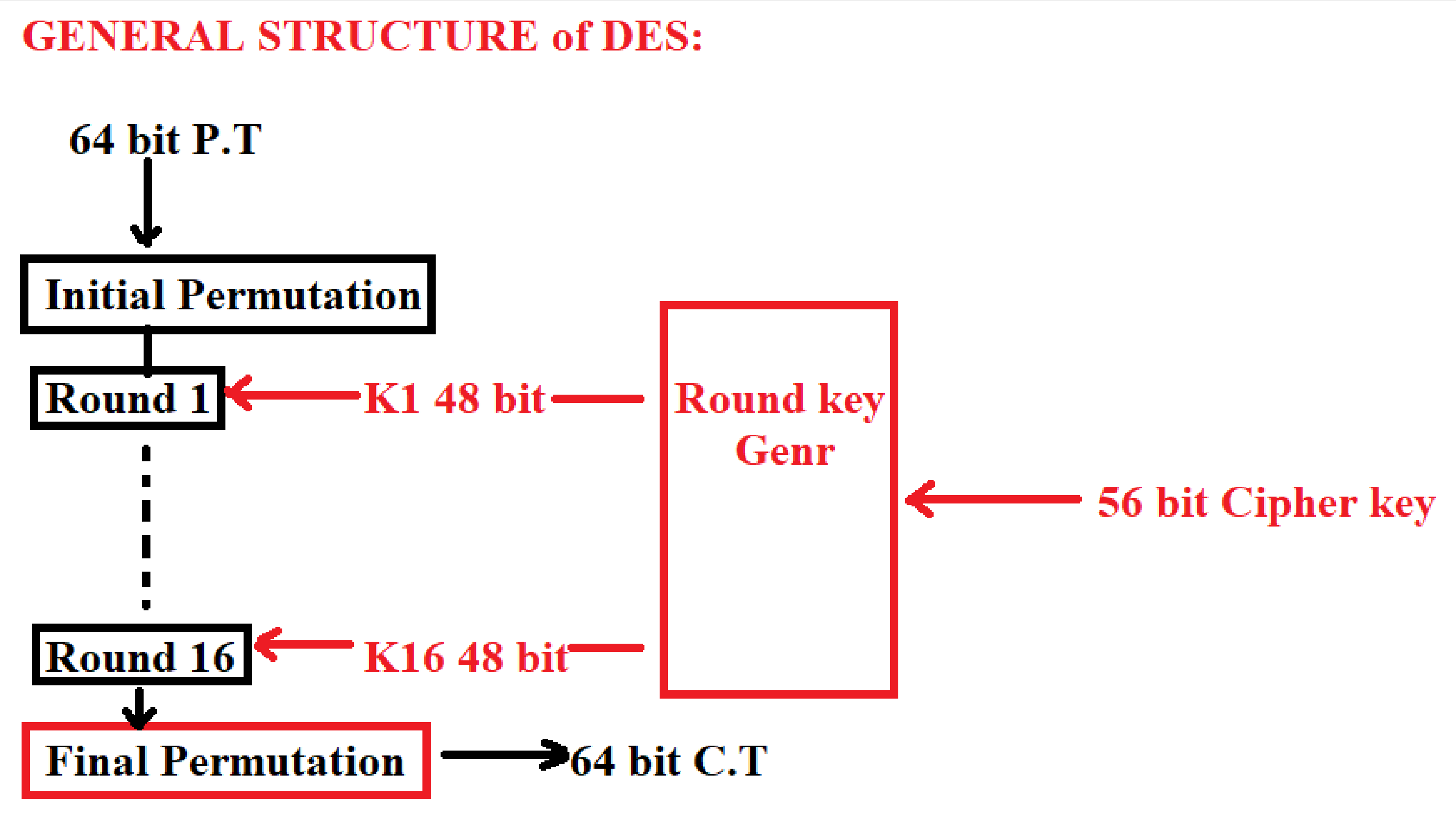
DES Encryption (2/2)

- This is followed by the phase of 16 rounds of the same function , involve substitution and permutation.
- The output of the last(16th)round consists of 64-bits that are the function of input plaintext and the key.
- The left and right halves of the output are swapped to produce the preoutput.
- Finally , preoutput is passed through a permutation (IP^{-1}), to produce the 64-bit output.

DES Encryption (Key)

- The right-hand portion, shows how 56-bit key is used.
- Initially , the key is passed through a permutation function.
- For each of the 16 rounds, a subkey (k_i) is produced by the combination of the left circular shift and the permutation. the permutation function is same for each round, but the different subkey is produced because of the repeated shifts of the key bits.

General Structure of DES



GENERAL STRUCTURE of DES:

DES step 2

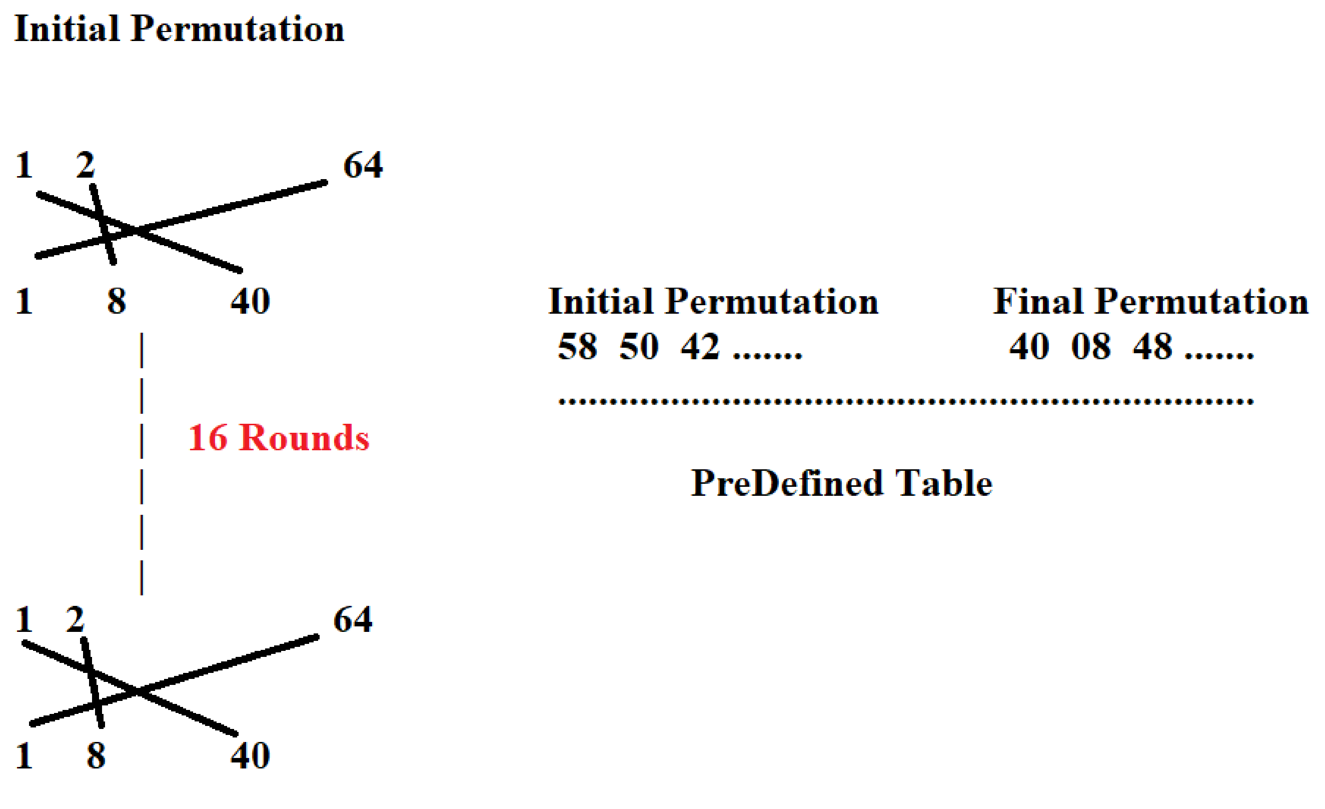
Initial Permutation (IP):

Occurs only once before the first round.

Final Permutation (FP):

Both initial and final permutation are straight P-Boxes that are inverses of each other. They have no cryptographic significance in DES. (Predefined)

Permutation's concept



Initial Permutation

Initial Permutation

58 50 42 ....

Final Permutation

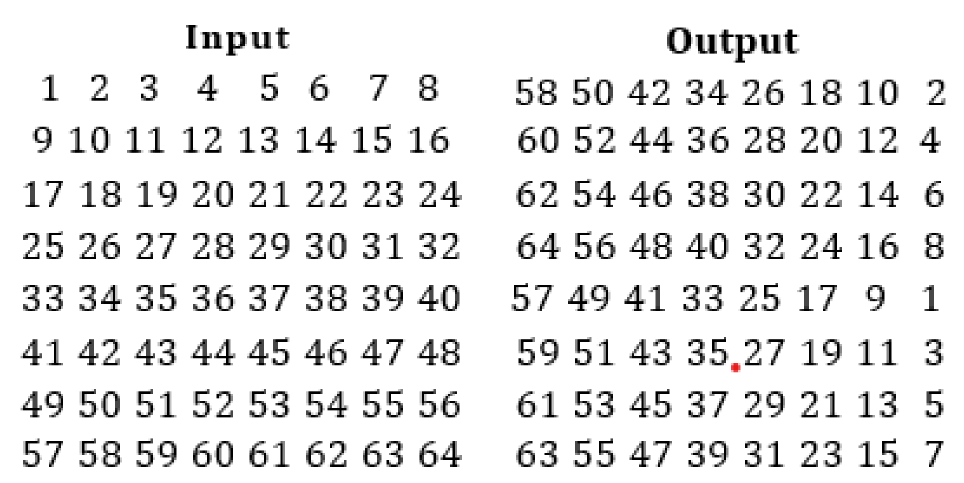
40 08 48 ....

....

PreDefined Table

INITIAL PERMUTATION: IP(FROM LAST TABLES)

- The 64-bit input data (message) block is first bitwise permutated (i.e., the bits within the block are rearranged)
- This is done using the following permutation table:

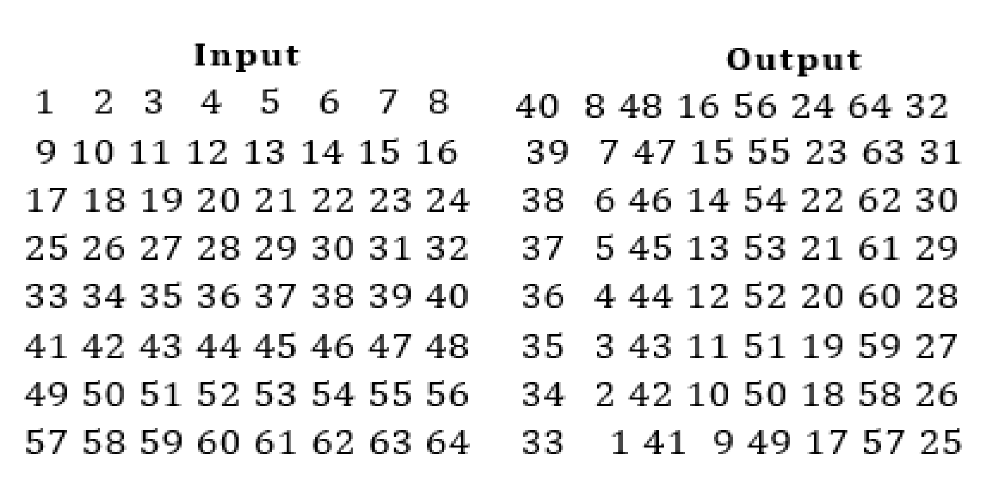


Input								Output							
1	2	3	4	5	6	7	8	58	50	42	34	26	18	10	2
9	10	11	12	13	14	15	16	60	52	44	36	28	20	12	4
17	18	19	20	21	22	23	24	62	54	46	38	30	22	14	6
25	26	27	28	29	30	31	32	64	56	48	40	32	24	16	8
33	34	35	36	37	38	39	40	57	49	41	33	25	17	9	1
41	42	43	44	45	46	47	48	59	51	43	35	27	19	11	3
49	50	51	52	53	54	55	56	61	53	45	37	29	21	13	5
57	58	59	60	61	62	63	64	63	55	47	39	31	23	15	7

- Example: 35th bit of input block is equal to the 41st bit of the output block.

FINAL PERMUTATION

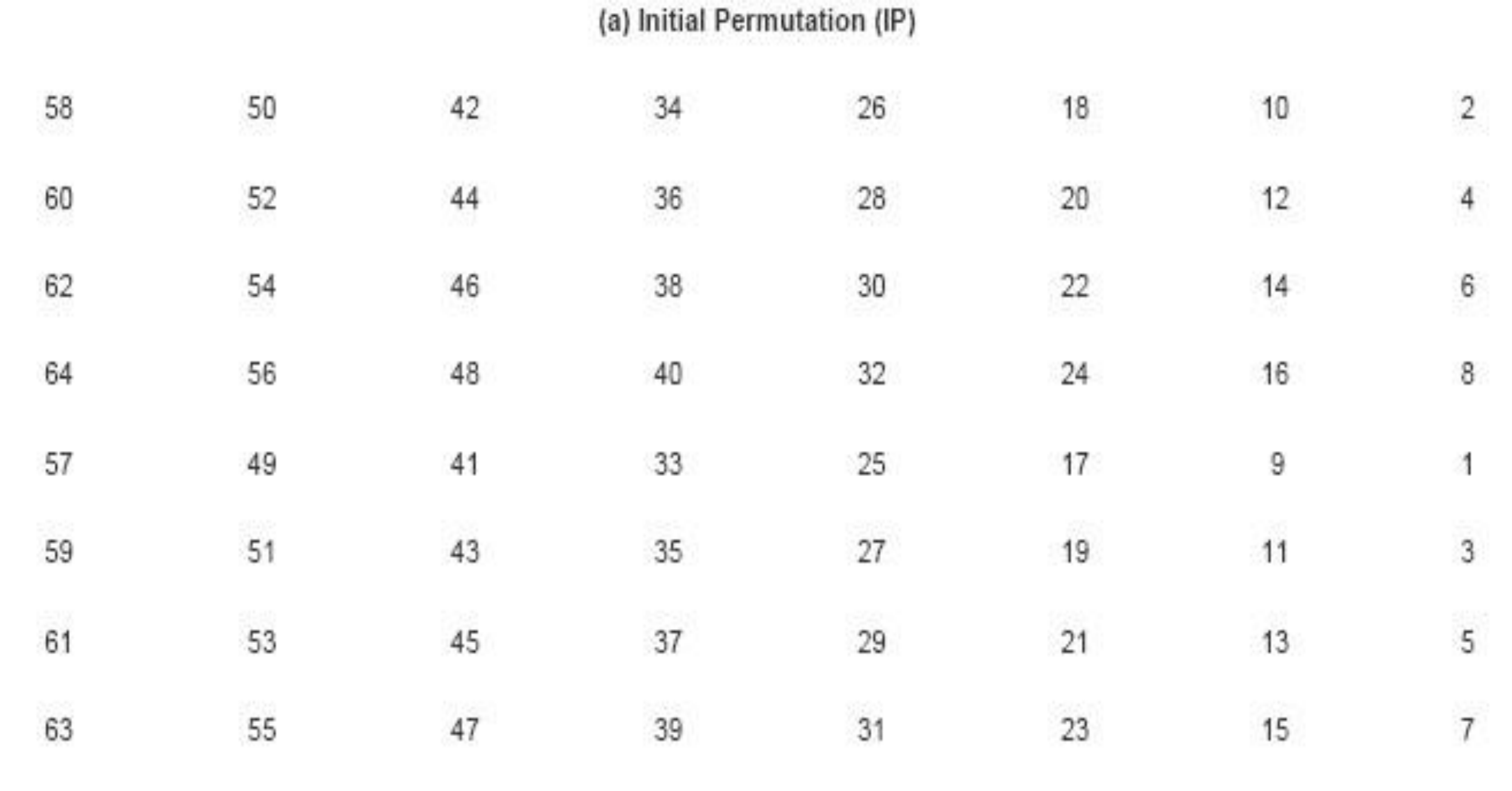
- The 64-bit output after 16 rounds is finally bitwise permutated (i.e., the bits within the block are rearranged)
- This is done using the following permutation table:



Input								Output							
1	2	3	4	5	6	7	8	40	8	48	16	56	24	64	32
9	10	11	12	13	14	15	16	39	7	47	15	55	23	63	31
17	18	19	20	21	22	23	24	38	6	46	14	54	22	62	30
25	26	27	28	29	30	31	32	37	5	45	13	53	21	61	29
33	34	35	36	37	38	39	40	36	4	44	12	52	20	60	28
41	42	43	44	45	46	47	48	35	3	43	11	51	19	59	27
49	50	51	52	53	54	55	56	34	2	42	10	50	18	58	26
57	58	59	60	61	62	63	64	33	1	41	9	49	17	57	25

- Example: 41st bit of Input block is equal to the 35th bit of the output block.

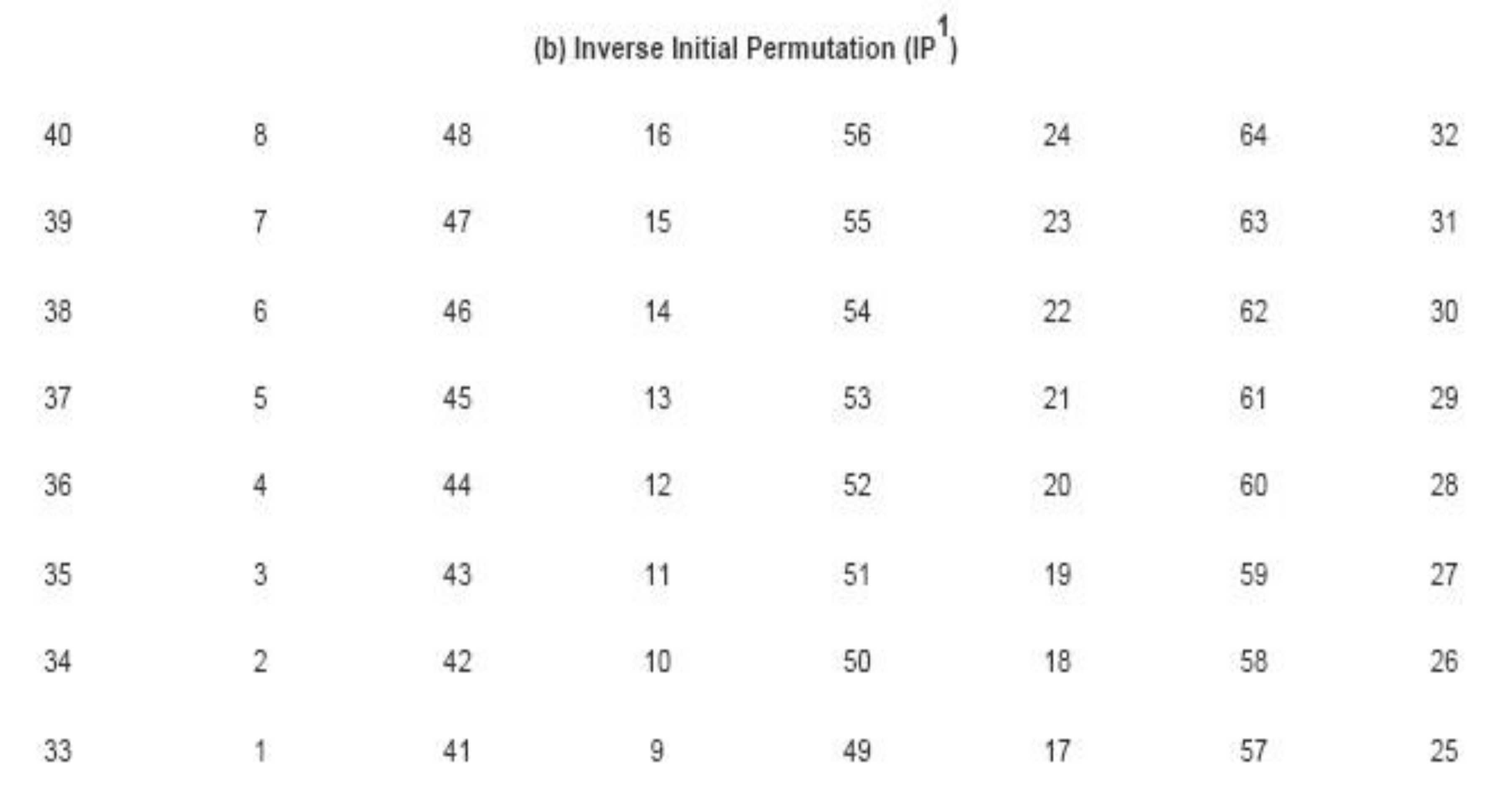
DES Block Cipher



(a) Initial Permutation (IP)

58	50	42	34	26	18	10	2
60	52	44	36	28	20	12	4
62	54	46	38	30	22	14	6
64	56	48	40	32	24	16	8
57	49	41	33	25	17	9	1
59	51	43	35	27	19	11	3
61	53	45	37	29	21	13	5
63	55	47	39	31	23	15	7

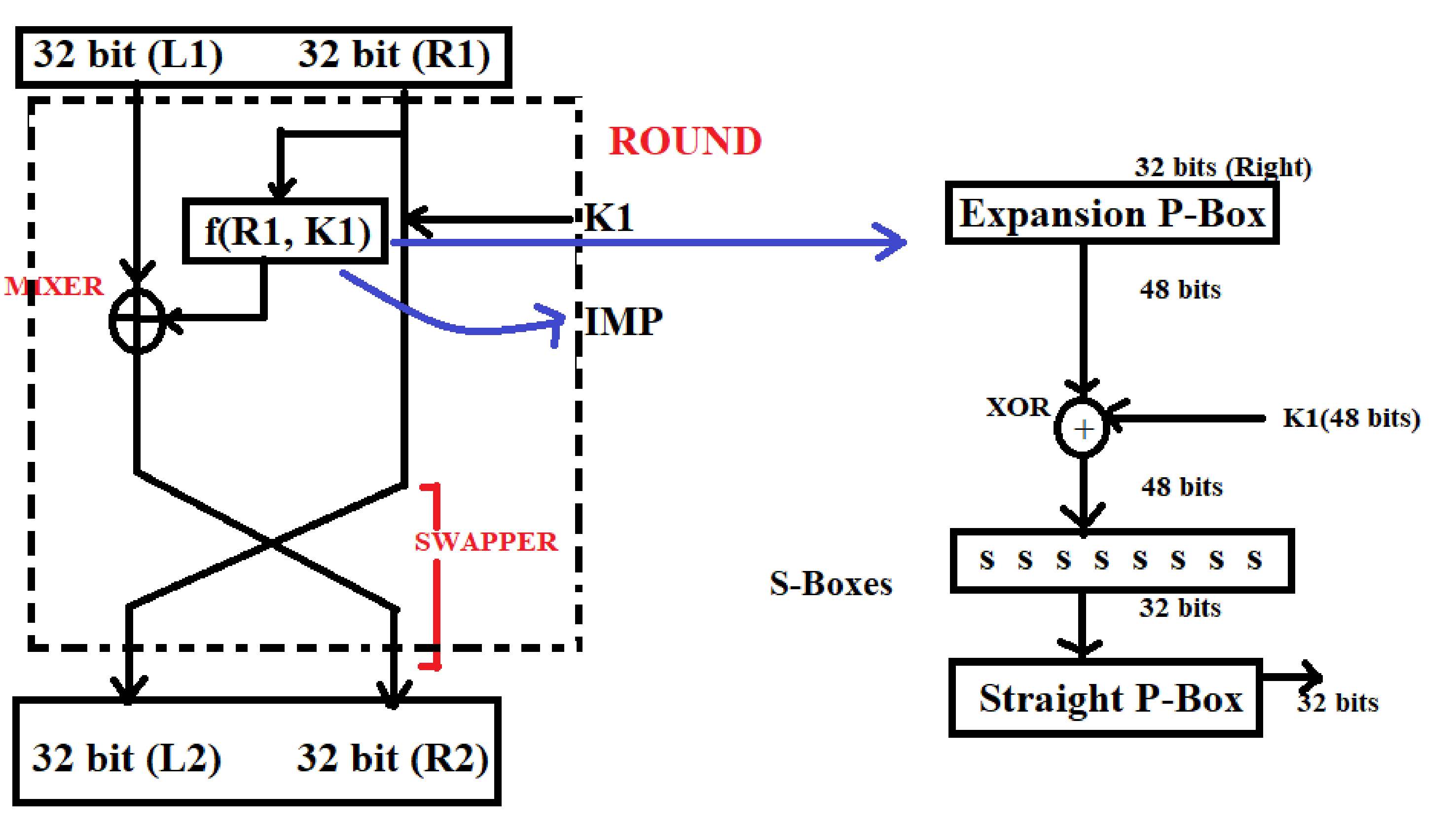
DES Block Cipher



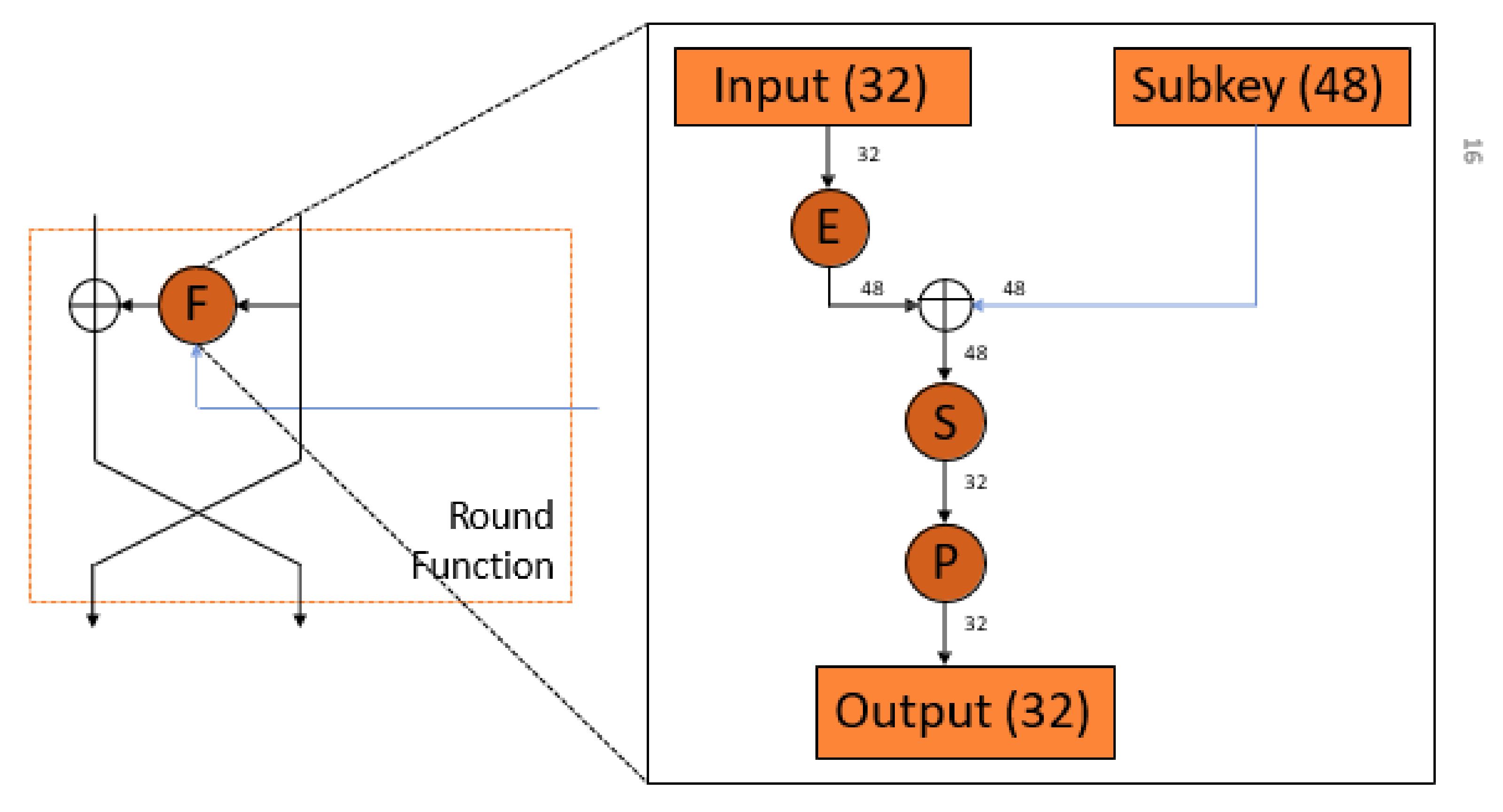
(b) Inverse Initial Permutation (IP^{1})

40	8	48	16	56	24	64	32
39	7	47	15	55	23	63	31
38	6	46	14	54	22	62	30
37	5	45	13	53	21	61	29
36	4	44	12	52	20	60	28
35	3	43	11	51	19	59	27
34	2	42	10	50	18	58	26
33	1	41	9	49	17	57	25

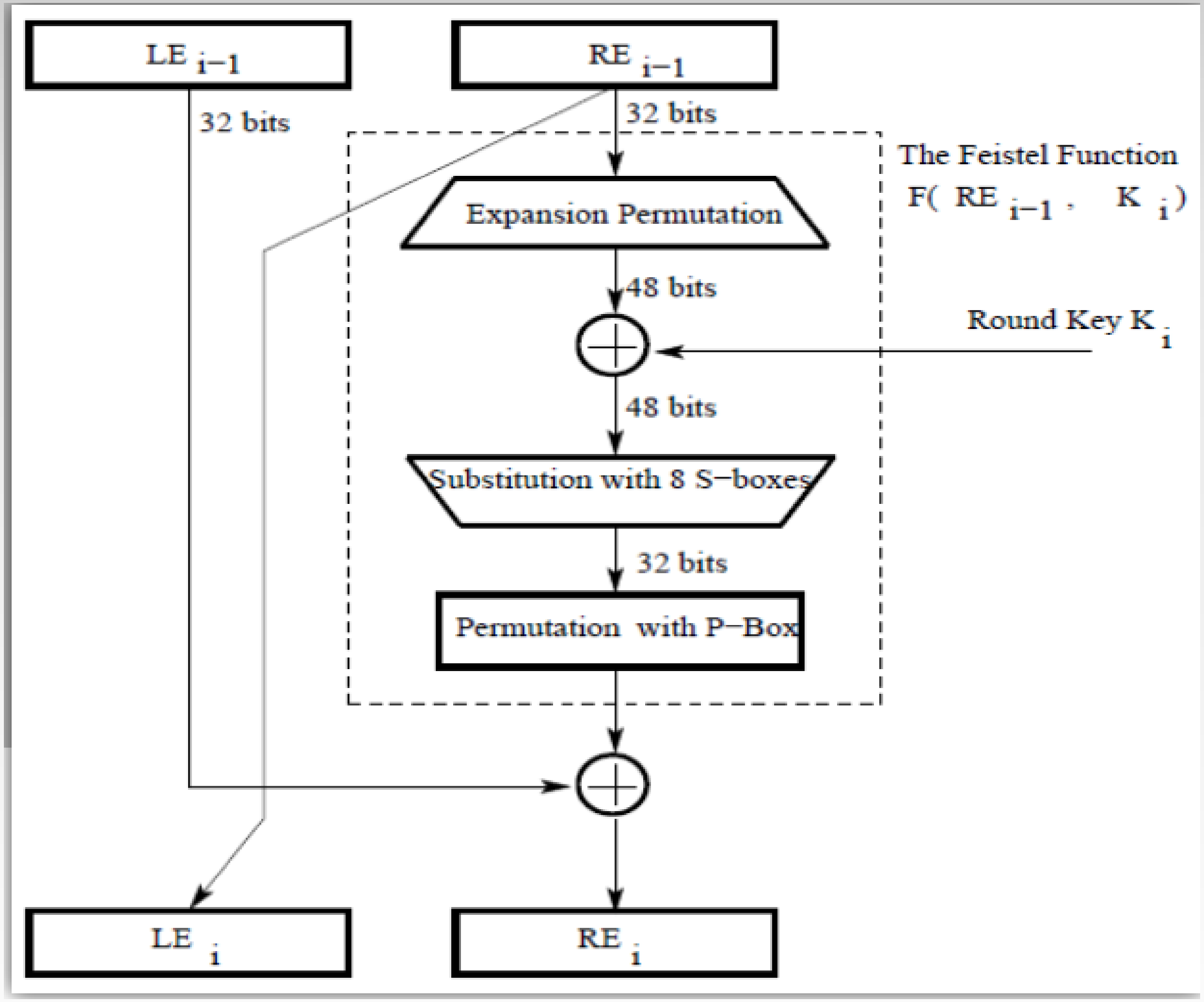
DES Step-3


32 bit (L1)
32 bit (R1)
ROUND
32 bits (Right)
Expansion P-Box
f(R1, K1)
K1
MIXER
IMP
48 bits
XOR
K1(48 bits)
48 bits
SWAPPER
S-Boxes
s s s s s s s s
32 bits
Straight P-Box
32 bits
32 bit (L2)
32 bit (R2)

DES CIPHER FUNCTION



DES Round Structure



THE SINGLE ROUND

- The left hand side of the diagram, the left and right halves of each 64-bit are treated as 32-bit quantities labelled as L(left) and R(right).
- The round key K_i is 48 bit. The R input is 32 bits.
- The R input is first expanded to 48 bits by using table that involves duplication of 16 of the R bits.
- The resulting 48 bits are XORed with k_i.
- This 48-bit result passes through a substitution function that produces a 32-bit output , which is permuted

DES Step-3

ROUNDS:
There are 16 rounds, and each round is based on Feistel Cipher structure.

DES Function:
Applies 48 bit key to the rightmost 32-bit to produce 32-bit o/p.

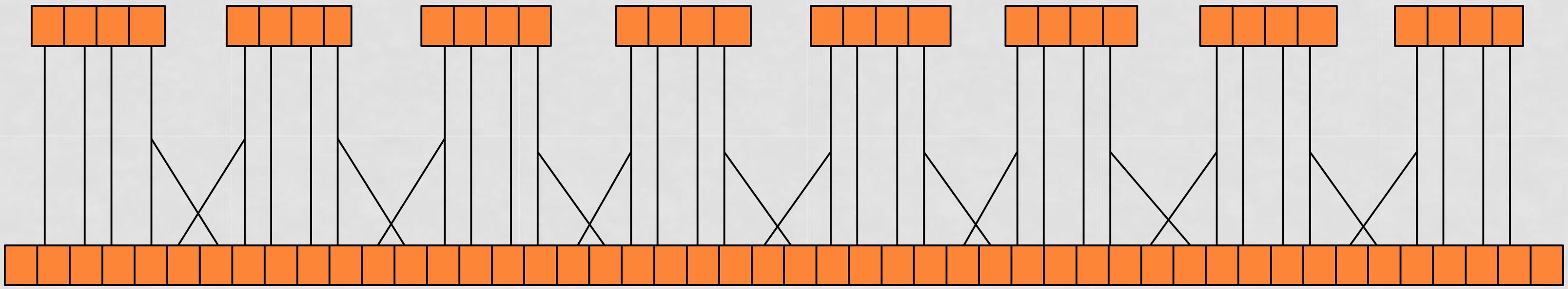
→ Expansion P-Box

→ Whitener

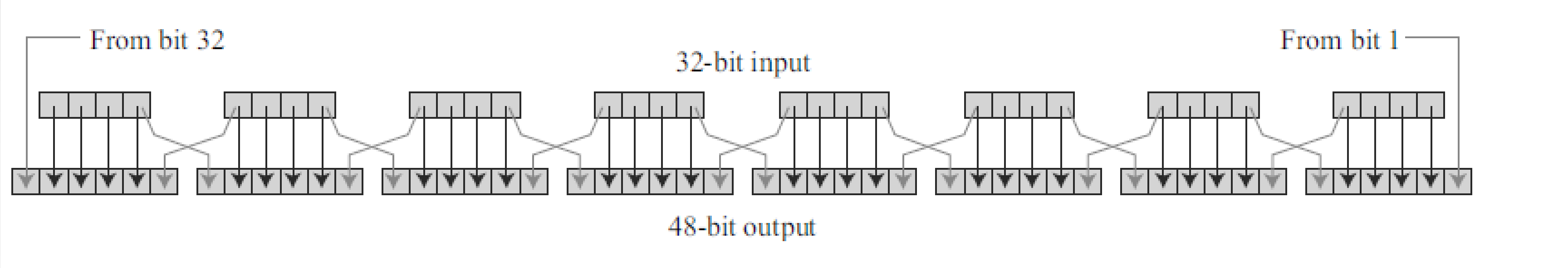
→ Group of S-Boxes

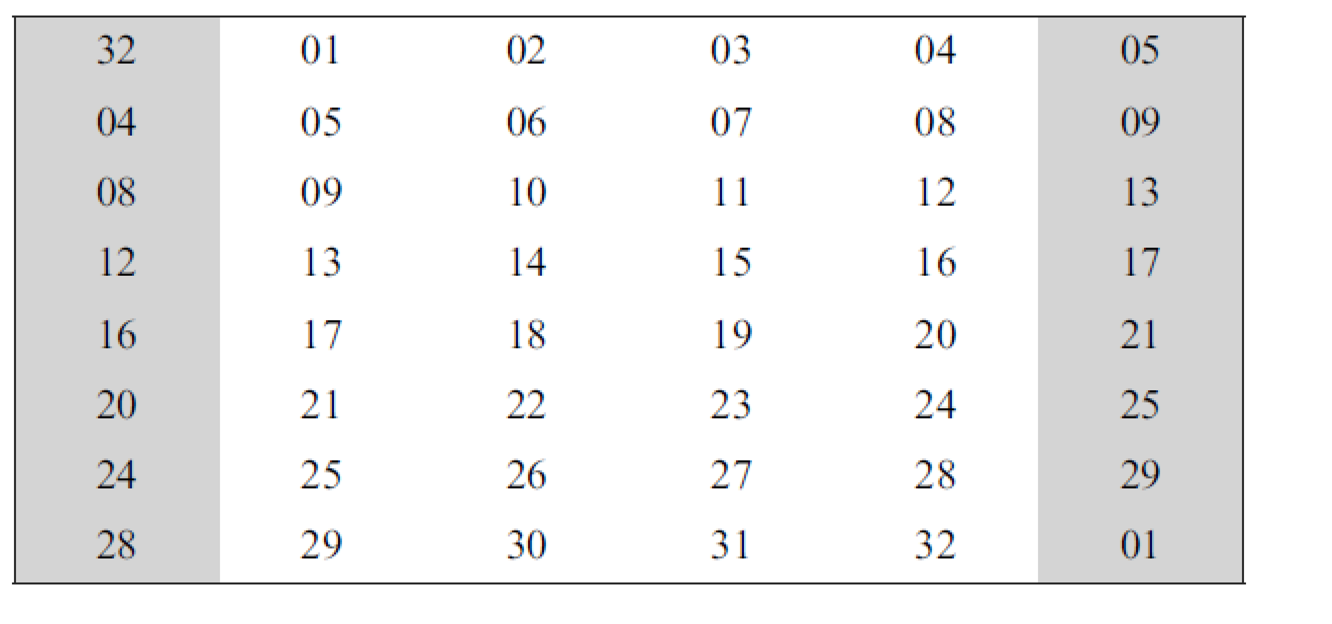
→ Straight P Box

EXPANSION PERMUTATION :E



Expansion Box

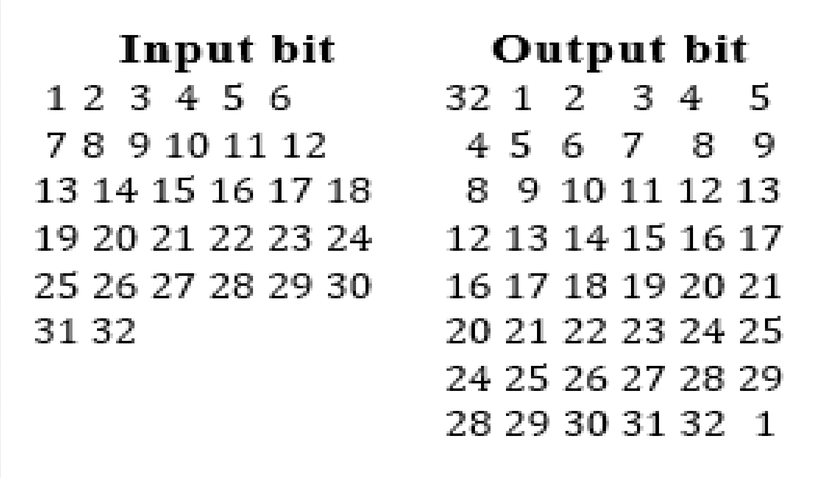




32	01	02	03	04	05
04	05	06	07	08	09
08	09	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	01

EXPANSION PERMUTATION: E(FROM LAST TABLE)

- The **expansion permutation** acts on the 32-bit input to the cipher function.
- It expands the 32-bit input block to a 48-bit output block by duplicating some input bits at specified positions
- The permutation is given by the following table:



Input bit	Output bit
1 2 3 4 5 6	32 1 2 3 4 5
7 8 9 10 11 12	4 5 6 7 8 9
13 14 15 16 17 18	8 9 10 11 12 13
19 20 21 22 23 24	12 13 14 15 16 17
25 26 27 28 29 30	16 17 18 19 20 21
31 32	20 21 22 23 24 25
	24 25 26 27 28 29
	28 29 30 31 32 1

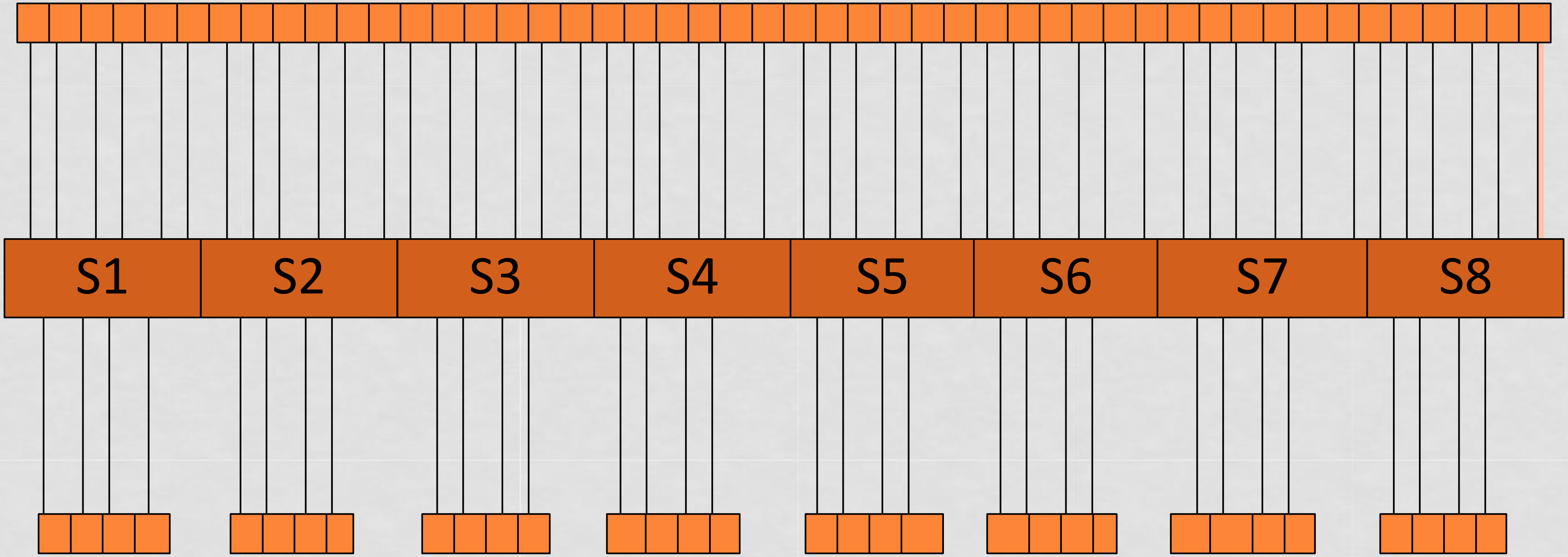
- Example: 46th bit of output block (counting from left, starting from bit 1) is equal to the 31st bit of input block

SUBSTITUTION BOXES: S

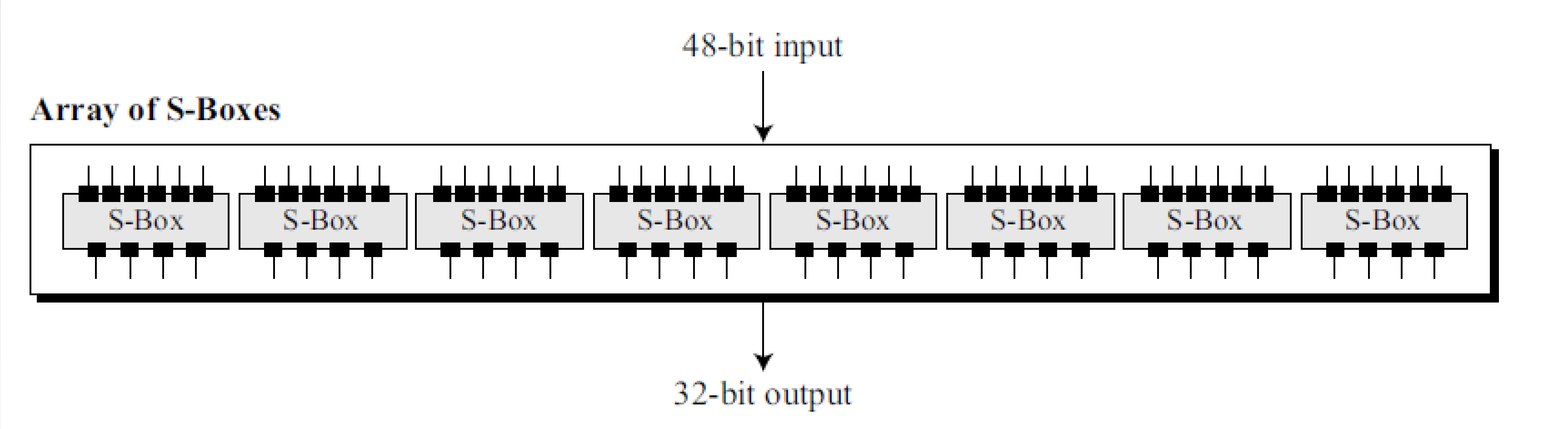
- The **substitution boxes (S-boxes)** map a 6-bit input block to a 4-bit output block

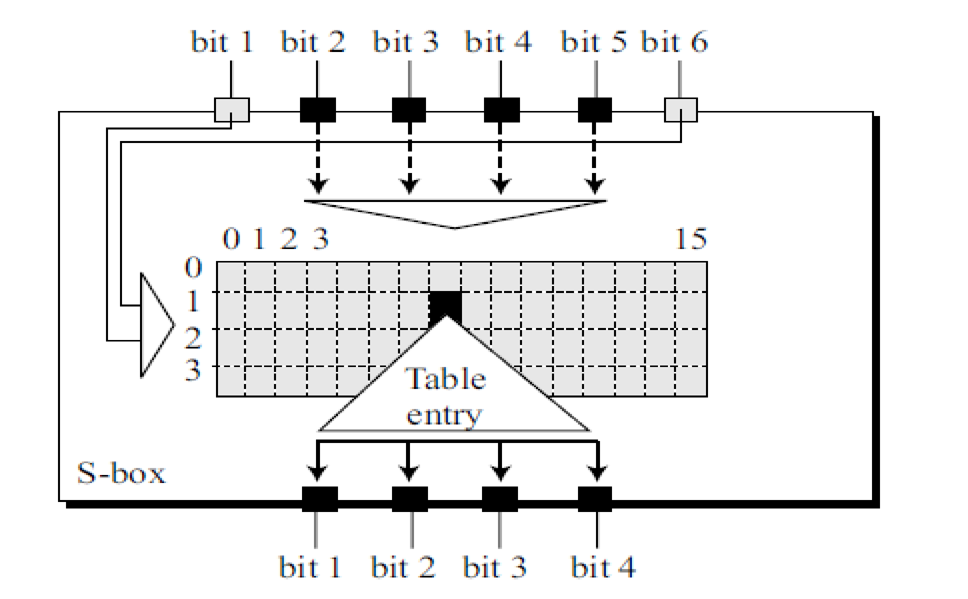
- There are 8 S-boxes, so the 48-bit input block is mapped to a 32-bit output block

SUBSTITUTION BOXES: S



S Box


48-bit input
Array of S-Boxes
S-Box
S-Box
S-Box
S-Box
S-Box
S-Box
S-Box
S-Box
32-bit output


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

S Box

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

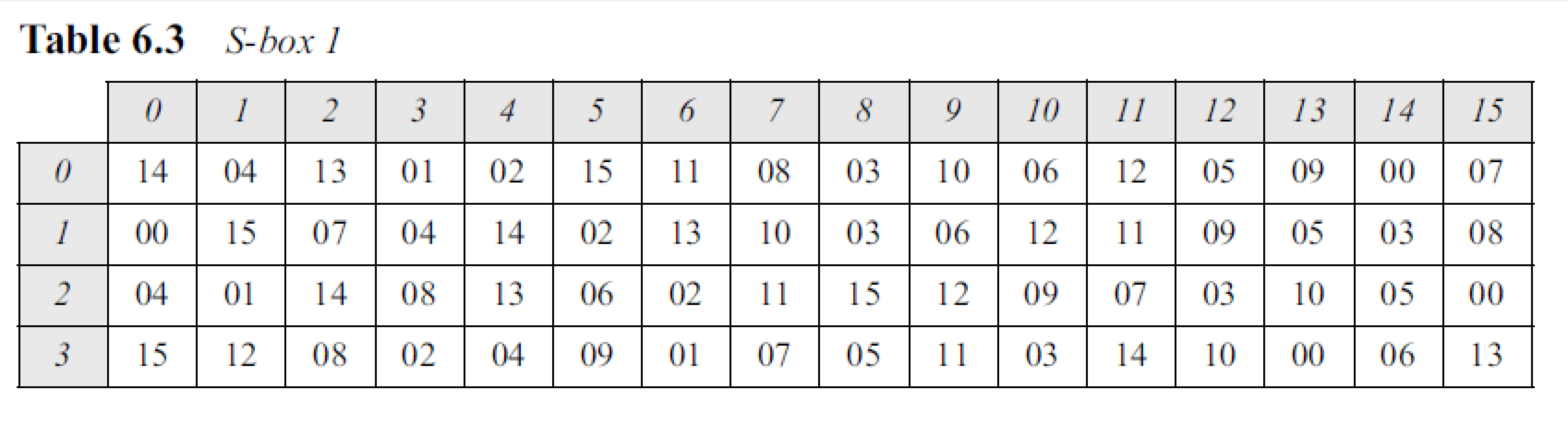


Table 6.3 *S-box 1*

	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

S Box

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

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, in Table 6.3 (S-box 1). The result is 12 in decimal, which in binary is **1100**. So the input 100011

yields the output 1100.

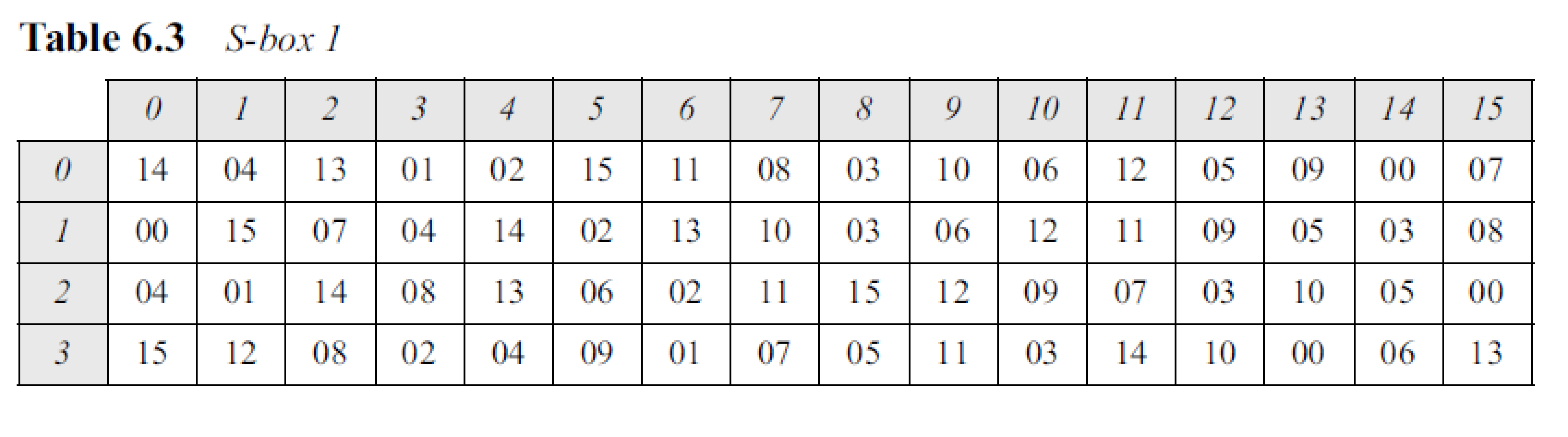


Table 6.3 *S-box 1*

	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

S Box

Example: The input to S-box 8 is 000000. What is the output?

Output: ???

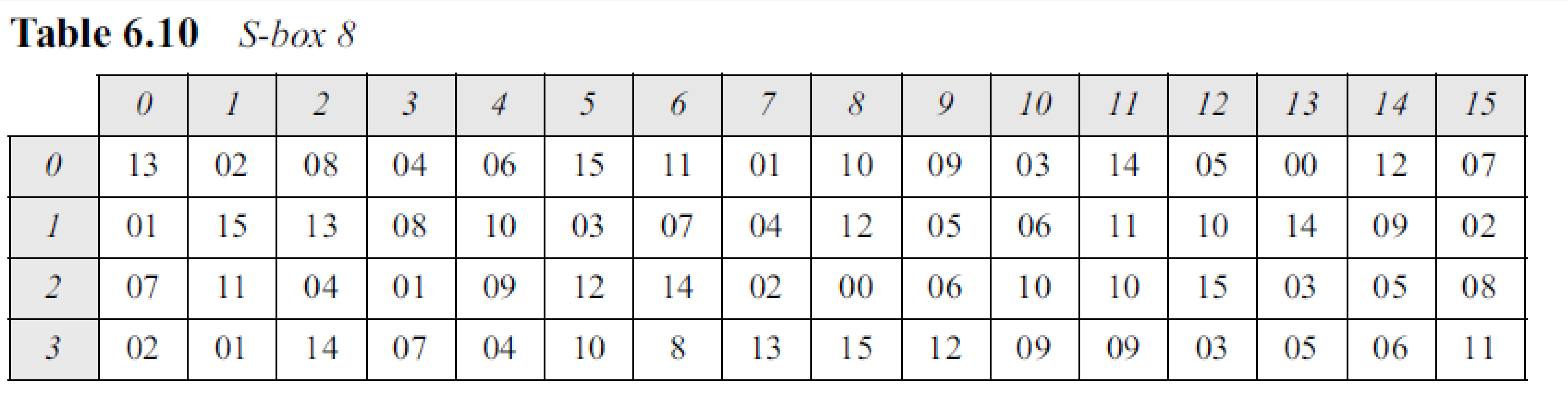


Table 6.10 *S-box 8*

	0	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*
0	13	02	08	04	06	15	11	01	10	09	03	14	05	00	12	07
1	01	15	13	08	10	03	07	04	12	05	06	11	10	14	09	02
2	07	11	04	01	09	12	14	02	00	06	10	10	15	03	05	08
3	02	01	14	07	04	10	8	13	15	12	09	09	03	05	06	11

S Box

Example: The input to S-box 8 is 000000. What is the output?

Output: If we write the first and the sixth bits together, we get 00 in binary, which is 0 in decimal. The remaining bits are 0000 in binary, which is 0 in decimal. We look for the value in row 0, column 0, in Table 6.10 (S-box 8). The result is 13 in decimal, which is 1101 in binary. So the input 000000 yields the output 1101**.**

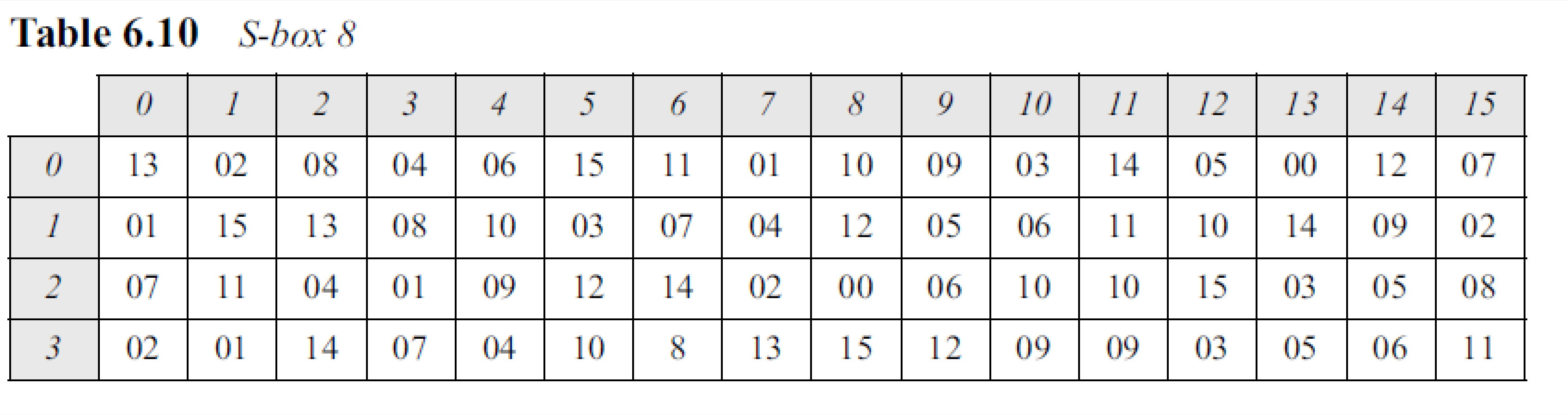


Table 6.10 *S-box 8*

	0	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*
0	13	02	08	04	06	15	11	01	10	09	03	14	05	00	12	07
1	01	15	13	08	10	03	07	04	12	05	06	11	10	14	09	02
2	07	11	04	01	09	12	14	02	00	06	10	10	15	03	05	08
3	02	01	14	07	04	10	8	13	15	12	09	09	03	05	06	11

PERMUTATION: P

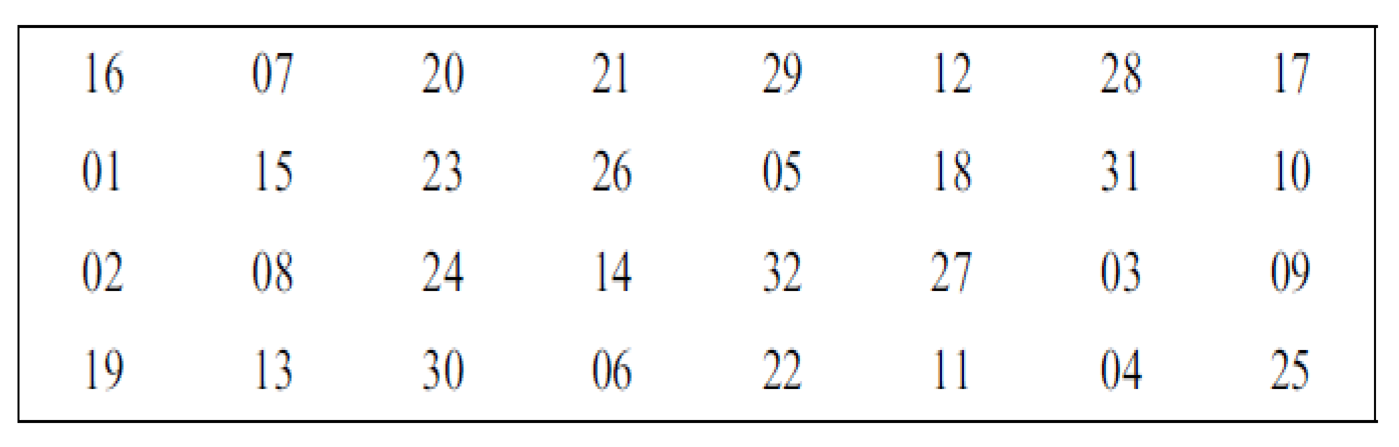
- The 32-bit output of the S-boxes is then bitwise permutated (i.e., the bits within the block are rearranged)
- This is done using the following permutation table:

Output	Input
1 2 3 4	16 7 20 21
5 6 7 8	29 12 28 17
9 10 11 12	1 15 23 26
13 14 15 16	5 18 31 10
17 18 19 20	2 8 24 14
21 22 23 24	32 27 3 9
25 26 27 28	19 13 30 6
29 30 31 32	22 11 4 25

- Example: 25th bit of output block is equal to the 19th bit of the input block.

DES 4-1: Straight Permutations

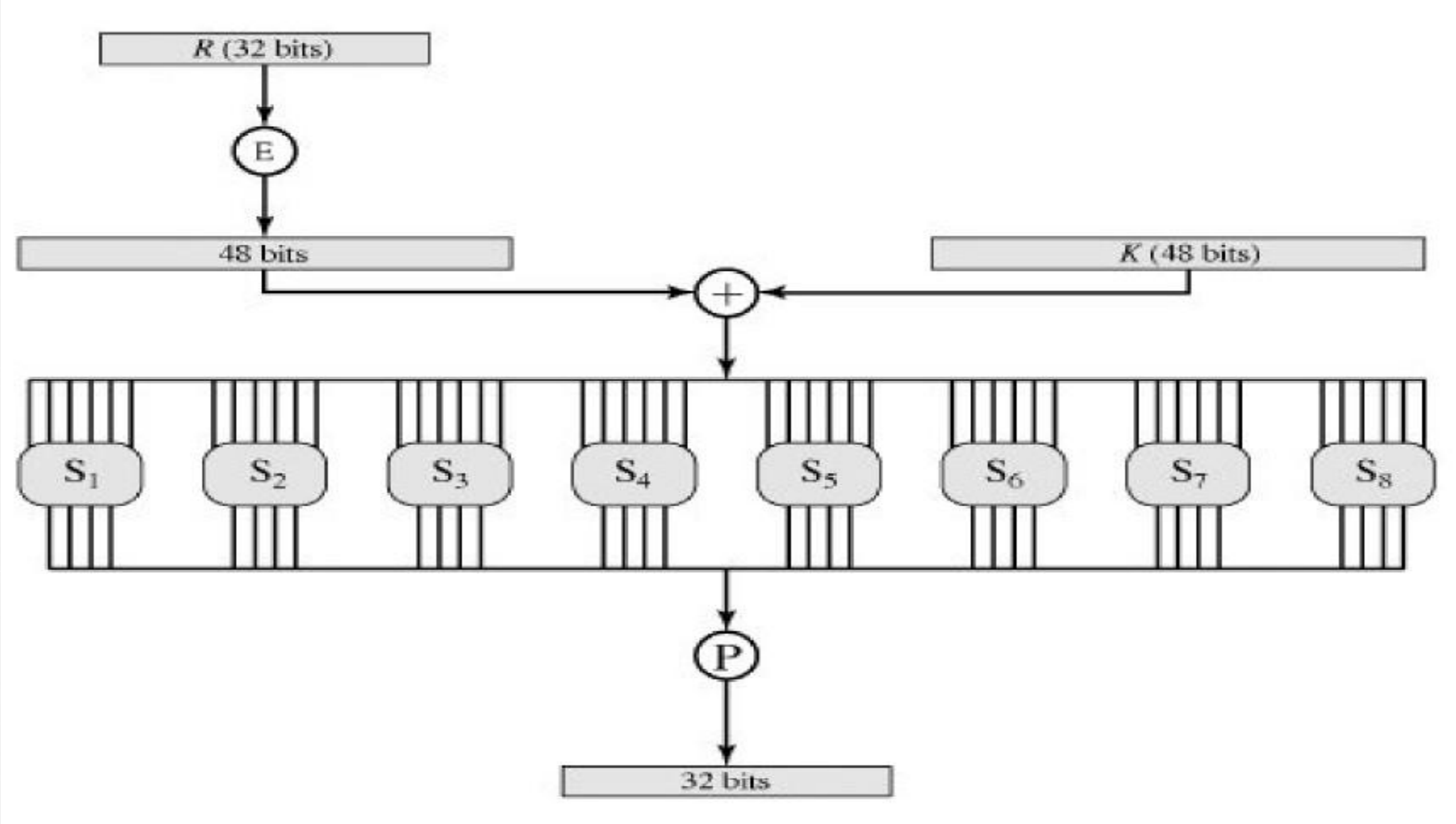
32 bit output from S-Box as Input



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

32 bit output from straight Permutations

SUBSTITUTION BOXES: S



KEY GENERATION:
Round key-generator create sixteen 48-bit keys out of a 56-bit cipher key. Actual key is of 64 bit from which 8 extra parity bits are dropped.

Parity Drop→ (8, 16, 24, 32, 40, 48, 56, 64)

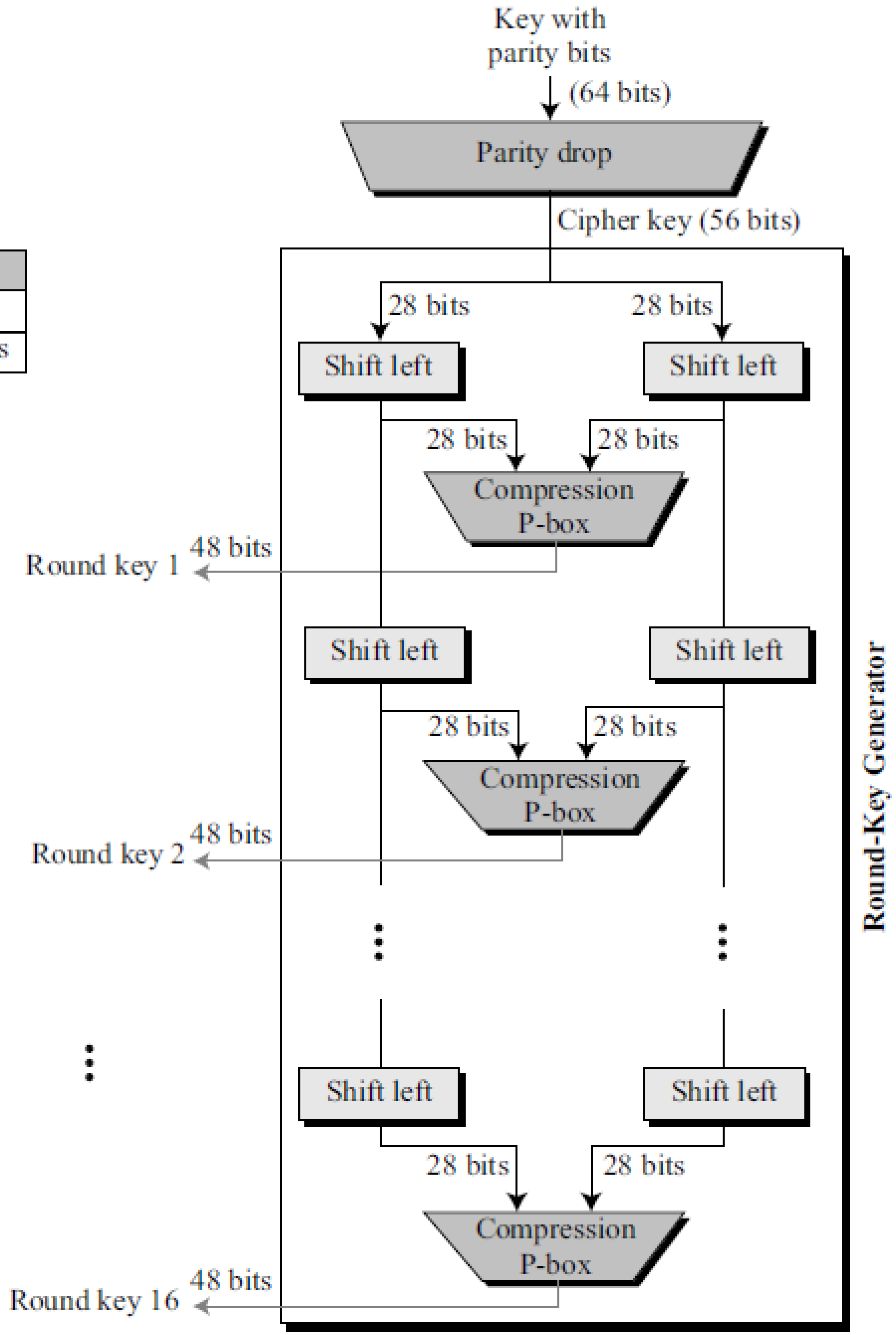
Shifting:

1, 2, 9, 16 → One bit

Others → Two bits.

Shifting

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



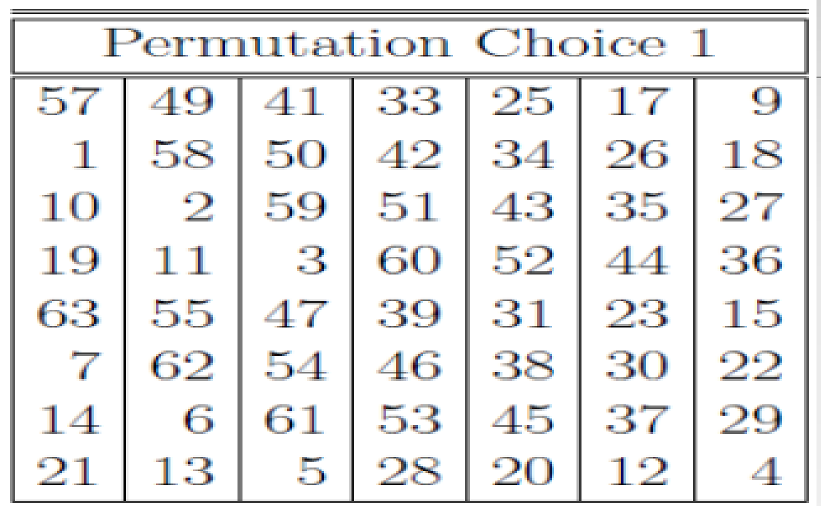
Round Key Generation

- The initial 64-bit key may be represented as 8 bytes, with the last bit of each byte used as a parity bit.
- The relevant 56 bits are subject to a permutation at the beginning before any round keys are generated. This is our Permutation Choice 1.
- At the beginning of each round, we divide the 56 relevant key bits into two 28 bit halves and circularly shift each half by one or two bits.
- For generating round key, we join together the two halves and apply a 56 bit to 48 bit contracting permutation (Permutation Choice 2) to the joined bit pattern. The resulting 48 bits constitute our round key.
- The two halves generated in each round are fed as the two halves going into the next round.

DES KEY SCHEDULE

- The 64-bit key is used as input to the algorithm.
- The key is first subjected to a permutation governed by the table labelled permuted choice one.
- The resulting 56-bit key is then treated as two 28-bit quantities, labelled C_0 and D_0.
- At each round, C_{i-1} and D_{i-1} are separately subjected to circular left shift or rotation , of 1 to 2 bits, these values serve as input to the next round.
- They also serve as input to permutation Choice Two, which produce 48-bit output that serve as input to the function $F(R_{i-1},K_i)$.

Permutation choice 1 of the Encryption Key:

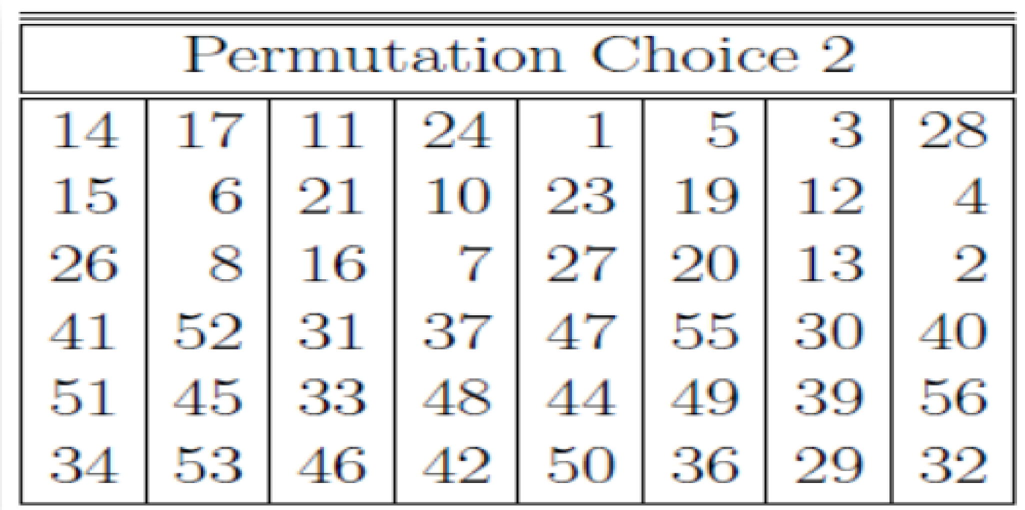


Permutation Choice 1

57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

Note that the bit positions assume that the key bits are addressed 1 through 64 in an 8-byte bit pattern. But note that the last bit of each byte is used as a parity bit. Also note that the permutation shown is not a table, in the sense that the rows and the columns do not carry any special and separate meanings. The permutation order for the bits is given by reading the entries shown from the upper left corner to the lower right corner.

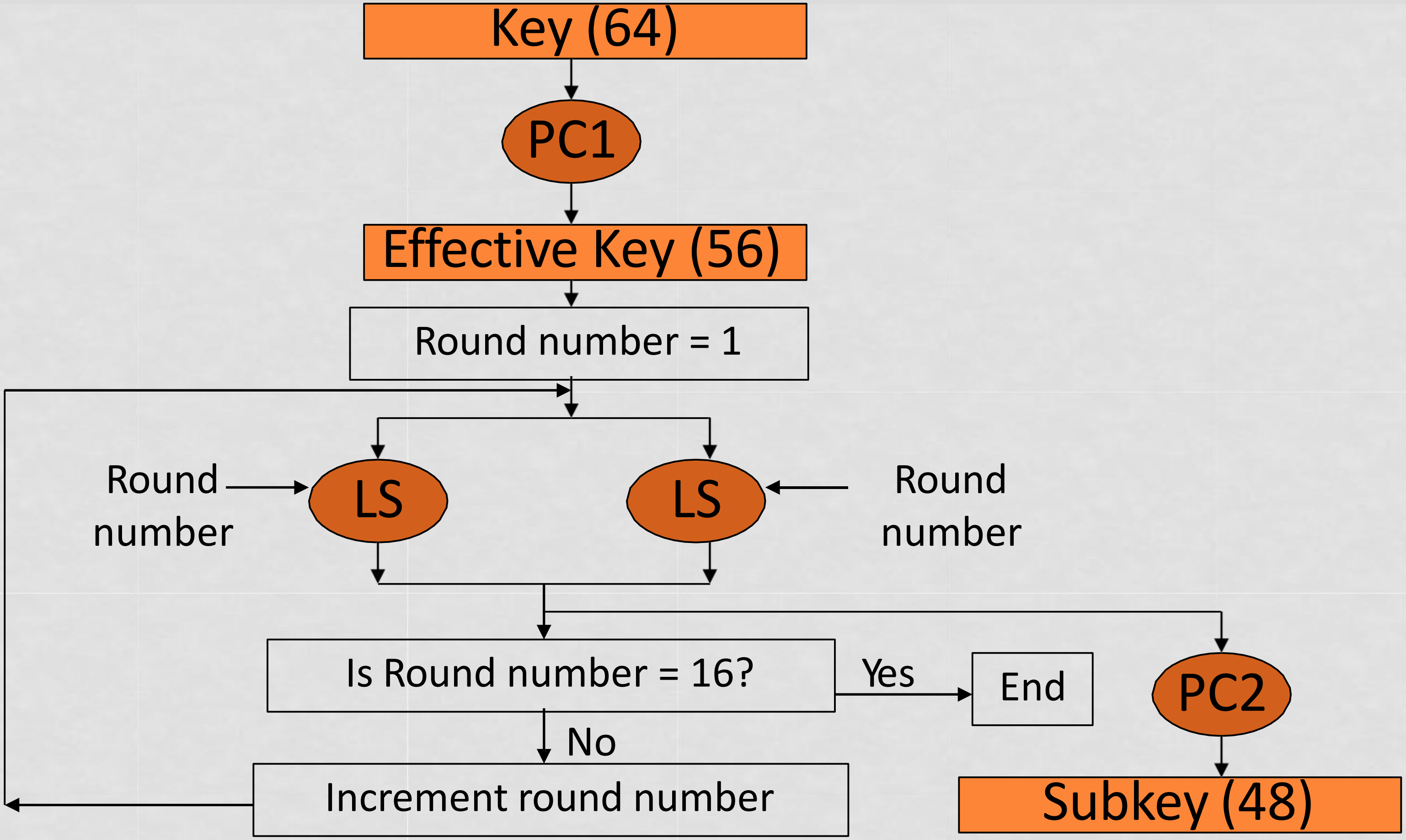
Contraction-permutation that Generates the 48-bit Round Key from the 56 Key:



Permutation Choice 2							
14	17	11	24	1	5	3	28
15	6	21	10	23	19	12	4
26	8	16	7	27	20	13	2
41	52	31	37	47	55	30	40
51	45	33	48	44	49	39	56
34	53	46	42	50	36	29	32

Also note that the permutation shown is not a table, in the sense that the rows and the columns do not carry any special and separate meanings. The permutation order for the bits is given by reading the entries shown from the upper left corner to the lower right corner.

DES KEY SCHEDULE



DES Decryption

- The decryption uses the same algorithm as encryption , except the application of subkeys are reversed.

THE STRENGTH OF DES

- The key length is 56 bits, there are 2^{56} possible keys, 7.2 X 10^{16} keys, brute force attack impractical.
- Assuming, on average half the key space has to be searched, a single machine performing on DES encryption per microsec would take more than 1000 years to break the cipher.
- In 1977, Deffie Hallman, postulated that the technology existed to build a parallel machine with 1 million encryption devices, each of which could perform one encryption per microsec.
- This would bring search time to about 10 hours , cost $20 million.

DES STRENGTH

- DES finally proved insecure in July 1998,
- When the Electronic Frontier Foundation(EFF) declared broken DES using “DES cracker”, cost $250,000 and publish the description of cracker , enables others to build their own, with decrease price of hardware and increase speed make DES worthless.

SUMMARY

- Types of Cryptography
- Asymmetric Key Cryptography
- Symmetric Key Cryptography
  - Requirements for secure symmetric encryption
  - Approaches for attacking a symmetric encryption scheme
- Block Ciphers vs. Stream Ciphers
- Data Encryption Standard (DES)
  - DES Encryption /Decryption
  - Strength/Weakness of DES