Information Security

Dr. Irfan Yousuf
Department of Computer Science (New Campus)
UET, Lahore
(Week 5: September 30 - October 04, 2024)

Outline

- Data Encryption Standard (DES)

Data Encryption Standard (DES)

- This algorithm adopted in 1977 by the National Institute of Standards and Technology (NIST).
- The algorithm itself is referred to as the Data Encryption Algorithm (DEA).
- For DES, data are encrypted in **64-bit blocks** using a **56-bit key**.
- The algorithm transforms 64-bit input in a series of steps into a 64-bit output. The same steps, with the same key, are used to reverse the encryption.

DES Encryption Algorithm

- The general structure of the DES consists of (1) key schedule, (2) round function and (3) initial and final permutation.
- **Step1:** Plaintext is broken into blocks of length 64 bits.
- **Step2:** The 64-bit block undergoes an initial permutation (IP) using **initial permutation** IP table, IP(M).
- **Step3:** The 64-bit permuted input is divided into two 32-bit blocks: left (L) and right (R). The initial values of the left and right blocks are denoted L_0 and R_0.

DES Encryption Algorithm

- **Step4:** There are 16 rounds of operations on the L and R blocks. During each round, the following formula is applied:
- $L_n = R_{n-1}$
  $R_n = L_{n-1}$ XOR $F(R_{n-1}, K_n)$

DES Encryption Algorithm

Step5: The function F(.) represents the heart of the DES algorithm. This function implements the following operations:

1-Expansion: The right 32-bit half-block is expanded to 48 bits using the expansion permutation (E) table, $E(R_{n-1})$.

2-Key mixing: The expanded result is combined with a subkey using an XOR operation. Sixteen 48-bit subkeys (one for each round) are derived from the main key using the key schedule, $K_n + E(R_{n-1})$.

3-Substitution: After mixing in the subkeys, the block is divided into eight 6-bit pieces and fed into the substitution boxes (S-boxes), which implements nonlinear transformation. Each 6-bit piece uses as an address in the S-boxes where the first and last bits are used to address the i^{th} row and the middle four bits to address the j^{th} column in the S-boxes. The output of each S-box is 4-bit length piece. The output of all eight S-boxes is then combined into 32 bit section.

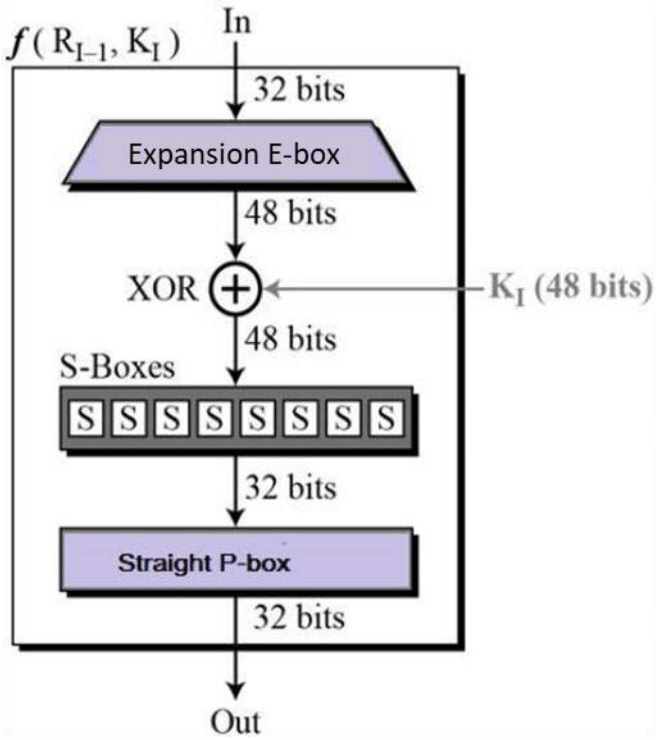
$K_n + E(R_{n-1}) = B_1B_2B_3B_4B_5B_6B_7B_8$

$S(Kn + E(Rn-1)) = S1(B_1)S2(B_2)S3(B_3)S4(B_4)S5(B_5)S6(B_6)S7(B_7)S8(B_8)$

4-Permutation: The 32 bits outputs from the S-boxes are rearranged using the P-box, $F=P(S(K_n + E(R_{n-1})))$

DES Encryption Algorithm

Step5: The function F(.) represents the heart of the DES algorithm. This function implements the following operations:

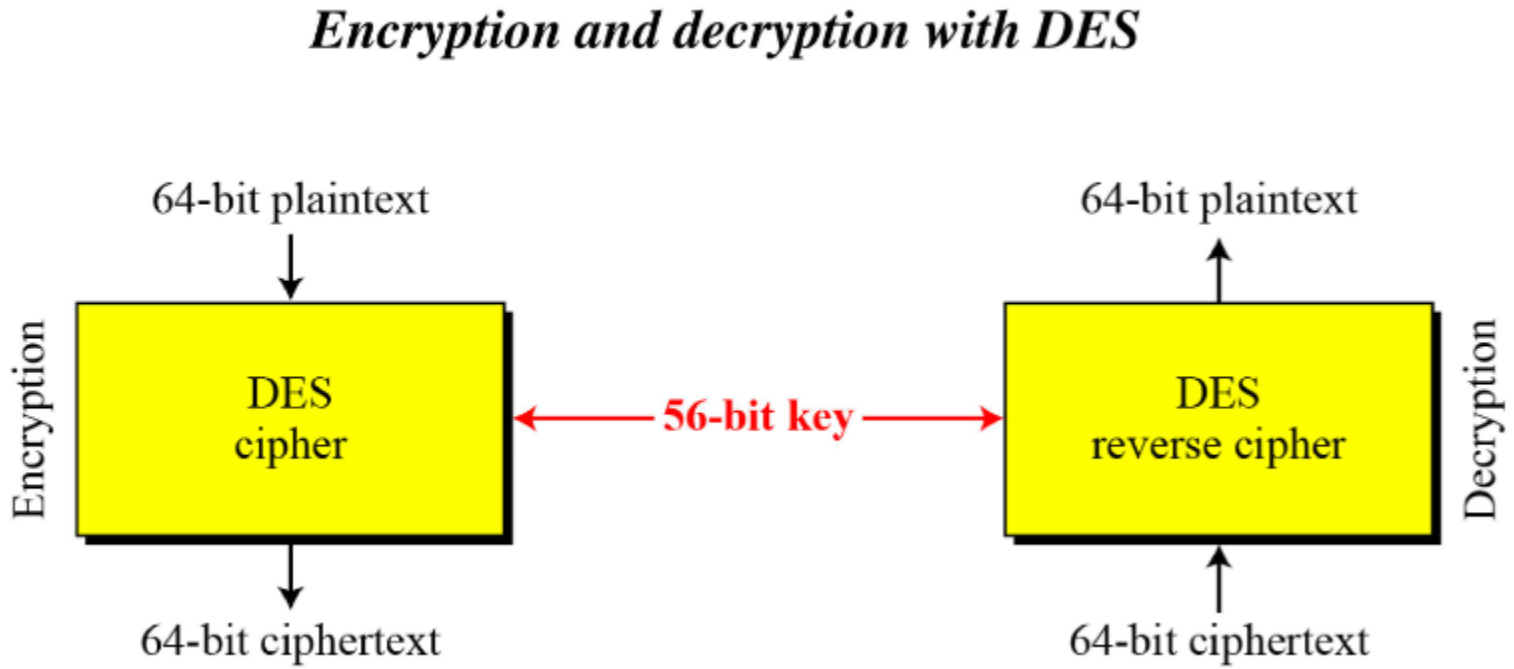


Single Round function (F) of the DES

DES Encryption Algorithm

- **Step6:** The results from the final DES round (i.e., L16 and R16) are recombined into a 64-bit value and rearranged using an inverse initial permutation (IP^{-1}) table. The output from IP^{-1} is the 64-bit ciphertext block.

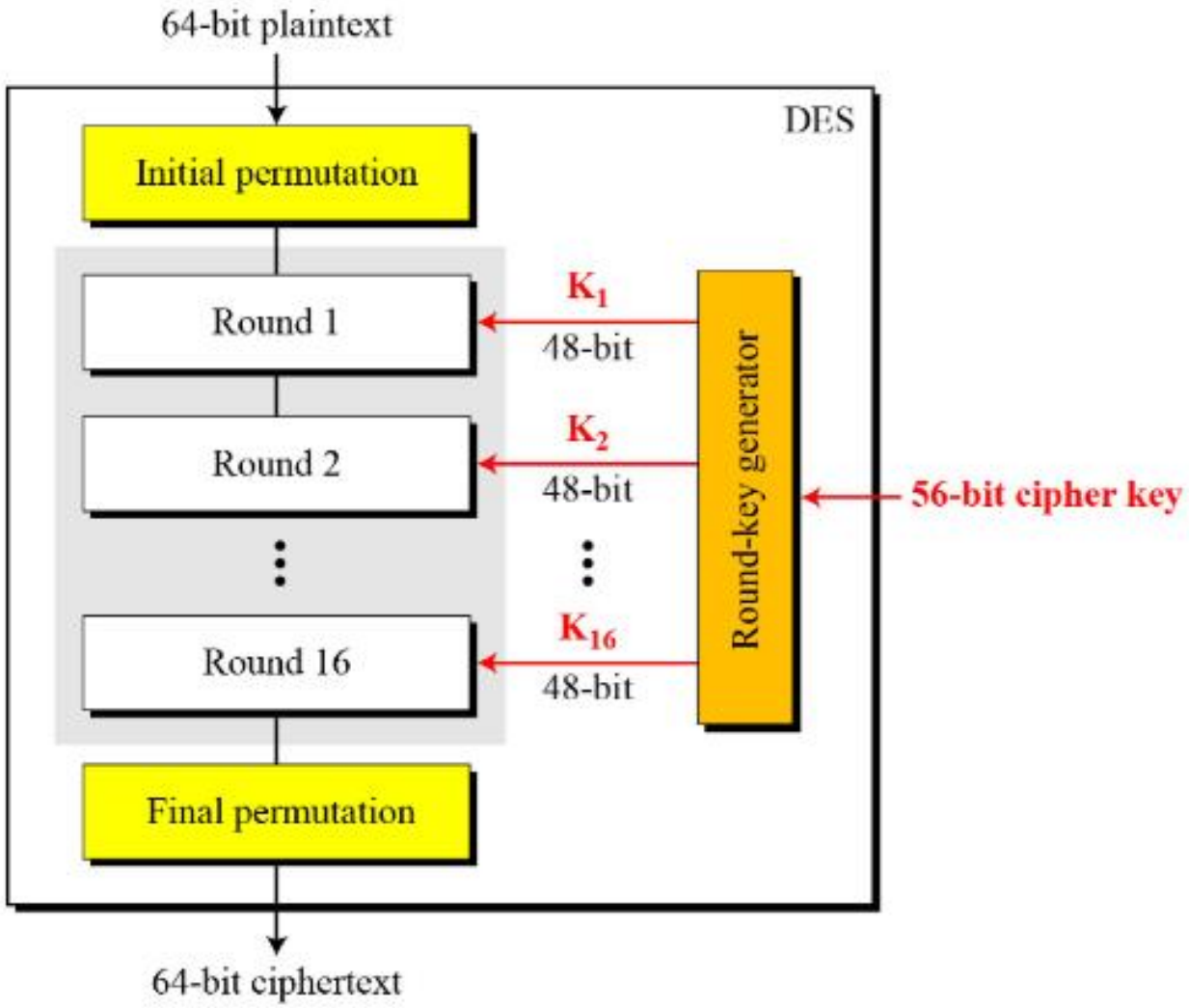
DES Encryption Algorithm



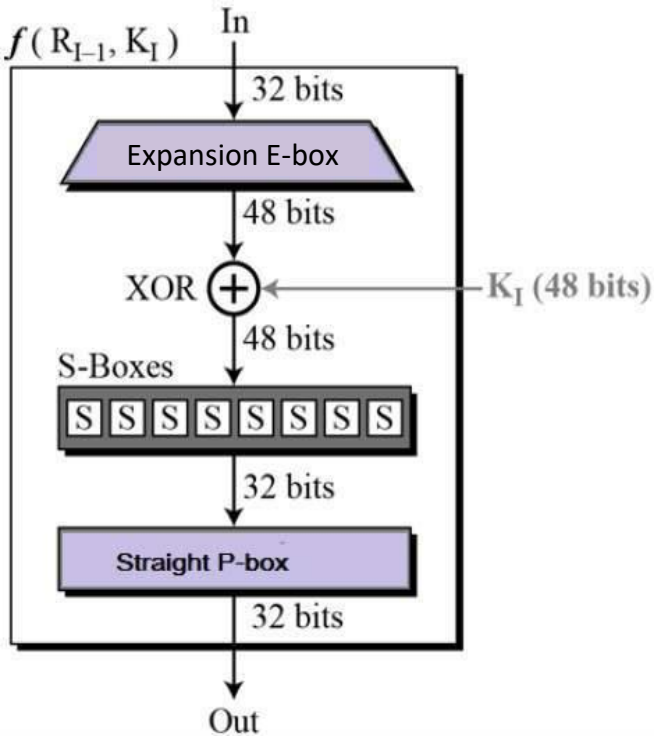
The encryption process is made of two permutations (P-boxes), which we call initial and final permutations, and sixteen Feistel rounds.

DES Encryption Algorithm

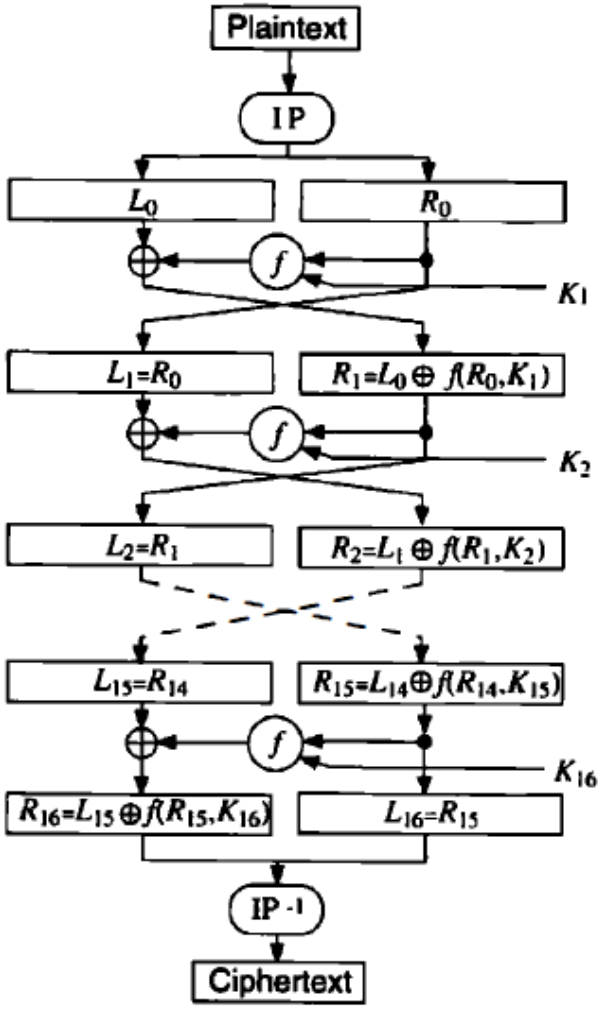
General structure of DES



DES Encryption Algorithm



Single Round function (F) of the DES



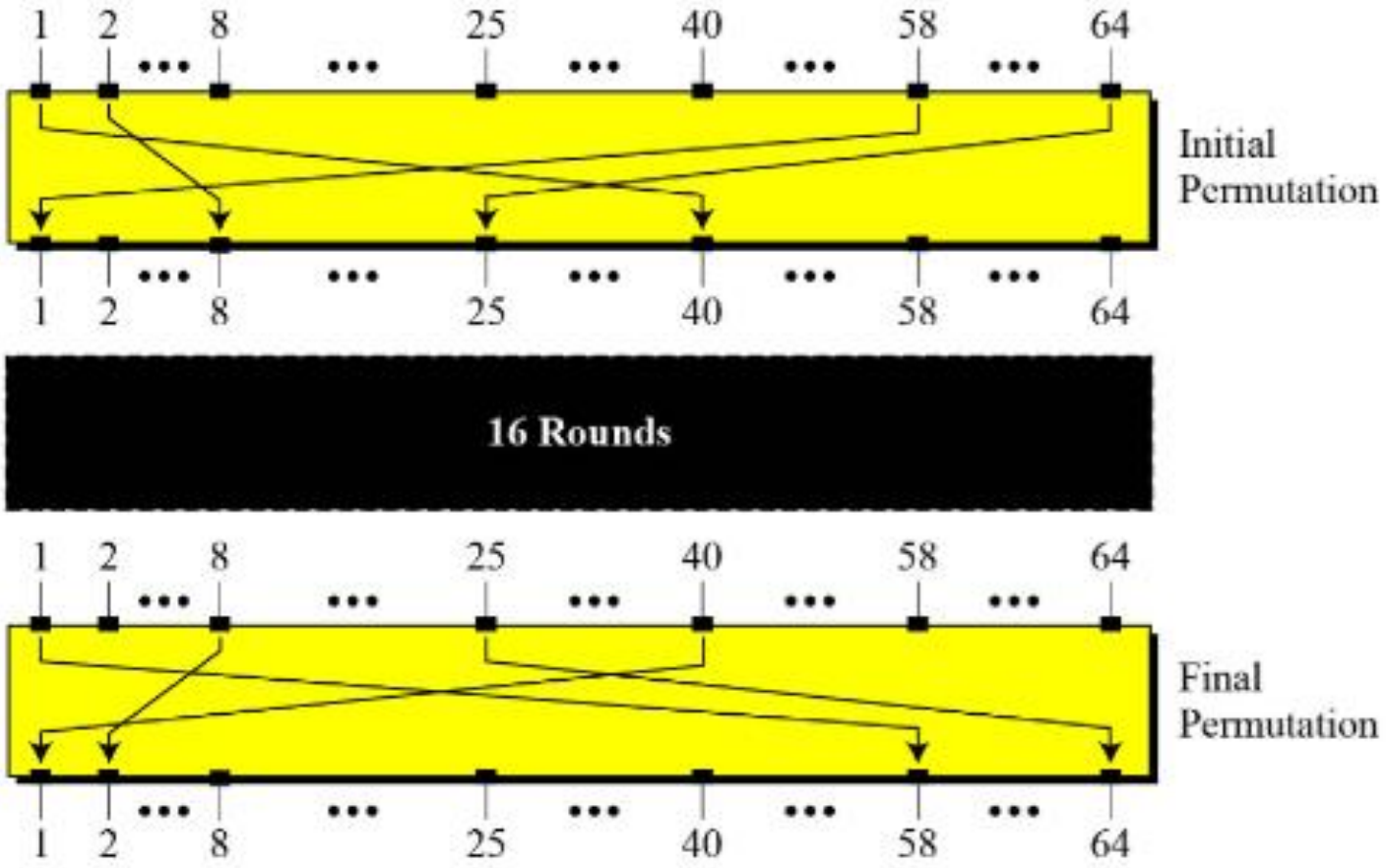
DES Encryption Flowchart

DES Encryption Algorithm

- **Step1:** Plaintext is broken into blocks of length 64 bits.
- **Step2:** The 64-bit block undergoes an initial permutation (IP) using **initial permutation** IP table, IP(M).
- **Step3:** The 64-bit permuted input is divided into two 32-bit blocks: left (L) and right (R). The initial values of the left and right blocks are denoted L_0 and R_0.
- **Step4:** There are 16 rounds of operations on the L and R blocks. During each round, the following formula is applied:
- $L_n = R_{n-1}$
  $R_n = L_{n-1}$ XOR $F(R_{n-1},K_n)$

DES Encryption Algorithm

Initial and final permutation steps in DES



DES Encryption Algorithm

Initial and final permutation tables

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

How to read this table?

The 58th bit of input x will be the first bit of output IP(x), the 50th bit of x is the second bit of IP(x), etc.

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

DES Encryption Algorithm

Example 1

Find the output of the initial permutation box when the input is given in hexadecimal as:

0x0000 0080 0000 0002

0000 0000 0000 0000 0000 0000 1000 0000
0000 0000 0000 0000 0000 0000 0000 0010

Solution

Only bit 25 and bit 64 are 1s; the other bits are 0s. In the final permutation, bit 25 becomes bit 64 and bit 63 becomes bit 15. The result is

0x0002 0000 0000 0001

DES Encryption Algorithm

Example 2

Prove that the initial and final permutations are the inverse of each other by finding the output of the final permutation if the input is

0x0002 0000 0000 0001

Solution

The input has only two 1s; the output must also have only two 1s. Using Table 6.1, we can find the output related to these two bits. Bit 15 in the input becomes bit 63 in the output. Bit 64 in the input becomes bit 25 in the output. So the output has only two 1s, bit 25 and bit 63. The result in hexadecimal is

0x0000 0080 0000 0002

DES Encryption Algorithm

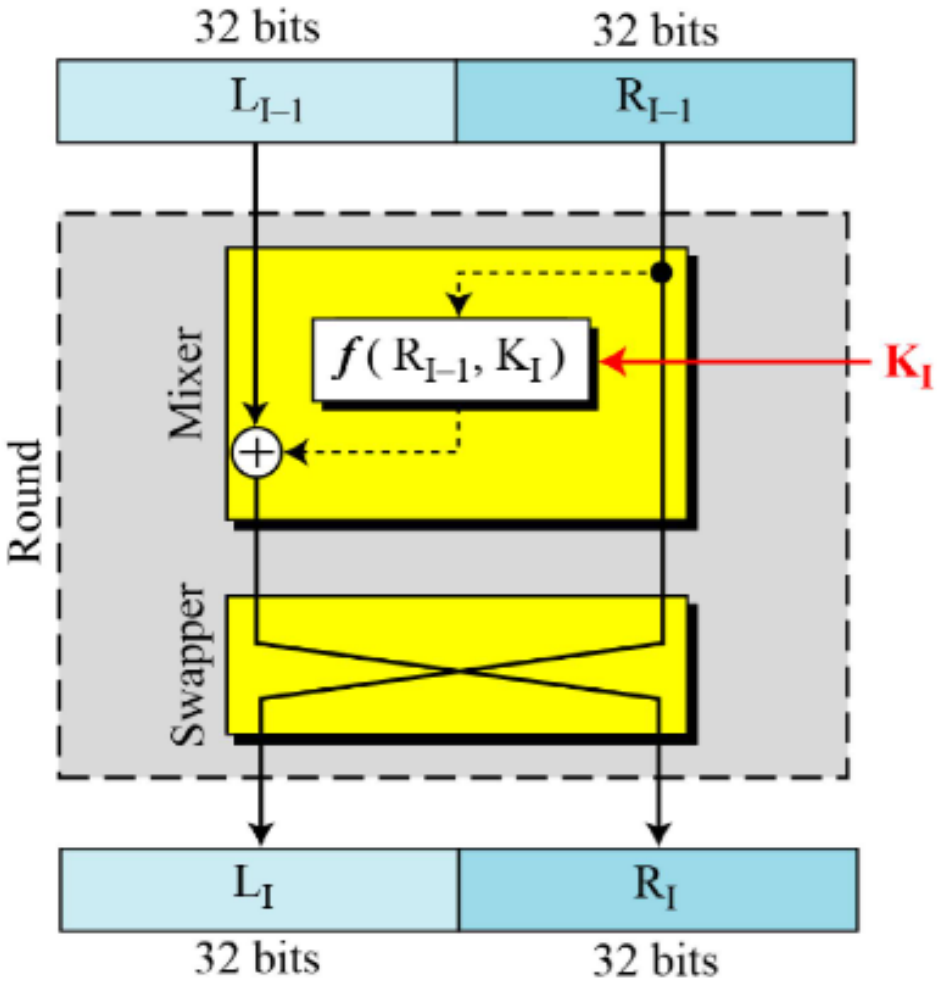
Plaintext (in binary):
0000 0001 0010 0011 0100 0101 0110 0111
1000 1001 1010 1111 1100 1101 1110 1111

After Initial Permutation (example):
1100 1100 1111 0000 1010 1010 1100 1100
1010 1010 1100 1100 1111 0000 1100 1100

DES Encryption Algorithm

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

A round in DES (encryption site)



DES Encryption Algorithm

Step5: The function F(.) represents the heart of the DES algorithm. This function implements the following operations:

1-Expansion: The right 32-bit half-block is expanded to 48 bits using the expansion permutation (E) table, $E(R_{n-1})$.

2-Key mixing: The expanded result is combined with a subkey using an XOR operation. Sixteen 48-bit subkeys (one for each round) are derived from the main key using the key schedule, $K_n + E(R_{n-1})$.

3-Substitution: After mixing in the subkeys, the block is divided into eight 6-bit pieces and fed into the substitution boxes (S-boxes), which implements nonlinear transformation. Each 6-bit piece uses as an address in the S-boxes where the first and last bits are used to address the i^{th} row and the middle four bits to address the j^{th} column in the S-boxes. The output of each S-box is 4-bit length piece. The output of all eight S-boxes is then combined into 32 bit section.

$K_n + E(R_{n-1}) = B_1B_2B_3B_4B_5B_6B_7B_8$

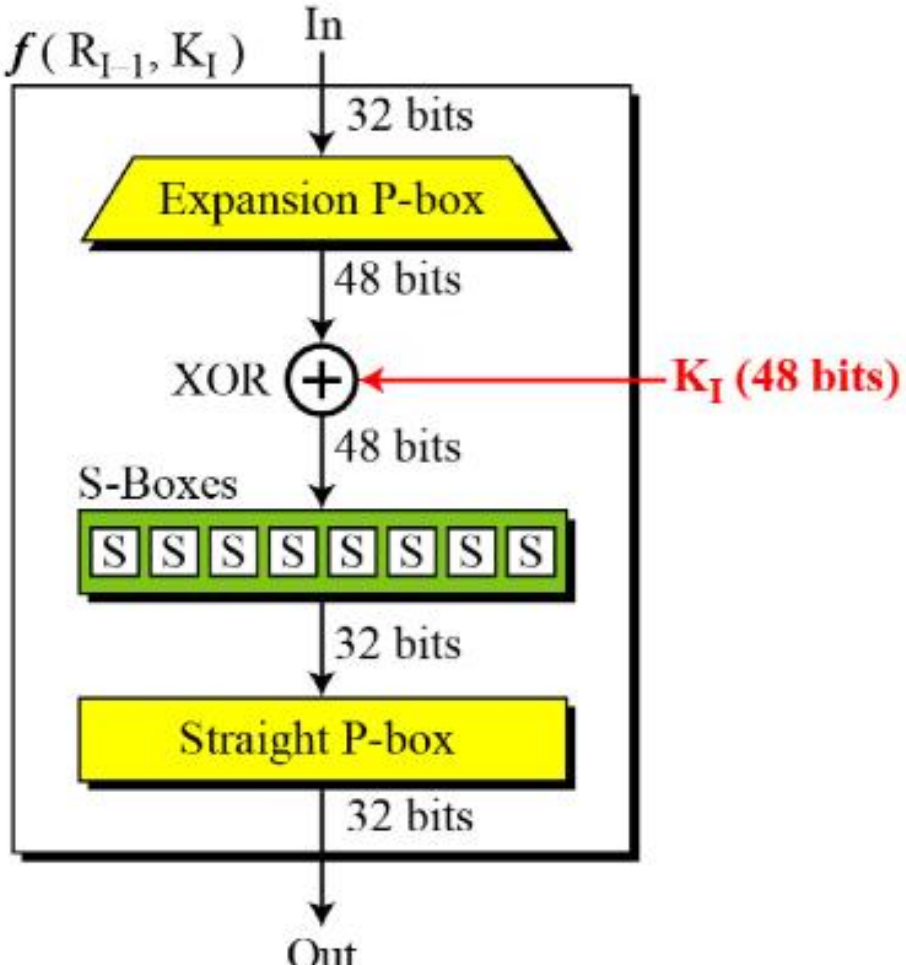
$S(Kn + E(Rn-1)) = S1(B_1)S2(B_2)S3(B_3)S4(B_4)S5(B_5)S6(B_6)S7(B_7)S8(B_8)$

4-Permutation: The 32 bits outputs from the S-boxes are rearranged using the P-box, $F=P(S(K_n + E(R_{n-1})))$

DES Encryption Algorithm

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.

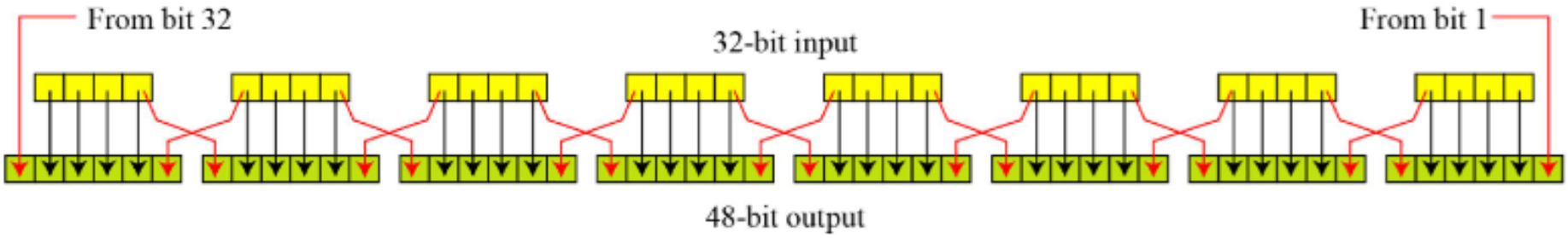
DES function



DES Encryption Algorithm

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



DES Encryption Algorithm

Although the relationship between the input and output can be defined mathematically,

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

DES Encryption Algorithm

Apply the expansion permutation on R0:

R0 (32 bits): 10101010 10101010 10101010 10101010

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

Expanded R0 (48 bits): 01010101 01011010 10100101 01101010 01011010 10101010

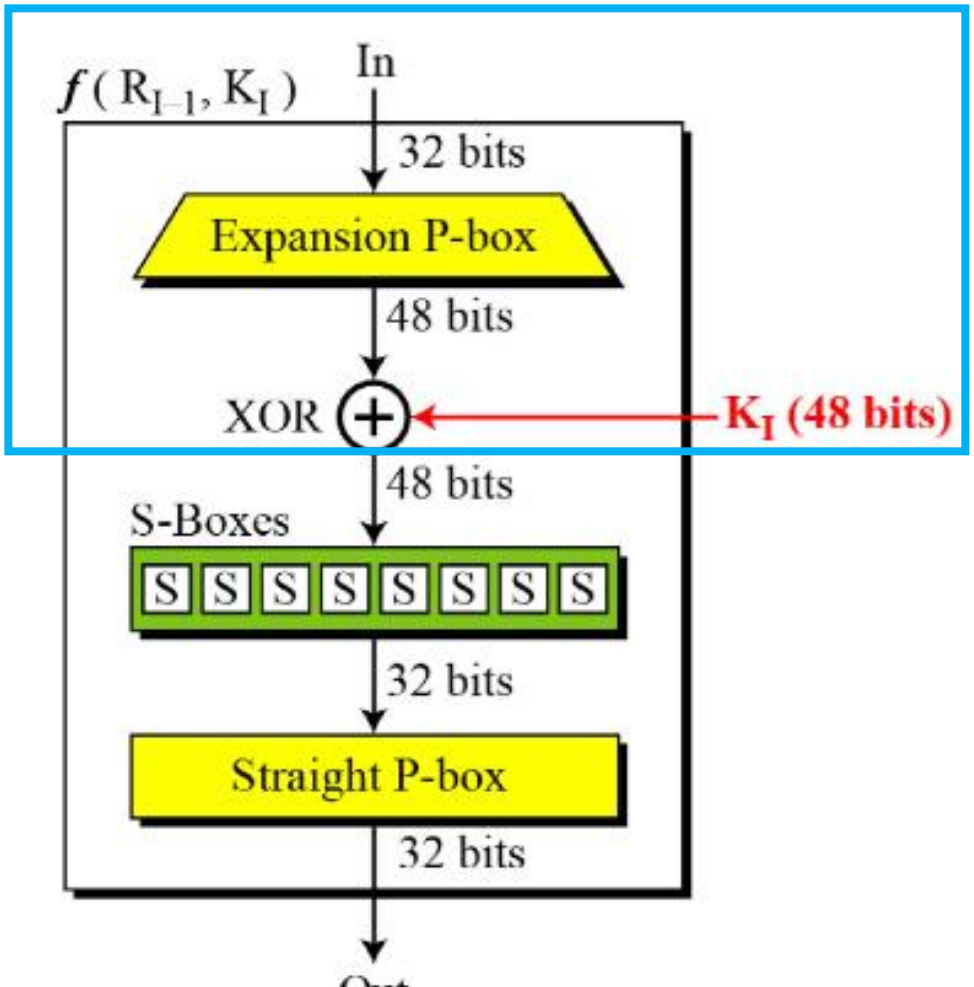
DES Encryption Algorithm

After the expansion permutation, DES uses the XOR operation on the expanded right section and the round key. Note that both the right section and the key are 48-bits in length. Also note that the round key is used only in this operation.

DES Encryption Algorithm

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.

DES function

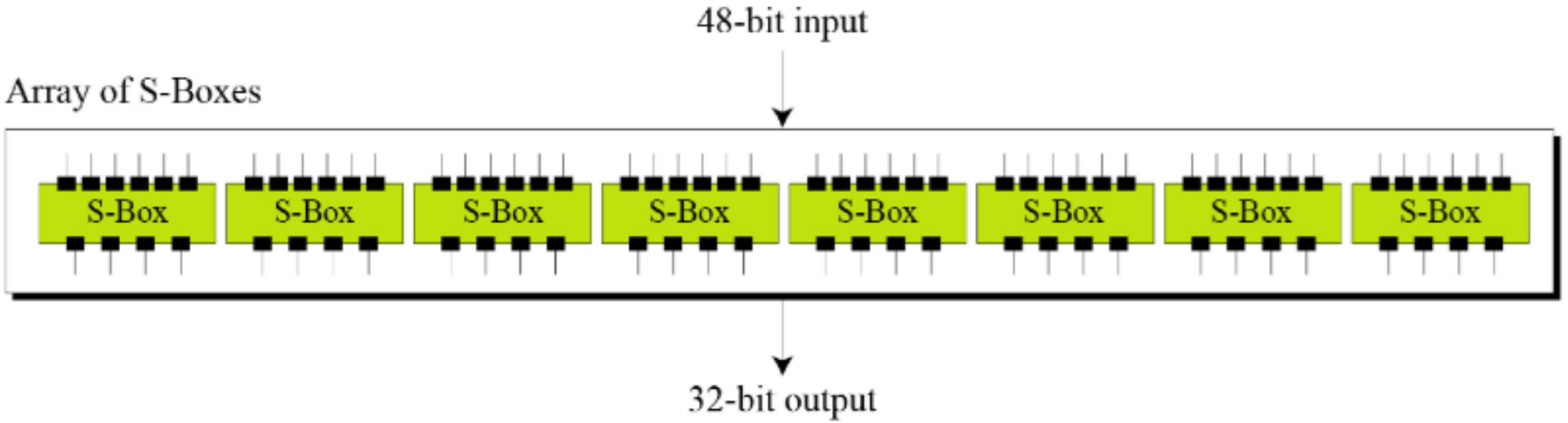


DES Encryption Algorithm

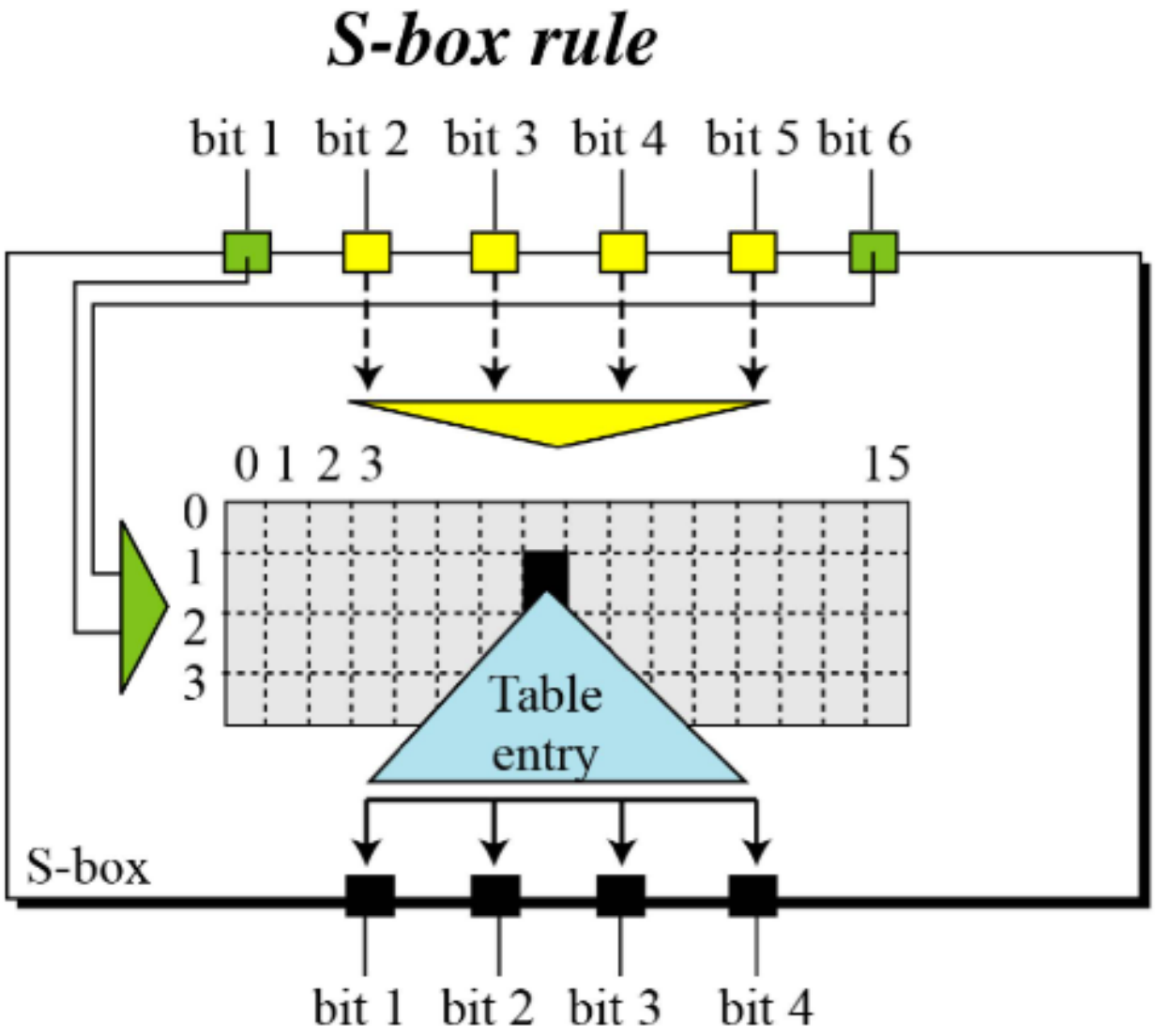
S-Boxes

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-boxes



DES Encryption Algorithm



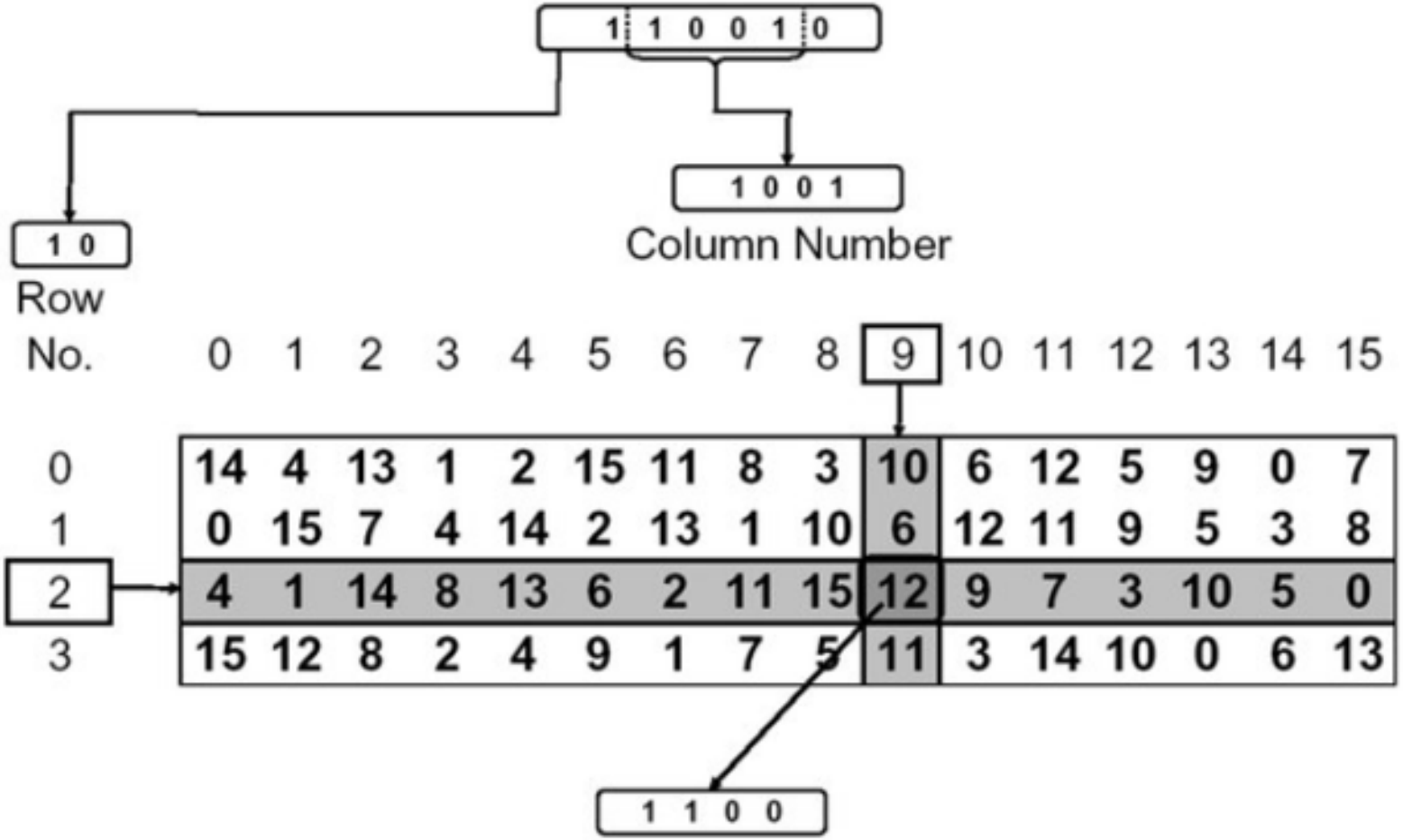
DES Encryption Algorithm

Table shows the permutation for S-box 1. For the rest of the boxes see the textbook.

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

DES Encryption Algorithm



Application of S-box in DES Algorithm

DES Encryption Algorithm

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

Solution

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.

DES Encryption Algorithm

After XOR operation:
10110101 00010000 00011111 01111000 01010000 00011101

101101 010001 000000 011111 011110 000101 000000 011101

DES has 8 S-boxes

DES Tables

S_1

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

S_2

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

S_3

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

S_4

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

S_5

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

S_6

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

S_7

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

S_8

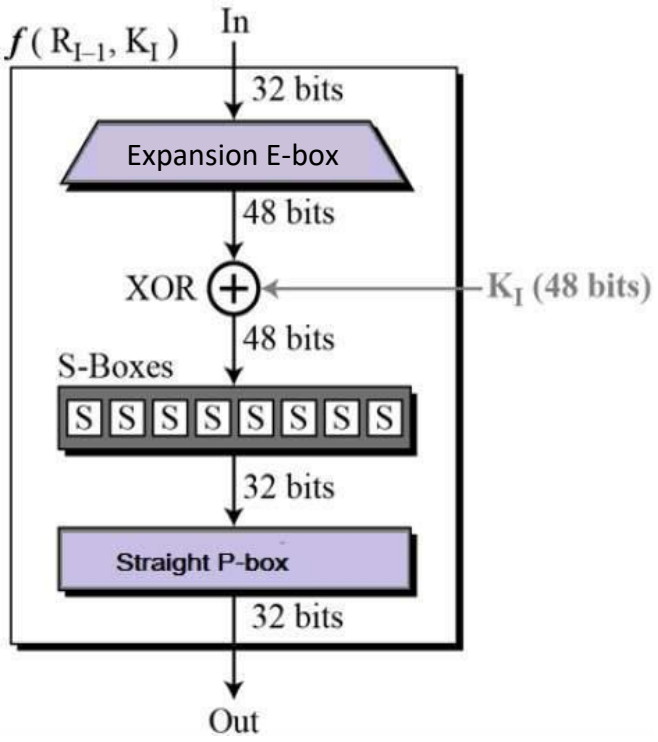
13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

DES Encryption Algorithm

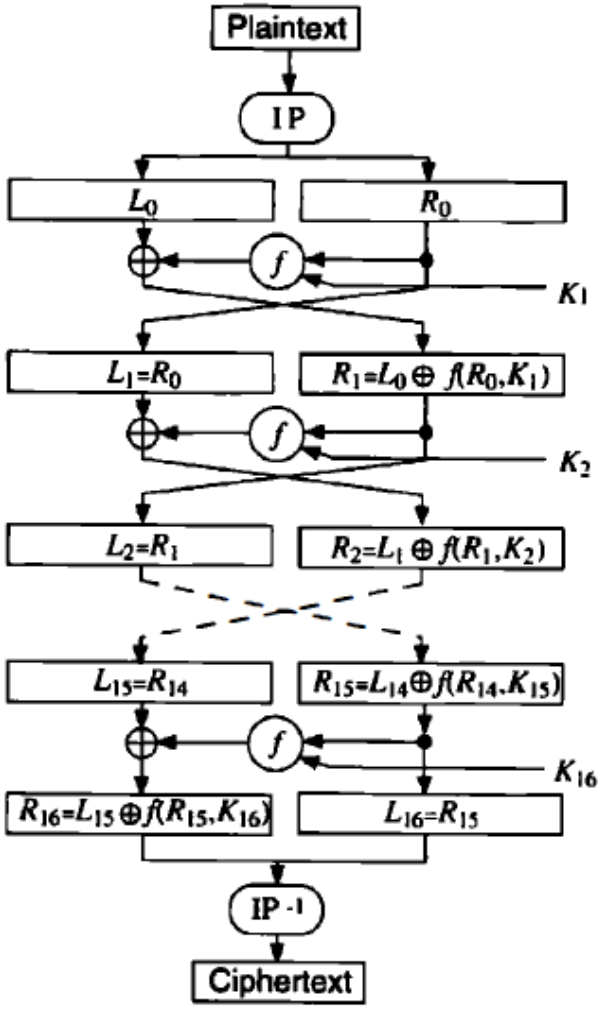
(d) Permutation Function (P)

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

DES Encryption Algorithm



Single Round function (F) of the DES



DES Encryption Flowchart

DES Encryption Algorithm

- **Step6:** The results from the final DES round (i.e., L16 and R16) are recombined into a 64-bit value and rearranged using an inverse initial permutation (IP^{-1}) table. The output from IP^{-1} is the 64-bit ciphertext block.

DES Tables

(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

(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

(c) Expansion Permutation (E)

32	1	2	3	4	5
4	5	6	7	8	9
8	9	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	1

(d) Permutation Function (P)

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

DES Tables

S_1

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

S_2

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

S_3

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

S_4

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

S_5

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

S_6

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

S_7

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

S_8

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

DES Key Schedule

Key schedule (generator):

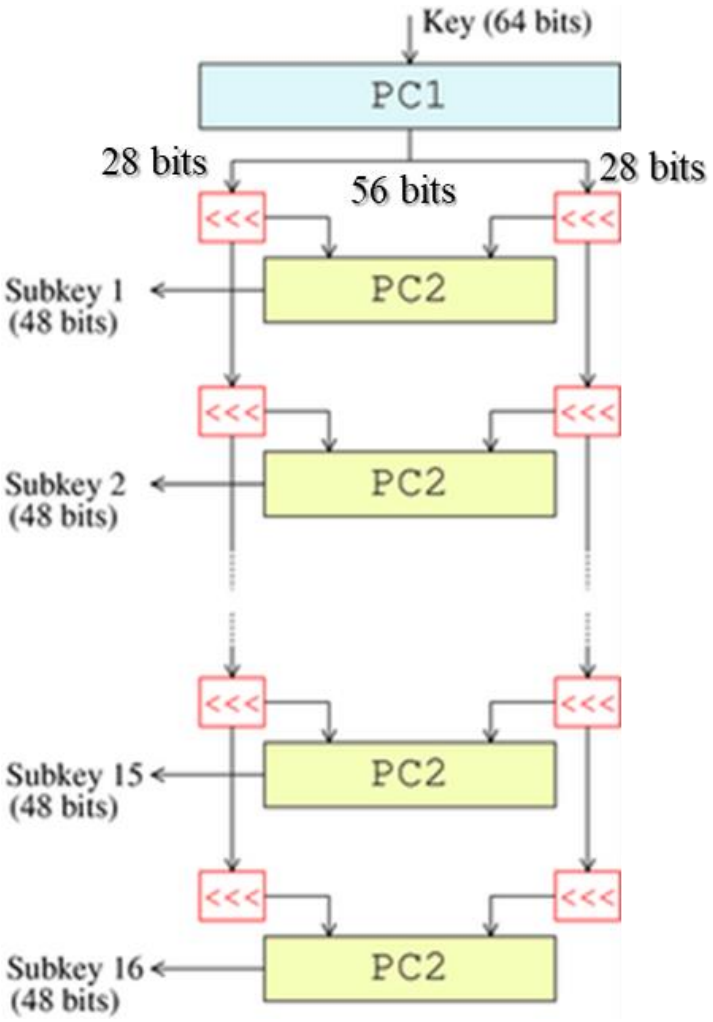
This algorithm generates the subkeys (K → K_1, K_2…K_{16}).

1- The 56 bits of the key are selected from the initial 64 by Permuted Choice 1 (PC1) table.

2- The 56 bits are divided into two 28-bit halves.

3- In each round, both halves are rotated left by one or two bits (specified for each round).

4- The 48 subkey bits are selected by Permuted Choice 2 (PC2) table (24 bits from the left half, and 24 from the right) and used in each round.



DES Key Schedule

(a) Input Key

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

(b) Permuted Choice One (PC-1)

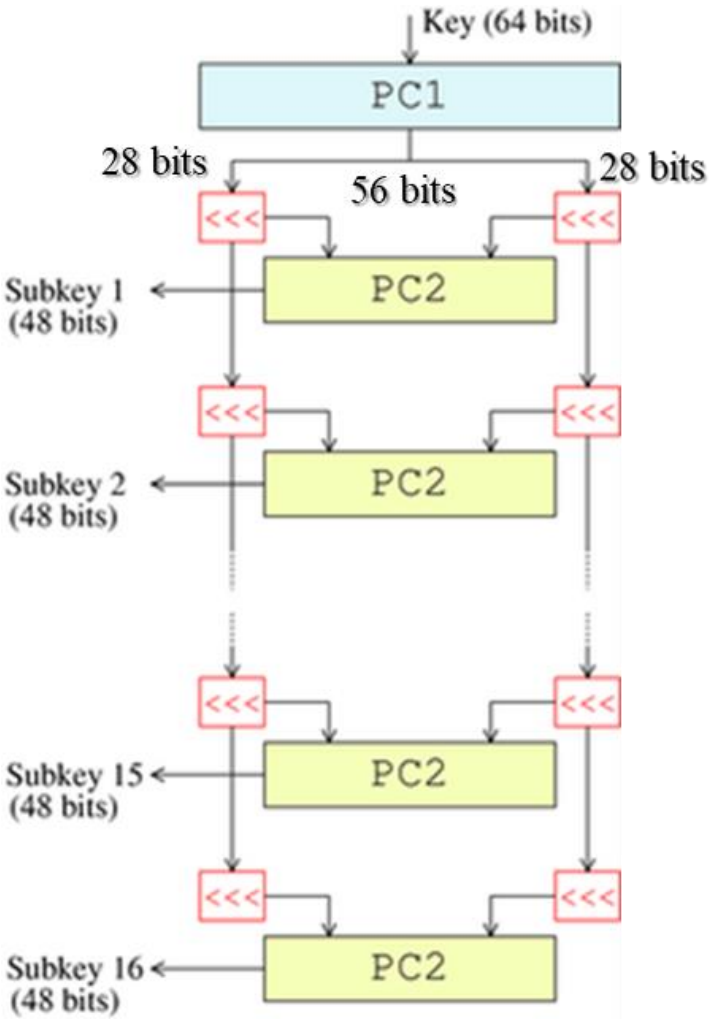
57	49	41	33	25	17	9
1	58	50	42	34	26	18
10	2	59	51	43	35	27
19	11	3	60	52	44	36
63	55	47	39	31	23	15
7	62	54	46	38	30	22
14	6	61	53	45	37	29
21	13	5	28	20	12	4

(c) Permuted Choice Two (PC-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

(d) Schedule of Left Shifts

Round Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Bits Rotated	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1



DES Decryption

The decryption algorithm uses the same steps exactly as in the encryption algorithm except that the application of the subkeys is reversed (i.e., in round1 use K_{16}, round2 use K_{15} and so on).

DES Example

https://page.math.tu-berlin.de/~kant/teaching/hess/krypto-ws2006/des.htm#:~:text=DES%20works%20by%20encrypting%20groups,key%20size%20is%2056%20bits.

https://herovired.com/learning-hub/blogs/des-algorithm-in-cryptography/

- Also read chapter 4 of Text Book

Summary

- DES