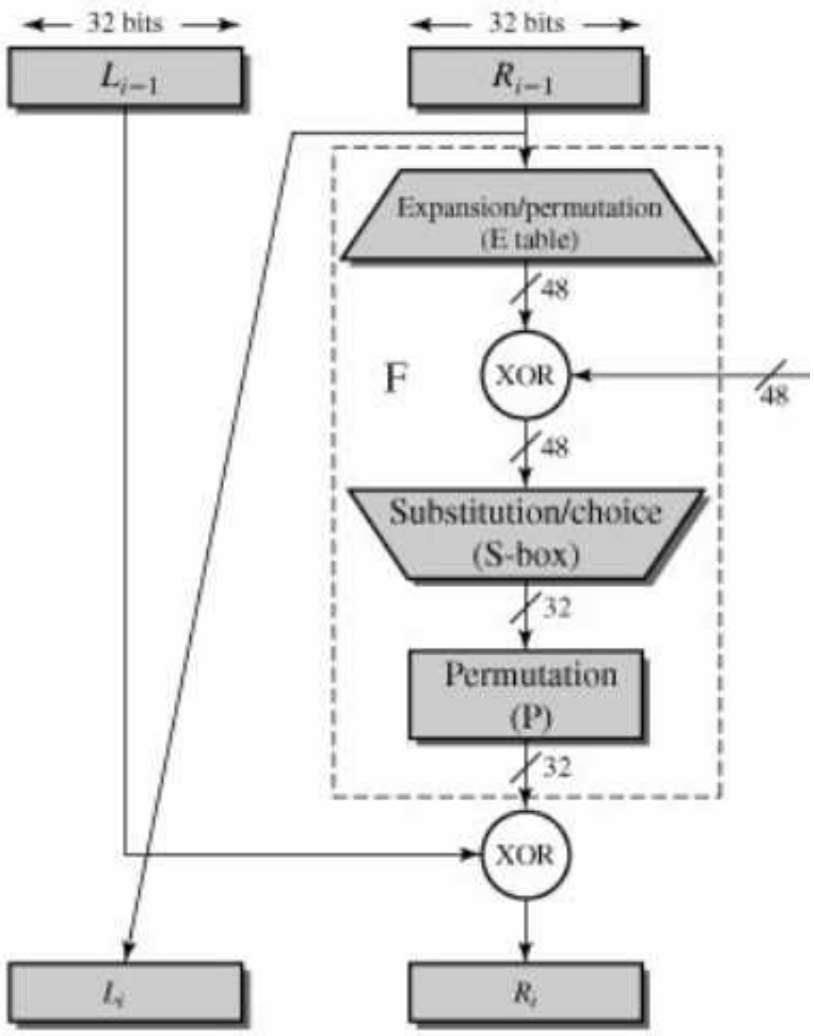
Details of Single Round

Figure below shows the internal structure of a single round. The left and right halves of each 64-bit intermediate value are treated as separate 32-bit quantities, labeled L (left) and R (right). As in any classic Feistel cipher, the overall processing at each round can be summarized in the following formulas:

$$L_i = R_{i-1}$$

$$R_i = L_{i-1} \times F(R_{i-1}, K_i)$$



1- The round key *K* is 48 bits. The *R* input is 32 bits. This *R* input is first expanded to 48 bits by using a table that defines a permutation plus an expansion that involves duplication of 16 of the *R* bits (table 1a (**E-table**))

(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	

2- The resulting 48 bits are XORed with *K*.

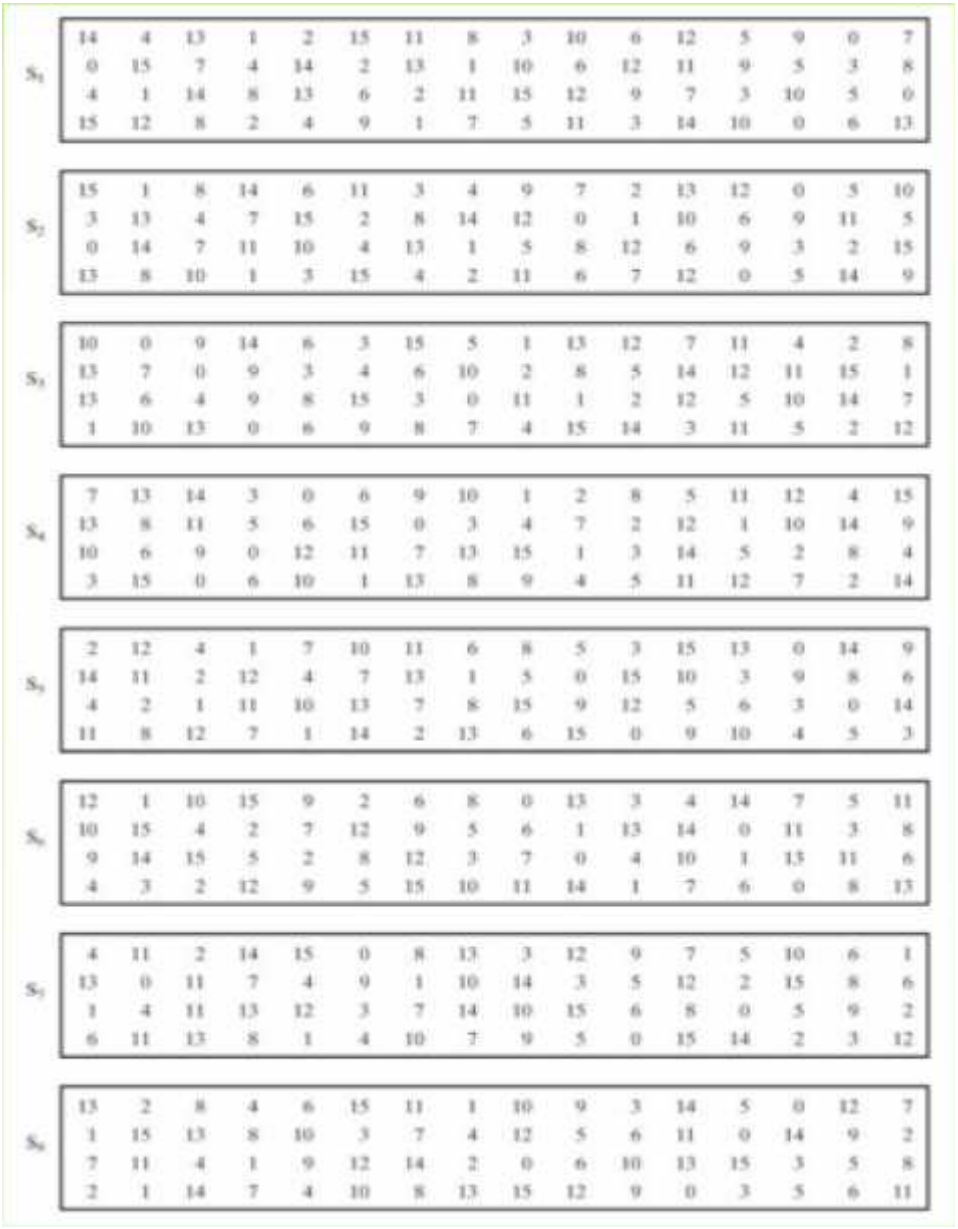
3- This 48-bit result passes through a substitution function that produces a 32-bit output, which is permuted as defined by (**S-boxes**).

The role of the S-boxes in the function F is illustrated in Figure 6a The substitution consists of a set of eight S-boxes, each of which accepts 6- bits as input and produces 4- bits as output. These transformations are as follows:

A- The first and last bits of the input to box S form a 2-bit binary number to select one of four substitutions defined by the four rows in the table for S*i*.

B- The middle four bits select one of the sixteen columns.

C- The decimal value in the cell selected by the row and column is then converted to its 4-bit representation(from S-boxes table) to produce the output.



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

Figure (6a) the s-box

Example:

In S*1* for input = 011001,

the row is 01 (row 1) and the column is 1100 (column 12).

The value in row 1, column 12 is 9, so the output is 1001.

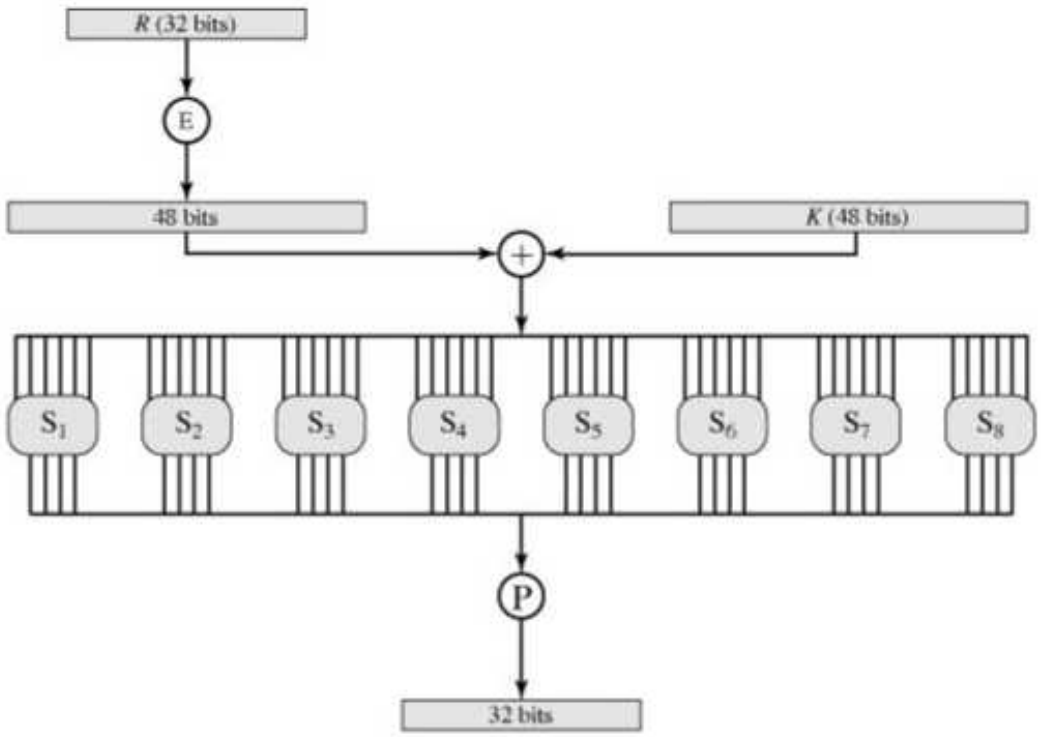
S (1100) = S (12) = 9 (base 2) = 1001 (base hex)

4- The 32-bit output from the eight S-boxes is then permuted.

The permutation function (p) is :

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

Calculation of F(R, K)



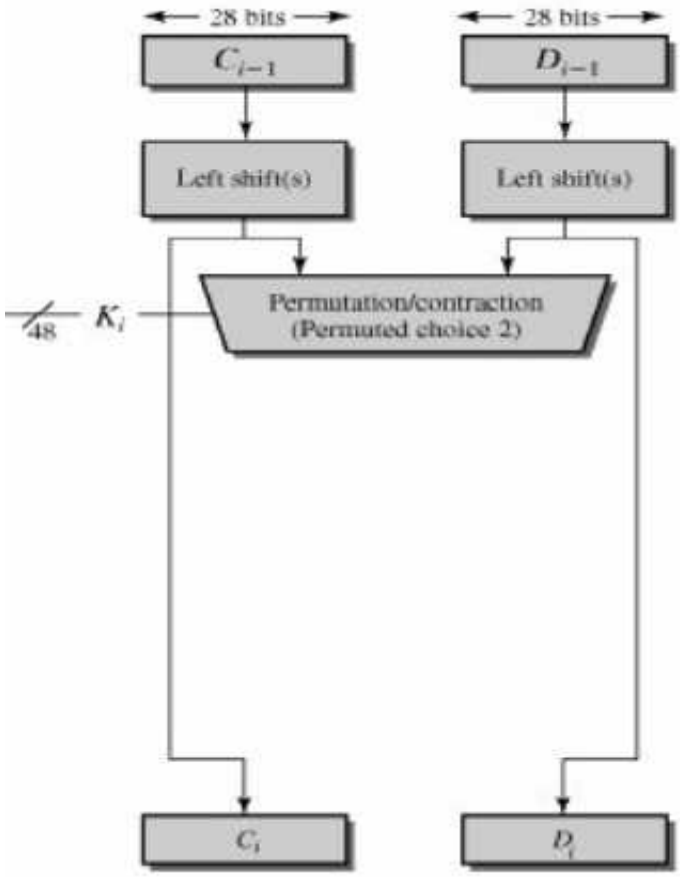
Example:

For single round the input = AAAAAAAAAAAAAAAA (in hex)

K= FF00FF0FF00FF0

Find the output of the round?

Key Generation:



1- 64-bit key is used as input to the algorithm. The bits of the key are numbered from 1 through 64, arranged (8x8) array .

(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	

2- The key is first sub Every eighth bit is ignored, as indicated by the lack of shading in, so only 56-bit is remain.

3- The key is subjected to a permutation governed by a table(b) labeled Permuted Choice One (PC-1).

(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		

4- The resulting 56-bit key is then treated as two 28-bit quantities, labeled C_o and D_o.

5- At each round, *Ci*-1 and *Di-1* are separately subjected to a circular left shift, or rotation, of 1 or 2 bits, as governed by Table d.

(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

6- These shifted values serve as input to the next round. They also serve as input *i*-1 to Permuted Choice Two (c), which produces a 48-bit output that serves as input to the function F(*Ri*-1, *Ki*-1).

(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				