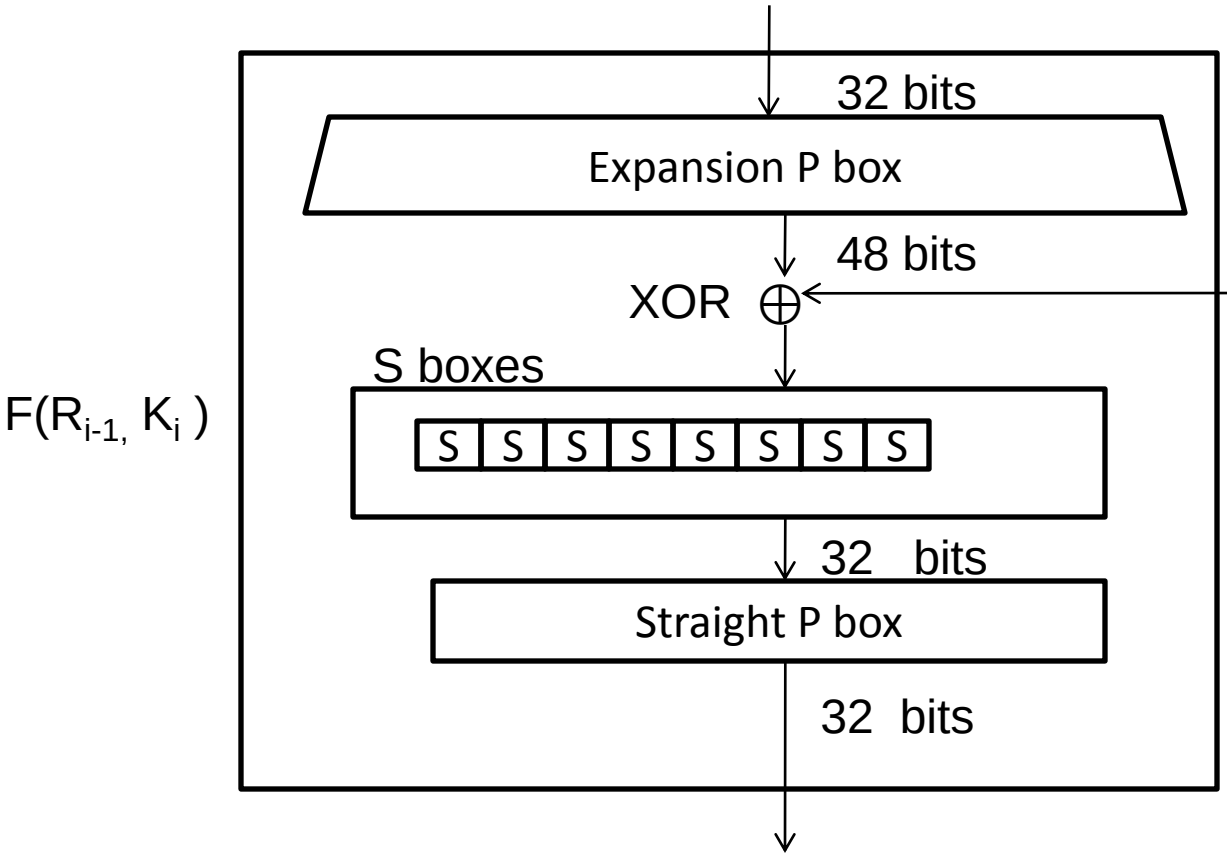
Network and Information Security
Lecture 14

B.Tech. Computer Engineering
Sem. VI.

Prof. Mrudang T. Mehta
Associate Professor
Computer Engineering Department
Faculty of Technology,
Dharmsinh Desai University, Nadiad

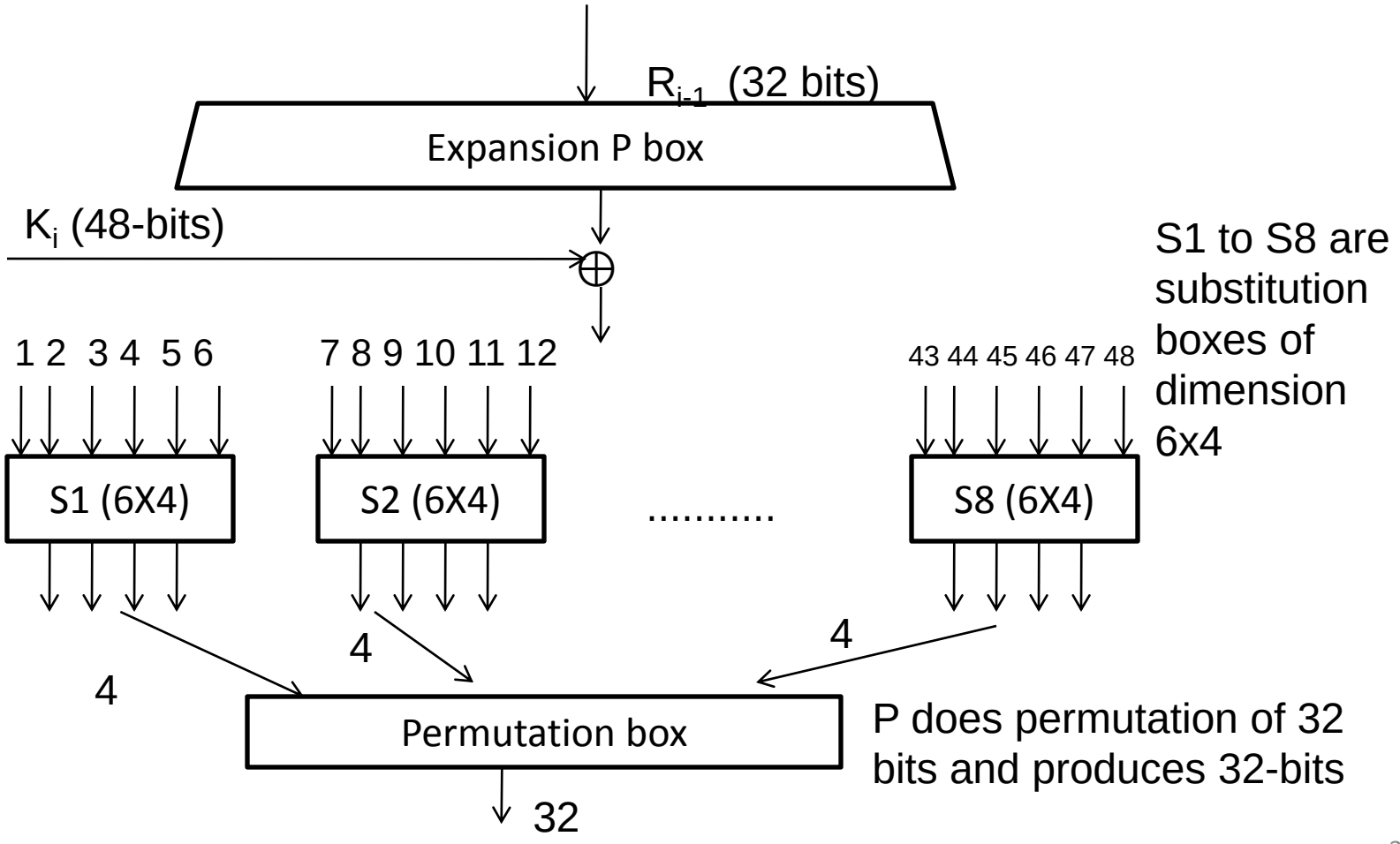
DES Function f



An expansion P box
A whitener (that adds key)
a group of S-boxes
A straight P-box

S-boxes

Round function, Input: R_{i-1} (32 bits), K_i (48 bits), Output: 32 bits



Expansion P-box table

Step 1

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

- First entry (1st row , 1st column) is 32 which indicates the index of bit from where we need to copy i.e. 1st bit of output is 32nd bit of input
- Thus by copying bits, we are able to generate 48 bits
- Basically we are adding redundancy i.e. We are creating 16 more bits by copying bits from certain position

- Step 2 (48 bits output of step 1) is ex-ored with 48-bits of round key K_i

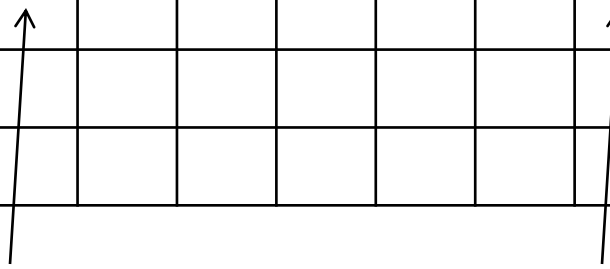
a	b	a $\oplus$ b
0	0	0
0	1	1
1	0	1
1	1	0

S-box

Step 3 S-box is of dimension of 6x4 which means it maps or substitutes 4 bits for 6 bits of input

6-bits => 2^6 =64 values are arranged in the following manner (4(rows) *16(columns)=64 , 4 bits entries)

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



Value from 0 to 15

Each output entry is of 4-bits (0 to 15)

- DES uses 8 S-boxes , each with a 6-bit input and a 4-bit output
- The 48-bit data from the second operation is divided into eight 6-bit chunks, and each chunk is fed into a box.
- The result of each box is a 4-bit chunk; when these are combined the result is a 32-bit text.
- The substitution in each box follows a pre-determined rule based on a 4-row by 16-column table.

- The combination of bits 1 and 6 of the input defines one of four rows;
- the combination of bits 2 through 5 defines one of the sixteen columns
- Each S-box has its own table (8-tables)

	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-box1

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

S-box2

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

S-box3

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

S-box4

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

S-box5

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

S-box6

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

S-box7

	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-box8

Example 1

The input to S-box 1 is 1<u>0001</u>1. What is the output?

If we write first and sixth bit together we get $(11)_2$ in binary.

11 in binary is 3 in decimal.

The remaining bits are $(0001)_2$ which is 1 in decimal.

Hence, we look in Row 3, and Column 1 in S-box1 table.

(3,1) = 12 = $(1100)_2$ => Output is 1100.

Example 2

The input to S-box 8 is 0<u>00000</u>0. What is the output?

If we write first and sixth bit together, we get 00 in binary, which is 0 in decimal.

The remaining bits are $(0000)_{2,}$ which is 0 in decimal.

Hence, we look in Row 0, and Column 0 in S-box8 table.

(0,0) = 13 = $(1101)_2$ => Output is 1101

Step 4: 32 bit output from step 3 is permuted to create 32 bits.

- P-box or permutation box is also given which simply does permutation of input bits.
- Straight permutation table

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

- Thus, F (function) creates 32 bits.
- The output 32 bits from F are ex-ored with L to create the right half of output.
  - $L_i = R_{i-1,}$ Left output of Round i is same as right input to round i-1.
  - $R_i = L_{i-1} \oplus F(R_{i-1,} k_i)$

- This process is repeated 16 times so that resultant cipher text can’t be crypt-analyzed easily.
- In round structure IP: Initial permutation, FP: Final permutation are just 64 bit permutations of the input 64 bits.
- FP and IP are inverses of each other. ($FP = IP^{-1}$)
- FP and IP have no cryptography significance in DES.

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

Example 1

Find the output of the final permutation box when the input is given in hexadecimal as: 0x0000008000000002

Represent hex in binary and find 1s

0000 0000 0000 0000 0000 0000 1000 0000000000000000 0000 0000 0000 0010

Only bit 25 and bit 63 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

Example 2

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

0x0002 0000 0000 0001.

0000 0000 0000 0010 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0001

- The input has only two 1s; the output must also have only two 1s.
- Using table, 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 0x0000008000000002