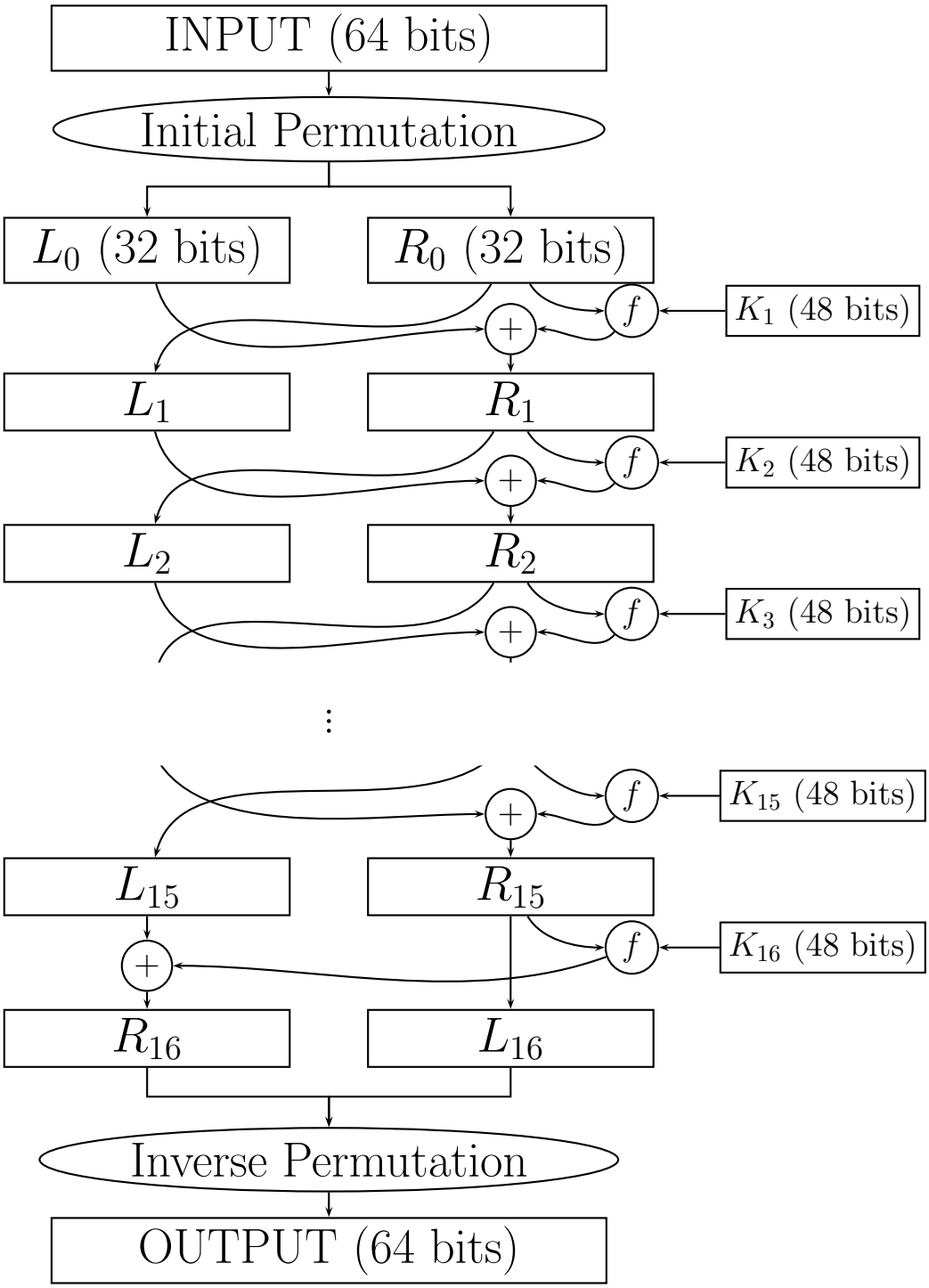
Data Encryption Standard (DES)

- Originally developed under the name Lucifer by IBM in the 60's and 70's
- Adopted by Federal government in July, 1977
- Reaffirmed in 1983, 1988, 1993, and 1999
- Most widely used cryptosystem in the world from 1977 to present
- Used by the U.S. Government to "protect the confidentiality of sensitive (unclassified) electronic information".
- Replaced by AES (Rijndael) in Summer of 2001

DES


INPUT (64 bits)
Initial Permutation
L_0 (32 bits)
R_0 (32 bits)
f
K_1 (48 bits)
L_1
R_1
K_2 (48 bits)
L_2
R_2
K_3 (48 bits)
K_{15} (48 bits)
L_{15}
R_{15}
K_{16} (48 bits)
R_{16}
L_{16}
Inverse Permutation
OUTPUT (64 bits)

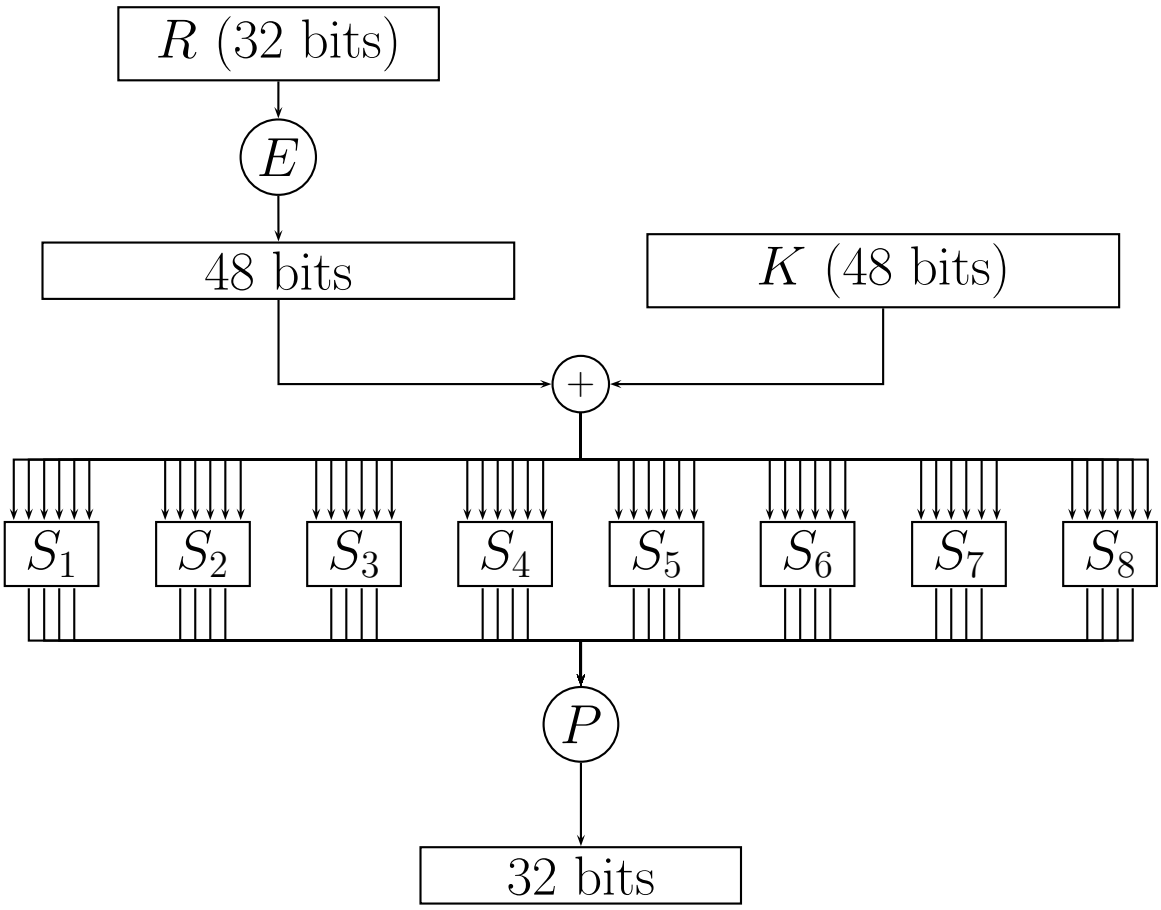
Initial Permutation

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

Inverse Permutation

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

The Cipher Function f



E Bit-Selection Table

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

Permutation 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

Selection Function S_1

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

Example: 110100

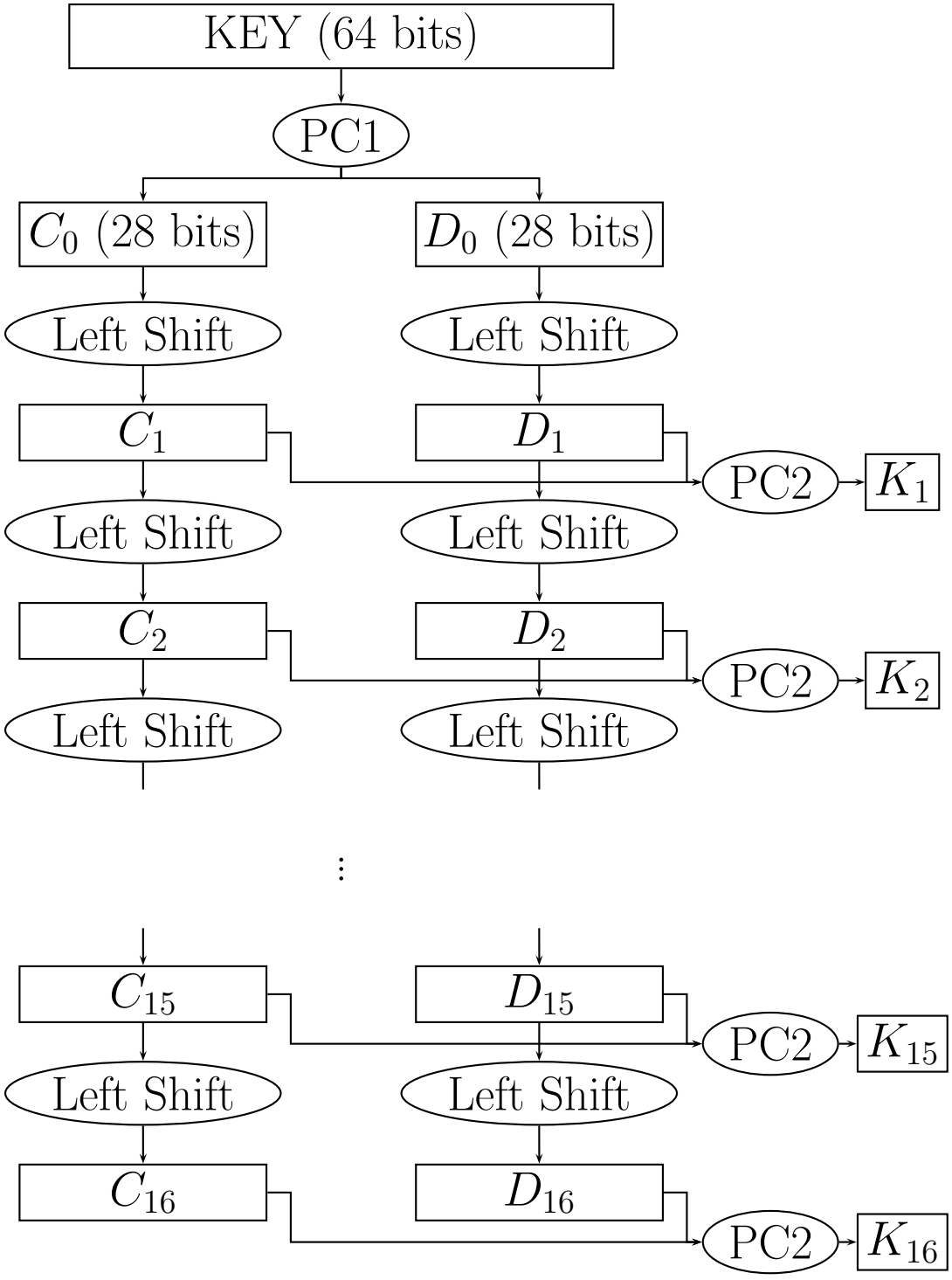
- Use first and last digit as row index: 10 (base 2) = 2
- Use middle four digits as column index: 1010 (base 2) = 10
- The number 9 appears in row 2, column 10
- 9 = 1001 (base 2)

$$S_1(110100) = 1001$$

Selection Functions S_2 Through S_8

		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
S_2:	0	15	1	8	14	6	11	3	4	9	7	2	13	12	0	5	10
	1	3	13	4	7	15	2	8	14	12	0	1	10	6	9	11	5
	2	0	14	7	11	10	4	13	1	5	8	12	6	9	3	2	15
	3	13	8	10	1	3	15	4	2	11	6	7	12	0	5	14	9
S_3:	0	10	0	9	14	6	3	15	5	1	13	12	7	11	4	2	8
	1	13	7	0	9	3	4	6	10	2	8	5	14	12	11	15	1
	2	13	6	4	9	8	15	3	0	11	1	2	12	5	10	14	7
	3	1	10	13	0	6	9	8	7	4	15	14	3	11	5	2	12
S_4:	0	7	13	14	3	0	6	9	10	1	2	8	5	11	12	4	15
	1	13	8	11	5	6	15	0	3	4	7	2	12	1	10	14	9
	2	10	6	9	0	12	11	7	13	15	1	3	14	5	2	8	4
	3	3	15	0	6	10	1	13	8	9	4	5	11	12	7	2	14
S_5:	0	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
	1	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
	2	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
	3	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3
S_6:	0	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
	1	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
	2	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
	3	4	3	2	12	9	5	15	10	11	14	1	7	6	0	8	13
S_7:	0	4	11	2	14	15	0	8	13	3	12	9	7	5	10	6	1
	1	13	0	11	7	4	9	1	10	14	3	5	12	2	15	8	6
	2	1	4	11	13	12	3	7	14	10	15	6	8	0	5	9	2
	3	6	11	13	8	1	4	10	7	9	5	0	15	14	2	3	12
S_8:	0	13	2	8	4	6	15	11	1	10	9	3	14	5	0	12	7
	1	1	15	13	8	10	3	7	4	12	5	6	11	0	14	9	2
	2	7	11	4	1	9	12	14	2	0	6	10	13	15	3	5	8
	3	2	1	14	7	4	10	8	13	15	12	9	0	3	5	6	11

Key Schedule Calculation



Number of Left Shifts

iteration	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
shifts	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

Permuted Choice, PC1

C_0:	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
D_0:	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

Permuted Choice, PC2

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

The Decrypting Process

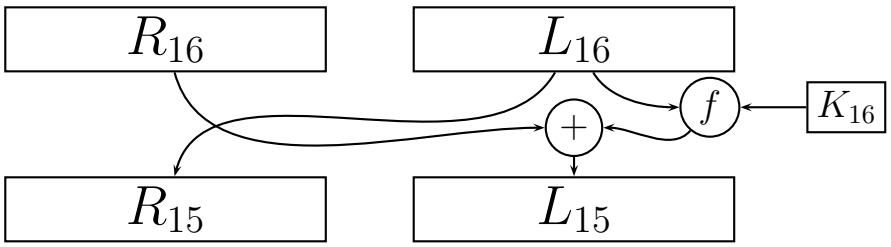
From the encrypting process we have that

$$L_{16} = R_{15} \quad \text{and} \quad R_{16} = L_{15} \oplus f(R_{15}, K_{16}).$$

Therefore

$$L_{15} = R_{16} \oplus f(L_{16}, K_{16}),$$

where $\oplus$ denotes bit-by-bit addition mod 2. Pictorially, we have



In other words, to decrypt a message, you may use the same mechanism that was used to encrypt, except apply the Key Schedule in reverse, meaning first use K_{16}, then K_{15}, and so on. This explains why L_{16} and R_{16} were reversed at the end of the encrypting process.

Variations of DES

- **triple-DES**: In effect, the input is encrypted three times. One such method uses the keys k_1, k_2, and k_3 to define

$$C = E_{k_3}(D_{k_2}(E_{k_1}(M)))$$

where E_k and D_k represent DES encryption and DES decryption, respectively.

- **DESX**: plaintext is bitwise XORed with 64 bits of additional key material before encryption with DES and the output is also bitwise XORed with another 64 bits of key material.