DES History

- In 1973, NBS (NIST) issues a public request for proposals for a national cipher standard, which must be
  - Secure
  - Public
  - Completely specified
  - Easy to understand
  - Available to all users
  - Economic and efficient in hardware
  - Able to be validated
  - Exportable
- IBM submitted LUCIFER (Feistel) (which was redesigned to become the DES)
- In 1977, adopted by NBS (NIST) as DES (Data Encryption Standard, Federal Information Processing Standard 46 (FIPS PUB 46))

DES History

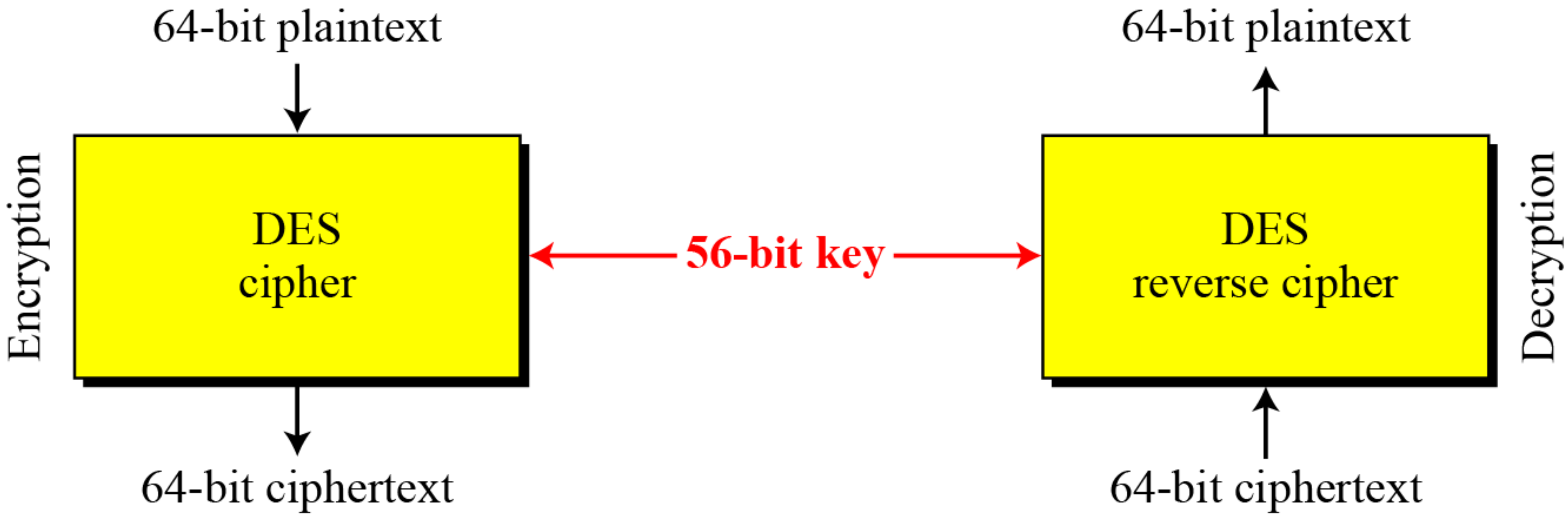
- Chronolgy
  - 1973: NBS publishes a first request for a standard encryption algorithm
  - 1974: NBS publishes a second request for encryption algorithms
  - 1975: DES is published in the Federal Register for comment
  - 1976: First and second workshop on DES
  - 1976: DES is approved as a standard
  - 1977: DES is published as a FIPS standard FIPS PUB 46
  - 1983: DES reaffirmed for the first time
  - 1986: Videocipher II, a TV satellite scrambling system based upon DES begins use by HBO
  - 1988: DES is reaffirmed for the second time as FIPS 46-1, superseding FIPS PUB 46
  - 1992: Biham and Shamir publish the first theoretical attack with less complexity than brute force: differential cryptanalysis. However, it requires an unrealistic 2^{47} chosen plaintexts
  - 1993: DES is reaffirmed for the third time as FIPS 46-2

DES History

- 1994: The first experimental cryptanalysis of DES is performed using linear cryptanalysis (Matsui, 1994)
- 1997: The DESCHALL Project breaks a message encrypted with DES for the first time in public
- 1998: The Electronic Frontier Foundation (EFF)'s DES cracker (Deep Crack) breaks a DES key in 56 hours
- 1999: Together, Deep Crack and distributed.net break a DES key in 22 hours and 15 minutes
- 1999: DES is reaffirmed for for the fourth time as FIPS 46-3, which specifies the preferred use of Triple DES, with single DES permitted only in legacy systems
- 2001: The Advanced Encryption Standard is published in FIPS 197
- 2002: The AES standard becomes effective
- 2004: The withdrawal of FIPS 46-3 (and a couple of related standards) is proposed in the Federal Register
- 2005: NIST withdraws FIPS 46-3

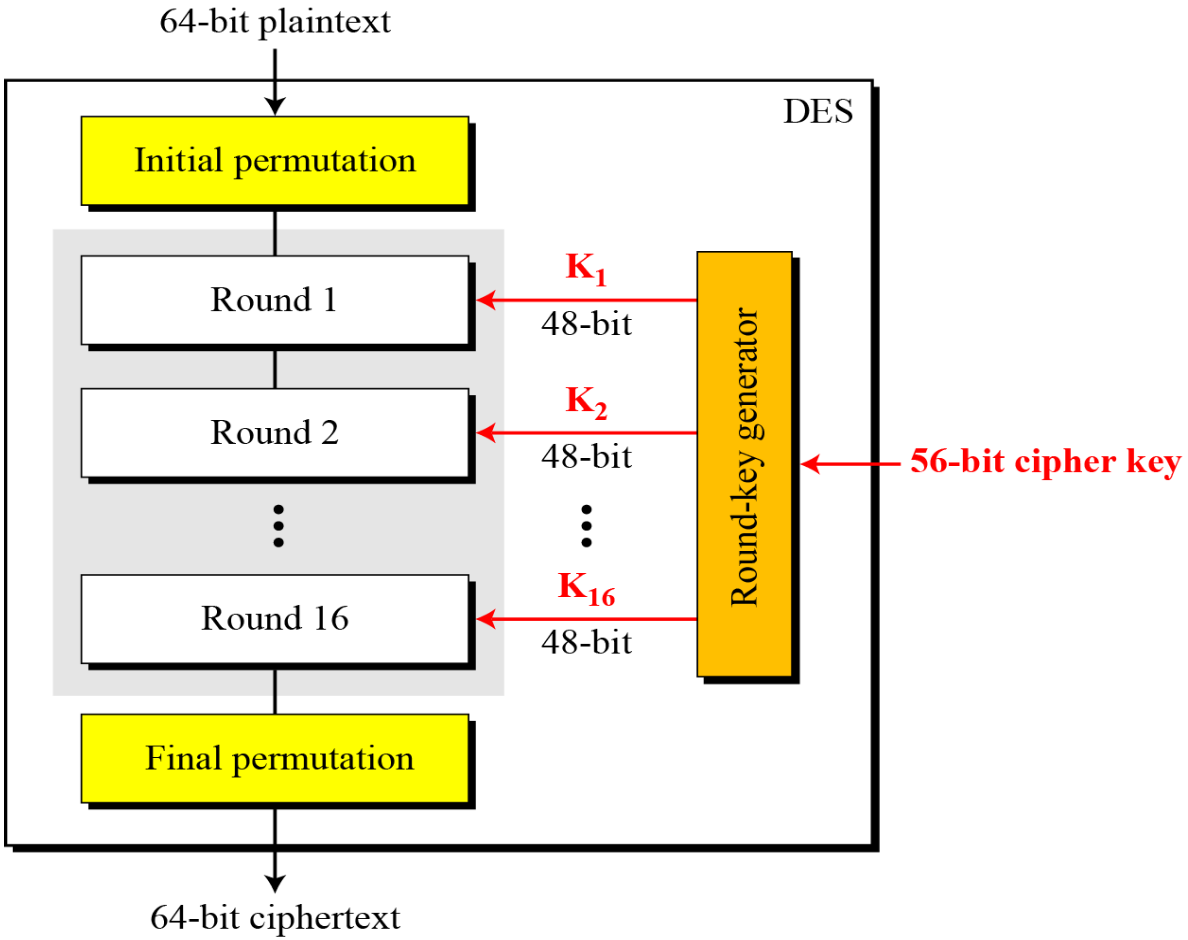
Encryption and Decryption with DES

DES is a block cipher, as shown in Figure

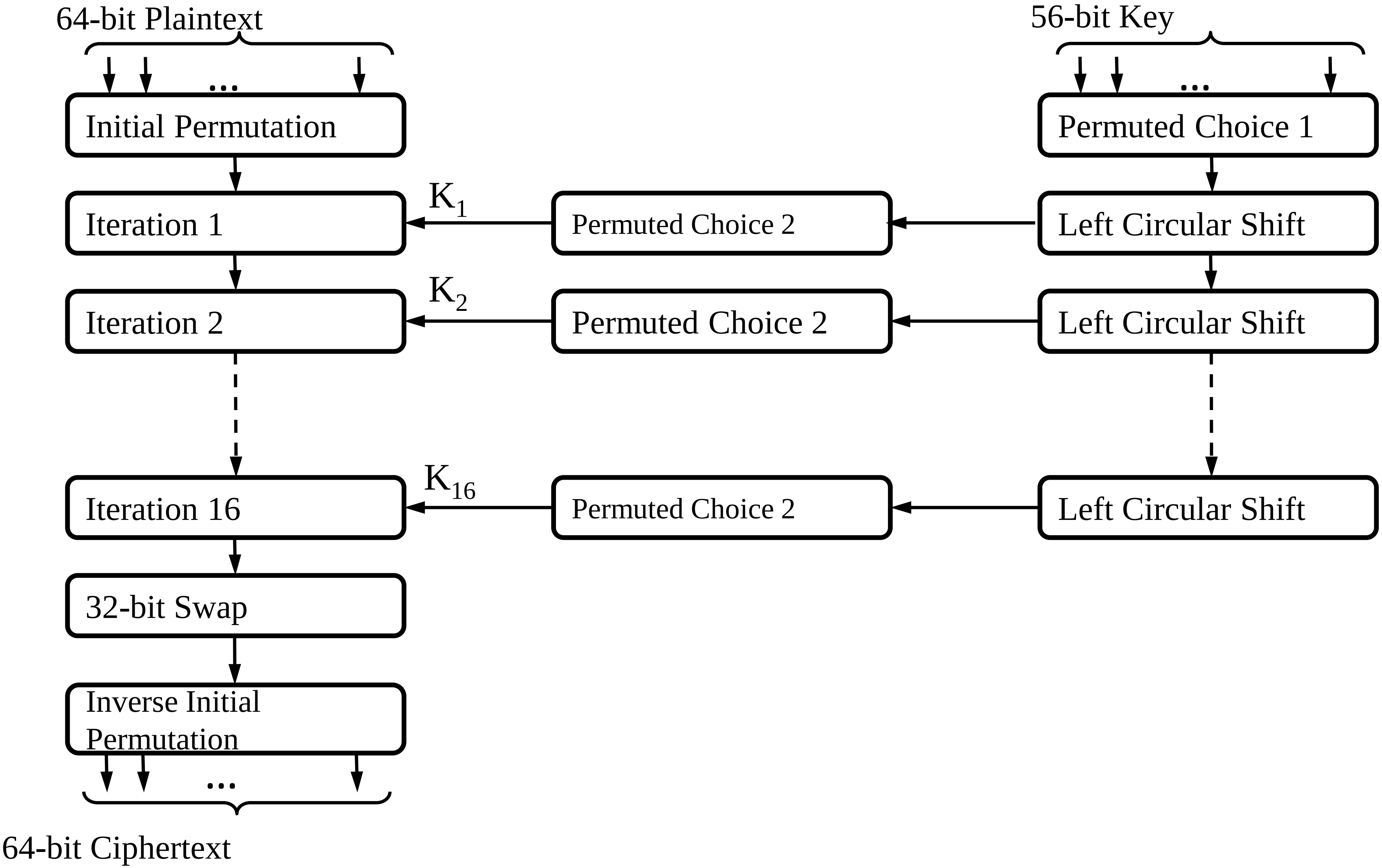


General Structure of DES

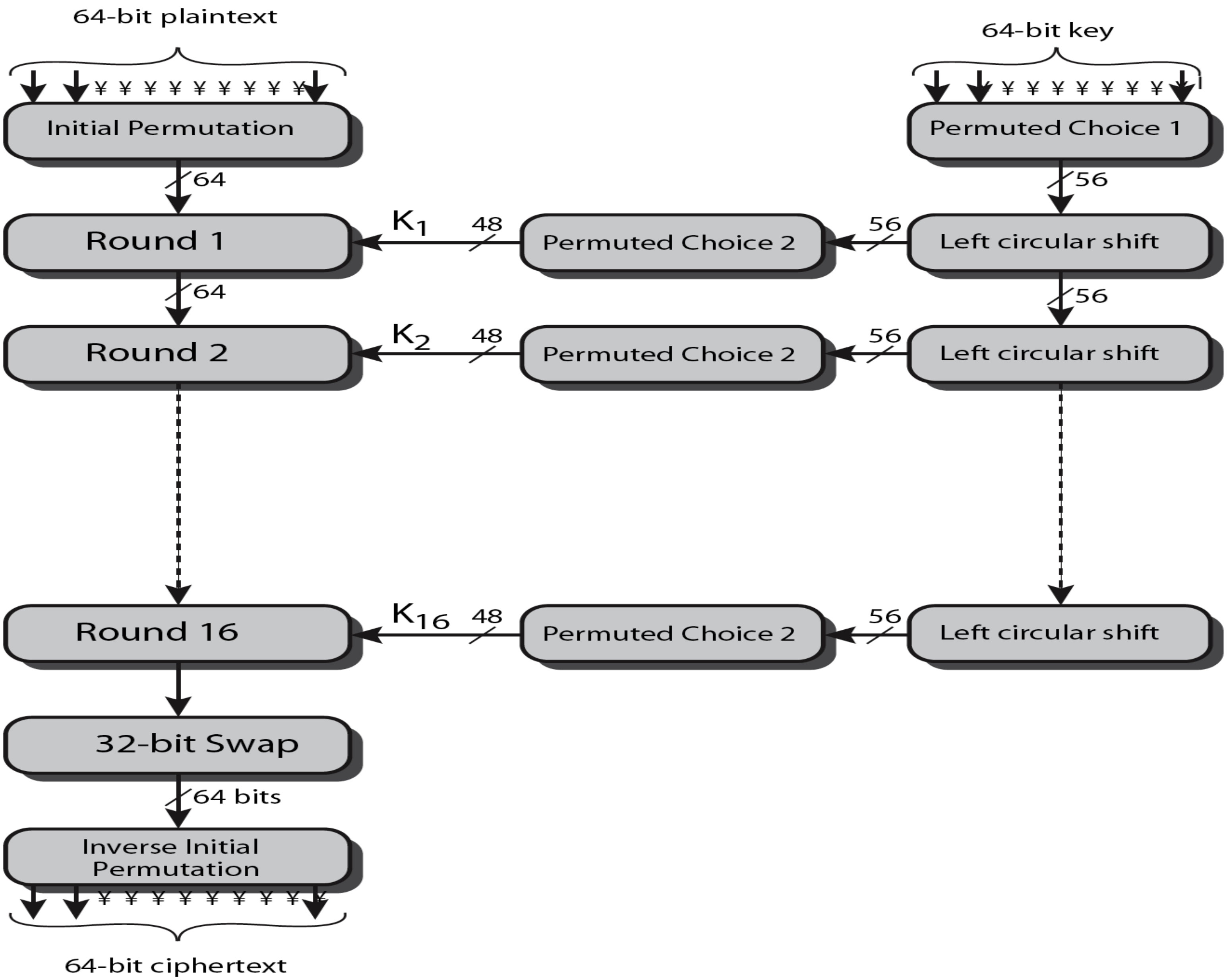
The encryption process is made of two permutations (P-boxes), which we call initial and final permutations, and sixteen Feistel rounds.



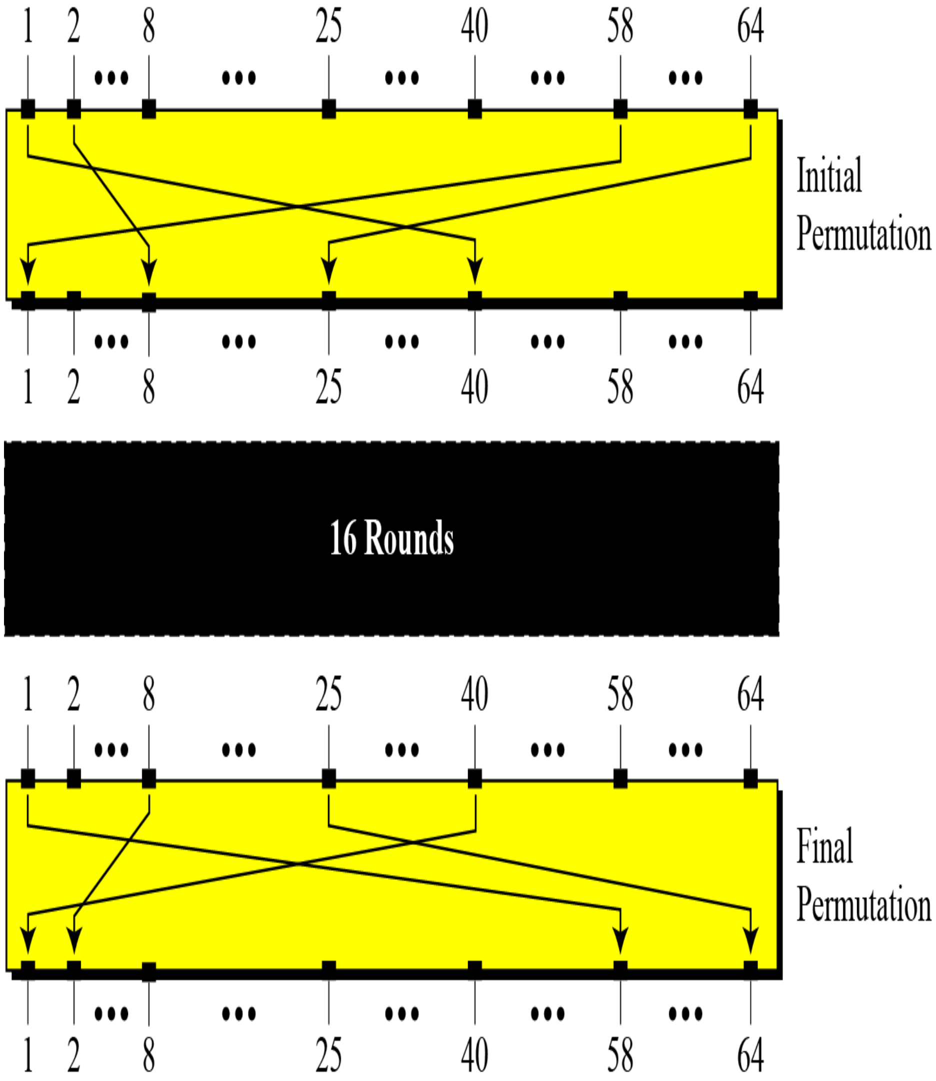
DES Encryption Overview

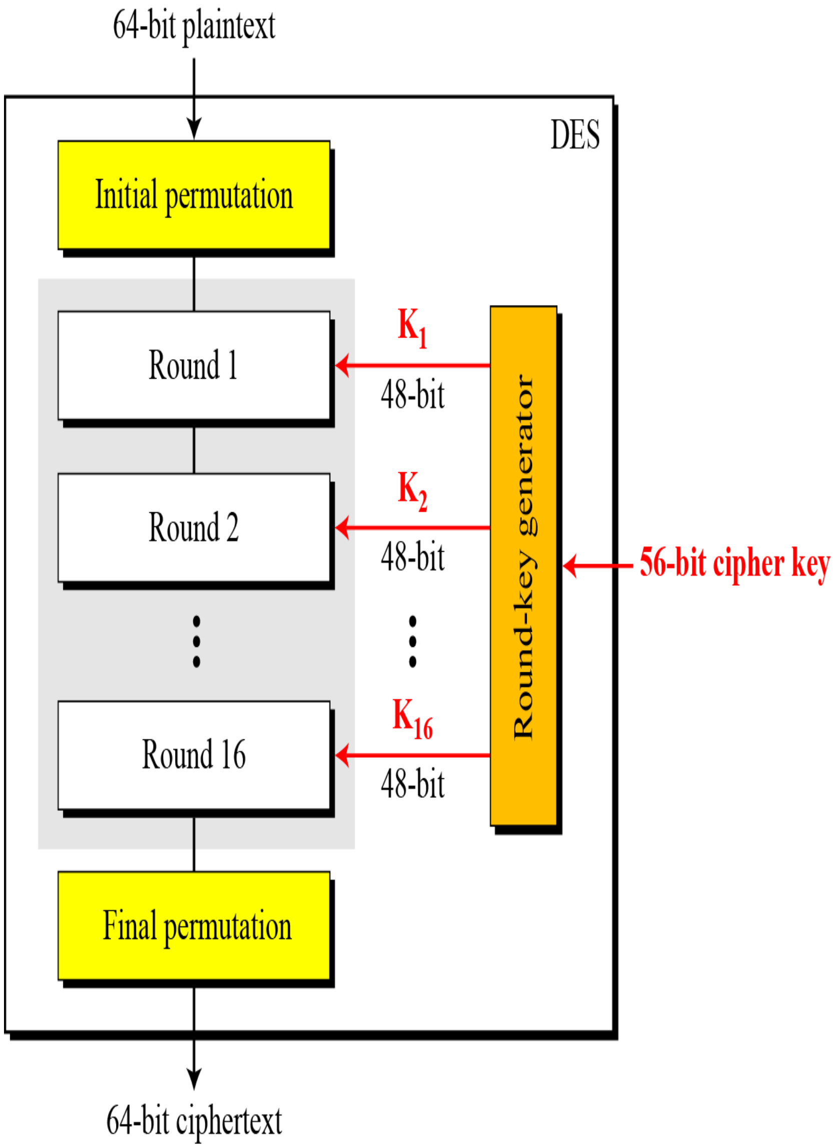


DES Encryption Overview

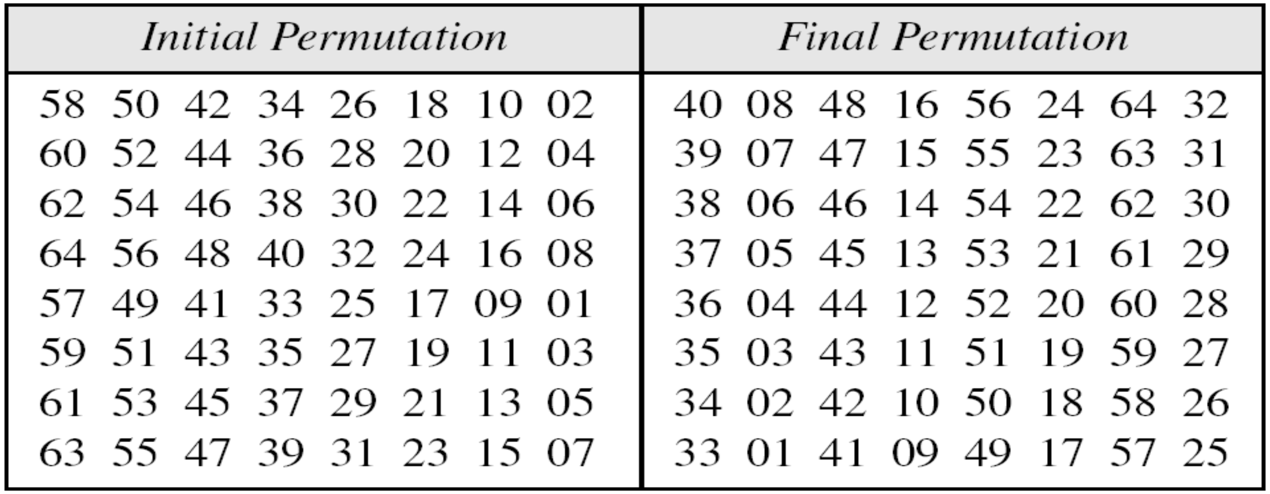


Initial and Final Permutation Steps in DES





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

Problem No. 1

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

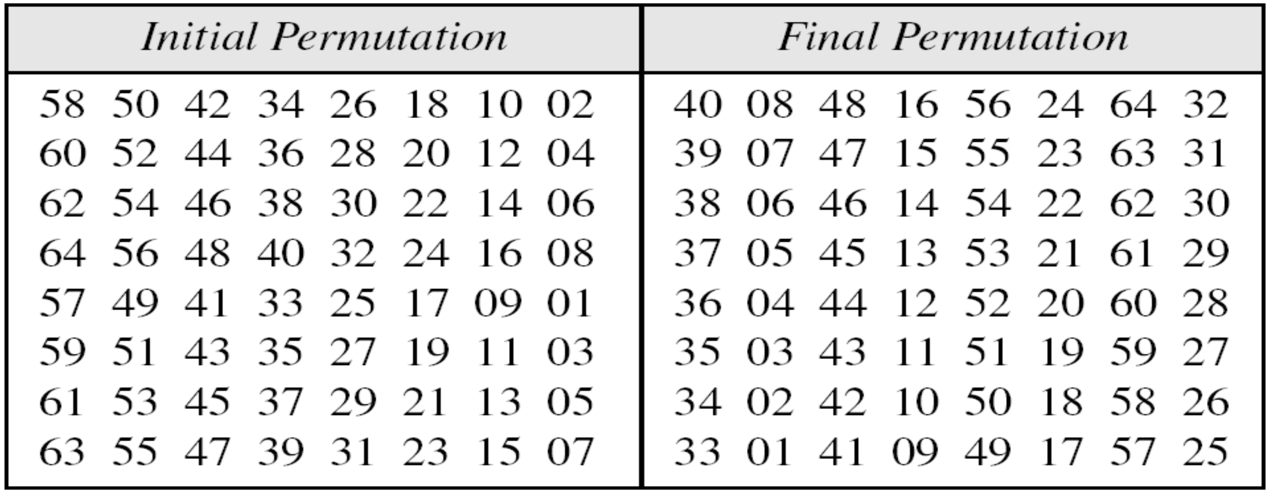
0x0002 0000 0000 0001

Input has only two 1s (Bit 15 and bit 64): the output must also have only two 1s(the nature straight permutation).

The 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.

0x0000 0080 0000 0002

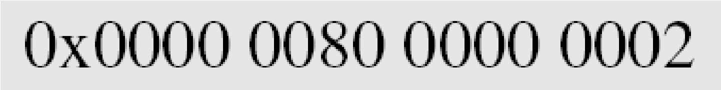
Problem No. 2



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

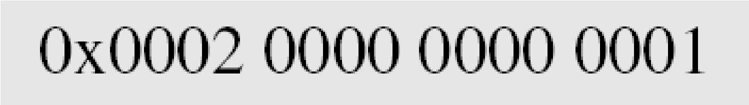
Problem No. 2

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



0x0000 0080 0000 0002

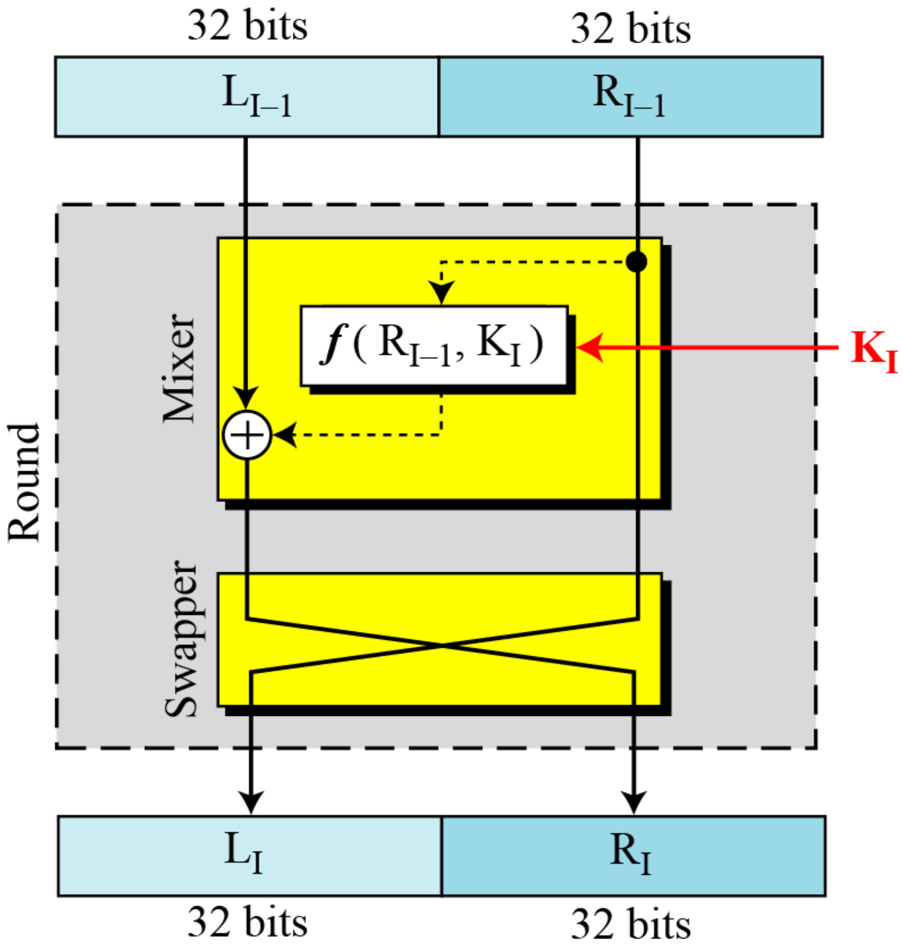
Solution: The bit 25 in the input becomes bit 64 in the output. Bit 63 in the input becomes bit 15 in the output. So the output has only two 1s, bit 15 and bit 64.



0x0002 0000 0000 0001

A round in DES (encryption site)

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



DES Contd.

- The computation consists of 16 iterations of a calculation
- The cipher function ***f*** operates on two blocks, one of 32 bits and one of 48 bits, and produces a block of 32 bits.
- The input block is then **LR**, 32 bit block **L** followed by a 32 bit block **R**.
- Let **K** be a block of 48 bits chosen from the 64-bit key. Then the output **L'R'** of an iteration with input **LR** is defined by:

  L' = R

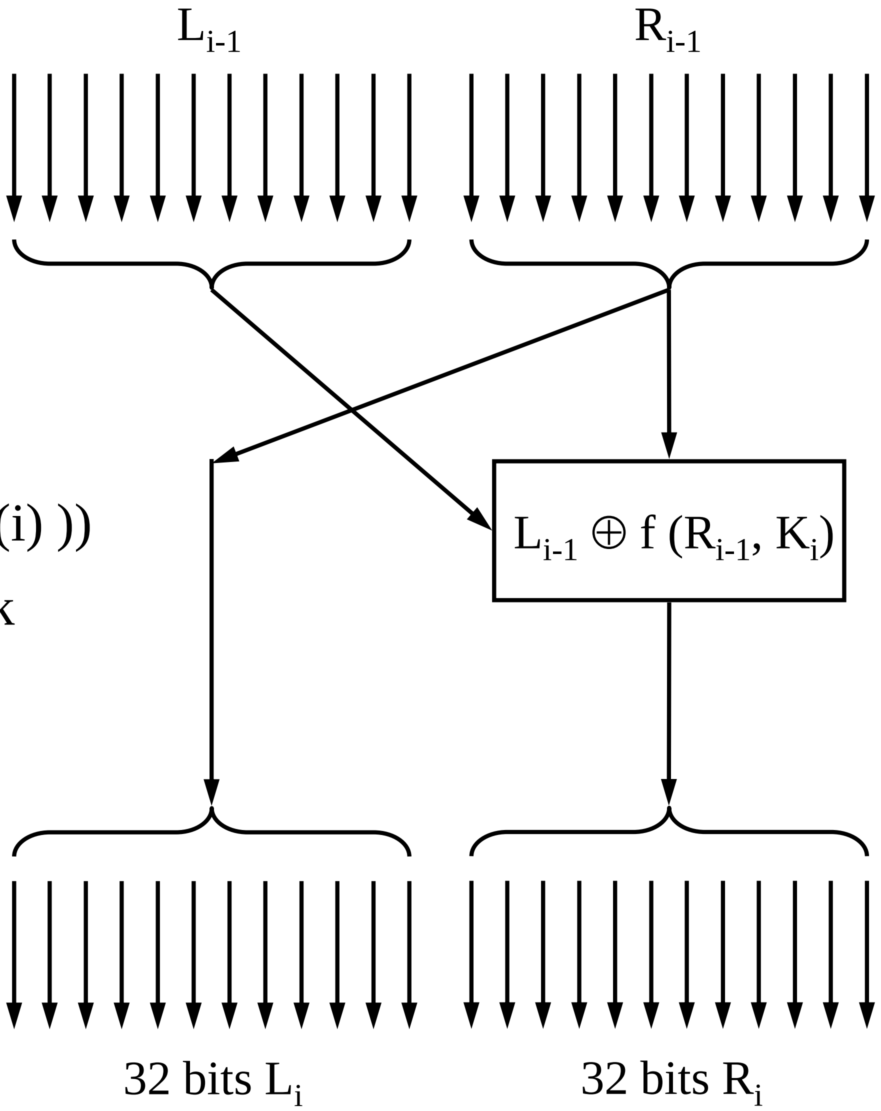
  R' = L (+) *f* (R,K)

- L'R' is the output of the 16th iteration then R'L' is the preoutput block.
- At each iteration a different block K of key bits is chosen from the 64-bit key designated by **KEY**.
- Let KS be a function which takes an integer *n* in the range from 1 to 16 and a 64-bit block KEY as input and yields as output a 48-bit block $\mathbf{K}_{n}$ which is a permuted selection of bits from KEY. That is

K_n = KS (*n*, KEY)

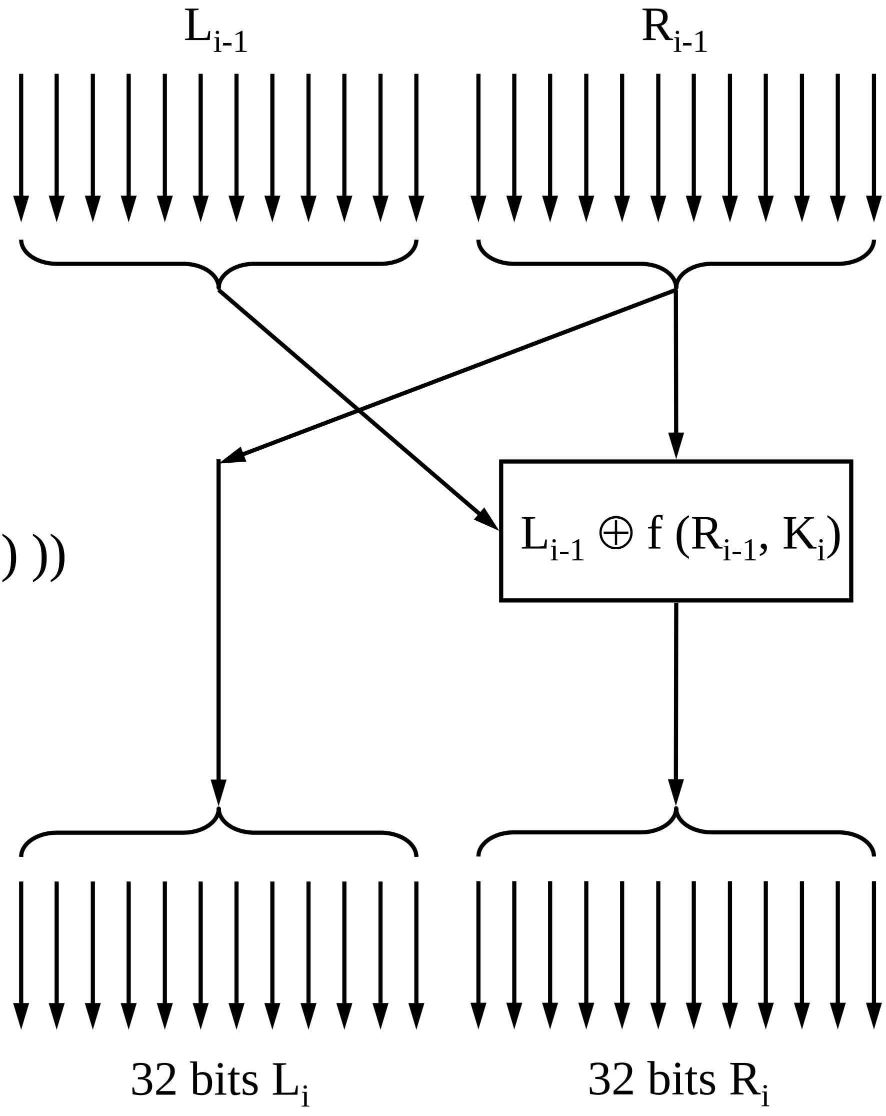
DES - Swapping of Left and Right Halves

- This can be described functionally as:
  - L(i) = R(i-1)
  - R(i) = L(i-1) ⊕ P(S(E(R(i-1)) ⊕ K(i)))
- This forms one round in an S-P network

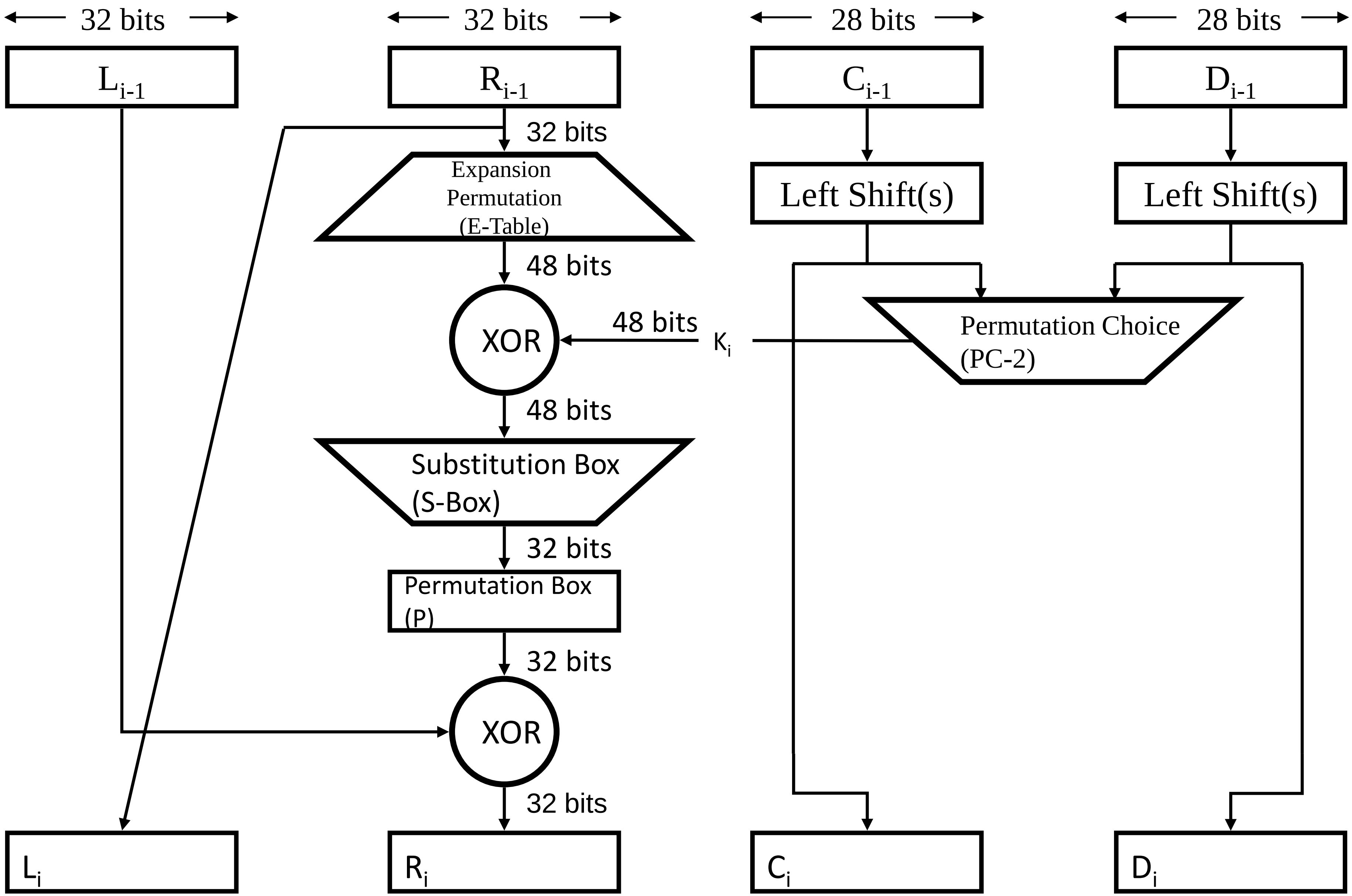


DES - Swapping of Left and Right Halves

- This can be described functionally as:
  - $L(i) = R(i-1)$
  - $R(i) = L(i-1) \oplus P(S(E(R(i-1)) \oplus K(i)))$
- This forms one round in an S-P network

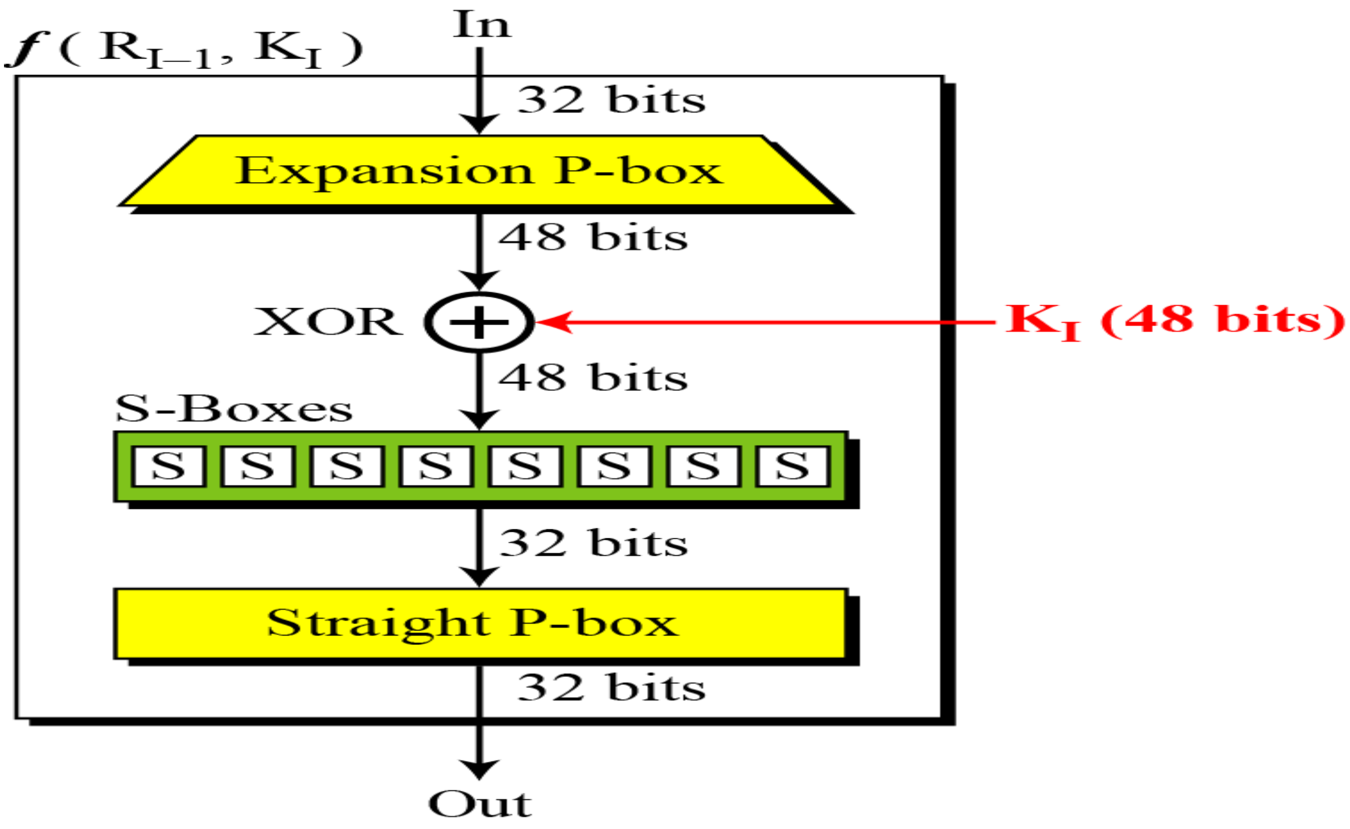


Details of Each Iteration



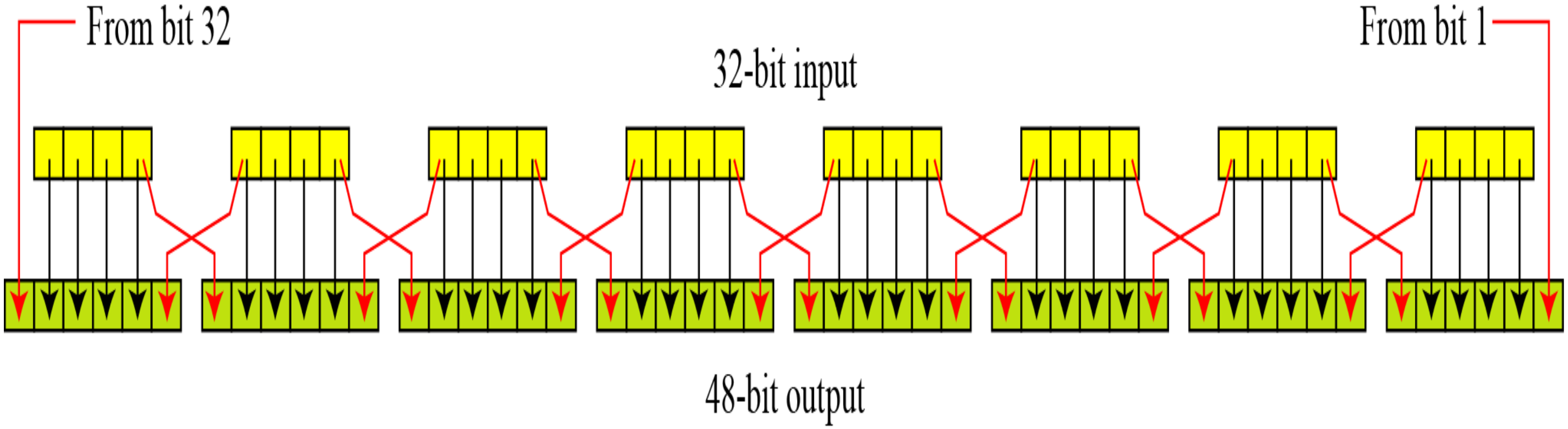
DES Function

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.



Expansion P-box

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.



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

Expansion Permutation Table

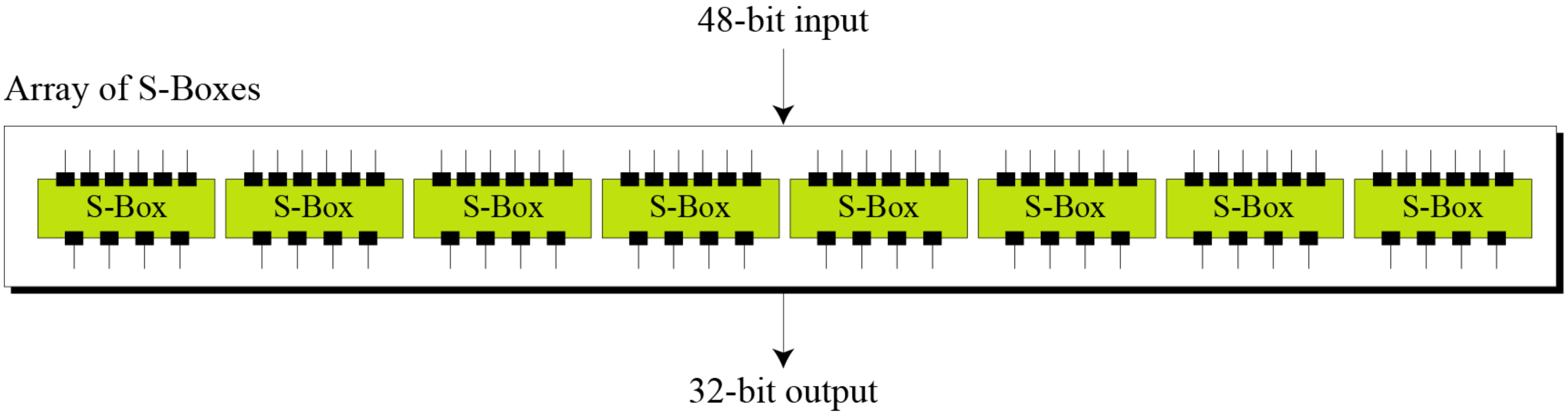
32	1	2	3	4	5
4	5	6	7	8	9
8	9	10	11	12	13
12	13	14	15	16	17
16	17	18	19	20	21
20	21	22	23	24	25
24	25	26	27	28	29
28	29	30	31	32	1

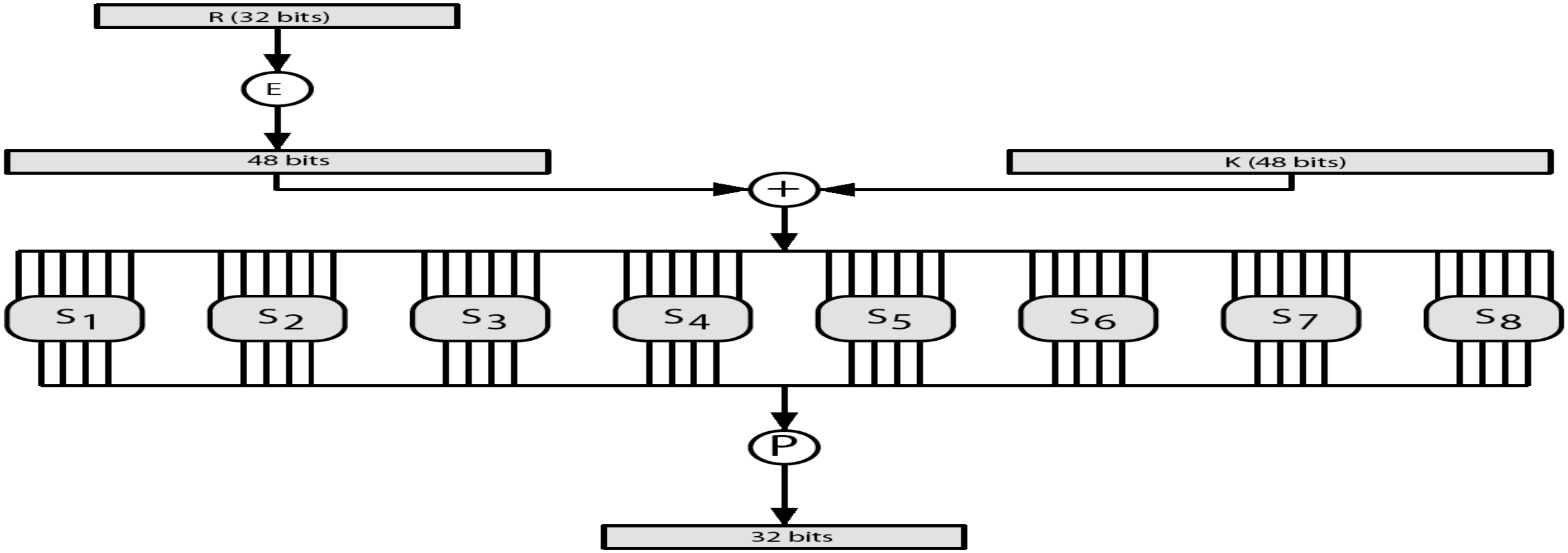
Whitener (XOR)

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.

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.



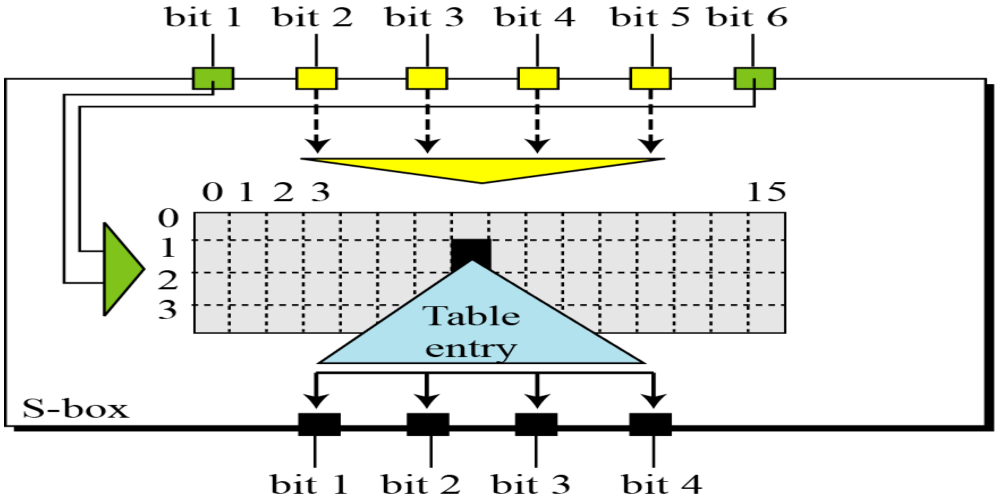


S-Box 1

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

	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

Problem No. 3
The input to S-box 1 is 100011.
What is the output?



S-Box Structure

S1

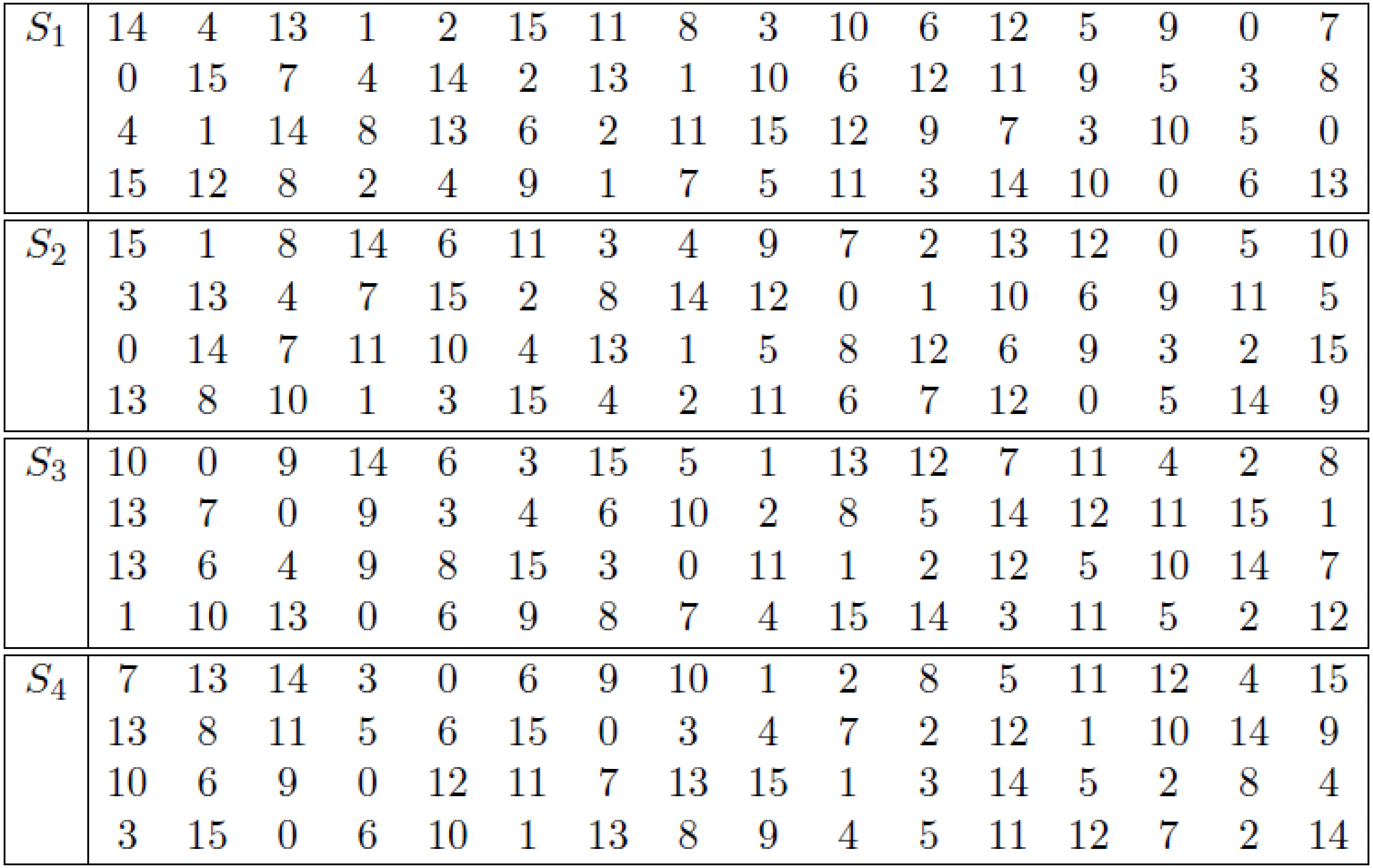
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
00(0)	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
01(2)	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
10(3)	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
11(4)	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

For example, for input 011011 the row is 01, that is row 1, and the column is determined by 1101, that is column 13. In row 1 column 13 appears 5 so that the output is 0101.

Solution to Problem No. 2

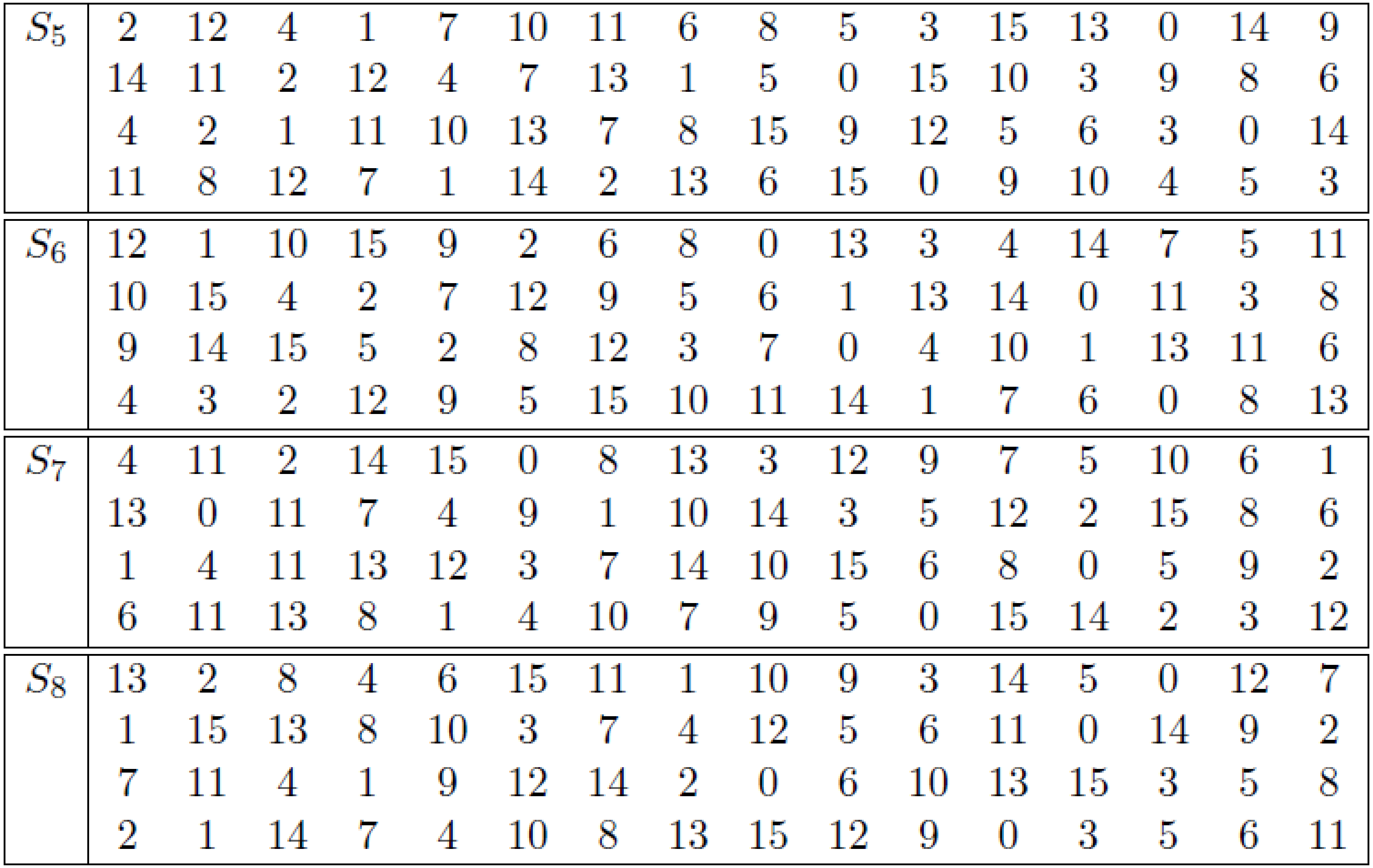
If we write the first and the sixth bits together, we get 11 in binary, which is 3 in decimal. The remaining bits are 0001 in binary, which is 1 in decimal. We look for the value in row 3, column 1, (S-box 1). The result is 12 in decimal, which in binary is 1100. So the input 100011 yields the output 1100.

S-Box Structure



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-Box Structure



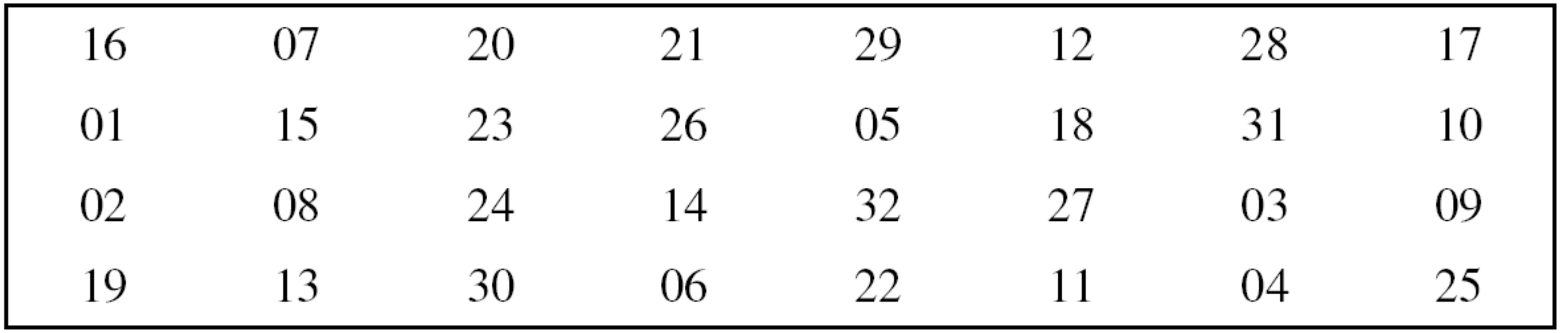
S_5	2	12	4	1	7	10	11	6	8	5	3	15	13	0	14	9
	14	11	2	12	4	7	13	1	5	0	15	10	3	9	8	6
	4	2	1	11	10	13	7	8	15	9	12	5	6	3	0	14
	11	8	12	7	1	14	2	13	6	15	0	9	10	4	5	3
S_6	12	1	10	15	9	2	6	8	0	13	3	4	14	7	5	11
	10	15	4	2	7	12	9	5	6	1	13	14	0	11	3	8
	9	14	15	5	2	8	12	3	7	0	4	10	1	13	11	6
	4	3	2	12	9	5	15	10	11	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

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

Solution

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.

Straight Permutation



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

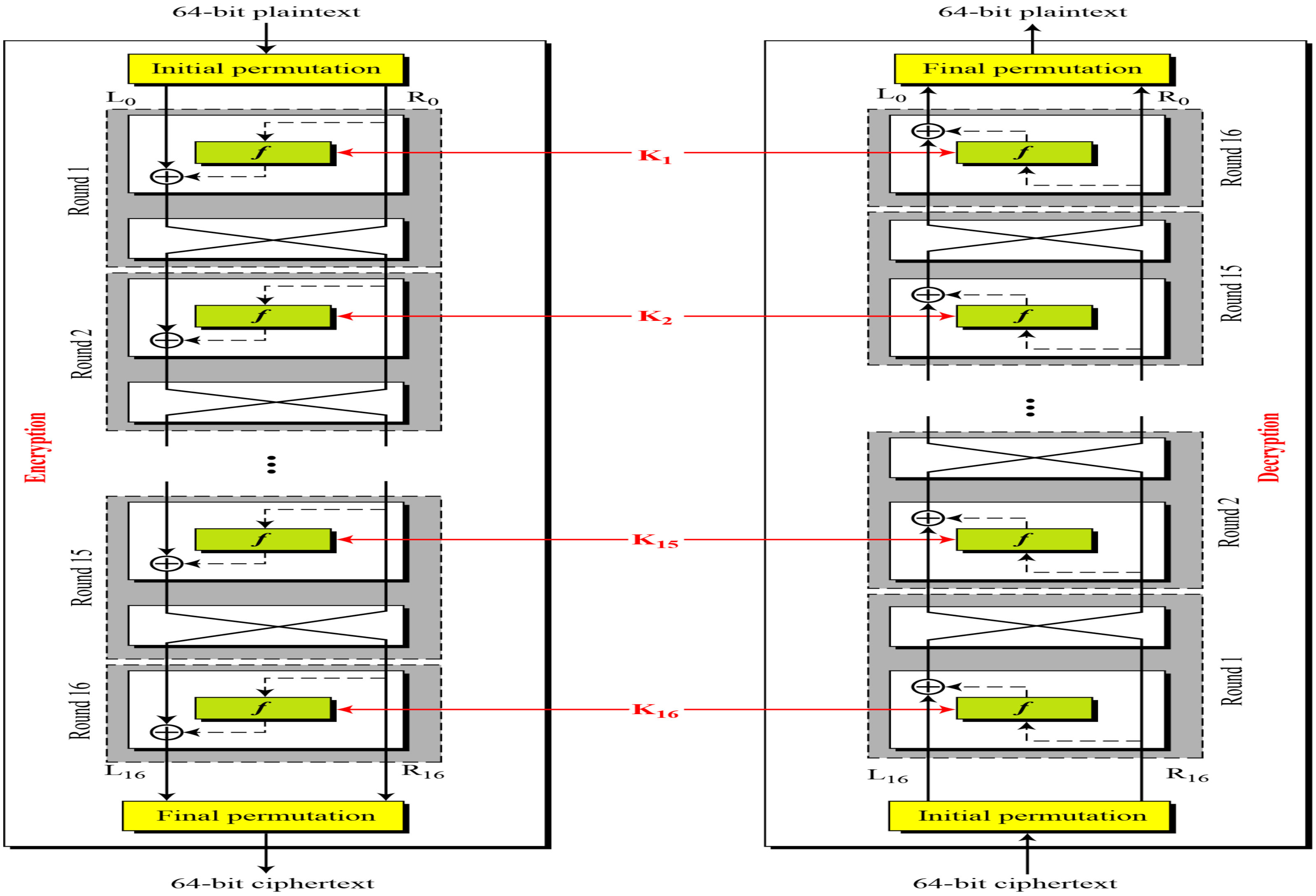
Cipher and Reverse Cipher

Using mixers and swappers, we can create the cipher and reverse cipher, each having 16 rounds.

First Approach

To achieve this goal, one approach is to make the last round (round 16) different from the others; it has only a mixer and no swapper.

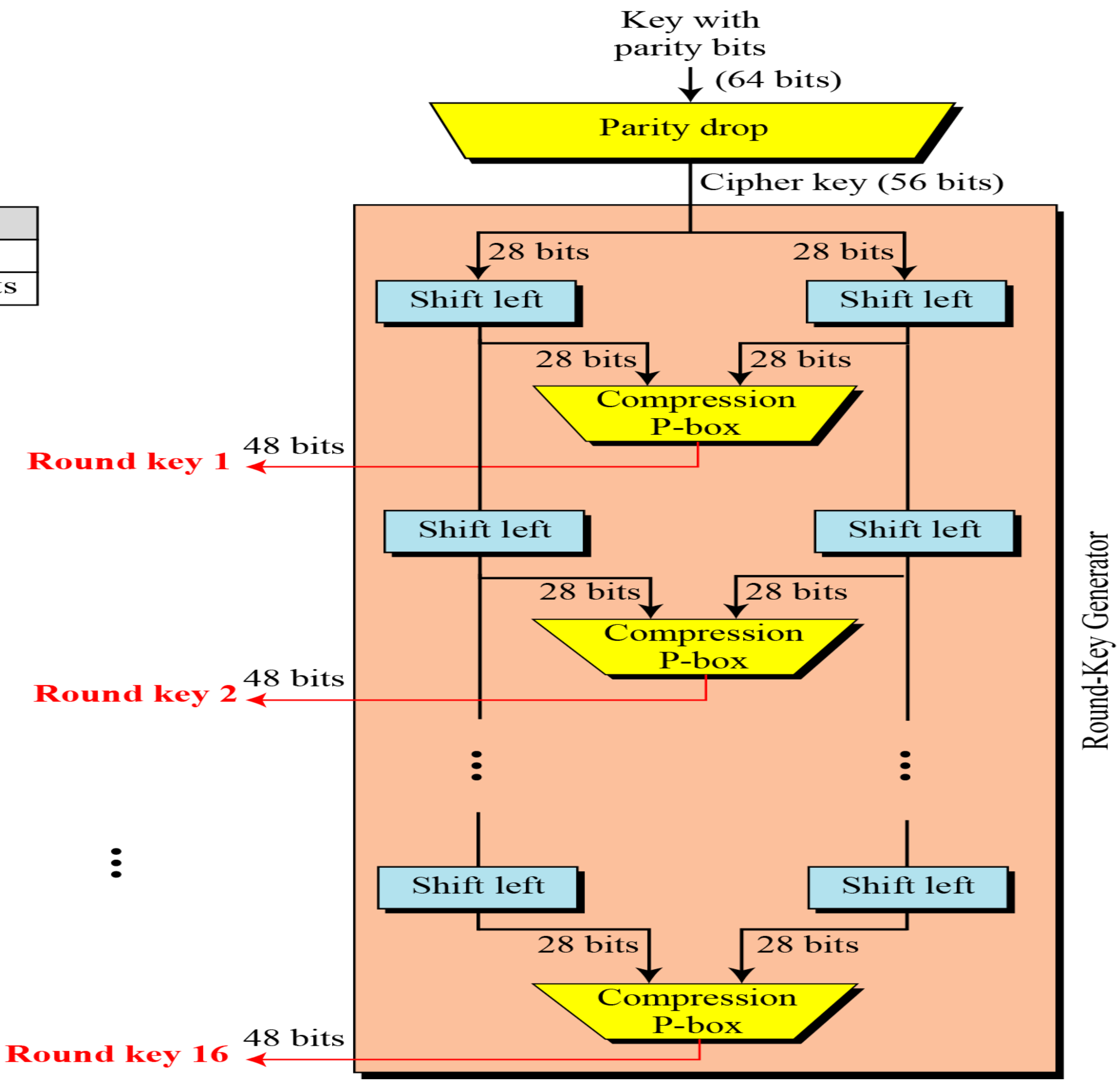
DES Cipher and Reverse Cipher for the First Approach



Key Generation

Shifting

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



Permutation 1

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

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

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

Permutation Choice 2

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

Key Rotation Schedule

Round Number	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Number of Left Shifts	1	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1
Total Number of Shifts	1	2	4	6	8	10	12	14	15	17	19	21	23	25	27	28

Iteration Corresponds to Left Shifts

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

Avalanche Effect

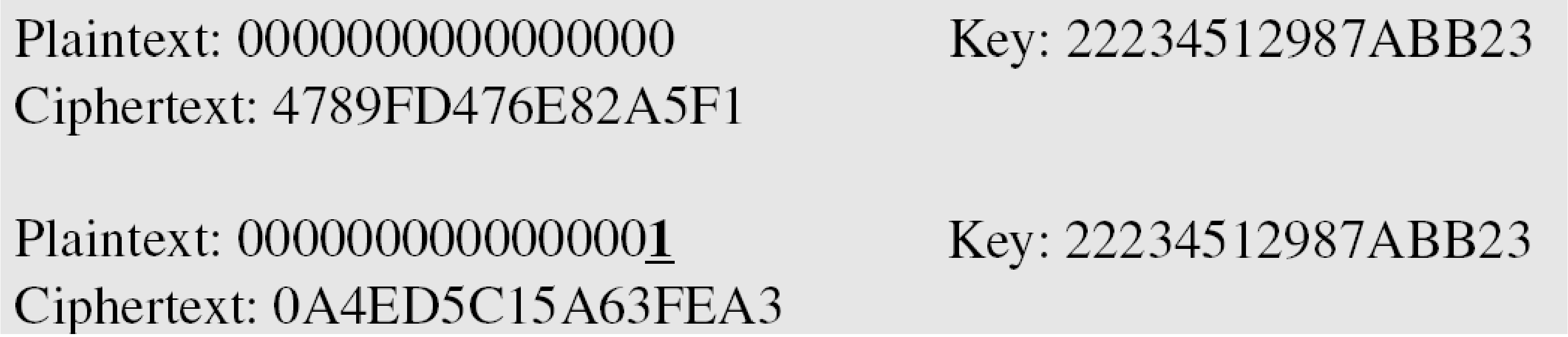
DES exhibits strong avalanche, where a change of **one** input or key bit results in changing approx **half** output bits

Avalanche Effect

DES exhibits strong avalanche, where a change of **one** input or key bit results in changing approx **half** output bits.

Two desired properties of a block cipher are the avalanche effect and the completeness.

To check the avalanche effect in DES, let us encrypt two plaintext blocks (with the same key) that differ only in one bit and observe the differences in the number of bits in each round.

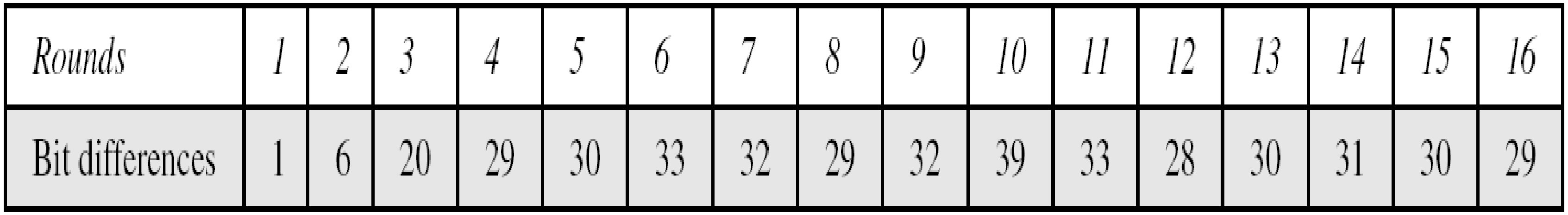


Plaintext: 0000000000000000 Key: 22234512987ABB23
Ciphertext: 4789FD476E82A5F1

Plaintext: 0000000000000001 Key: 22234512987ABB23
Ciphertext: 0A4ED5C15A63FEA3

Avalanche Effect Contd.

Although the two plaintext blocks differ only in the rightmost bit, the ciphertext blocks differ in 29 bits. This means that changing approximately 1.5 percent of the plaintext creates a change of approximately 45 percent in the ciphertext.



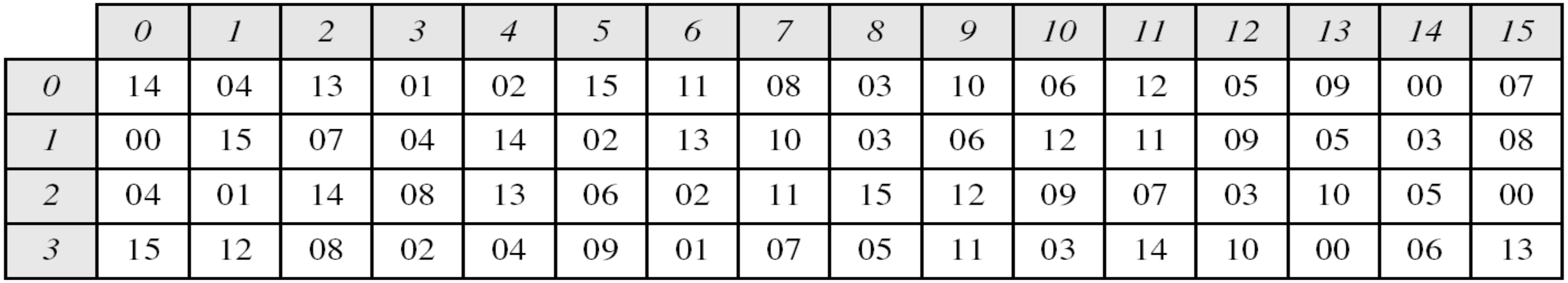
Rounds	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*
Bit differences	1	6	20	29	30	33	32	29	32	39	33	28	30	31	30	29

Completeness Effect

Completeness effect means that each bit of the ciphertext needs to depend on many bits on the plaintext. The diffusion and Confusion produced by P-boxes and S-boxes in DES shows a very strong completeness effect.

Design Criteria of S-boxes

The design provides confusion and diffusion of bits from each round to the next.



	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

- The entries of each row are permutation of values are between 0 and 15.

- If there is a single bit change in the input, two or more bits will be changed in the output.

- If two inputs to an S-box differ only in the two middle bits (bit 3 and bit 4), the output must differ in at least two bits.

Design Criteria of S-boxes Contd.

If two inputs to an S-box differ in the first two bits (bit 1 and bit 2) and are the same in the last two bits (5 and 6), the two outputs must be different.

There are only 32 6-bit input word pairs(X_i and X_j) in which $X_i \Theta X_j \neq (000000)2$. These 32 input pairs create 32 4-bit output word pair. If we create the difference between the 32 output pair $d = Y_i \Theta Y_j$, no more than 8 of these d should be the same.

Design Criteria for P-boxes

Between two rounds of S-boxes, there are one straight P-box (32 bit to 32 bit) and one Expansion P-box (32 bit to 48 bit). These two P-boxes together provide diffusion of bits.

- Each S-box input comes from the output of a different S- box (in the previous round).

- The four outputs from each S-box go to six different S-boxes (in the next round).

- No two output bits from an S-box go the same S-boxes (in the next round).

- For each S-box, the two output bits go to the first or last two bits of an S-box in the next round. The other two output bits go the middle bits of an S-box in the next round.

Design Criteria for P-boxes Contd.

If we number the eight S-boxes, S_1, S_2, S_3, …, S_8

- An output of S_{j-2} goes to one of the first two bits of S_j (in the next round).
- An output bit from S_{j-1} goes to one of the last two bits of S_j (in the next round).
- An output of S_{j+1} goes to one of the two middle bits of S_j (in the next round)

DES Weaknesses

During the last few years critics have found some weakness in DES

Weakness in S-Boxes

- At least three weaknesses are mentioned in the literature for S-boxes
- In S-box 4, the last three output bits can be derived in the same way as the first output bit by complementing some of the input bits.
- Two specifically chosen inputs to an S-box array can create the same output.
- It is possible to obtain the same output in a single round by changing bits in only three neighboring S-boxes.

Weakness in P-boxes

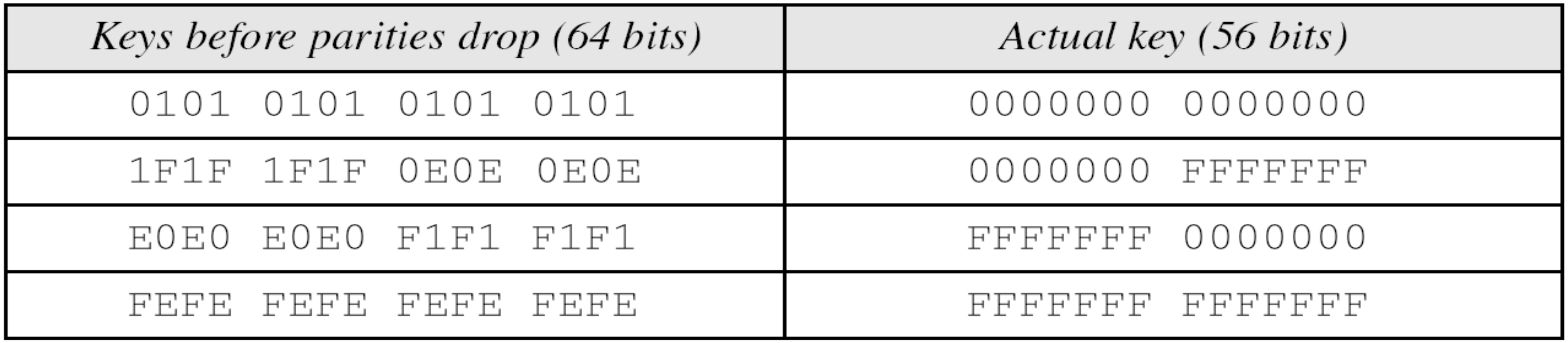
- It is not clear why the designer of DES used the initial and final permutation; these have no security benefits.
- In the expansion permutation (inside the function), the first and fourth bits of every 4-bit series are repeated.

Number of Rounds

DES uses sixteen rounds of Feistel ciphers. the ciphertext is thoroughly a random function of plaintext and ciphertext.

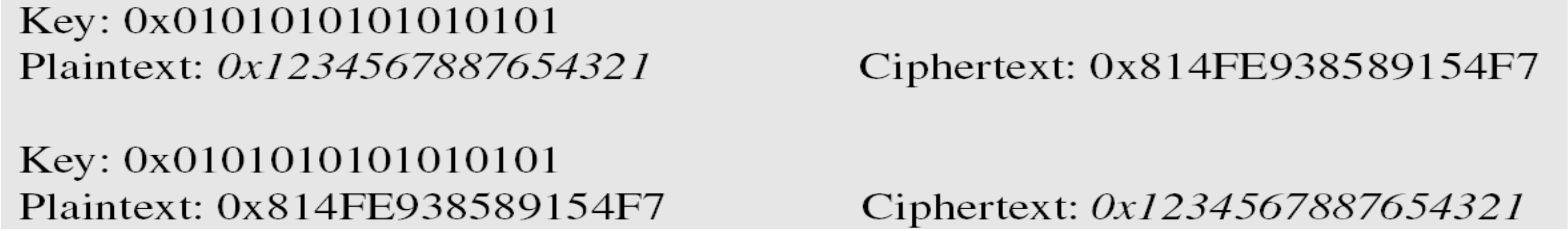
Weakness in Keys

Table 6.18 *Weak keys*



Keys before parities drop (64 bits)	*Actual key (56 bits)*
0101 0101 0101 0101	0000000 0000000
1F1F 1F1F 0E0E 0E0E	0000000 FFFFFFF
E0E0 E0E0 F1F1 F1F1	FFFFFFF 0000000
FEFE FEFE FEFE FEFE	FFFFFFF FFFFFFF

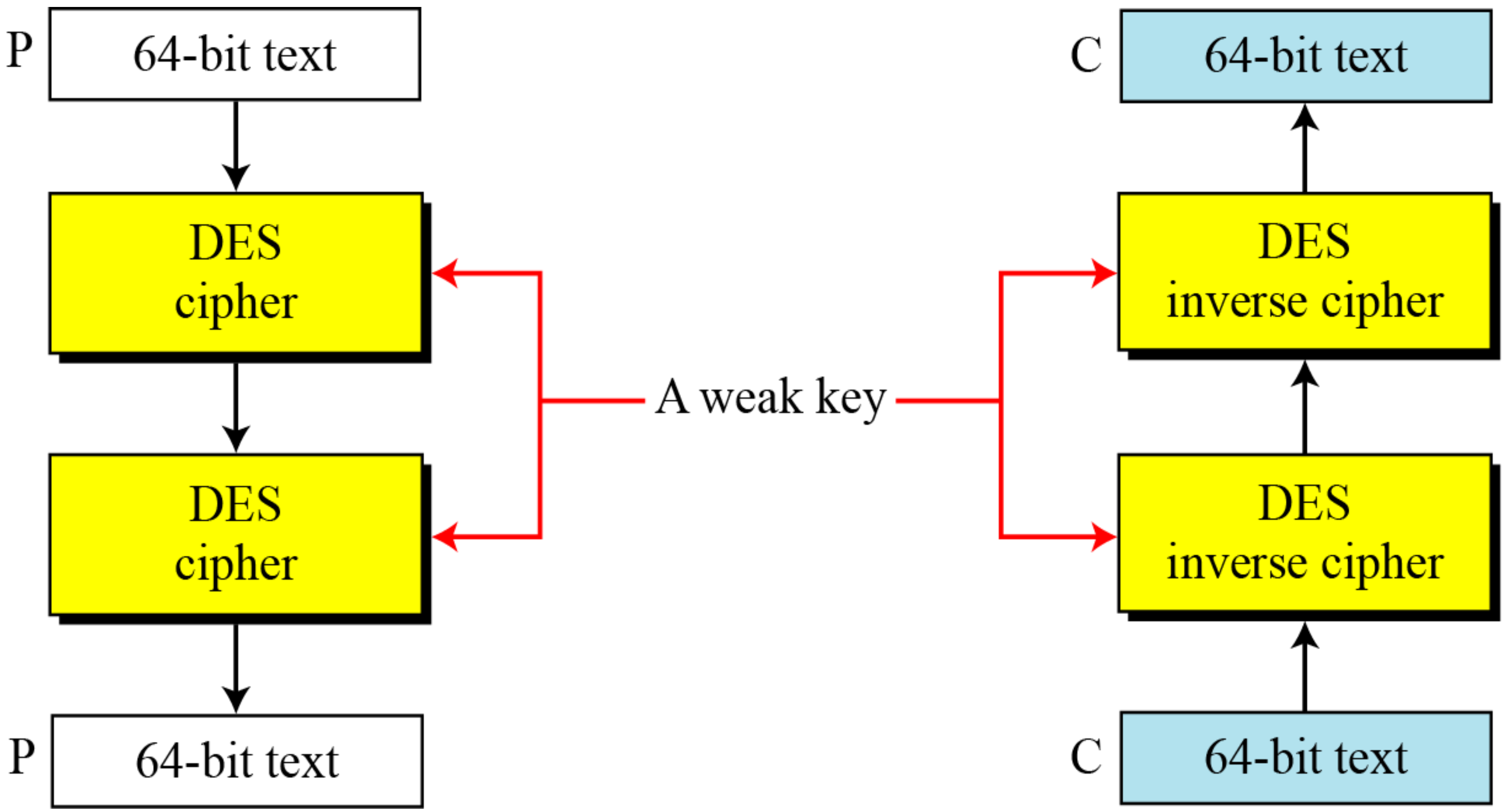
Let us try the first weak key in Table 6.18 to encrypt a block two times. After two encryptions with the same key the original plaintext block is created. Note that we have used the encryption algorithm two times, not one encryption followed by another decryption.



Key: 0x0101010101010101
Plaintext: *0x1234567887654321* Ciphertext: 0x814FE938589154F7

Key: 0x0101010101010101
Plaintext: 0x814FE938589154F7 Ciphertext: *0x1234567887654321*

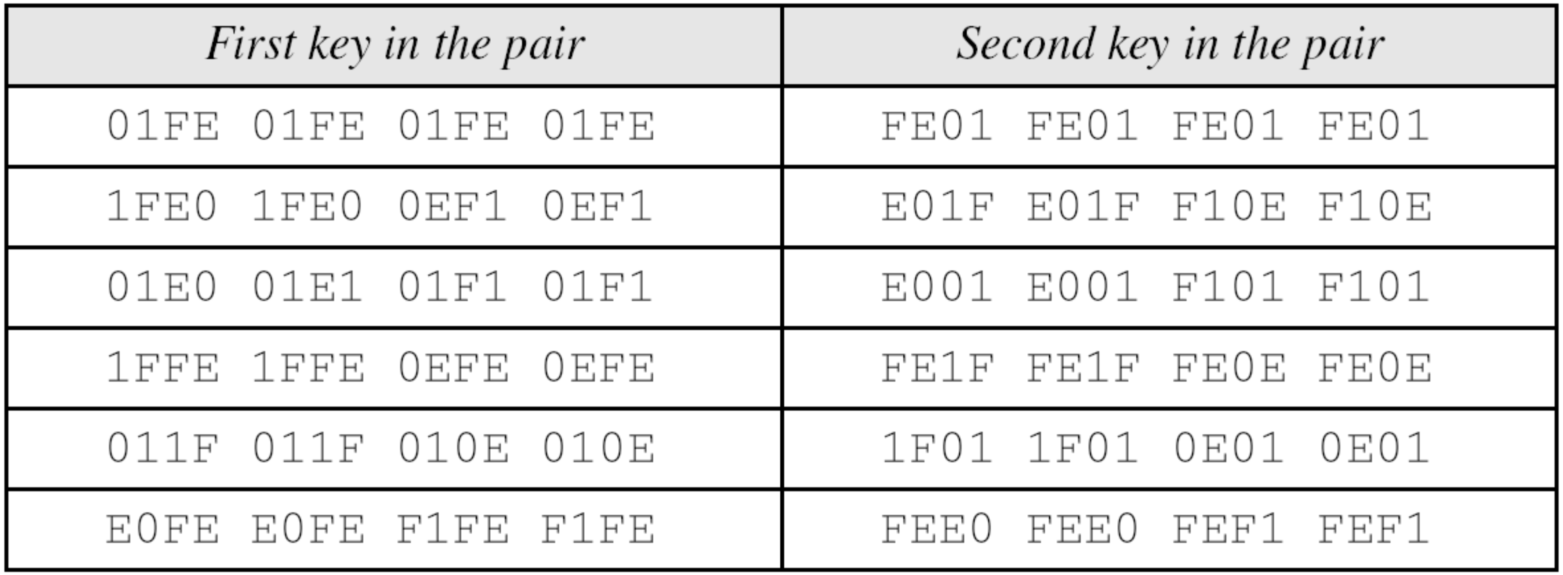
Double Encryption and Decryption with a Weak Key



Semi Weak Keys

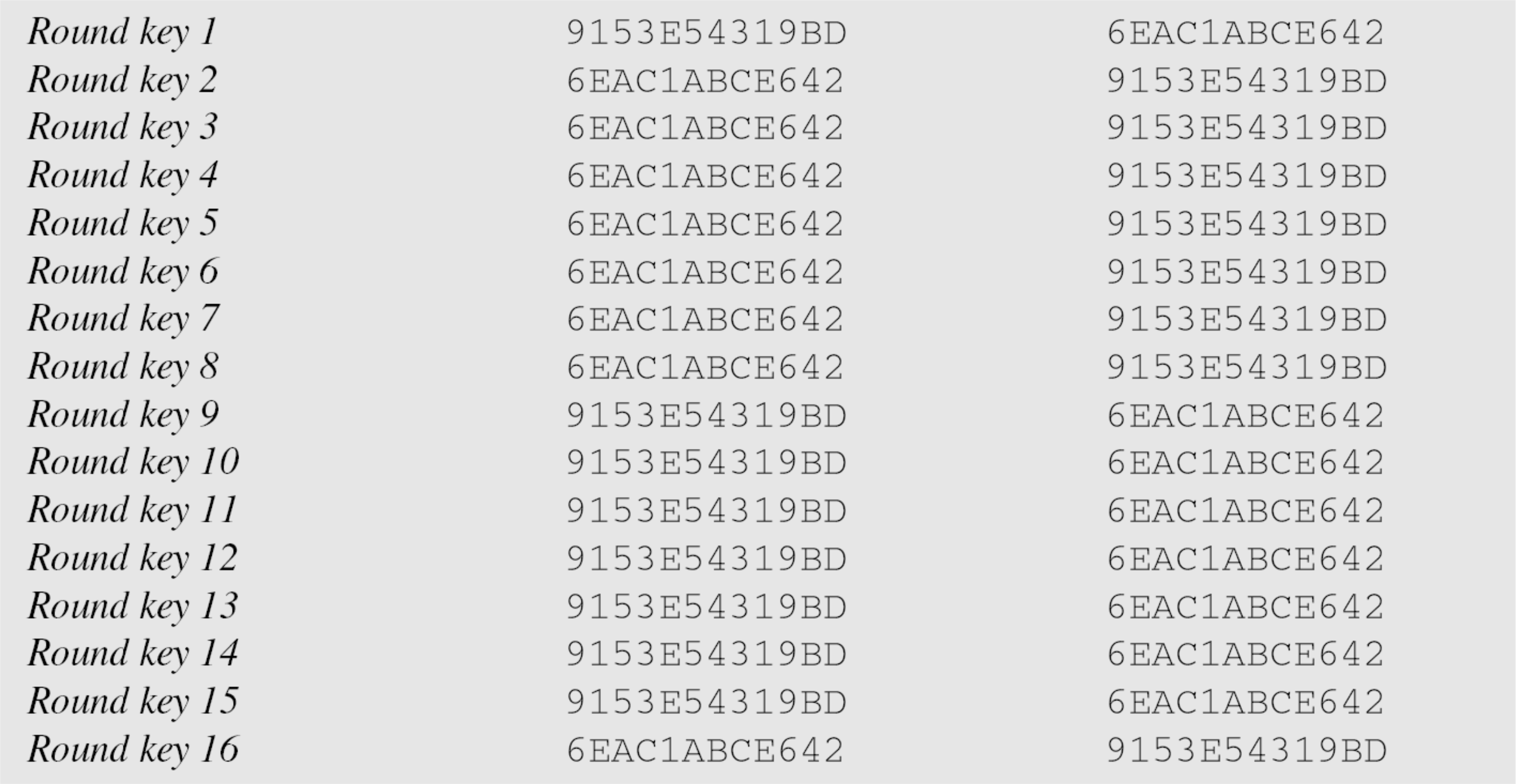
There are six key pairs that are called semi weak keys and they are shown in the below table. A semi weak key creates only two different round keys and each of them is repeated eight times. In addition, the round keys created from each pair are the same with different orders.

Table 6.19 *Semi-weak keys*

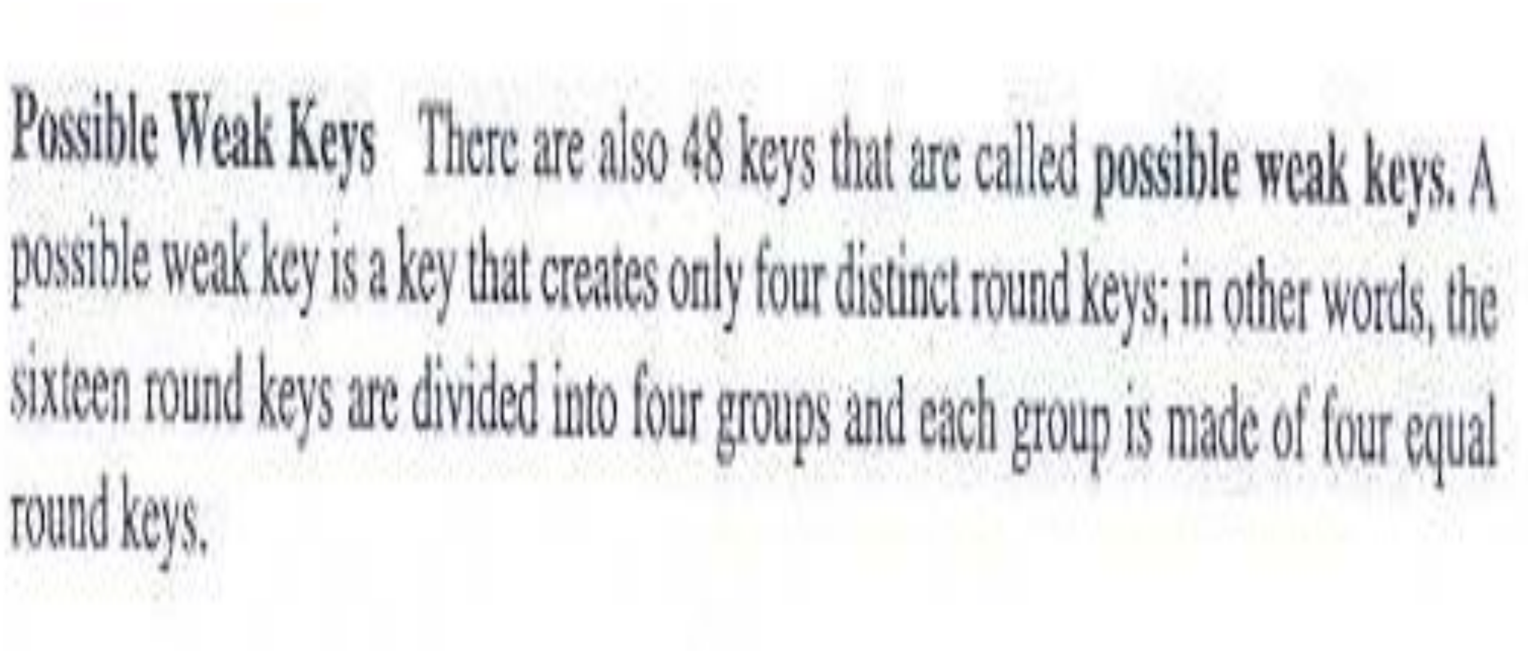


First key in the pair	*Second key in the pair*
01FE 01FE 01FE 01FE	FE01 FE01 FE01 FE01
1FE0 1FE0 0EF1 0EF1	E01F E01F F10E F10E
01E0 01E1 01F1 01F1	E001 E001 F101 F101
1FFE 1FFE 0EFE 0EFE	FE1F FE1F FE0E FE0E
011F 011F 010E 010E	1F01 1F01 0E01 0E01
E0FE E0FE F1FE F1FE	FEE0 FEE0 FEF1 FEF1

Semi Weak Keys



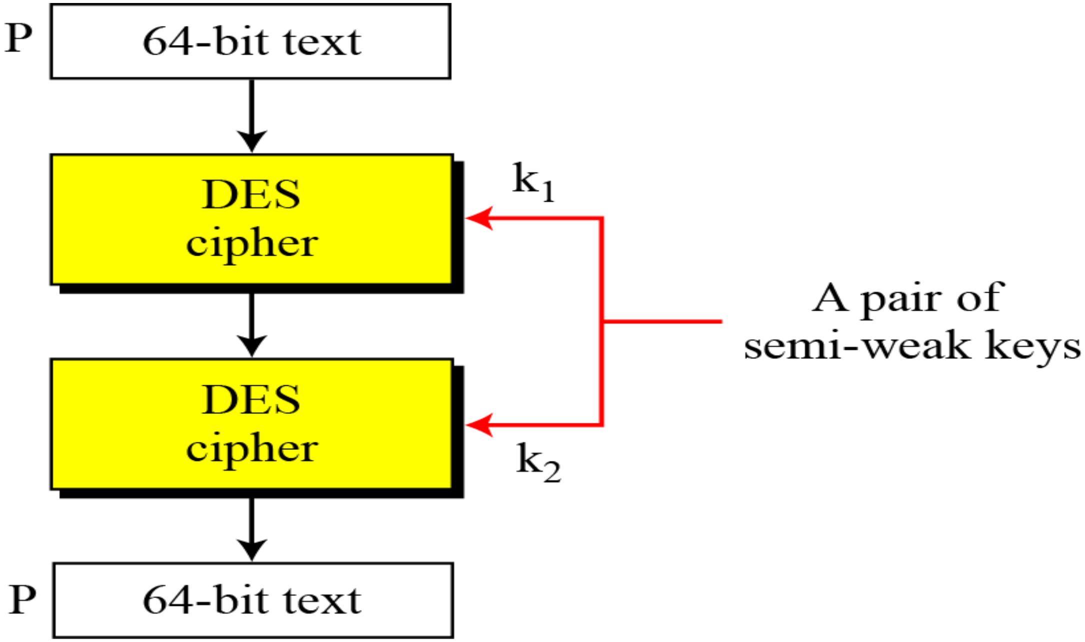
Round key 1	9153E54319BD	6EAC1ABCE642
Round key 2	6EAC1ABCE642	9153E54319BD
Round key 3	6EAC1ABCE642	9153E54319BD
Round key 4	6EAC1ABCE642	9153E54319BD
Round key 5	6EAC1ABCE642	9153E54319BD
Round key 6	6EAC1ABCE642	9153E54319BD
Round key 7	6EAC1ABCE642	9153E54319BD
Round key 8	6EAC1ABCE642	9153E54319BD
Round key 9	9153E54319BD	6EAC1ABCE642
Round key 10	9153E54319BD	6EAC1ABCE642
Round key 11	9153E54319BD	6EAC1ABCE642
Round key 12	9153E54319BD	6EAC1ABCE642
Round key 13	9153E54319BD	6EAC1ABCE642
Round key 14	9153E54319BD	6EAC1ABCE642
Round key 15	9153E54319BD	6EAC1ABCE642
Round key 16	6EAC1ABCE642	9153E54319BD



Possible Weak Keys There are also 48 keys that are called **possible weak keys.** A possible weak key is a key that creates only four distinct round keys; in other words, the sixteen round keys are divided into four groups and each group is made of four equal round keys.

Semi Weak Keys

A pair of semi-weak keys in Encryption and Decryption

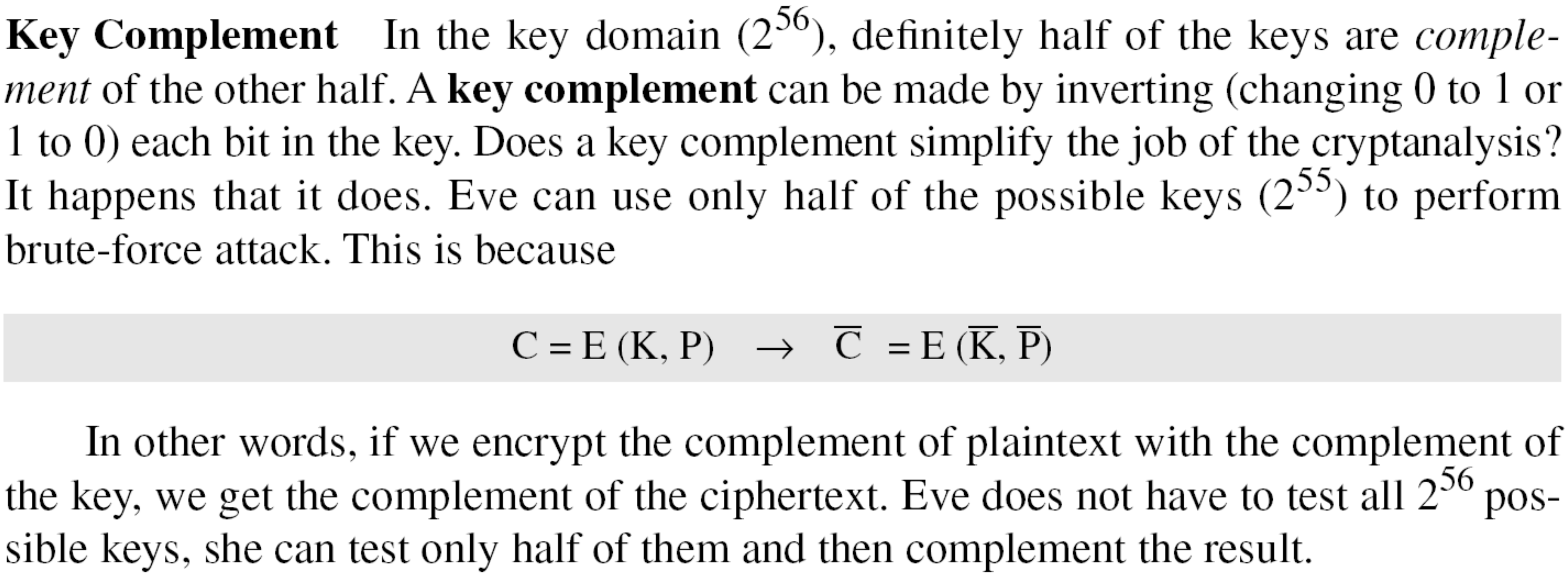


What is the probability of randomly selecting a weak, a semi-weak, or a possible weak key?

Solution

DES has a key domain of 2^{56}. The total number of the above keys are 64 (4 + 12 + 48). The probability of choosing one of these keys is 8.8×10^{-16}, almost impossible.

Key Complement



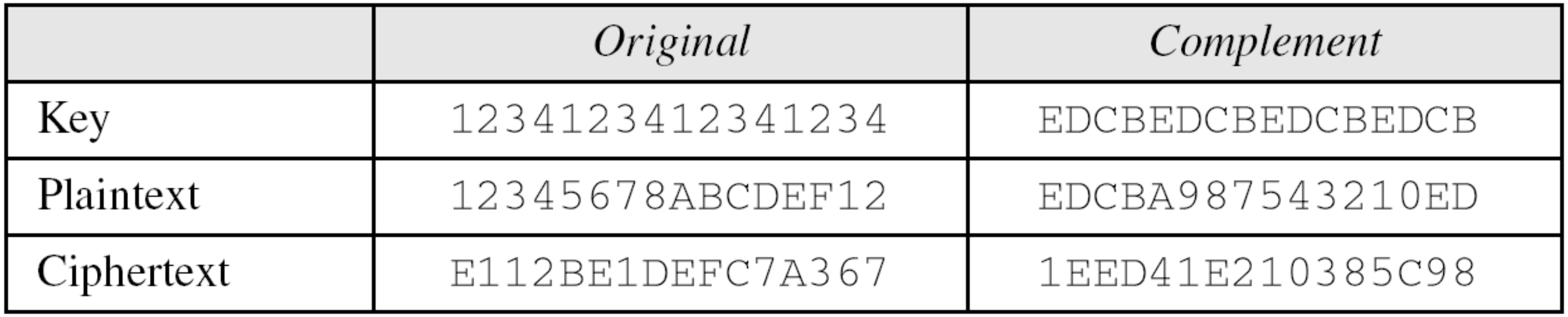
Key Complement In the key domain (2^{56}), definitely half of the keys are *complement* of the other half. A **key complement** can be made by inverting (changing 0 to 1 or 1 to 0) each bit in the key. Does a key complement simplify the job of the cryptanalysis? It happens that it does. Eve can use only half of the possible keys (2^{55}) to perform brute-force attack. This is because

$$C = E(K, P) \quad \rightarrow \quad \overline{C} = E(\overline{K}, \overline{P})$$

In other words, if we encrypt the complement of plaintext with the complement of the key, we get the complement of the ciphertext. Eve does not have to test all 2^{56} possible keys, she can test only half of them and then complement the result.

Let us test the claim about the complement keys. We have used an arbitrary key and plaintext to find the corresponding ciphertext. If we have the key complement and the plaintext, we can obtain the complement of the previous ciphertext (Table 6.20).

Table 6.20 *Results for Example 6.10*



	Original	*Complement*
Key	1234123412341234	EDCBEDCBEDCBEDCB
Plaintext	12345678ABCDEF12	EDCBA987543210ED
Ciphertext	E112BE1DEFC7A367	1EED41E210385C98