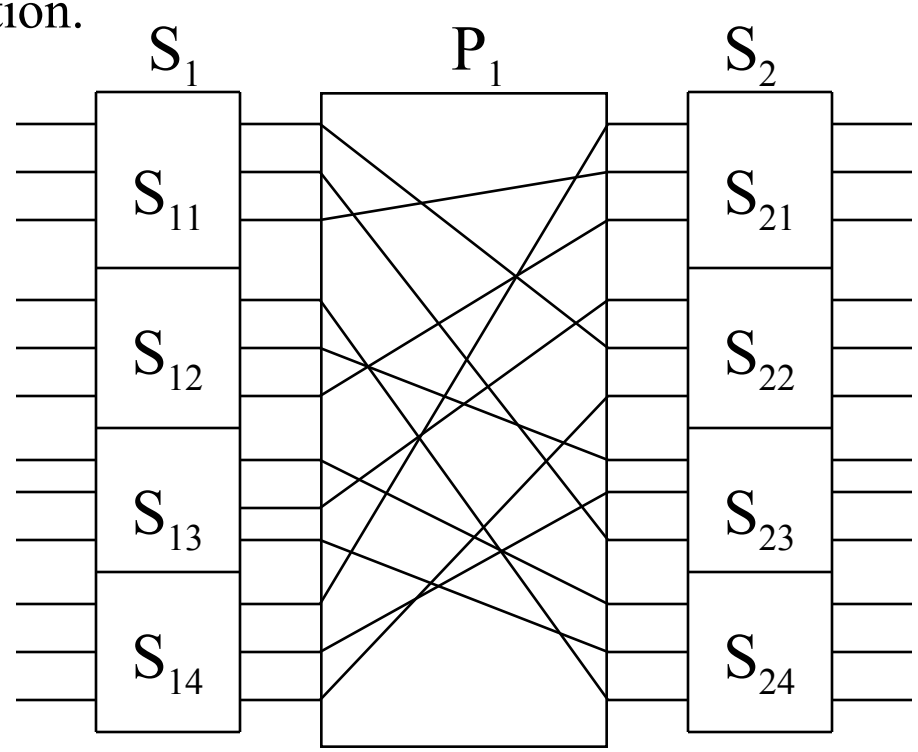
4. Data Encryption Standard (DES)

4.1. Substitution-Permutation Cipher

Shannon proposed composing different kinds of functions to create "mixing transformations", which randomly distribute the meaningful messages uniformly over the set of all possible ciphertext messages. Mixing transformations could be created, for example, by applying a transformation followed by an alternating sequence of substitutions and simple linear operations.

This approach is embodied in the LUCIFER cipher, designed at IBM by Feistl. LUCIFER uses a transformation that alternately applies substitutions and transposition.



The substitutions S_i are broken into 4 smaller substitutions S_{i1} ,...,S_{i4} each operating on a 3-bit subblock to reduce the complexity of the circuits.

Because the permutation P_i shuffle all 12-bit of plaintext can conceivable affect each bit of ciphertext.

4. Data Encryption Standard (DES)

4.2. The Data Encryption Standard (DES)

In 1977 the National Bereau of Standards announced a ***Data Encryption Standard*** to be used in unclassified U.S. Government applications. The encryption algorithm was developed at IBM, and was the outgrowth of LUCIFER.

DES enciphers 64-bit blocks of data with a 56-bit key. There is disagreement over whether a 56-bit key is sufficiently strong; we shall discuss controversy after describing the algorithm.

The algorithm - which is used both to encipher and to decipher - is summarized in ***Figure 4.1***. An input block T is first transposed under an initial permutation IP, $T_0=IP(T)$. After it has passed through 16 iterations of a function f, it is transposed under the inverse permutation IP^{-1} to give the final result. The permutations IP and IP^{-1} are given in Table 4.1 and Table 4.2, respectively. These tables read left-to-right, top-to-bottom. For example IP transpose $T=t_1t_2,...,t_{64}$ into $T_0=t_{58}t_{50},...,t_7$. All tables are fixed.

Between the initial and final transpositions, the algorithm performs 16 iteration of a function f that combines substitution and transposition. Let T_i denote the result of the ith iteration, and let L_i and R_i denote the left and right halves of T_i, respectively; that is $T_i = L_iR_i$, where

$$L_i = t_1,...,t_{32},\ R_i = t_{33},...,t_{64}.$$

Then

$$L_i = R_{i-1},\ R_i = L_{i-1}+f(R_{i-1},K_i),$$

where "+" is the exclusive-or operation and K_i is a 48-bit key.

4. Data Encryption Standard (DES)

4.2. The Data Encryption Standard (DES)

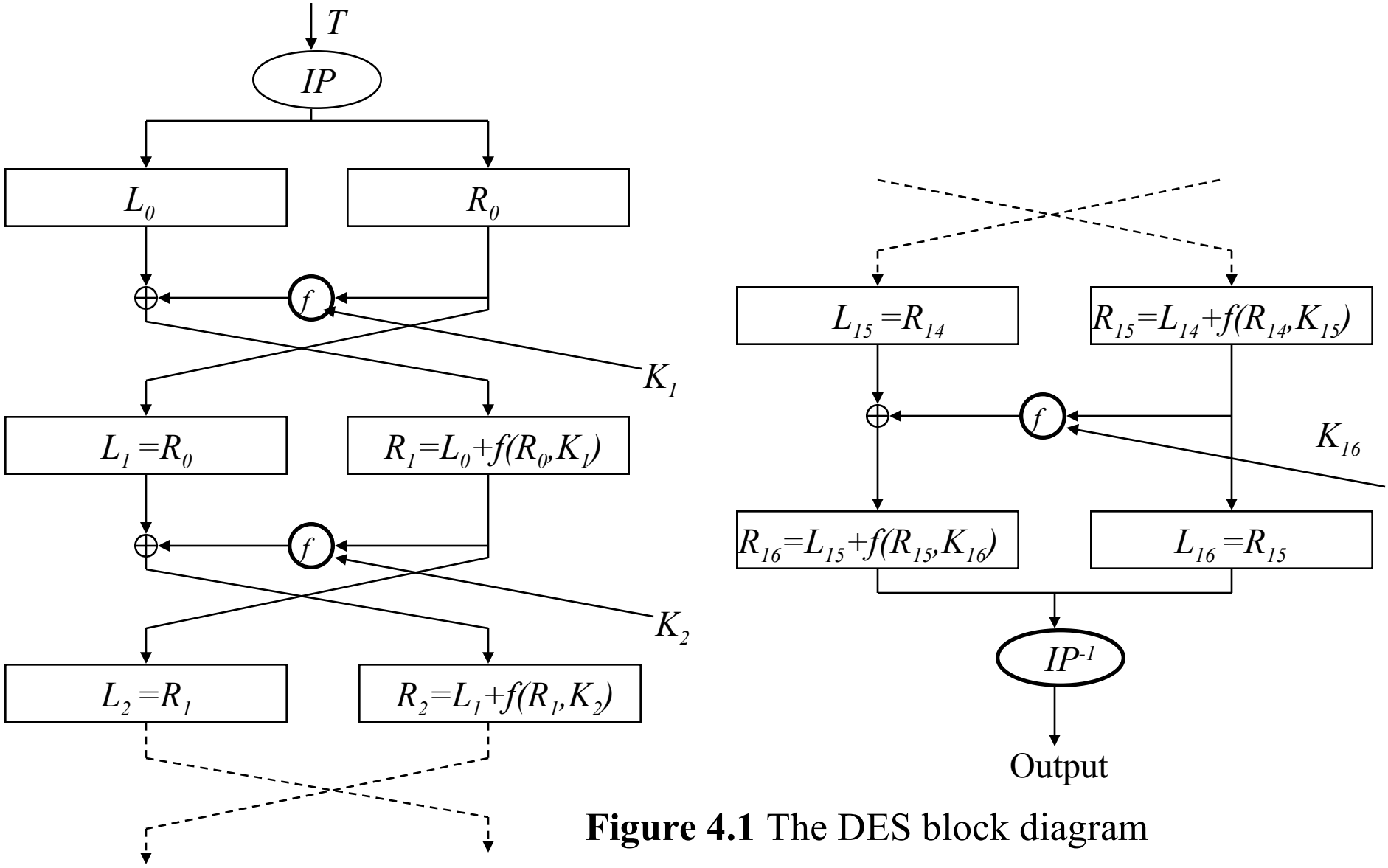


Figure 4.1 The DES block diagram

4. Data Encryption Standard (DES)

4.2. The Data Encryption Standard (DES)

Table 4.1. 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

Table 4.2. Final 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

Table 4.3. Bit-selected table *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

Table 4.3. Permutation *P*

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

4. Data Encryption Standard (DES)

4.3. DES: The function f

The next ***Figure 4.2*** shows the function $f(R_{i-1},K_i)$. First R_{i-1} is expanded to a 48-bit block $E(R_{i-1})$ using the bit-selection table E shown in Table 4.3. This table is used in the permutation tables, exempt that bits of R_{i-1} are selected more than once; thus given $R_{i-1}= r_1 r_2...r_{32}$, $E(R_1)=r_{32}r_1r_2...r_{32}r_1$. Next, the exclusive-or $E(R_{i-1})$ of and K_i is calculated and the result broken into eight 6-bit blocks $B_1B_2,...,B_8$, where $E(R_{i-1})+ +K_i=B_1B_2,...,B_8$. Each *6*-bit block B_i is then used as input to selection (substitution) function (S-box) S_j, which returns a 4-bit block $S_j(B_j)$. These blocks are concatenated together, and the resulting 32-bit block is transposed by permutation P shown in Table 4.4. Thus, the block returned by $f(R_{i-1},K_i)$ is $P(S_1(B_1)...S_8(B_8))$.

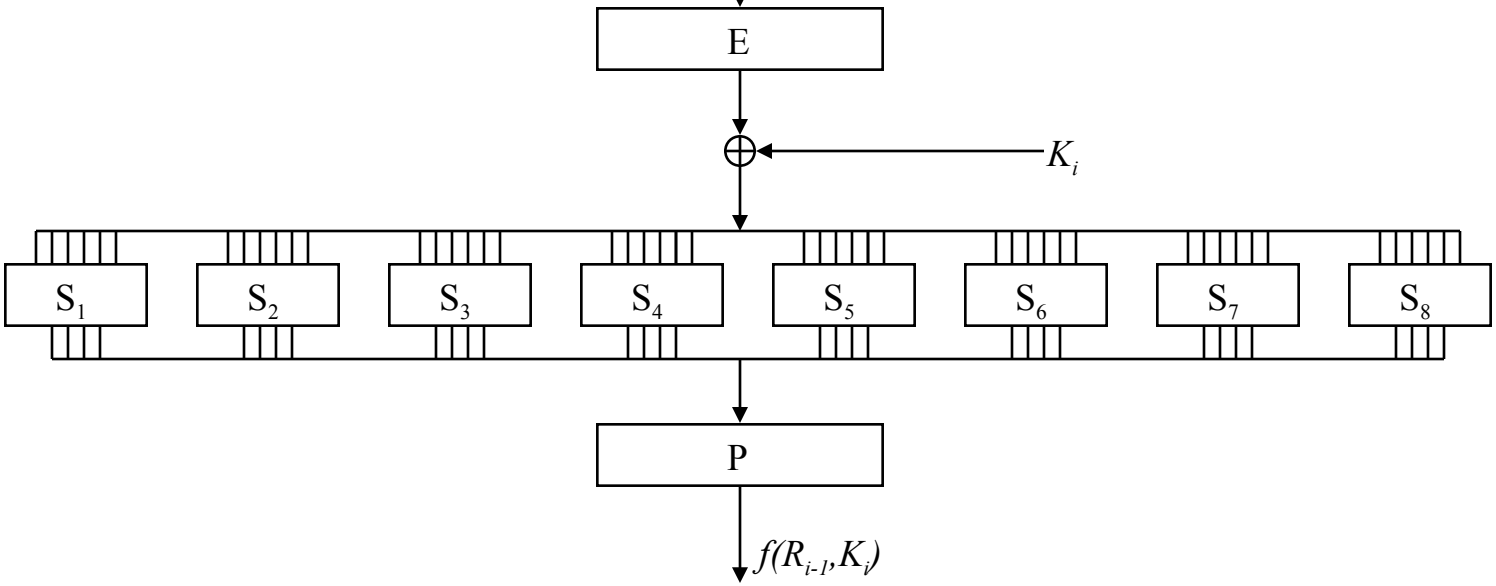


Figure 4.2. The function $f(R_{i-1},K_i)$.

4. Data Encryption Standard (DES)

4.4. DES: S-boxes

Each *S*-box S_j maps a 6-bit block $B_j=b_1b_2b_3b_4b_5b_6$ into a 4-bit block as defined in ***Table 4.5***.

Table 4.5. Selection functions (S-boxes)

	Column																
Row	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	S_1
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13	
...								...									...

This is done as follows: The integer corresponding b_1b_6 selects a row in the table, while the integer corresponding $b_2b_3b_4b_5$ selects a column. The value of $S_j(B_j)$ is then the 4-bit representation of the integer in that row and column.

Example 4.1. If B=010100, then returns the value in row 0, column 10; this is 6, which is represented as 0110.

4. Data Encryption Standard (DES)

4.5. Key calculation

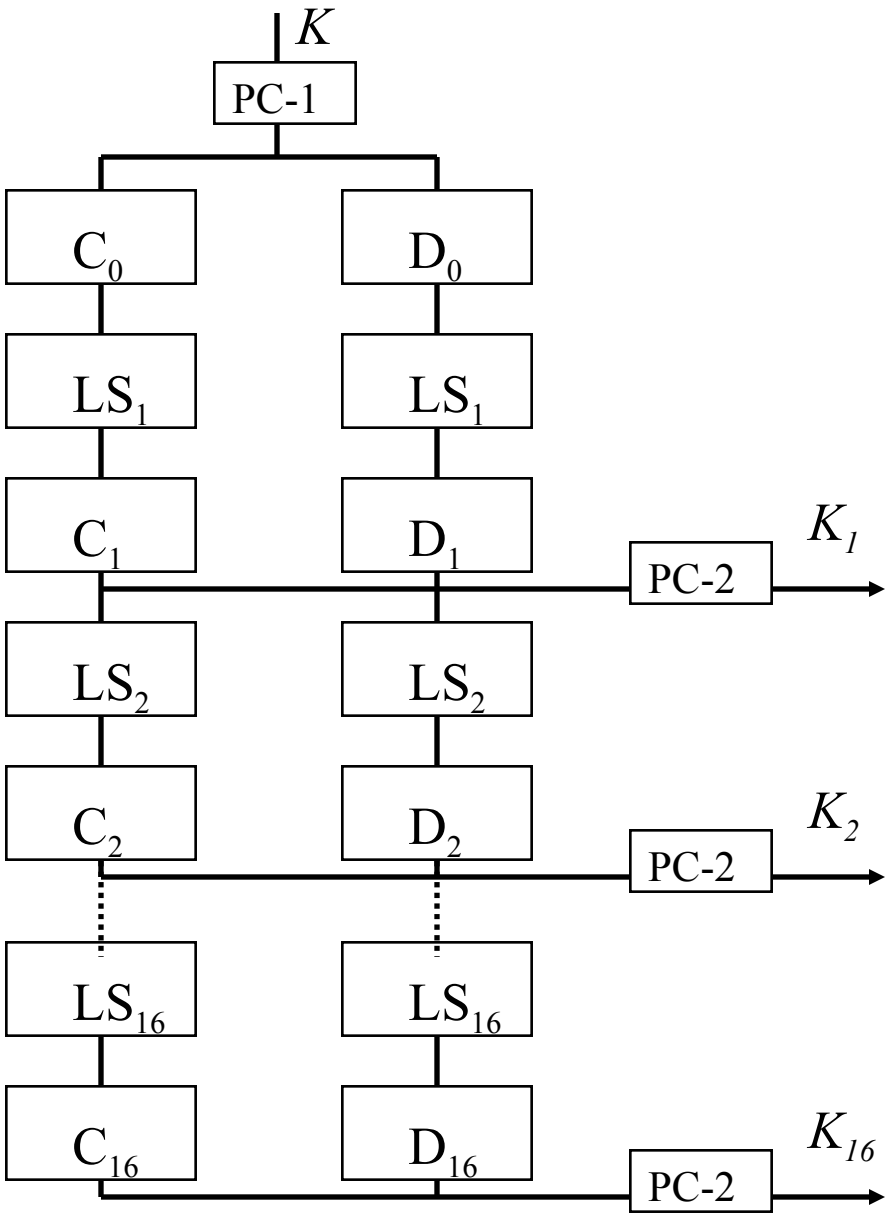


Table 4.6. Key permutation PC-1

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

Table 4.7. Key permutation 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

Figure 4.3. Key schedule calculation.

4. Data Encryption Standard (DES)

4.5. Key calculation

Each iteration *i* uses a different 48-bit key K_i derived from the initial key *K*. Figure 4.3. illustrates how this is done. K is input as a 64-bit block, with 8 parity bits in positions 8, 16,...,64. The permutation PC-1 (permuted choice 1) discards the parity bits and transposes the remaining 56 bits as shown in ***Table 4.6***. The result PC-1(K) is then split into two halves *C* and *D* of 28 bits each. The blocks *C* and *D* are then successively shifted left to derive each key K_i. Letting C_i and D_i denotes the value of *C* and *D* used to derive K_i , we have

$$C_i = LS_j(C_{i-1}),\ D_i = LS_j(D_{i-1}),$$

where LS_i is a left circular shift by the number of positions shown in ***Table 4.8***, and C_0 D_0 are the initial values of *C* and *D*. Key K_i is then given by

$$K_j = PC\text{-}2(C_i, D_i).$$

Table 4.8. Key schedule of left shifts LS.

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
~~Nmuber Shifts~~	~~1~~	~~1~~	~~2~~	~~2~~	~~2~~	~~2~~	~~2~~	~~2~~	~~1~~	~~2~~	~~2~~	~~2~~	~~2~~	~~2~~	~~2~~	~~1~~

4. Data Encryption Standard (DES)

4.5. Deciphering and Implementation

Deciphering. Deciphering is performed using the same algorithm, exept that K_{16} is used in the first iteration, K_{15} in the second, and so on, with K_1 used in the *16*th iteration. This is because the final permutation IP^{-1} is the inverse of the initial permutation IP, and

$$R_{i-1}=L_i,\ L_{i-1}= R_i+f(L_i,K_i).$$

Note that whereas the order of keys is reversed, the algorithm itself is not.

Implementation. DES has been implemented both in software and in hardware. Hardware implementations achieve encryption rates of several million bps (bits per second).

Two major DES's weaknesses.

1.***Key size***: 56 bits may not provide adequate security.

2.***S-boxes***: The S-boxes may have hidden trapdoors.

4. Data Encryption Standard (DES)

4.6. DES: Weak and Semiweak Keys

It must realized that the cryptographic strength, would be reduced if the internal keys at each round were the same. For this reason, the condition $K(1)=K(2)= \ldots +K(16)$ should be avoided. There is, however, a set of ***weak keys*** within DES which satisfy the above condition. This occurs whenever the bits in register *C* are all ones or zeros, and the bits in register *D* are all ones or zeros. In this case, C1 and D1 are, respectively,

$$k_{49}=k_{41}=k_{33}=\ldots=k_{57}=0 \text{ or } 1$$

and

$$k_{55}=k_{47}=k_{39}=\ldots=k_{63}=0 \text{ or } 1.$$

Thus there are four weak keys altogether and they are represented by the following (parity-adjusted) external keys:

01 01 01 01 01 01 01 01
1F 1F 1F 1F 1F 1F 1F 1F
E0 E0 E0 E0 E0 E0 E0 E0
FE FE FE FE FE FE FE FE

There is another set of keys defined as ***semiweak***. These have the property that only two different internal keys are produced, each occurring eight times. A semiweak key occurs whenever

1. Register C or D contains bit pattern 0101...0101 or 1010...1010.

2. The other register (D or C) contains bit pattern 0000...0000, 1111...1111, 0101...0101, 1010...1010.

4. Data Encryption Standard (DES)

4.7. DES: Implementation

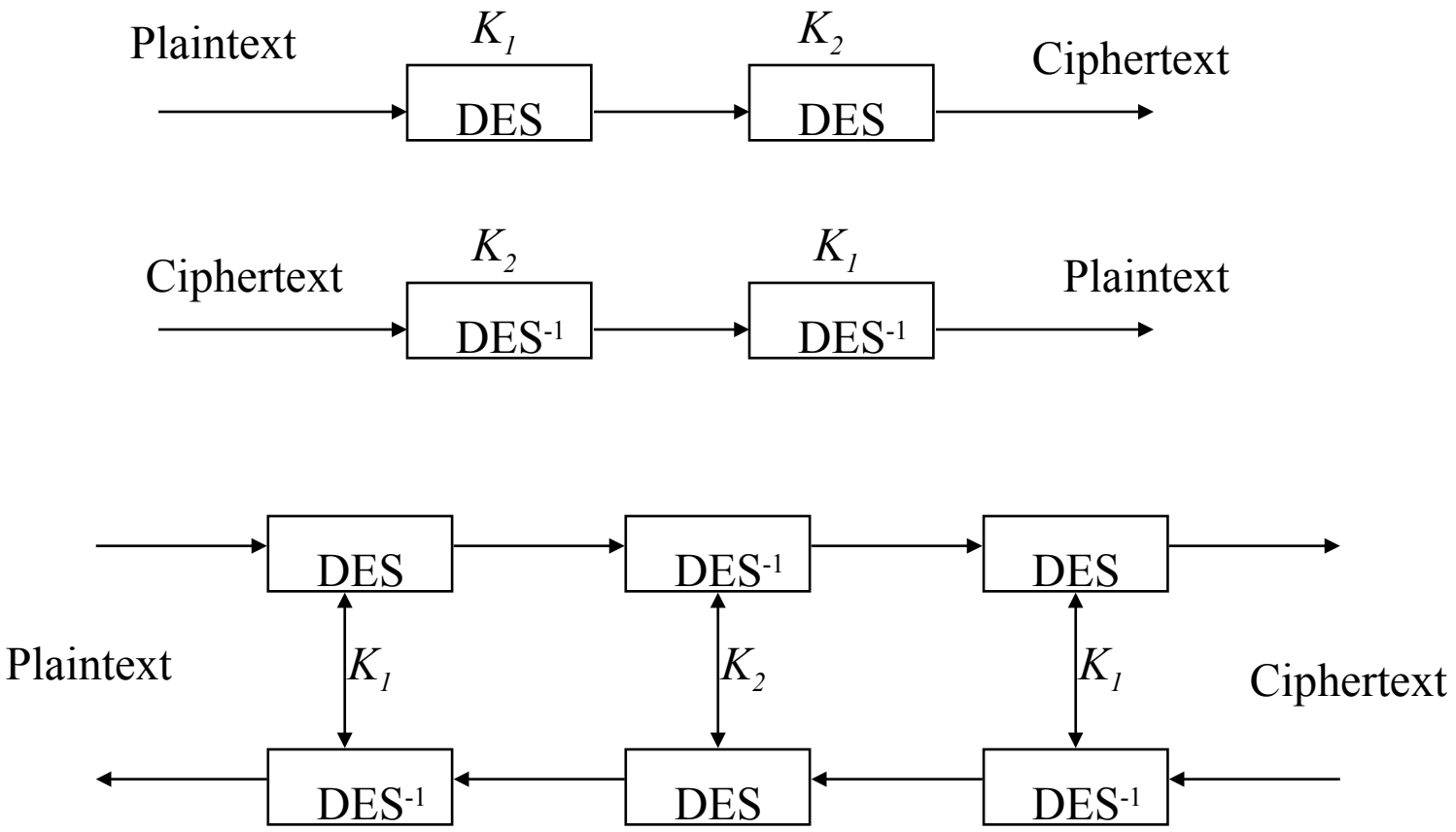


Figure 4.4. Multiple Encipherment with DES.