Date: Monday, September 23, 2002
Prof.: Dr Jean-Yves Chouinard

Design of Secure Computer Systems CSI4138/CEG4394

Notes on the Data Encryption Standard (DES)

The Data Encryption Standard (DES) has been developed as a cryptographic standard for general use by the public. DES was designed with the following objectives in mind [NIS77, Pfl89]:

1. High level of security
2. Completely specified and easy to understand
3. Cryptographic security do not depend on algorithm secrecy
4. Adaptable to diverse applications
5. Economical hardware implementation
6. Efficient (e.g. high data rates)
7. Can be validated
8. Exportable

1 Data Encryption Algorithm

- Substitution-permutation algorithm:
  - 64-bit input and output blocks
  - 56-bit key (with an additional 8 parity bits)
  - information data is cycled 16 times through a set of substitution and permutation transformations: highly non-linear input-output relationship
- Very high throughput rates achievable (up to 100 $Mbits/s$)
- Availability of economical hardware to implement DES
- Low to medium security applications (e.g. secure speech communications)

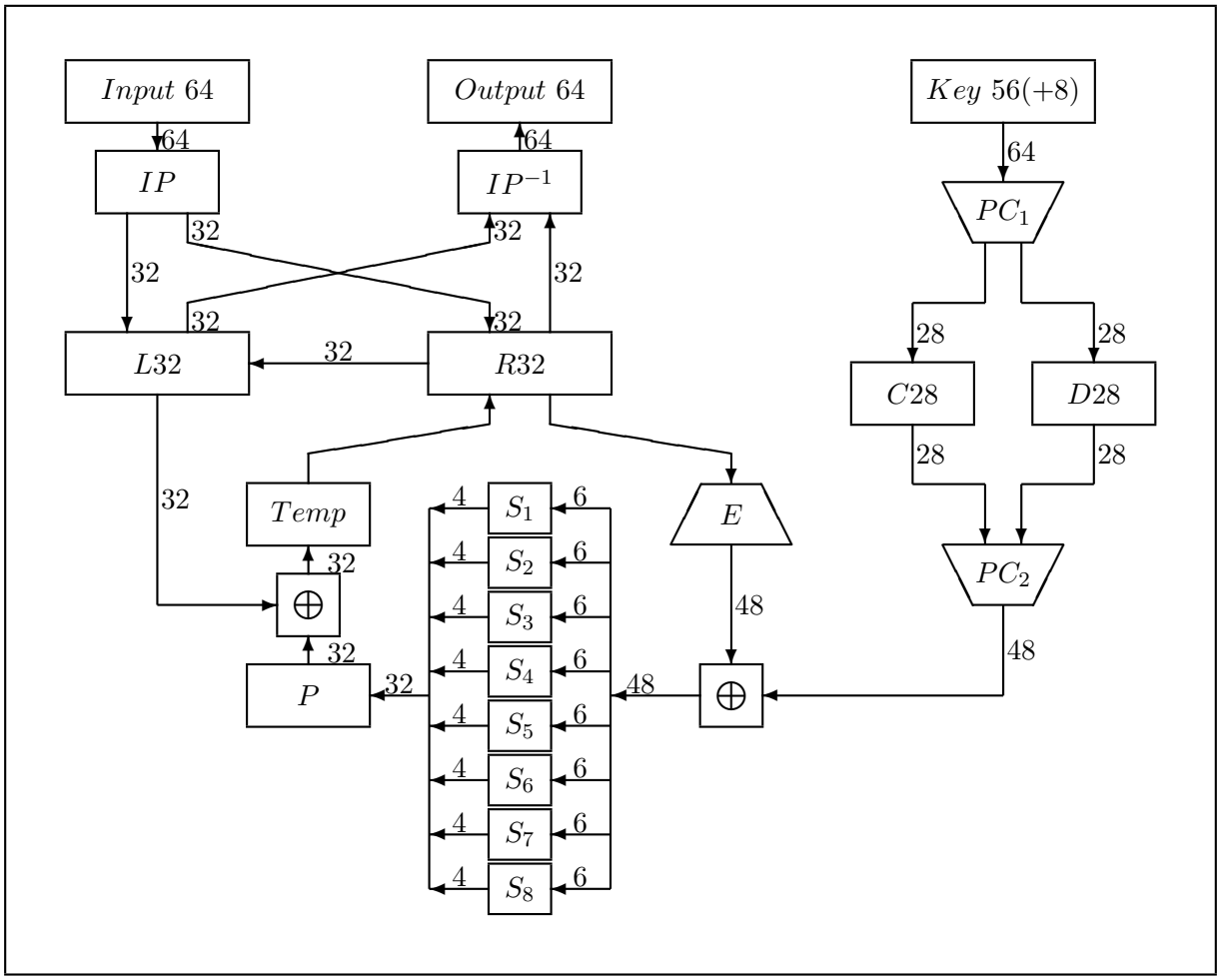


Figure 1: DES encryption/decryption algorithm.

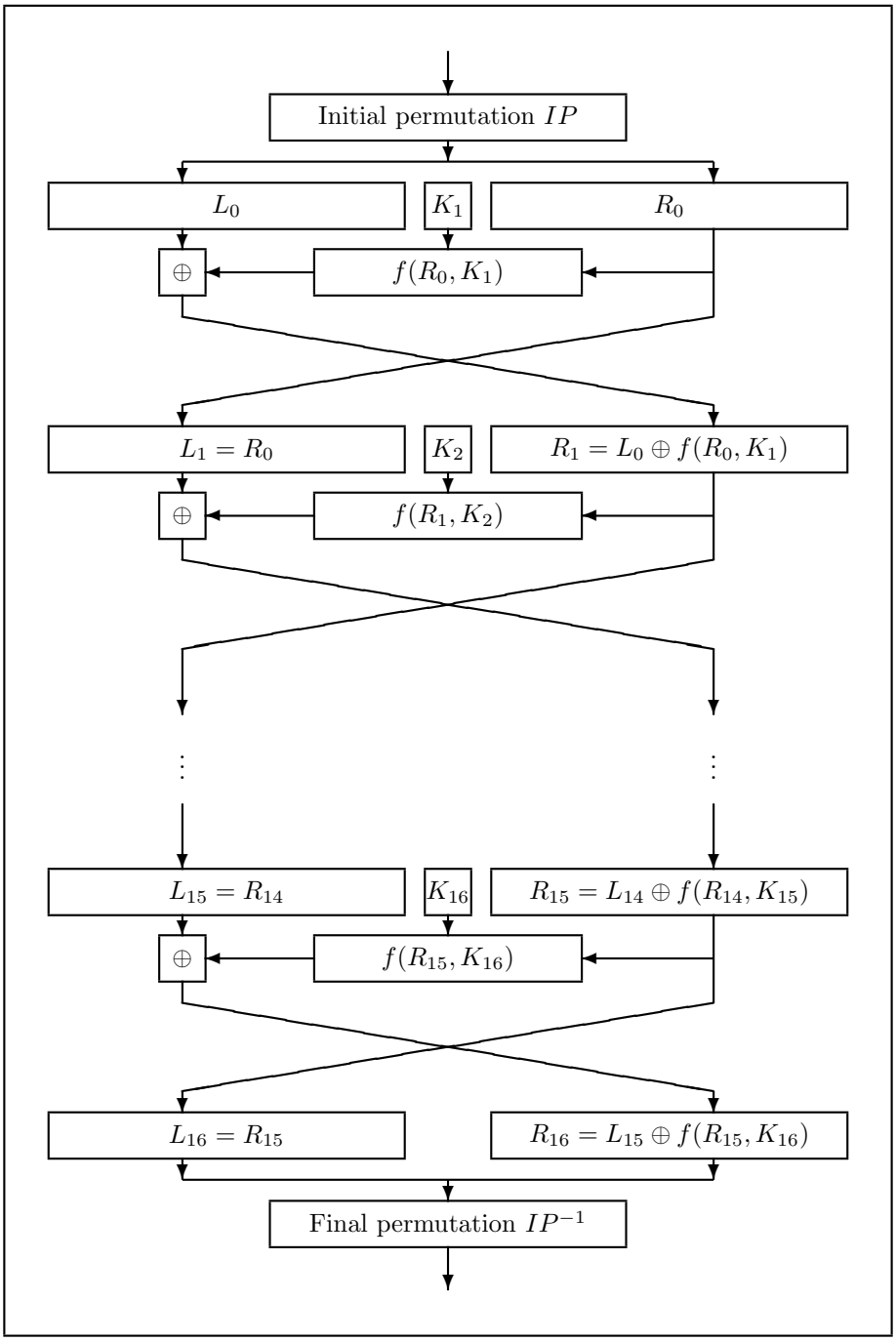


Figure 2: DES sequence of substitution and permutation transformations.

Table 1: Initial IP and inverse initial IP^{-1} permutation tables.

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

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 2: Expansion permutation E and permutation P tables.

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

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

Table 3: S-boxes (substitution boxes) tables.

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

Table 4: Permuted choices PC_1 and PC_2 tables.

Permuted choice 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

Permuted choice 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

Table 5: Key schedule of left shifts (encryption).

Cycle i	Amount of left shifts
1	1
2	1
3	2
4	2
5	2
6	2
7	2
8	2
9	1
10	2
11	2
12	2
13	2
14	2
15	2
16	1

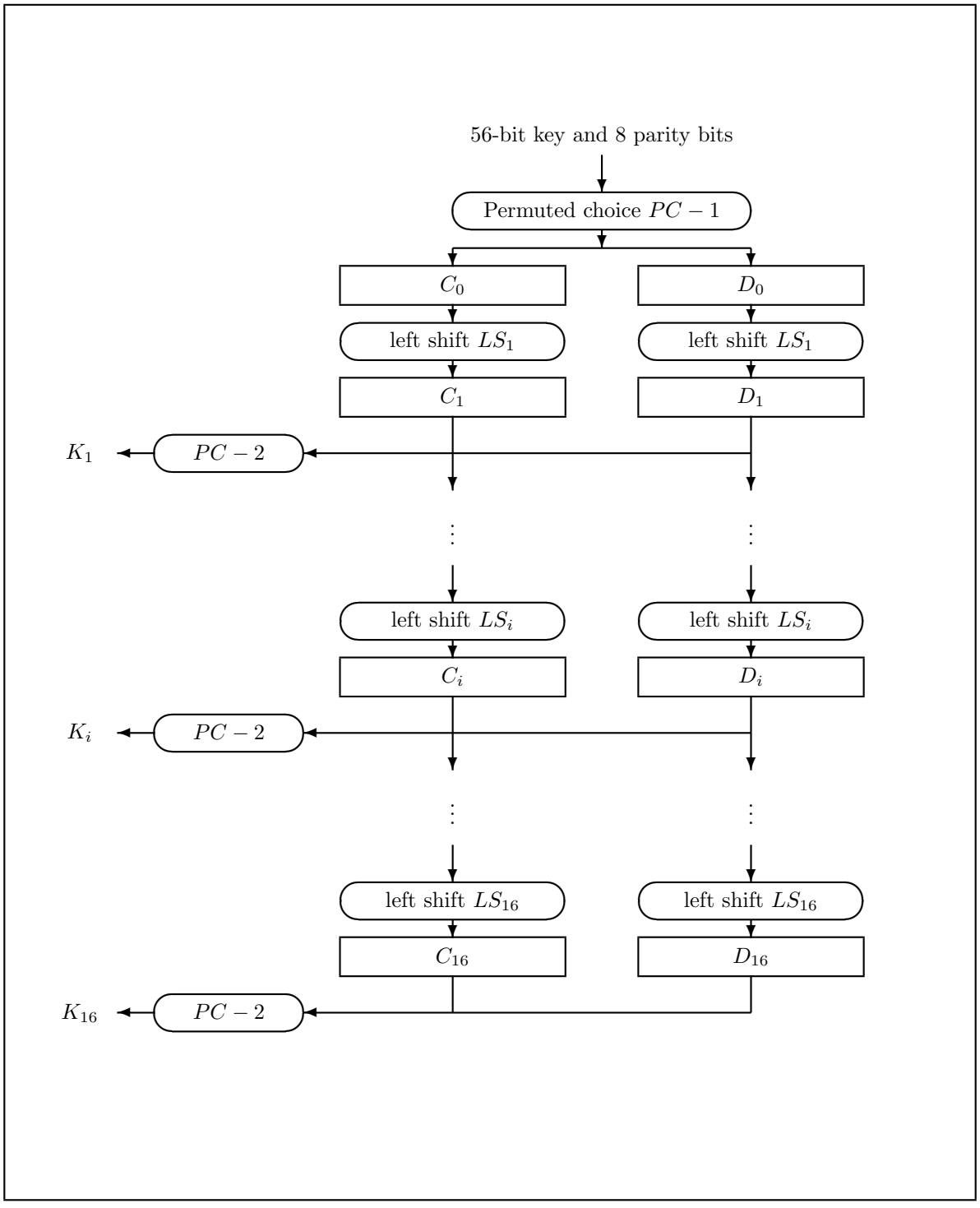


Figure 3: Sub-key schedule calculation.

2 Diffusion properties of DES

The Data Encryption Standard algorithm diffuses the encipherment transformation over the whole 64-bit ciphertext within the 16 substitution and transposition transformation rounds (or cycles). The table 6 (taken from [DP84]) gives the left and right register contents at each cycle i (i.e. L_i and R_i, $i = 1, \ldots, 16$). The plaintext message

$$M_1 = \text{“}0000000000000000''$$

while the second message

$$M_2 = \text{“}0000000000000001''$$

differs only by one bit, that is;

$$d_H(M_1, M_2) = 1$$

and the encipherment key

$$K = \text{“}08192A3B4C5D6E7F''.$$

It can be easily seen that, as the plaintext goes through the series of substitution and permutation transformations, the Hamming distance between the contents of the left and right registers increases from 1 to about half of the 64 bits, indicating the diffusion effect of DES.

Table 6: Hamming distance d_H between the L and R register contents as a function of the DES sub-key cycle (from Davis and Price).

Cycle i	Left register L_i	Right register R_i	Left register L_i	Right register R_i	d_H
1	0 0 0 0 0 0 0 0	0 0 0 0 0 0 0 0	0 0 0 0 0 0 8 0	0 0 0 0 0 0 0 0	1
2	0 0 0 0 0 0 0 0	A F 0 D 6 8 F D	0 0 0 0 0 0 0 0	A F 0 D 6 8 7 D	1
3	A F 0 D 6 8 F D	C E 0 A 3 6 E A	A F 0 D 6 8 7 D	C E 3 A 3 2 E 2	5
4	C E 0 A 3 6 E A	0 B D C C 5 F E	C E 3 A 3 2 E 2	8 1 B D E D 5 F	15
5	0 B D C C 5 F E	5 D 1 8 1 C C 3	8 1 B D E D 5 F	F 8 C A 3 9 B 2	26
6	5 D 1 8 1 C C 3	1 7 4 4 B 9 7 8	F 8 C A 3 9 B 2	A 9 9 4 B 9 1 8	26
7	1 7 4 4 B 9 7 8	9 B 1 C B 0 D 8	A 9 9 4 B 9 1 8	3 E 9 D 0 5 D 8	22
8	9 B 1 C B 0 D 8	7 A E 8 C 7 E 0	3 E 9 D 0 5 D 8	2 3 F 4 8 D F F	26
9	7 A E 8 C 7 E 0	A 2 A C 7 B 3 F	2 3 F 4 8 D F F	9 D 5 8 D C F B	34
10	A 2 A C 7 B 3 F	5 8 0 E 5 1 E F	9 D 5 8 D C F B	3 F 0 7 6 3 0 3	34
11	5 8 0 E 5 1 E F	2 8 6 5 D B D 4	3 F 0 7 6 3 0 3	9 7 A E 4 A E 3	35
12	2 8 6 5 D B D 4	B F 6 8 F 7 7 C	9 7 A E 4 A E 3	6 8 A B B 6 1 2	37
13	B F 6 8 F 7 7 C	8 F 5 7 F 6 2 9	6 8 A B B 6 1 2	0 9 D 9 C 3 9 8	32
14	8 F 5 7 F 6 2 9	0 8 2 7 B 2 4 0	0 9 D 9 C 3 9 8	4 4 B 0 4 3 5 C	31
15	0 8 2 7 B 2 4 0	F 2 D E B F A C	4 4 B 0 4 3 5 C	3 1 9 F D 4 B 8	29
16	F 2 D E B F A C	1 6 C D 0 E B 8	3 1 9 F D 4 B 8	4 2 6 8 4 F F 9	24

After the DES encryption transformation is completed on both plaintext messages, the resulting ciphertexts are equal to:

$$C_1 = \text{“25DDAC3E96176467}''$$

and

$$C_2 = \text{“1BDD183F1626FB43}''$$

and the Hamming distance is:

$$d_H(C_1, C_2) = 22$$

Here, changing only plaintext bit did resulted in changing 22 of the ciphertext bits.

Table 7 illustrates the effect of diffusion on a set of 16 plaintext messages, which differs only by one bit, with the same key K:

$$K = \text{“0123456789ABCDEF}''$$

The average Hamming distance over the resulting ciphertexts is:

$$d_{H(\text{average})} = \frac{1}{16} \sum_{i=2}^{17} d_H(C_1, C_i) = 31.06$$

Table 7: Hamming distance $d_H(C_1, C_i)$ between a set of pair of ciphertexts when the plaintexts differ by a single bit, i.e. $d_H(M_1, M_i) = 1$, for the same key K (from Davis and Price).

Index	Plaintext M_i	Ciphertext C_i	$d_H(C_1, C_i)$
1	A B C D E F A B C D E F A B C D	C D E 8 7 2 D 4 A 4 7 1 3 4 6 F	
2	8 B C D E F A B C D E F A B C D	C D 3 D 0 A A 4 C 4 0 2 4 B 4 A	29
3	A 9 C D E F A B C D E F A B C D	8 0 1 F 8 A 2 9 6 8 B C 4 4 7 3	38
4	A B D D E F A B C D E F A B C D	5 D 9 8 C 4 7 D D D B A 6 F 3 0	36
5	A B C F E F A B C D E F A B C D	9 9 8 9 5 6 2 A 8 4 F 4 0 1 C 9	26
6	A B C D 6 F A B C D E F A B C D	6 7 C 2 6 9 F 2 5 4 2 7 9 1 F 9	30
7	A B C D E B A B C D E F A B C D	F 8 C 9 8 F 7 9 A D C 0 6 E A 4	33
8	A B C D E F D B C D E F A B C D	8 7 D 3 2 4 0 A B B F 4 4 0 7 4	34
9	A B C D E F A 9 C D E F A B C D	D B 9 9 8 B 6 7 0 4 6 C D C E 7	30
10	A B C D E F A B E C E F A B C D	2 F 6 E 5 4 7 0 E 4 E 3 5 1 A C	25
11	A B C D E F A B C C E F A B C D	B 5 3 E 4 2 D E 3 0 F 9 7 A D 0	29
12	A B C D E F A B C D 6 F A B C D	4 F 4 0 6 7 7 2 6 B 3 5 B 0 1 4	28
13	A B C D E F A B C D E 7 A B C D	A B 1 5 5 2 8 9 6 6 0 C 6 0 B 2	35
14	A B C D E F A B C D E F 2 B C D	5 B D A 9 3 F 7 D 4 2 7 B 8 D 2	30
15	A B C D E F A B C D E F A F C D	9 8 5 3 C 5 1 1 E D 5 6 8 8 7 E	34
16	A B C D E F A B C D E F A B D D	7 0 A A 2 4 0 7 9 5 9 F 0 4 B 1	34
17	A B C D E F A B C D E F A B C 5	8 9 2 B E C 4 7 C 9 7 1 2 B E 3	26

Table 8 shows the diffusion property, but this time for 16 different keys, which differing by one bit from each other, over the same plaintext message M where

$$M = \text{“ABCDEFABCDEFABCD}''$$

The average Hamming distance is equal to:

$$d_{H(\text{average})} = \frac{1}{16} \sum_{i=2}^{17} d_H(C_1, C_i) = 32.88$$

Table 8: Hamming distance $d_H(C_1, C_2)$ between a set of pair of ciphertexts when the keys differ by a single bit ($d_H(K_1, K_i) = 1$) for the same plaintext M (from Davis and Price).

Index	Key K_i	Ciphertext C_i	$d_H(C_1, C_i)$
1	0 1 2 3 4 5 6 7 8 9 A B C D E F	C D E 8 7 2 D 4 A 4 7 1 3 4 6 F	
2	8 1 2 3 4 5 6 7 8 9 A B C D E F	1 B 7 3 F E 8 B C 0 B 8 8 6 0 6	35
3	0 2 2 3 4 5 6 7 8 9 A B C D E F	0 F 9 2 F 6 0 D 2 F D 4 D 8 B 7	32
4	0 1 6 2 4 5 6 7 8 9 A B C D E F	3 1 A F D 8 C 5 4 F B F 4 B C D	37
5	0 1 2 6 4 5 6 7 8 9 A B C D E F	C 7 9 F 5 9 6 3 D 4 6 5 A 7 E E	29
6	0 1 2 3 0 5 6 7 8 9 A B C D E F	3 6 5 2 9 C C 1 0 7 1 7 A 3 8 9	39
7	0 1 2 3 4 6 6 7 8 9 A B C D E F	7 F 3 5 F 7 E 6 C E C 5 7 E E 3	30
8	0 1 2 3 4 5 4 6 8 9 A B C D E F	C 9 F 3 F D 9 2 6 0 C 6 8 1 8 A	27
9	0 1 2 3 4 5 6 4 8 9 A B C D E F	E 6 9 2 8 3 2 2 E E 8 B 9 A 6 9	36
10	0 1 2 3 4 5 6 7 A 8 A B C D E F	0 9 9 7 A 5 A F 6 E 4 E 1 4 6 0	37
11	0 1 2 3 4 5 6 7 8 A A B C D E F	C 7 B 4 1 C 4 F 3 8 D 9 A F 7 A	31
12	0 1 2 3 4 5 6 7 8 9 E A C D E F	4 B 4 A A 2 0 A B 2 1 4 4 D D D	30
13	0 1 2 3 4 5 6 7 8 9 A 8 C D E F	1 2 9 9 7 E E 8 0 0 1 B D 2 7 C	32
14	0 1 2 3 4 5 6 7 8 9 A B 4 C E F	4 2 D 1 7 B 7 D F 5 3 4 3 B 7 9	28
15	0 1 2 3 4 5 6 7 8 9 A B C E E F	8 7 F 6 4 9 3 C 4 C 8 9 8 3 2 7	33
16	0 1 2 3 4 5 6 7 8 9 A B C D 6 F	F C 3 0 F 6 F 7 6 B D 4 2 5 9 2	31
17	0 1 2 3 4 5 6 7 8 9 A B C D E C	1 C B D E C 4 B 7 9 B C A 7 A 1	39

3 DES Weaknesses

3.1 Key space size

In DES, the key consists in a 56-bit vector providing a key space $\mathcal{K}$ of $2^{56} = 7.2058 \times 10^{16}$ elements. In an exhaustive search known-plaintext attack, the cryptanalyst will obtain the solution after 2^{55}, or 3.6029×10^{16} trials, on average.

In 1977, Diffie and Hellman[DH77] have shown that a special purpose multiple parallel processor consisting of 10^6 intergrated circuits, each one trying a key every $1\mu\ s$, could determine the key used in about 10 hours on average in a known-plaintext attack. The cost of such a multiple processor machine would have been around $50,000,000 in 1977 [Pfl89]. If such a machine was used 365 days a year, 24 hours a day, amortizing the price over the number of key solutions obtained, then the price per solution would have been about $20,000 per solution.

Diffie and Hellman argued that if the key length was increased from 56 to 64 bits, it would make the DES algorithm secure even for "*intelligence agencies budgets...*" [Sim92], while decreasing the key length from 56 to only 48 would make DES "*vulnerable to attack by almost any reasonable sized organization*" [Sim92]. The key length is thus a very critical parameter to the security of DES.

3.2 Complement property

Another possible weakness of DES lies in the complement property of the DES algorithm. Let M be a 64-bit plaintext message to be encrypted into a 64-bit ciphertext C using the 56-bit key K:

$$C = DES_K(M)$$

The complement property of DES [Pfl89] indicates that the bit-by-bit modulo-2 complement of the ciphertext C, i.e. $\bar{C}$, can be obtained from the plaintext M and key K as:

$$\begin{aligned} \bar{C} &= DES_{\bar{K}}(\bar{M}) \\ \bar{C} &= \overline{DES_K(M)} \end{aligned}$$

Since complementing the ciphertext vector $\bar{C}$ takes much less time than actually performing the DES encryption transformation, the exhaustive key search attack can be reduced almost by half.

3.3 DES weak keys

The DES algorithm generates from the 56-bit key K a set, or sequence, of 16 distinct 48-bit sub-keys which are then used in each round of substitution and permutation transformation of DES. However, if the left and right registers C_i and D_i of the sub-key schedule calculation branch are filled with "0" or "1", the sub-keys will be identical:

$$k_1 = k_2 = \ldots = k_{16}$$

The encryption and decryption processes being the same except for the order of sub-keys, when such *weak keys* are employed, enciphering a plaintext messages M twice will result in the original plaintext message [DP84]:

$$DES_K[DES_K(M)] = M$$

The weak keys of the DES are listed hereafter:

$$\begin{array}{lcl}
K_1 & = & 0\ 1 \quad 0\ 1 \quad 0\ 1 \quad 0\ 1 \quad 0\ 1 \quad 0\ 1 \quad 0\ 1 \quad 0\ 1 \\
K_2 & = & \mathrm{F\ E} \quad \mathrm{F\ E} \quad \mathrm{F\ E} \quad \mathrm{F\ E} \quad \mathrm{F\ E} \quad \mathrm{F\ E} \quad \mathrm{F\ E} \quad \mathrm{F\ E} \\
K_3 & = & 1\ \mathrm{F} \quad 1\ \mathrm{F} \quad 1\ \mathrm{F} \quad 1\ \mathrm{F} \quad 0\ \mathrm{E} \quad 0\ \mathrm{E} \quad 0\ \mathrm{E} \quad 0\ \mathrm{E} \\
K_4 & = & \mathrm{E}\ 0 \quad \mathrm{E}\ 0 \quad \mathrm{E}\ 0 \quad \mathrm{E}\ 0 \quad \mathrm{F}\ 1 \quad \mathrm{F}\ 1 \quad \mathrm{F}\ 1 \quad \mathrm{F}\ 1
\end{array}$$

3.4 DES semi-weak key pairs

Another property observed in the DES algorithm is the existence of *semi-weak pairs of keys* for which the pattern of alternating *zeroes* and *ones* in the two sub-key registers C_i and D_i. This results in the first key, say K_1, producing the sub-key sequence: k_1, k_2, ..., k_{16}, while the second key of the pair, K_2, generates the inverse sub-key sequence: k_{16}, k_{15}, ..., k_1. Thus the encryption of message M by key K_1 followed by a second encryption with key K_2 will give the original message M:

$$DES_{K_2}[DES_{K_1}(M)] = M$$

The semi-weak keys of the DES are [DP84]:

$$\begin{array}{lcl}
K_{1,1} & = & 0\ 1 \quad \mathrm{F\ E} \quad 0\ 1 \quad \mathrm{F\ E} \quad 0\ 1 \quad \mathrm{F\ E} \quad 0\ 1 \quad \mathrm{F\ E} \\
K_{1,2} & = & \mathrm{F\ E} \quad 0\ 1 \quad \mathrm{F\ E} \quad 0\ 1 \quad \mathrm{F\ E} \quad 0\ 1 \quad \mathrm{F\ E} \quad 0\ 1 \\
K_{2,1} & = & 1\ \mathrm{F} \quad \mathrm{E}\ 0 \quad 1\ \mathrm{F} \quad \mathrm{E}\ 0 \quad 0\ \mathrm{E} \quad \mathrm{F}\ 1 \quad 0\ \mathrm{E} \quad \mathrm{F}\ 1 \\
K_{2,2} & = & \mathrm{E}\ 0 \quad 1\ \mathrm{F} \quad \mathrm{E}\ 0 \quad 1\ \mathrm{F} \quad \mathrm{F}\ 1 \quad 0\ \mathrm{E} \quad \mathrm{F}\ 1 \quad 0\ \mathrm{E} \\
K_{3,1} & = & 0\ 1 \quad \mathrm{E}\ 0 \quad 0\ 1 \quad \mathrm{E}\ 0 \quad 0\ 1 \quad \mathrm{F}\ 1 \quad 0\ 1 \quad \mathrm{F}\ 1 \\
K_{3,2} & = & \mathrm{E}\ 0 \quad 0\ 1 \quad \mathrm{E}\ 0 \quad 0\ 1 \quad \mathrm{F}\ 1 \quad 0\ 1 \quad \mathrm{F}\ 1 \quad 0\ 1 \\
K_{4,1} & = & 1\ \mathrm{F} \quad \mathrm{F\ E} \quad 1\ \mathrm{F} \quad \mathrm{F\ E} \quad 0\ \mathrm{E} \quad \mathrm{F\ E} \quad 0\ \mathrm{E} \quad \mathrm{F\ E} \\
K_{4,2} & = & \mathrm{F\ E} \quad 1\ \mathrm{F} \quad \mathrm{F\ E} \quad 1\ \mathrm{F} \quad \mathrm{F\ E} \quad 0\ \mathrm{E} \quad \mathrm{F\ E} \quad 0\ \mathrm{E} \\
K_{5,1} & = & 0\ 1 \quad 1\ \mathrm{F} \quad 0\ 1 \quad 1\ \mathrm{F} \quad 0\ 1 \quad 0\ \mathrm{E} \quad 0\ 1 \quad 0\ \mathrm{E} \\
K_{5,2} & = & 1\ \mathrm{F} \quad 0\ 1 \quad 1\ \mathrm{F} \quad 0\ 1 \quad 0\ \mathrm{E} \quad 0\ 1 \quad 0\ \mathrm{E} \quad 0\ 1 \\
K_{6,1} & = & \mathrm{E}\ 0 \quad \mathrm{F\ E} \quad \mathrm{E}\ 0 \quad \mathrm{F\ E} \quad \mathrm{F}\ 1 \quad \mathrm{F\ E} \quad \mathrm{F}\ 1 \quad \mathrm{F\ E} \\
K_{6,2} & = & \mathrm{F\ E} \quad \mathrm{E}\ 0 \quad \mathrm{F\ E} \quad \mathrm{E}\ 0 \quad \mathrm{F\ E} \quad \mathrm{F}\ 1 \quad \mathrm{F\ E} \quad \mathrm{F}\ 1
\end{array}$$

4 Differential and linear cryptanalysis

Traditional cryptanalysis of block ciphers such as the Data Encryption Standard rely on such known plaintext methods as doing exhaustive search over the whole key space. While this type of brute force cryptanalytic attack may seem practical on conventional single DES encryption, it becomes impractical to perform on double DES and triple DES enciphering implementations. More sophisticated cryptanalysis methods have been proposed in the recent years to reduce the computational complexity of a brute force attack. Two such methods are differential cryptanalysis and linear cryptanalysis cryptanalysis. Differential cryptanalysis is briefly described in section 4.1 and linear cryptanalysis cryptanalysis in section 4.2.

4.1 Differential cryptanalysis

Differential cryptanalysis has been proposed since 1990 to break block ciphers such as DES and its predecessor LUCIFER. While successful for breaking LUCIFER, differential cryptanalysis is still, at least for the time being, of "academic" interest for breaking the 16-round full-fledged DES. The reason why DES is resistant against differential cryptanalysis is that while differential cryptanalysis has been known to the general public for less than ten years, its techniques were known to the DES developers in the seventies. Neverthless differential cryptanalysis, as linear cryptanalysis, is one of the most promising cryptanalysis methods.

Differential cryptanalysis involves the analysis of the distribution of the difference (modulo-2 bit per bit) between two plaintexts X_1 and X_2 and the two ciphertexts Y_1 and Y_2 resulting from their encryption. Here the plaintexts X_1 and X_2 are in fact the 32-bit contents of the right register prior the extension permutation $E(X)$ in a DES round. The two ciphertexts Y_1 and Y_2 are the 32-bit output from the standard permutation $P(C)$ after the substitution boxes.

Figure 4 shows a single round of DES encryption. Let ΔX represent the difference of the two known (and chosen) plaintexts X_1 and X_2:

$$\Delta X = X_1 \oplus X_2$$

where $X_1 \oplus X_2$ represents the addition modulo-2 bit by bit of the 2 plaintext vectors. In a chosen plaintext attack, the two plaintexts X_1 and X_2 are chosen such as to give a desired plaintext difference ΔX.

Since $\Delta X = X_1 \oplus X_2$ and $A = E(X)$ is simply an expansion permutation of the plaintext bits A, then the difference ΔA is also known:

$$\begin{aligned} \Delta A &= A_1 \oplus A_2 \\ \Delta A &= E(X_1) \oplus E(X_2) \\ \Delta A &= E(\Delta X) \end{aligned}$$

At each DES round, the unknown 48-bit subkey K_i is added to the 48-bit vector A at the output of the expansion permutation box:

$$\begin{aligned} B_1 &= A_1 \oplus K_i \qquad \text{and} \\ B_2 &= A_2 \oplus K_i \end{aligned}$$

Since the 48-bit subkey K_i is secret, the two 48-bit vectors B_1 and B_2 are also unknown. However, their difference ΔB is known!

$$\begin{aligned}
\Delta B &= B_1 \oplus B_2 \\
\Delta B &= (A_1 \oplus K_i) \oplus (A_2 \oplus K_i) \\
\Delta B &= A_1 \oplus A_2 \\
\Delta B &= \Delta A \\
\Delta B &= E(\Delta X)
\end{aligned}$$

So, by chosing the plaintexts X_1 and X_2 and therefore their difference ΔX, one finds the inputs to the 8 substitution boxes even if the subkeys are unknown.

Now working backward from known ciphertexts Y_1 and Y_2 obtained from the encryption of the above plaintexts X_1 and X_2, we can also determine their difference ΔY [1]:

$$\Delta Y = Y_1 \oplus Y_2$$

Both Y_1 and Y_2 vectors are permuted versions of the 32-bit outputs C_1 and C_2 of the substitution boxes:

$$\begin{aligned}
Y_1 &= P(C_1) \qquad \text{and} \\
Y_2 &= P(C_2)
\end{aligned}$$

or, expressing the substitution boxes outputs C_1 and C_2 as a function of the ciphertexts Y_1 and Y_2:

$$\begin{aligned}
C_1 &= P^{-1}(Y_1) \qquad \text{and} \\
C_2 &= P^{-1}(Y_2)
\end{aligned}$$

Finally, the difference at the output of the substitution boxes ΔC is:

$$\begin{aligned}
\Delta C &= C_1 \oplus C_2 \\
\Delta C &= (P^{-1}(Y_1)) \oplus (P^{-1}(Y_2)) \\
\Delta C &= P^{-1}(\Delta Y)
\end{aligned}$$

Differential cryptanalysis compares the distribution of the difference ΔX for a plaintext pair X_1 and X_2 with the distribution of the ciphertext difference ΔY for the corresponding ciphertext pair Y_1 and Y_2. In a chosen plaintext-ciphertext attack, the plaintext is chosen such as to provide the desired difference ΔX. It exploit the fact that the plaintext differences ΔX and the ciphertext differences ΔY are not equally likely. Some differences in plaintext pairs have a higher probability of causing difference in ciphertext pairs than others.

[1] As we know the actual ciphertexts are obtained by adding Y to the previous (and known) contents of the left register.

For each of the 8 DES substitution boxes, we can construct a table of joint plaintext and ciphertext differences (see Table 9 below) where each row represents a given plaintext difference ΔX and each column represent a given ciphertext difference ΔY. The entry $p_{i,j}$ in Table 9 represents the number of occurences that a given plaintext difference ΔX_i has produced a given ciphertext difference ΔY_j.

Table 9: Plaintext and ciphertext differences relative frequencies.

	ΔY_1	$\cdots$	ΔY_j	$\cdots$
ΔX_1	$p_{1,1}$	$\cdots$	$p_{1,j}$	$\cdots$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$
ΔX_i	$p_{i,1}$	$\cdots$	$p_{i,j}$	$\cdots$
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$

Biham and Shamir [BS93] have demonstrated that a full-fledged 16-round DES cryptanalysis requires 2^{47} chosen plaintext-ciphertext pairs or 2^{55} known plaintext-ciphertext pairs with 2^{37} DES operations, thus making this type attack on DES not practical yet.

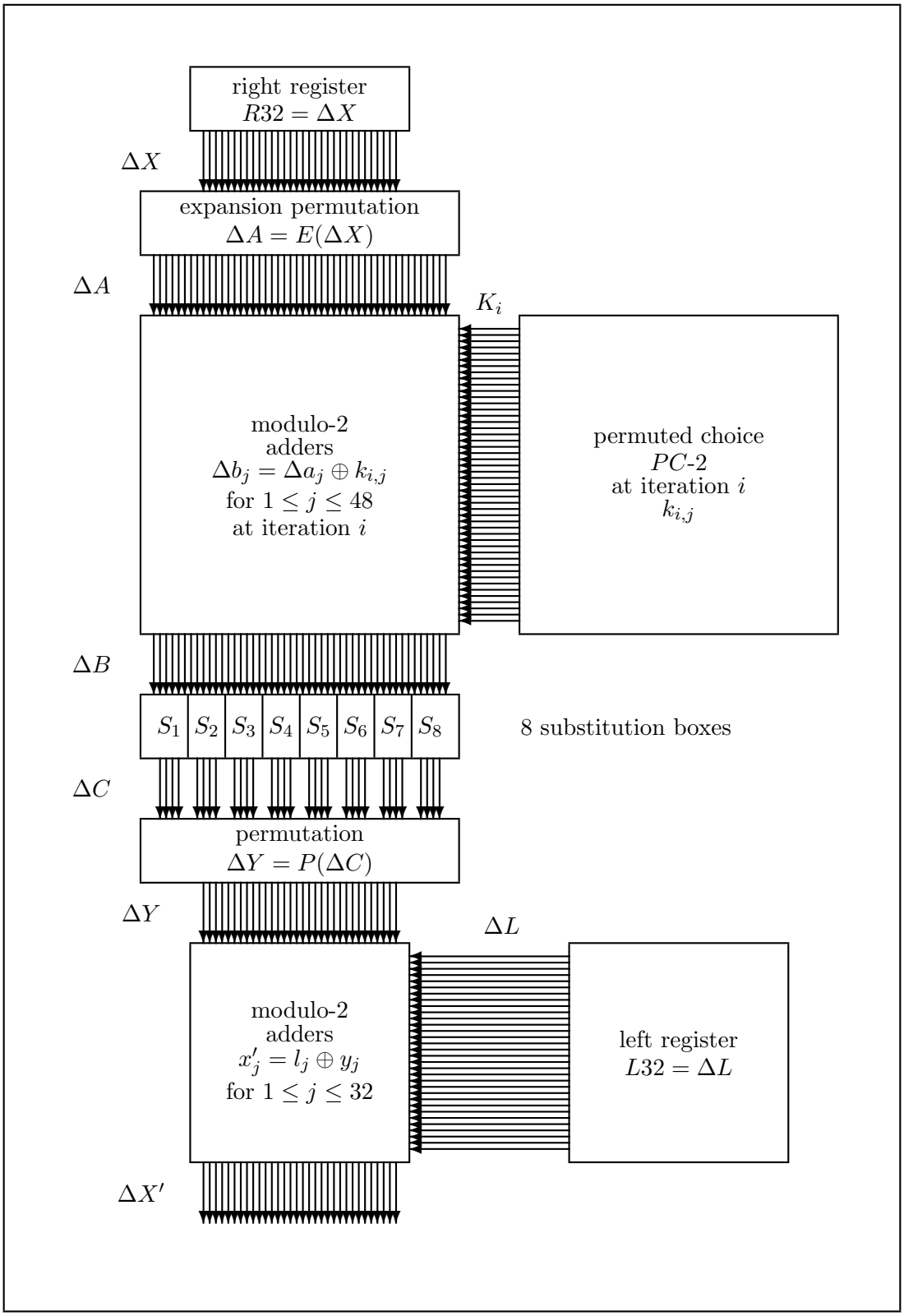


Figure 4: Differential cryptanalysis of a single DES encryption round.

4.2 Linear cryptanalysis

A method to break a block cipher such as DES is linear cryptanalysis. It basically consists in trying to represent (or approximate) a round of DES encryption with a linear transformation. Figure 5 illustrates how linear cryptanalysis can be applied on a single round of DES encryption.

In the known plaintext attack both the plaintext M and the corresponding ciphertext C are known. Since the output $IP(M)$ after the initial permutation function IP is known, one knows also the contents of the left and right registers.

Let $X = x_1, x_2, \ldots, x_{32}$ be the 32 bits contents of the right register. These 32 bits go through an expansion permutation $A = E(X)$: the resulting 48-bit vector $A = a_1, a_2, \ldots, a_{48}$ is added modulo-2 bit by bit with the 48-bit subkey $K_i = k_{i,1}, k_{i,2}, \ldots, k_{i,48}$ at the i^{th} iteration from the permuted choice transformation $PC2$.

The 48-bit vector $B = b_1, b_2, \ldots, b_{48}$ are then passed through the 8 DES substitution boxes $\{S_k\}_{k=1,\ldots,8}$ where each 6-bit input vector $(b_1, b_2, b_3, b_4, b_5, b_6)$ is replaced, or substituted, with a 4-bit output vector (c_1, c_2, c_3, c_4). The 32-bit vector $C = c_1, c_2, \ldots, c_{32}$ is transformed through a standard permutation P and the 32-bit vector $Y = y_1, y_2, \ldots, y_{32}$ is then added to the contents of the left register. The right register is updated with the resulting 32-bit vector.

$$\begin{aligned} Y &= P(C) \qquad \text{and} \\ C &= P^{-1}(Y) \end{aligned}$$

As can be seen in figure 5, if one knows the input (plaintext after initial permutation) X then the output:

$$A = E(X)$$

of the expansion permutation is also known. However, because the subkey $K_i = k_{i,j}$, for $j = 1, \ldots, 48$ at iteration i (we can begin with $i = 1$), is secret one cannot determine the sum at the output of the modulo-2 adders.

$$b_j = a_j \oplus k_{i,j} \qquad \text{for } 1 \leq j \leq 48$$

The bits at the output of the adders ($\{b_j\}_{j=1,\ldots,48}$) constitute the 48 input bits for the 8 substitution boxes $\{S_k\}$.

Now, working backward from the contents of the left register L and the previous contents of the right register X' (in fact the temporary register $TEMP32$ from a previous iteration of DES), one can determine the 32 bits vector Y. Since Y is the result of a standard permutation of the output of the substitution boxes:

$$C = P^{-1}(Y)$$

the 32 bits $c_1, c_2, \ldots, c_{32}$ at the output substitution boxes are also determined.

The substitution boxes $\{S_k\}_{k=1,\ldots,8}$ should be random and unbiased. For any 6-bit input b_1, b_2, b_3, b_4, b_5, and b_6, there should be a uniform distribution of output bits. Now, by the construction of the table of all possible 64 input vectors of a substitution box, each input bit $b_i = 0$ half of the time, and $b_i = 1$ the other half. In other words, we can say that each of the 6 input bits equals 0 with a probability $p = \frac{1}{2}$ and each of the 4 output bits is equal to 0 with $p = \frac{1}{2}$.

However, one can infer the actual input to a substitution box if it is possible to exploit the relationship between its inputs and outputs. For instance, if we observe the 4 bits c_1, c_2, c_3, and c_4

at the output of a substitution box S_k, and add them together modulo-2, then for the 64 different input vectors b_1, b_2, b_3, b_4, b_5, and b_6, the result will be $c_1 \oplus c_2 \oplus c_3 \oplus c_4 = 0$ half of the time (32 times) and $c_1 \oplus c_2 \oplus c_3 \oplus c_4 = 1$ the remaining 32 times (each of the 16 output values, or vectors, appearing 4 times in a given substitution table.

It has been observed that the input-output relationship of the substitution boxes is not always unbiased. For instance, DES substitution box S_5 is the most biased of the substitution boxes and this can be exploited to deduct the key.

Table 10 shows the relationships between the 6-bit input b_{25}, b_{26}, b_{27}, b_{28}, b_{29}, and b_{30} and the 4-bit output c_{17}, c_{18}, c_{19}, and c_{20} inside substitution box S_5. From Table 10, one can observe that even if input bit $b_{26} = 0$ half of the time, i.e., with a probability $p = \frac{1}{2}$, and that the modulo-2 addition $c_1 \oplus c_2 \oplus c_3 \oplus c_4 = 0$ also with $p = \frac{1}{2}$, the following equality:

$$b_{26} = c_1 \oplus c_2 \oplus c_3 \oplus c_4$$

is true only 12 times out of 64, or assuming equiprobable inputs, with a probability $p = \frac{12}{64} = \frac{3}{16}$. These 12 occurrences are indicated in the last column of Table 10. The observation that the probability of occurence of $b_{26} = c_1 \oplus c_2 \oplus c_3 \oplus c_4$ is $p = \frac{3}{16}$ instead of the expected probability of $p = \frac{1}{2}$ is used to help breaking DES. Then with a probability $p = \frac{3}{16}$:

$$\begin{aligned} b_{26} &= c_1 \oplus c_2 \oplus c_3 \oplus c_4 \\ a_{26} \oplus k_{i,26} &= c_1 \oplus c_2 \oplus c_3 \oplus c_4 \end{aligned}$$

But since $A = E(X)$ then $a_{26} = x_{17}$ (from the expansion function). Similarly, knowing the mapping of the standard permutation function $Y = P(C)$ we can replace c_{17}, c_{18}, c_{19}, and c_{20} by the known ciphertext (one round) values y_3, y_8, y_{14}, and y_{25}. Therefore, with a probability $p = \frac{3}{16}$,

$$\begin{aligned} k_{i,26} &= a_{26} \oplus c_1 \oplus c_2 \oplus c_3 \oplus c_4 \\ k_{i,26} &= a_{26} \oplus c_{17} \oplus c_{18} \oplus c_{19} \oplus c_{20} \\ k_{i,26} &= x_{17} \oplus y_3 \oplus y_8 \oplus y_{14} \oplus y_{25} \end{aligned}$$

since the one-round plaintext X and ciphertext Y pair are known this provides a clue that the subkey bit $k_{i,26}$ is the complement of the plaintext and ciphertext bits function $x_{17} \oplus y_3 \oplus y_8 \oplus y_{14} \oplus y_{25}$.

This one-round analysis has to be generalized to the 16 rounds of DES. This is possible, because the contents of the right register at the second iteration is a function of the results of the first iteration.

Linear cryptanalysis of DES is still not practical since it requires 2^{47} known plaintext-ciphertext pairs for solving a single key bit (out of the 56). A second key bit can be obtained by reversing the plaintext and ciphertext.

It has been shown [Sch96] that using a linear approximation of a 14 round DES and estimating (guessing) the 6 subkeys bits: $k_{i,25}$, $k_{i,26}$, $k_{i,27}$, $k_{i,28}$, $k_{i,29}$, $k_{i,30}$, corresponding to the 6 input bits of substitution box S_5, for rounds 2 and 14. This is equivalent to performing 2^{12} linear cryptanalysis in parallel but does provides a total of 26 key bits! This reduces the key space search from 2^{56} in an exhaustive search to a very small key space of only $2^{30} = 1,073,741,824$.

Table 10: Input output relationships of substitution box S_5 (beginning).

6-bit input b_1 b_2 b_3 b_4 b_5 b_6 / b_{25} b_{26} b_{27} b_{28} b_{29} b_{30}	output	4-bit output c_1 c_2 c_3 c_4 / c_{17} c_{18} c_{19} c_{20}	test
0 0 0 0 0 0	2	0 0 1 0	
0 0 0 0 0 1	14	1 1 1 0	
0 0 0 0 1 0	12	1 1 0 0	$b_{26} = c_1 \oplus c_2 \oplus c_3 \oplus c_4$
0 0 0 0 1 1	11	1 0 1 1	
0 0 0 1 0 0	4	0 1 0 0	
0 0 0 1 0 1	2	0 0 1 0	
0 0 0 1 1 0	1	0 0 0 1	
0 0 0 1 1 1	12	1 1 0 0	$b_{26} = c_1 \oplus c_2 \oplus c_3 \oplus c_4$
0 0 1 0 0 0	7	0 1 1 1	
0 0 1 0 0 1	4	0 1 0 0	
0 0 1 0 1 0	10	1 0 1 0	$b_{26} = c_1 \oplus c_2 \oplus c_3 \oplus c_4$
0 0 1 0 1 1	7	0 1 1 1	
0 0 1 1 0 0	11	1 0 1 1	
0 0 1 1 0 1	13	1 1 0 1	
0 0 1 1 1 0	6	0 1 1 0	$b_{26} = c_1 \oplus c_2 \oplus c_3 \oplus c_4$
0 0 1 1 1 1	1	0 0 0 1	
0 1 0 0 0 0	8	1 0 0 0	$b_{26} = c_1 \oplus c_2 \oplus c_3 \oplus c_4$
0 1 0 0 0 1	5	0 1 0 1	
0 1 0 0 1 0	5	0 1 0 1	
0 1 0 0 1 1	0	0 0 0 0	
0 1 0 1 0 0	3	0 0 1 1	
0 1 0 1 0 1	15	1 1 1 1	
0 1 0 1 1 0	15	1 1 1 1	
0 1 0 1 1 1	10	1 0 1 0	
0 1 1 0 0 0	13	1 1 0 1	$b_{26} = c_1 \oplus c_2 \oplus c_3 \oplus c_4$
0 1 1 0 0 1	3	0 0 1 1	
0 1 1 0 1 0	0	0 0 0 0	
0 1 1 0 1 1	9	1 0 0 1	
0 1 1 1 0 0	14	1 1 1 0	$b_{26} = c_1 \oplus c_2 \oplus c_3 \oplus c_4$
0 1 1 1 0 1	8	1 0 0 0	$b_{26} = c_1 \oplus c_2 \oplus c_3 \oplus c_4$
0 1 1 1 1 0	9	1 0 0 1	
0 1 1 1 1 1	6	0 1 1 0	

Table 11: Input output relationships of substitution box S_5 (end).

6-bit input b_1 b_2 b_3 b_4 b_5 b_6 / b_{25} b_{26} b_{27} b_{28} b_{29} b_{30}	output	4-bit output c_1 c_2 c_3 c_4 / c_{17} c_{18} c_{19} c_{20}	test
1 0 0 0 0 0	4	0 1 0 0	
1 0 0 0 0 1	11	1 0 1 1	
1 0 0 0 1 0	2	0 0 1 0	
1 0 0 0 1 1	8	1 0 0 0	
1 0 0 1 0 0	1	0 0 0 1	
1 0 0 1 0 1	12	1 1 0 0	$b_{26} = c_1 \oplus c_2 \oplus c_3 \oplus c_4$
1 0 0 1 1 0	11	1 0 1 1	
1 0 0 1 1 1	7	0 1 1 1	
1 0 1 0 0 0	10	1 0 1 0	$b_{26} = c_1 \oplus c_2 \oplus c_3 \oplus c_4$
1 0 1 0 0 1	1	0 0 0 1	
1 0 1 0 1 0	13	1 1 0 1	
1 0 1 0 1 1	14	1 1 1 0	
1 0 1 1 0 0	7	0 1 1 1	
1 0 1 1 0 1	2	0 0 1 0	
1 0 1 1 1 0	8	1 0 0 0	
1 0 1 1 1 1	13	1 1 0 1	
1 1 0 0 0 0	15	1 1 1 1	
1 1 0 0 0 1	6	0 1 1 0	
1 1 0 0 1 0	9	1 0 0 1	
1 1 0 0 1 1	15	1 1 1 1	
1 1 0 1 0 0	12	1 1 0 0	
1 1 0 1 0 1	0	0 0 0 0	
1 1 0 1 1 0	5	0 1 0 1	
1 1 0 1 1 1	9	1 0 0 1	
1 1 1 0 0 0	6	0 1 1 0	
1 1 1 0 0 1	10	1 0 1 0	
1 1 1 0 1 0	3	0 0 1 1	
1 1 1 0 1 1	4	0 1 0 0	$b_{26} = c_1 \oplus c_2 \oplus c_3 \oplus c_4$
1 1 1 1 0 0	0	0 0 0 0	
1 1 1 1 0 1	5	0 1 0 1	
1 1 1 1 1 0	14	1 1 1 0	$b_{26} = c_1 \oplus c_2 \oplus c_3 \oplus c_4$
1 1 1 1 1 1	3	0 0 1 1	

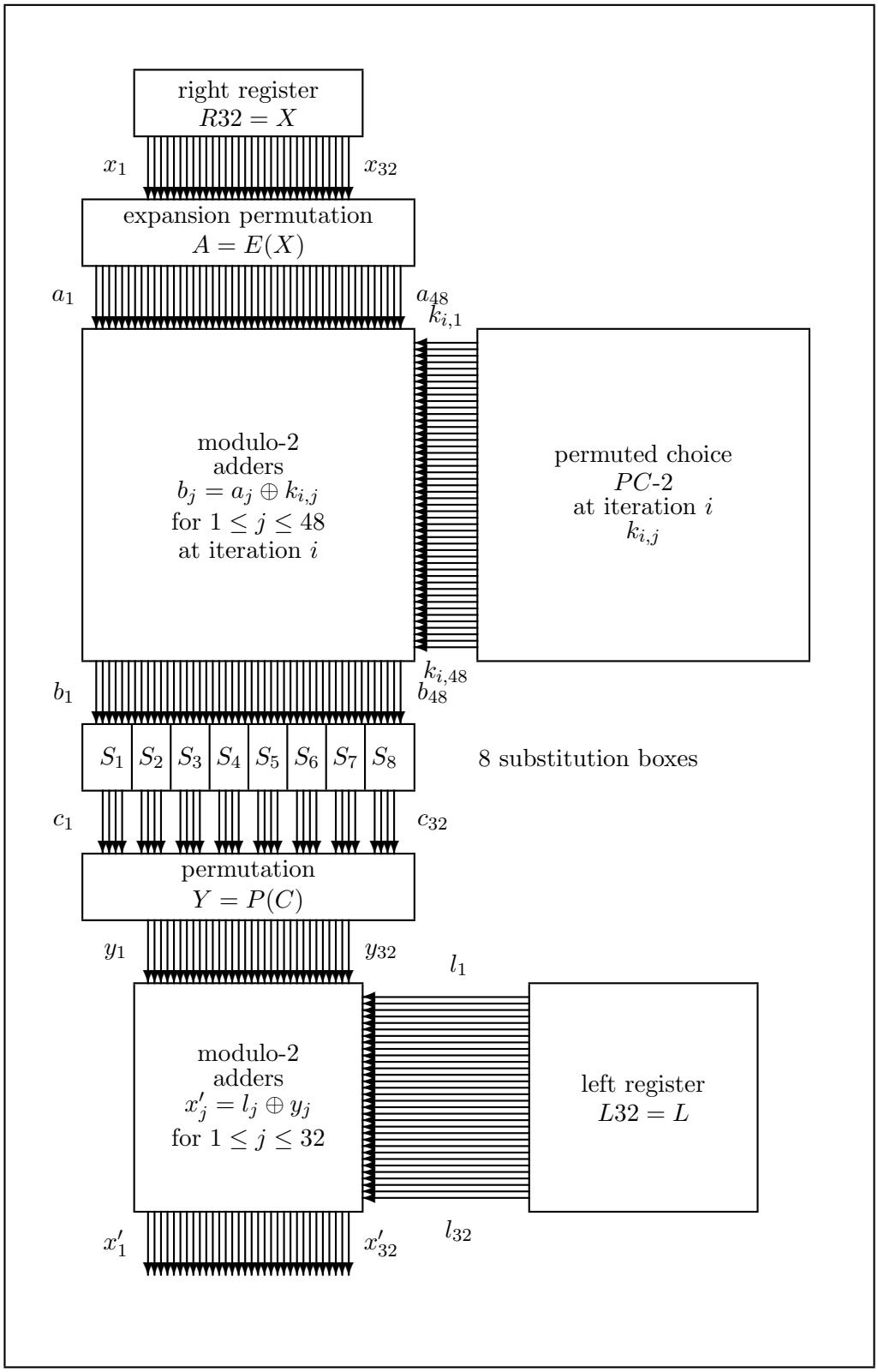


Figure 5: Linear approximation of a DES encryption round.

5 Modes of operation of the Data Encryption Standard (DES)

The Data Encryption Standard (DES) can be implemented into four different modes of operation [**?**]: two block cipher modes whereas the plaintext data bits are enciphered as 64-bit data blocks, as well as two stream cipher modes where the plaintext data bits are encrypted individually.

- Block cipher modes:
  - Electronic Codebook Mode (ECB)
  - Cipher Block Chaining Mode (CBC)
- Stream cipher modes:
  - Output Feedback Mode (OFB)
  - Cipher Feedback Mode (CFB)

Note that the two stream cipher modes of operation of DES can be used to encrypted plaintext data sub-blocks of s bits (i.e. $1 \leq s \leq 64$) instead of individual bits, such as sub-blocks of 8 bits ($s = 8$), for instance.

5.1 Electronic Codebook Mode (ECB):

The simplest method of implementing the Data Encryption Standard is the Electronic Codebook Mode. For the Electronic Codebook Mode, or in short ECB mode, the plaintext message stream M is broken into blocks of 64 bits and then encrypted using DES algorithm with key K:

$$\text{message stream:} \quad M = \underbrace{m_1, \ldots, m_{64}}_{\text{block } M_1}, \underbrace{m_{65}, \ldots, m_{128}}_{\text{block } M_2}, m_{129}, \ldots$$

Each 64-bit plaintext block M_i will result into a 64-bit ciphertext block C_i:

$$C_i = DES_K(M_i)$$

and the decryption is done by applying the inverse transformation $DES_K^{-1}(\bullet)$with the same encryption key K:

$$M_i = DES_K^{-1}(C_i) = DES_K^{-1}[DES_K(M_i)]$$

However, there is a weakness with the Electronic Codebook Mode: for the same encryption key K, 2 identical "slices" of plaintext, that is $M_i = M_j$ will result in two identical ciphertext "slices" $C_i = C_j$.

5.2 Cipher Block Chaining Mode (CBC):

The Cipher Block Chaining Mode of operation of the Data Encryption Standard can be used to prevent the repetition of such ciphertext "slices". For the Cipher Block Chaining Mode, the 64-bit ciphertext block C_i is a function of both the 64-bit input block M_i as well as the previous 64-bit ciphertext block C_{i-1}. The encryption transformation is given by:

$$C_i = DES_K(M_i \oplus C_{i-1})$$

where C_0 is an arbitrary (and secret) 64-bit initialization vector (sometimes referred to as IV). The decipherment is performed by applying the decryption transformation $DES_K^{-1}(\bullet)$ to the present ciphertext block C_i, and then, by adding (modulo-2) the previous decrypted cipher block C_{i-1}:

$$\begin{aligned} M_i &= DES_K^{-1}(C_i) \oplus C_{i-1} \\ &= DES_K^{-1}[DES_K(M_i \oplus C_{i-1})] \oplus C_{i-1} \\ &= M_i \oplus C_{i-1} \oplus C_{i-1} \\ &= M_i \end{aligned}$$

Using Cipher Block Chaining Mode, a plaintext block M_i will be enciphered differently depending on the previous ciphertext block C_{i-1} which in turn, depend on the previous plaintext block M_{i-1} and the preceding cipher block C_{i-2}, and so on.

For both Electronic Codebook Mode and Cipher Block Chaining Mode, a single channel error in the communication channel will results in many errors after decryption: this is due to the diffusion nature of the Data Encryption Standard algorithm (on the average, a single bit change in the received ciphertext block C_i leads to about 32 bits in error after applying the decryption transformation $DES_K^{-1}(C_i)$.

5.3 Stream Cipher Modes of Operation of DES

The Data Encryption Standard algorithm can also be implemented as stream cipher modes, encrypting plaintext data bits one by one (or more generally s plaintext data bits into s ciphertext data bits). Here we will consider only the case where $s = 1$.

The two stream cipher modes of DES are: Cipher Feedback Mode and Output Feedback Mode. In both cases, the plaintext data bits $m_1, \ldots, m_i, \ldots$ are encrypted individually one by one by adding modulo-2 a secret sequence $k_1, \ldots, k_i, \ldots$:

$$c_i = m_i \oplus k_i$$

and the decryption transformation consists in adding once more the same binary sequence k_1, ..., k_i, ...:

$$\begin{aligned} m_i &= c_i \oplus k_i \\ &= (m_i \oplus k_i) \oplus k_i \\ &= m_i \end{aligned}$$

An advantage of stream cipher modes over block cipher modes is that encryption and decryption transformation can begin without having to wait for a complete 64-bit block. Also each plaintext symbol (and ciphertext symbol) can be encrypted and decrypted as they are entered for transmission in the communication link.

5.4 Output Feedback Mode (OFB):

The implementation of the DES Output Feedback Mode is very similar to the Cipher Feedback Mode of operation except that the 64-bit register is not fed by the previous ciphertext data bits $c_1, \ldots, c_i, \ldots$ but instead by the binary sequence at the output of the DES encryption box.

Note that for the Output Feedback Mode, there is no error propagation since for this stream cipher mode, the content of the 64-bit register at the receiving end (used for DES decryption) generates a random sequence which is no longer a function of the received (and possibly corrupted) ciphertext bits from the transmission channel.

5.5 Cipher Feedback Mode (CFB):

The implementation of the DES Cipher Feedback Mode is very similar to the Output Feedback Mode of operation except that the 64-bit register is fed by the previous ciphertext data bits $c_1, \ldots, c_i, \ldots$ instead of the binary sequence at the output of the DES encryption box. In the Cipher Feedback Mode, each ciphertext bit (i.e. $c_{i-1}, c_{i-2}, \ldots$) is fed back to a 64-bit register one at the time. One bit of the resulting ciphertext k_i is then added modulo-2 with the incoming message bit to give the cipher bit c_i, which is sent through the communication link.

The error propagation behavior of the Cipher Feedback Modemode of operation is very similar to that of the Cipher Block Chaining Mode, since in both cases a single channel transmission error will affect a ciphertext bit as well as about half of 64-bit (this time in the 64-bit shift register in the decryption box).

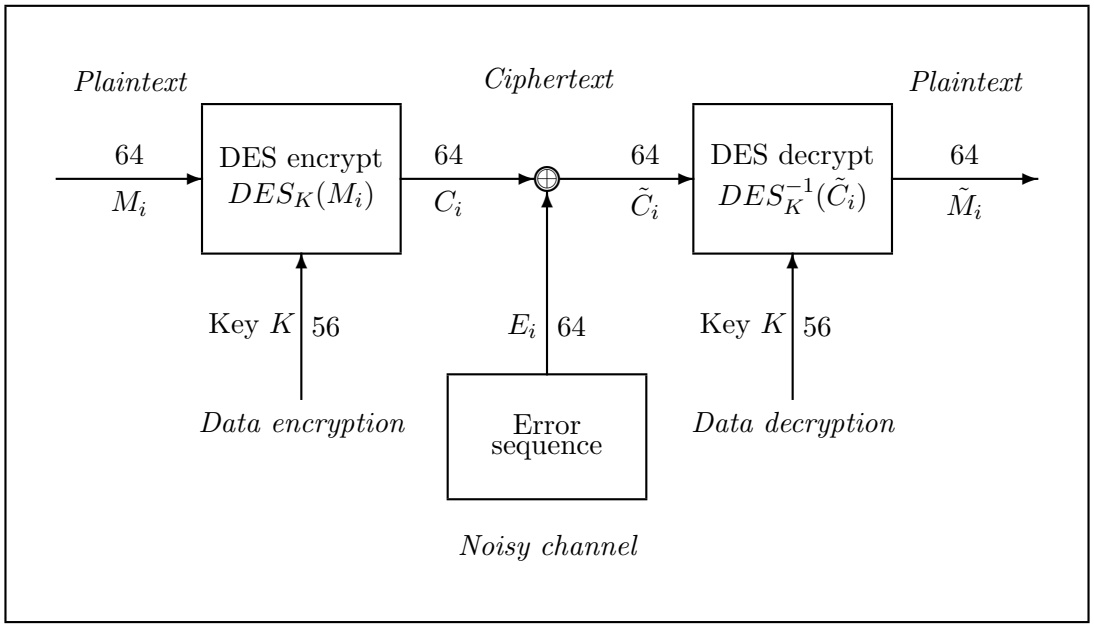


Figure 6: Electronic Codebook (ECB) mode of operation of DES.

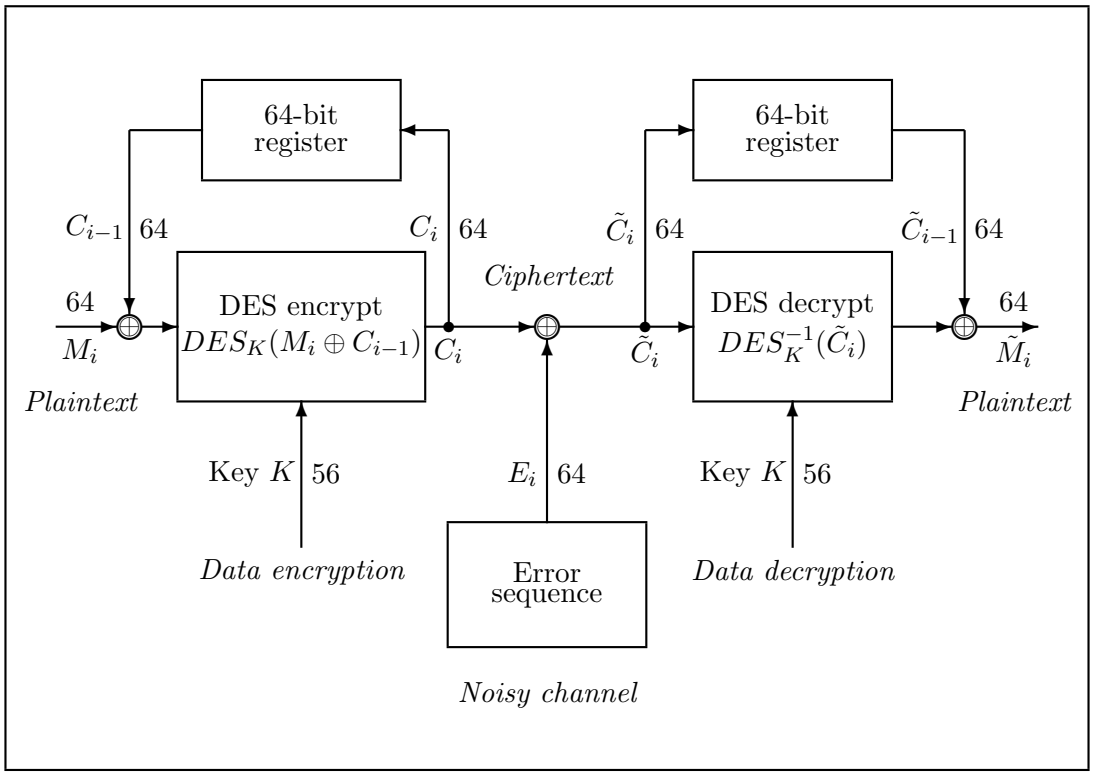


Figure 7: Cipher Block Chaining (CBC) mode of operation of DES.

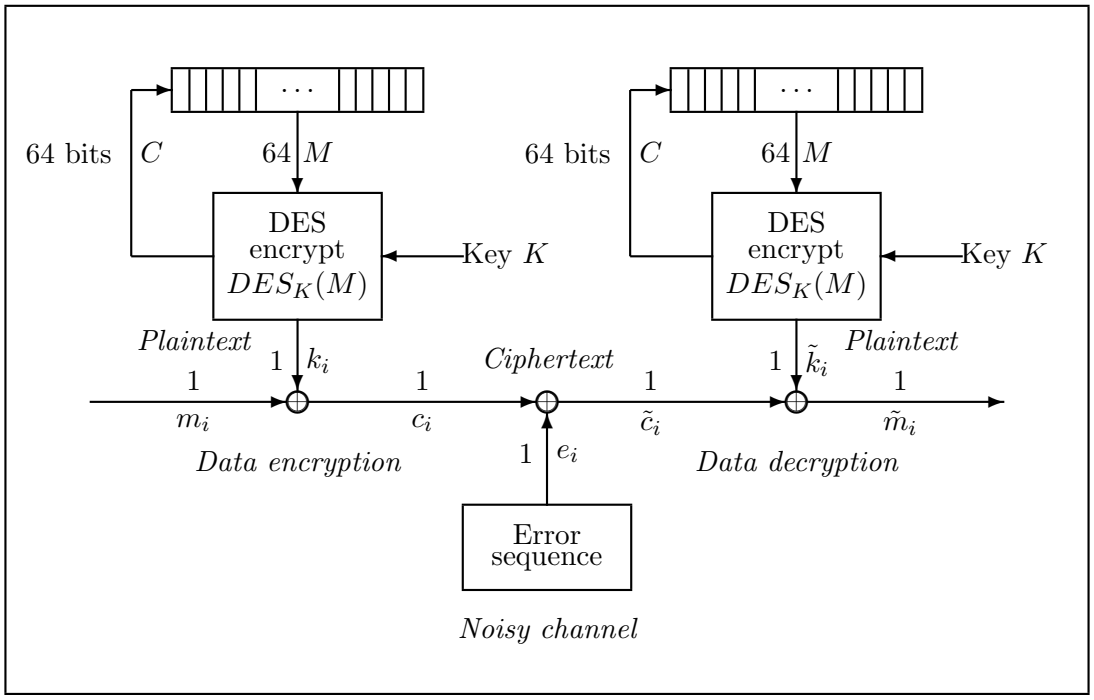


Figure 8: Output Feedback (OFB) mode of operation of DES.

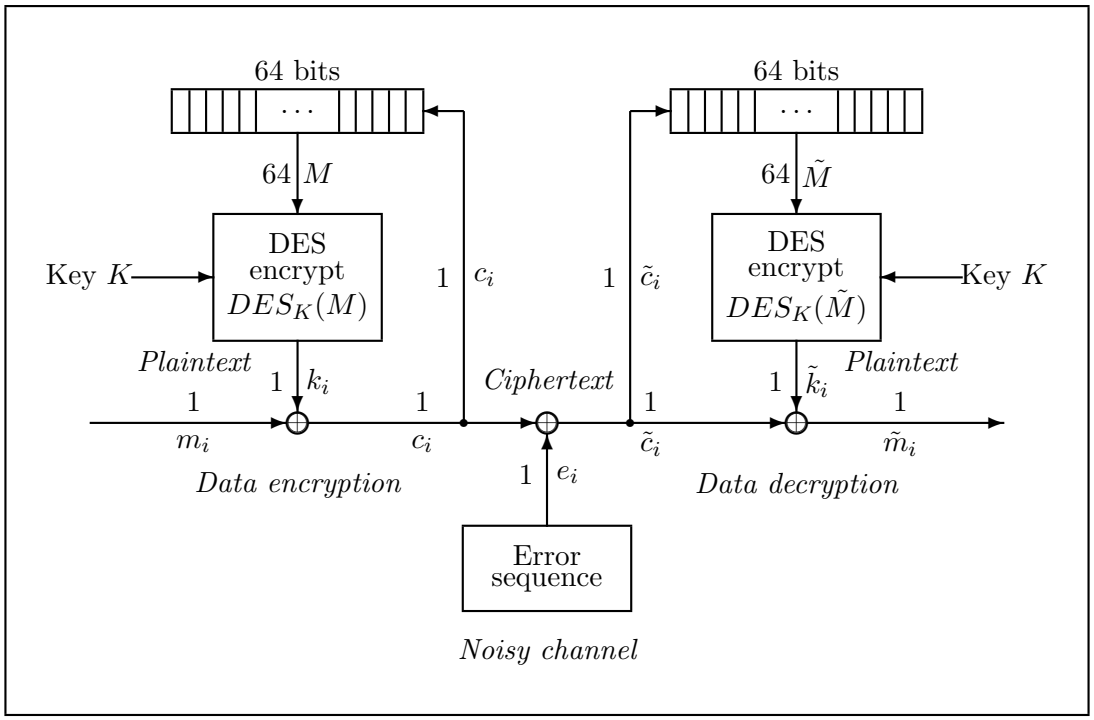


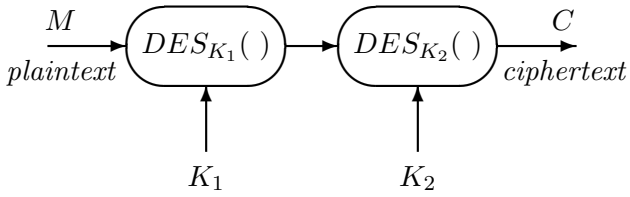
Figure 9: Cipher Feedback (CFB) mode of operation of DES.

6 Multiple DES encryption

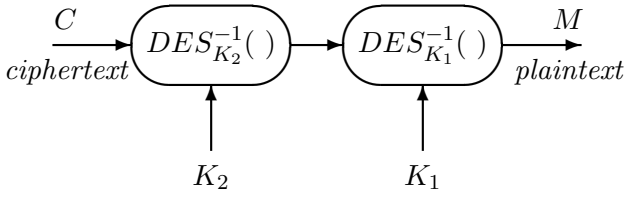
As we have seen, the Data Encryption Standard is vulnerable to a brute force cryptanalysis attack by performing an exhaustive key space search. Multiple DES encipherment provide additionnal cryptographic strength to DES by using more than a single key. In this section, we will see how to protect information using double DES and triple DES (subsections 6.1 and 6.2 respectively) enciphering.

6.1 Double DES encryption

Double DES encryption of a plaintext message M is achieved by applying the DES encryption transformation on the message with 56-bit key K_1 and then applying DES encryption on the resulting 64-bit block with a second 56-bit key K_2. The decryption of a ciphertext C from a double DES is obtained by applying the DES decryption transformation twice using first the last encryption key, i.e., K_2, and then the first one, K_1. Figure 10 illustrates the double DES encryption and decryption processes.



Double DES encryption



Double DES decryption

Figure 10: Double DES encryption and decryption.

$$
\begin{aligned}
C &= DES_{K_2}\left[DES_{K_1}(M)\right] \qquad \text{(double DES encryption)} \\
M &= DES^{-1}_{K_1}\left[DES^{-1}_{K_2}(C)\right] \qquad \text{(double DES decryption)}
\end{aligned}
$$

Although there are 2^{56} choices for key K_1 and 2^{56} choices for key K_2 which lead to an overall choice of 2^{112} pairs of keys (K_1, K_2), the cryptographic strength fo double DES is not as dramatically increased as it may appear.

Double DES is prone to what is refered to as a *meet-in-the-middle* attack. This type of known-plaintext attack was first presented by Diffie and Hellmann [DH77] and requires essentially two known plaintext-ciphertext pairs: (M_1, C_1) and (M_2, C_2) each pair obtained with the same double DES key pair (K_1, K_2):

$$\begin{aligned} C_1 &= DES_{K_2}\left[DES_{K_1}(M_1)\right] \qquad \text{and} \\ C_2 &= DES_{K_2}\left[DES_{K_1}(M_2)\right] \end{aligned}$$

The attack consists in computing, sorting by increasing value and recording in a table the 2^{56} encrypted 64-bit blocks $\{X_i\}$ obtained with all the possible keys on the known plaintext M_1.

$$X_i = DES_{K_i}(M_1) \qquad \text{for } 1 \leq i \leq 2^{56}$$

The next step consists of using the second key, K_2, to compute the 2^{56} different 64-bit decryption blocks $\{Y_j\}_{j=1,\ldots,2^{56}}$ using the known ciphertext C_1 (from the known (M_1, C_1) plaintext-ciphertext pair):

$$Y_j = DES^{-1}_{K_j}(C_1) \qquad \text{for } 1 \leq j \leq 2^{56}$$

For each decryption, the value of Y_j is compared with the table of sorted solutions $\{X_i\}_{i=1,\ldots,2^{56}}$. Since

$$C_1 = DES_{K_2}\left[DES_{K_1}(M_1)\right]$$

then there are values of $X_i = DES_{K_i}(M_1)$ that match $Y_j = DES^{-1}_{K_j}(C_1)$ and, out of these, one of them which is the desired solution. However, there are 2^{112} ways to choose the (K_1, K_2) pair of double DES keys but the double DES ciphertext can take only one of the 2^{64} values of the ciphertext space. Therefore, there are, on average, $\frac{2^{112}}{2^{64}} = 2^{48}$ pairs of keys (K_1, K_2) which will produce to the known ciphertext C_1 from the corresponding known plaintext M_1.

Now, using the second known plaintext-ciphertext pair (M_2, C_2), the cryptanalyst can encrypt M_2 with the suspected key pair $DES_{K_j}\left[DES_{K_i}(M_2)\right]$ and compare the result with the known ciphertext C_2. If indeed $C_2 = DES_{K_j}\left[DES_{K_i}(M_2)\right]$ then the cryptanalyst is pretty sure that the actual key pair is $(K_i, K_j) = (K_1, K_2)$ and that he, or she, has broken the double DES. In fact, the probability of succeeding twice matching X_i and Y_j with the same pair (K_i, K_j) wrongfully (that is, when $(K_i, K_j) \neq (K_1, K_2)$) is very unlikely: the probability of a false solution is

$$p_{false} = \frac{2^{112}}{2^{64} \times 2^{64}} = 2^{-16}$$

Note that each time a new plaintext-ciphertext pair is used the calculations of X_i and Y_j requires the storage of $2^{56} = 7.20576 \times 10^{16}$ DES enciphered blocks of 64 bits each, or 5.76461×10^{17} bytes. This means that breaking double DES with two pairs of plaintext-ciphertext requires the precomputation of 2^{56} encryptions of $X_i = DES_{K_i}(M)$ for each pair, or about 2^{57} encryptions. The decryptions $Y_j = DES^{-1}_{K_j}(C)$ are done simultaneously.

The cryptanalyst may want more assurance about the results of the meet-in-the-middle attack. Using a third known plaintext-ciphertext pair, say (M_3, C_3), he can reduce the probability of false double DES breaking to:

$$p_{false} = \frac{2^{112}}{2^{64} \times 2^{64} \times 2^{64}} = 2^{-80}$$

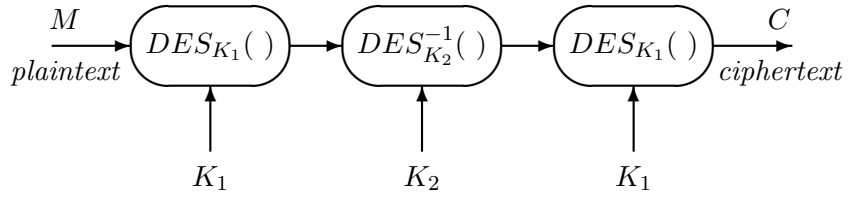
6.2 Triple DES encryption

6.2.1 Triple DES encryption with 2 keys

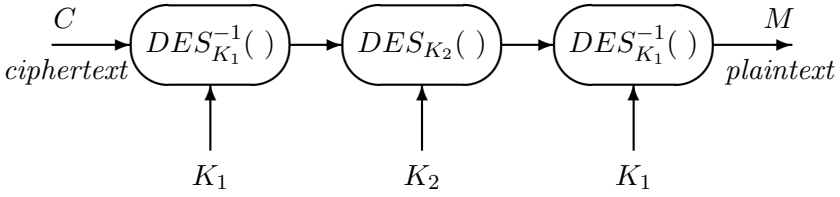
To prevent a *meet-in-the-middle* type of attack, a third DES encryption box may be used in cascade with two distinct keys K_1 and K_2 as shown in figure 11. Triple DES encryption and decryption using two different keys are performed as:

$$C = DES_{K_1}\left\{DES_{K_2}^{-1}\left[DES_{K_1}(M)\right]\right\} \qquad \text{(triple DES encryption)}$$

$$M = DES_{K_1}^{-1}\left\{DES_{K_2}\left[DES_{K_1}^{-1}(C)\right]\right\} \qquad \text{(triple DES decryption)}$$



Triple DES encryption (2 keys)



Triple DES decryption (2 keys)

Figure 11: Triple DES encryption and decryption using 2 keys.

The second encryption box is really a DES decryption box: this arrangement is used to allow triple DES to be used as single DES but has no effect on the cryptographic strength of triple DES. It is possible to use triple DES encryption to communicate with a single key DES user, as a conventional single DES, by setting the two keys equal to $K = K_1 = K_2$.

Since the second transformation is in fact a decryption transformation (that is, with the reverse order of the sequence of sixteen 48-bit subkeys), the cascade of encryption and decryption using the same key K: $DES_K^{-1}\left[DES_K(M)\right]$ is simply M and applying DES encryption once more provides the desired single DES encrypted cryptogram $C = DES_K(M)$. The cryptogram C can then be decrypted using the conventional single DES: $DES_K^{-1}(C)$.

$$C = DES_K\left\{DES_K^{-1}\left[DES_K(M)\right]\right\} \qquad \text{(triple DES encryption)}$$

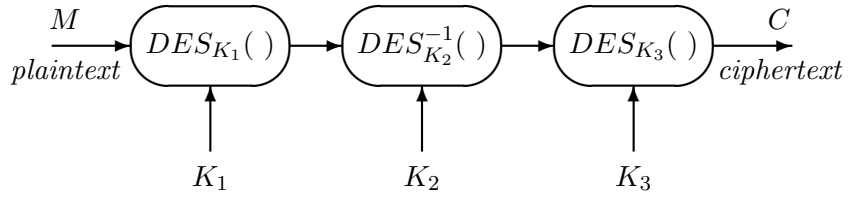
$$
\begin{aligned}
C &= DES_K(M) \\
M &= DES_K^{-1}(C) \qquad \text{(single DES decryption)}
\end{aligned}
$$

6.2.2 Triple DES encryption with 3 keys

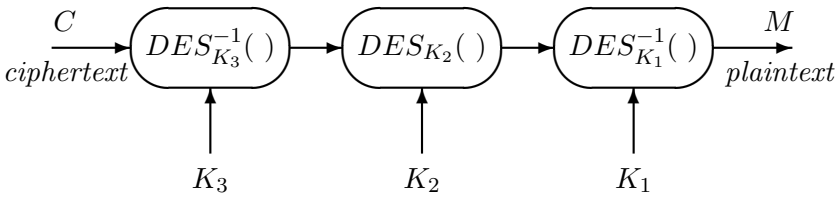
Even if there is no known method to break triple DES with two different keys, some still prefer to use triple DES encryption with three different keys (see 12 below). The plaintext encryption and ciphertext decryption are then obtained as:

$$
\begin{aligned}
C &= DES_{K_3}\left\{DES_{K_2}^{-1}\left[DES_{K_1}(M)\right]\right\} \qquad \text{(triple DES encryption)} \\
M &= DES_{K_1}^{-1}\left\{DES_{K_2}\left[DES_{K_3}^{-1}(C)\right]\right\} \qquad \text{(triple DES decryption)}
\end{aligned}
$$

Since three different 56-bit keys are used, that is K_1, K_2 and K_3, this requires a total of 168 key bits. Once again, it is possible to use triple DES encryption to encrypt single DES cryptograms. This time it is done by simply repeating the same key 3 times: $K = K_1 = K_2 = K_3$.



Triple DES encryption (3 keys)



Triple DES decryption (3 keys)

Figure 12: Triple DES encryption and decryption with 3 different keys.

There is no practically feasible attacks on triple DES. It is estimated [Cop94] that an exhaustive search will require about $2^{112} = 5.1923 \times 10^{33}$ computations!

Triple DES with this configuration is used on some Internet communications to ensure secure transfer of information over a computer communication network.

References

[BS93] E. Biham and A. Shamir. *Differential Cryptanalysis of the Data Encryption Standard*. Springer-Verlag, New-York, 1993.

[Cop94] D. Coppersmith. The Data Encryption Standard (DES) and its Strength Against Attacks. *IBM Journal of Research and Development*, May 1994.

[DH77] W. Diffie and M.E. Hellman. Exhaustive Cryptanalysis of the NBS Data Encryption Standard. *Computer*, 10(6):74–84, June 1977.

[DP84] D.W. Davies and W.L. Price. *Security for Computer Networks*. John Wiley and Sons, New-York, 1984.

[NIS77] NIST. Data Encryption Standard. Technical Report FIPS PUB 46-3, National Institute of Standards and Technology, Washington DC, January 1977. (reaffirmed 25 October 1999).

[Pfl89] C.P. Pfleeger. *Security in Computing*. Prentice-Hall, Englewood Cliffs, New-Jersey, 1989.

[Sch96] B. Schneier. *Applied Cryptography: Protocols, Algorithms, and Source Code in C (second edition)*. John Wiley and Sons, New-York, 1996.

[Sim92] G.J. Simmons. *Contemporary Cryptography: The Science of Information Integrity*. IEEE Press, New-York, 1992.