Uğur Kellecioğlu

30467504728

CMPE 325 HW2

Instructor: Haydar Çukurtepe

HOMEWORK

4.10 (20pts)

Compute the bits number 4, 17, 41, and 45 at the output of the first round of the DES decryption, assuming that the ciphertext block is composed of all ones and the external key is composed of all ones.

Main key K = 11111111 11111111 11111111 11111111 11111111 11111111 11111111 → 56 bits

K1 = K2 =...= K16 =1111111 1111111 1111111 1111111 1111111 1111111 1111111(48 bits)

C = 11111111 11111111 11111111 11111111 11111111 11111111 11111111 11111111

C Left = 11111111 11111111 11111111 11111111

C Right = 11111111 11111111 11111111 11111111

32 bit to 48 bit

E(Right) = 11111111 11111111 11111111 11111111 11111111 11111111

Inside of the function F,

E(RD0) + K16 = 00000000 00000000 00000000 00000000 00000000 00000000 00000000

Permutation

B = 11101111101001110010110001001101

P(B)= 11011000110110001101101110111100

R1= 00100111 00100111 00100100 01000011

L1= 11111111 11111111 11111111 11111111

so → 1111111**1** 11111111 **1**1111111 11111111 00100111 **0**0100**0**111 00100100 01000011

4 → 1

17 → 1

41 → 0

45 → 0

4.11 This problem provides a numerical example of encryption using a one-round version of DES. We start with the same bit pattern for the key K and the plaintext, namely:

Hexadecimal notation:	0 1 2 3 4 5 6 7 8 9 A B C D E F
Binary notation:	0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111

a. Derive K_1, the first-round subkey.
b. Derive L_0, R_0.
c. Expand R_0 to get $E[R_0]$, where $E[\cdot]$ is the expansion function of Table S.1.
d. Calculate $A = E[R_0] \oplus K_1$.
e. Group the 48-bit result of (d) into sets of 6 bits and evaluate the corresponding S-box substitutions.
f. Concatenate the results of (e) to get a 32-bit result, B.

g. Apply the permutation to get $P(B)$.
h. Calculate $R_1 = P(B) \oplus L_0$.
i. Write down the ciphertext.

M = 0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111

K = 0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111

From the original 64 bit key -> 56 bit key

K+ = 1111 0000 1100 1100 1010 1010 0000 1010 1010 1100 1100 1111 0000 0000

C_0 = 1111000011001100101010100000

D_0 = 1010101011001100111100000000

C_1 = 1110 0001 1001 1001 0101 0100 0001 D_1 = 0101 0101 1001 1001 1110 0000 0001	We apply PC-2 to Kn = CnDn K_1 = 0000 1011 0000 0010 0110 0111 1001 1011 0100 1001 1010 0101

M = 0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111

IP = 1100 1100 0000 0000 1100 1100 1111 1111 1111 0000 1010 1010 1111 0000 1010 1010

L_0 = 1100 1100 0000 0000 1100 1100 1111 1111

R_0 = 1111 0000 1010 1010 1111 0000 1010 1010

$L_n = R_{n-1}$

$R_n = L_{n-1} + f(R_{n-1}, K_n)$

K_1 = 0000 1011 0000 0010 0110 0111 1001 1011 0100 1001 1010 0101

$L_1 = R_0$ = 1111 0000 1010 1010 1111 0000 1010 1010

$R_1 = L_0 + f(R_0, K_1)$ = 1100 1100 0000 0000 1100 1100 1111 1111 xor $f(R_0, K_1)$

E(R0)= 0111 1010 0001 0101 0101 0101 0111 1010 0001 0101 0101 0101

K1 + E(R0) = 011100 010001 011100 110010 111000 010101 110011 110000

B1 = 0 1110 0 → row is 00 = 0, rest is 1110 = column number 14 → 0000

B2 = 0 1000 1 → row is 01 = 1, rest is 1000 = column number 8 → 1100

B3 = 0 1110 0 → row is 00 = 0, rest is 1110 = column number 14 → 0010

B4 = 1 1001 0 → row is 10 = 2, rest is 1001 = column number 9 → 0001

B5 = 1 1100 0 → row is 10 = 2, rest is 1100 = column number 12 → 0110

B6 = 0 1010 1 → row is 01 = 1, rest is 1010 = column number 10 → 1101

B7 = 1 1001 1 → row is 11 = 3, rest is 1001 = column number 9 → 0101

B8 = 1 1000 0→ row is 10 = 2, rest is 1000 = column number 8 → 0000

$S_1(B_1)S_2(B_2)S_3(B_3)S_4(B_4)S_5(B_5)S_6(B_6)S_7(B_7)S_8(B_8)$ = 00001100001000010110110101010000

After P

f = 1001 0010 0001 1100 0010 0000 1001 1100

$R_1 = L_0 + f(R_0 , K_1)$

R1 = 1100 1100 0000 0000 1100 1100 1111 1111 + 1001 0010 0001 1100 0010 0000 1001 1100

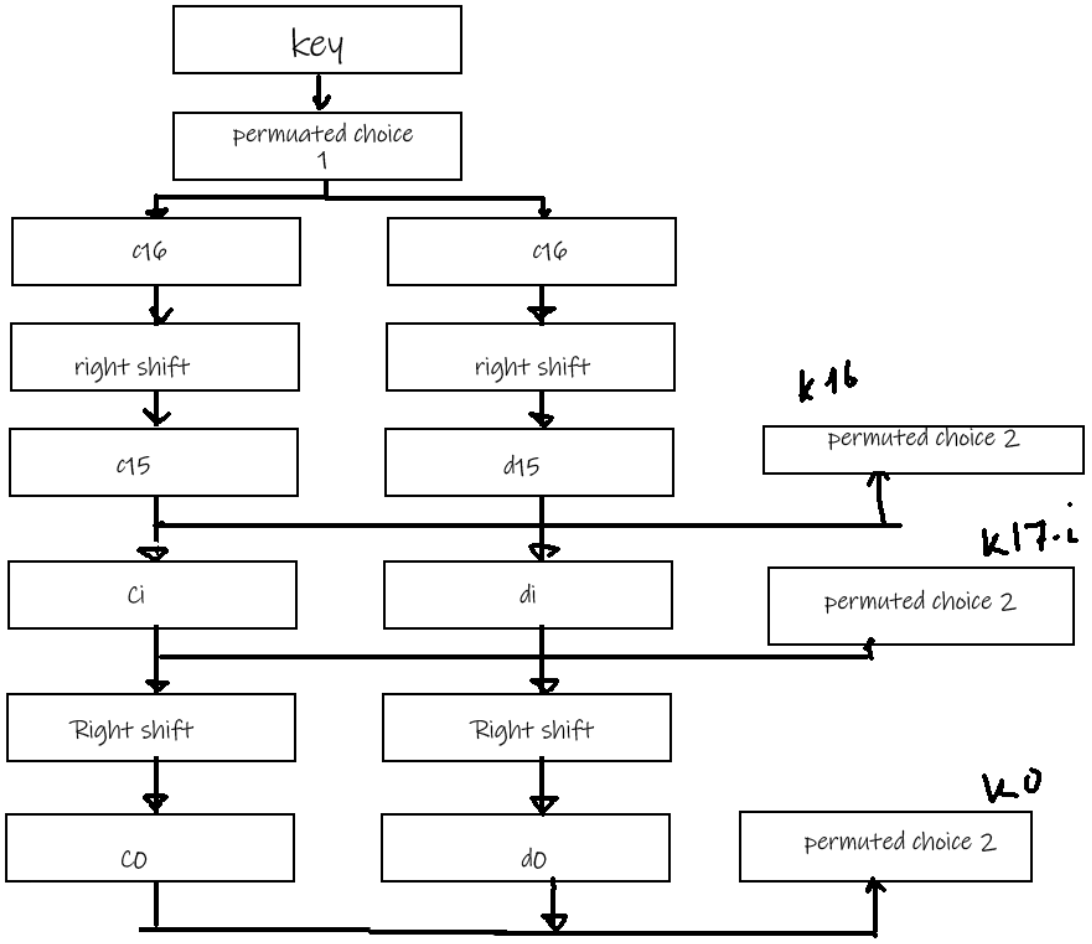
= 0101 1110 0001 1100 1110 1100 0110 0011

R1L1 = 01011110000111001110110001100011111100001010101011110000101010 10

IP-1 =0000 0001 0110 0011 0101 0100 0111 0110 1101 1000 1010 1111 1100 1101 1010 1110

1635476D8AFCDAE

4.13 When using the DES algorithm for decryption, the 16 keys ($K_1, K_2, \ldots, K_{16}$) are used in reverse order. Therefore, the right-hand side of Figure S.1 is not valid for decryption. Design a key-generation scheme with the appropriate shift schedule (analogous to Table S.3d) for the decryption process.



1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	1	2	2	2	2	2	2	1	2	2	2	2	2	2	1

4.18

M = 01000110
K = 1010000010

Subkeys

After P10 K = 1000001100

L_0 = 10000 R_0 = 01100

Left shift 1

L_0 = 00001 R_0 = 11000

After P8 we get K_1 = 10100100

Left shift 2 of Left shift 1

L_0 = 00100 R_0 = 00011

After P8 we get K_2 = 01000011

} We get our Keys

Message

Initial permutation of M = 11000001 = IP

Round 1

L = 1100 R = 0001

- Expansion permutation 10000010 = EP
- EP ⊕ K_2 = 11000001 (L | R)

S-box 0 (01) S-box 1 (10) => After P4 1010 => XOR => 0101

fk_1 = 01100001 switch func = 00010110

Round 2

- Let's split switch into 2

  L = 0001 R = 0110

- Apply expansion EP_2 = 00111100

- $EP_2 \oplus K_2 = \underbrace{1001}_{L}\ \underbrace{1000}_{R}$

- $\frac{Sbox_0}{11}$ $\frac{Sbox_1}{11}$ => After P_4 = 1111 => XOR => 1110

fk_2 is 11100110

IP^{-1} = Result = 01101101