Si funksionon DES

Des pranon si input mesazhin dhe një çelës 64-bit si hyrje për enkriptim dhe dekriptim, nga të cilat përdoren vetëm 56-bit. 16 nënçelësat, me 48-bit secili, do të krijohen më pas nga ky çelës 56-bit.

Psh. çelësi ynë i hyrjes:
00010011 00110100 01010111 01111001 10011011 10111100 11011111 11110001

Hapi i parë është ndërrimi i çelësit duke përdorur tabelën

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

1111000 0110011 0010101 0101111 0101010 1011001 1001111 0001111

Ndarja e çelësit

1111000 0110011 0010101 0101111	0101010 1011001 1001111 0001111

Numri i përsëritjes së

ndryshimeve të numrave majtas

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

C0	1111000011001100101010101111	D0	0101010101100110011110001111
C1	1110000110011001010101011111	D1	1010101011001100111100011110
C2	1100001100110010101010111111	D2	0101010110011001111000111101
C3	0000110011001010101011111111	D3	0101011001100111100011110101
C4	0011001100101010101111111100	D4	0101100110011110001111010101
C5	1100110010101010111111110000	D5	0110011001111000111101010101
C6	0011001010101011111111000011	D6	1001100111100011110101010101

C7	11001010101011111111100001100	D7	01100111100011110101010110
C8	00101010101111111110000110011	D8	10011110001111010101011001
C9	01010101011111111100001100110	D9	00111100011110101010110011
C10	0101010111111111000011001 1001	D10	11110001111010101011001100
C11	0101011111111000011001100101	D11	11000111101010101100110011
C12	0101111111100001100110010101	D12	00011110101010110011001111
C13	0111111110000110011001010101	D13	01111010101011001100111100
C14	1111111000011001100101010101	D14	11101010101011001100111110001
C15	1111100001100110010101010111	D15	10101010101100110011111000111
C16	11110000110011001010101011111	D16	01010101011001100111100011111

Formojmë 16 çelësat përfundimtarë duke aplikuar një ndryshim tjetër

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

Për shembull, çelësi ynë **C1 D1** 1110000 1100110 0101010 1011111 1010101 0110011 0011110 0011110 do të bëhet:	
K 1 =	000110 110000 001011 101111 111111 000111 000001 110010
K 2 =	011110 011010 111011 011001 110110 111100 100111 100101
K 3 =	010101 011111 110010 001010 010000 101100 111110 011001
K 3 =	011100 101010 110111 010110 110110 110011 010100 011101
K 5 =	011111 001110 110000 000111 111010 110101 001110 101000
K 6 =	011000 111010 010100 111110 010100 000111 101100 101111
K 7 =	111011 001000 010010 110111 111101 100001 100010 111100
K 8 =	111101 111000 101000 111010 110000 010011 101111 111011
K 9 =	111000 001101 101111 101011 111011 011110 011110 000001
K 10 =	101100 011111 001101 000111 101110 100100 011001 001111
K 11 =	001000 010101 111111 010011 110111 101101 001110 000110
K 12 =	011101 010111 000111 110101 100101 000110 011111 101001
K 13 =	100101 111100 010111 010001 111110 101011 101001 000001
K 14 =	010111 110100 001110 110111 111100 101110 011100 111010
K 15 =	101111 111001 000110 001101 001111 010011 111100 001010
K 16 =	110010 110011 110110 001011 000011 100001 011111 110101

Hapi 2: Kodimi i çdo blloku 64-bitësh të mesazhit

Gjëja e parë që duhet të bëjmë është të aplikojmë një IP(initial permutation) të ndërrimit fillestar në çdo bllok prej 64 bitësh, sipas tabelës:

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

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

Tani e ndajmë bllokun **IP** në dy pjesë majtas L_0 dhe djathtas R_0:

L_0 = 1100 1100 0000 0000 1100 1100 1111 1111

R_0 = 1111 0000 1010 1010 1111 0000 1010 1010

Tani ne procedojmë me 16 përsëritje, për 1<= n <=16, duke përdorur një funksion **f** i cili funksionon në dy blloqe -- një bllok të dhënash prej 32 bitësh dhe një çelës K_n prej 48 bitësh -- për të prodhuar një bllok me 32 bit .

$$L_n = R_{n-1}$$
$$R_n = L_{n-1} \oplus f(R_{n-1}, K_n)$$

Psh për n=1

K_1 = 000110 110000 001011 101111 111111 000111 000001 110010
$L_1 = R_0$ = 1111 0000 1010 1010 1111 0000 1010 1010
$R_1 = L_0 \oplus f(R_0, K_1)$

Pra, si funksionon funksioni f ?

Për të llogaritur **f**, fillimisht zgjerojmë çdo bllok $\mathbf{R_{n-1}}$ nga 32 bit në 48 bit. Kjo bëhet duke përdorur një tabelë përzgjedhjeje që përsërit disa nga bitet në $\mathbf{R_{n-1}}$.
Kjo tabelë përzgjedhëse **E** ka një bllok hyrës 32 bit ($\mathbf{R_{n-1}}$) dhe një bllok dalje 48 bit.

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

Shembull: Ne kalkulojmë $\mathbf{E(R_0)}$ nga $\mathbf{R_0}$ si më poshtë:

$\mathbf{R_0}$ = 1111 0000 1010 1010 1111 0000 1010 1010
$\mathbf{E(R_0)}$ = 011110 100001 010101 010101 011110 100001 010101 010101

Hapi tjetër në llogaritjen e **f** është XOR-i i daljes $\mathbf{E(R_{n-1})}$ me çelësin $\mathbf{K_n}$:

$\mathbf{K_n \oplus E(R_{n-1})}$

Shembull: Për $\mathbf{K_1}$, $\mathbf{E(R_0)}$, kemi:

K1 = 000110 110000 001011 101111 111111 000111 000001 110010
$E(R_0)$ = 011110 100001 010101 010101 011110 100001 010101 010101
$K_1+E(R_0)$ = 011000 010001 011110 111010 100001 100110 010100 100111

Deri në këtë pikë ne kemi zgjeruar $\mathbf{R_{n-1}}$ nga 32 bit në 48 bit, duke përdorur tabelën e përzgjedhjes, dhe XOR e kemi bërë rezultatin me çelësin $\mathbf{K_n}$

Tani i kemi 48 bit, që do të përdoren për kutitë **S**.

Një kuti **S** merr si hyrje 6 bit dhe jep 4 bit dalje që do të zëvendësojë hyrjen 6 bit. Kemi 8 grupe me 6 bit

$\mathbf{K_n \oplus E(R_{n-1}) = B_1\ B_2\ B_3\ B_4\ B_5\ B_6\ B_7\ B_8}$,

Kutia $\mathbf{S_1}$ funksionon si më poshtë:

S1

Row No.	Column Number															
	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

6 bitet e parë: 011000
Përcaktimi i rreshtit - miret biti i parë dhe i fundit: 00 = 0 (me bazë 10)
Përcaktimi i kolonës - miren 4 bitet në mes: 1100 = 12 (me bazë 10)
Kjo përcakton daljen; 5 në binar 0101, kështu që dalja është 0101.
Prandaj S1(011000) = 0101.

Tabelat që përcaktojnë funksionet **S1,...,S8** janë si më poshtë:

S1

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

S2

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

S3

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

S4

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

S5

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

S6

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

S7

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

S8

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

Në shembullin tonë ne marrim si dalje të tetë kutive S :

K_1 + **E**(R_0) = 011000 010001 011110 111010 100001 100110 010100 100111.

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

0101 1100 1000 0010 1011 0101 1001 0111

Faza e fundit në llogaritjen e **f** është të bëjmë një ndryshim **P** të daljes **S**-box për të marrë vlerën përfundimtare të **f**:

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

Nga shembulli ynë marrim se:

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

0101 1100 1000 0010 1011 0101 1001 0111

f = 0010 0011 0100 1010 1010 1001 1011 1011

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

= 1100 1100 0000 0000 1100 1100 1111 1111
⊕ 0010 0011 0100 1010 1010 1001 1011 1011
= 1110 1111 0100 1010 0110 0101 0100 0100

Në raundin tjetër, do të kemi $L_2 = R_1$, që është blloku që sapo kemi llogaritur, dhe më pas duhet të llogarisim $R_2 = L_1 \oplus f(R_1 , K_2)$, e kështu me radhë për 16 raunde. Në fund të raundit të gjashtëmbëdhjetë kemi blloqet L_{16} dhe R_{16} . Më pas e kthejmë rendin e dy blloqeve në bllokun 64-bit

R_{16} L_{16} dhe aplikoni një ndryshim përfundimtar IP^{-1} siç përcaktohet nga tabela e mëposhtme:

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

Shembull:

L_{16} = 0100 0011 0100 0010 0011 0010 0011 0100
R_{16} = 0000 1010 0100 1100 1101 1001 1001 0101

Ne e kthejmë rendin e këtyre dy blloqeve dhe aplikojmë ndryshimin përfundimtar në

$R_{16}L_{16}$ = 00001010 01001100 11011001 10010101 01000011 01000010 00110010 00110100

IP^{-1} = 10000101 11101000 00010011 01010100 00001111 00001010 10110100 00000101

I cili në hexadecimal format është

85E813540F0AB405.

M = 0123456789ABCDEF, C = 85E813540F0AB405.