Laborator 4

Algoritmul de criptare DES (Data Encryption Standard)

Algoritmul DES este cel mai folosită metodă de criptare din acest moment în întreaga lume. Pentru foarte mult timp şi aproape pentru toată lumea din domeniul criptografiei DES şi criptare sigură" au fost sinonime. În ciuda loviturii primite de la organizaţia EFF (Electronic Frontier Foundation) ce a creat o maşină în valoare de 220.000 $ special pentru „spargerea" algoritmului DES, acesta continuă să fie folosit de guvernul SUA şi de bancile comerciale în continuare prin noua sa extensie denimită „triple-DES".

1. Scurt istoric

În data de 15 mai 1973, în timpul mandatului preşedintelui Richard Nixon, organizaţia guvernamentală americană NBS (National Bureau of Standards), a publicat o notă în registrul federal (echivalentul Monitorului Oficial) solicitând pentru un algoritm de criptare ce va facilita protecţia datelor în timpul transmiterii acestora sau atunci când sunt stocate digital. Nota explica de ce criptarea este importantă în viitorul apropiat al SUA:

În ultima decadă, a fost observată o creştere accelerată în acumularea şi comunicarea datelor digitale de către guvern, industrie şi de către alte numeroase organizaţii din sectorul privat. Conţinutul acestor date stocate sau transmise este nu de multe ori foarte valoros şi uneori secret . În acest moment întâlnim în mod uzual transmisii de date ce constituie transferuri bancare de ordinul milioanelor de dolari, vânzări sau cumpărări de bunuri ce ţin de securitatea naţională, mandate de arestare sau de condamnare, rezervarea de locuri la cursele aeriene, fişiere ce conţin date despre pacienţi şi tratamentul lor, şamd.

Importanţa mare din punct de vedere al volumului, valorii şi confidenţialităţii acestor date, transmise de organizaţiile particulare sau guvernamentale, a dus la îngrijorare în ceea ce priveşte la expunerea acestor date la acces şi utilizare neautorizate.

Este cunoscut că criptarea este singura metodă de protecţie a acestor date în timpul transmiterii sau stocării lor. […] Organizaţia noastră, în rolul ei de a stabili standarde precum şi de a juta diversele organizaţii private sau guvernamentale în accesarea tehnologiilor moderne, va evalua toate propunerile venite înainte de a lua orice decizie."

Din acest moment NBS a aşteptat un răspuns ce a venit în data de 6 august 1974, atunci când IBM a trimis un algoritm candidat ce a fost dezvoltat şi folosit intern sub numele de LUCIFER. După ce a evaluat algoritmul cu ajutorul agenţiei NSA (National Security Agency), NBS a adoptat o variantă modificată a algoritmului LUCIFER drept noul standard pentru criptografia americanş şi l-a denumit DES, în 15 iulie 1977.

DES a fost adoptat rapid pentru datele de tip analogic, mai ales în domeniul telefoniei. În acelaşi timp, industria bancară, care este cel mai mare utilizator al algoritmului in SUA, a adoptat DES ca standard pentru criptarea datelor bancare. Standardizarea a fost făcută cu ajutorul asociaţiei guvernamentale ANSI (American National Standards Institute) şi adoptată sub numele de X3.92 în 1980.

2. Introducere în DES

Algoritmul DES foloseşte numere binare folosite de calculatoarele moderne. Fiecare grup de 4 biţi reprezintă un număr hexazecimal.
Ex:
0001 – 1
0010 – 2
0011 – 3
1010 - A
1011 – B
1111 – F

Algoritmul DES acţionează asupra grupurilor de 64 de biţi, sau altfel spus asupra unui număr hexazecimal de 16 cifre. Pentru criptare DES foloseşte chei ce aparent au lungimea de 64 de biţi, din care sunt folosiţi însă doar 56.

De exemplu, daca vom cripta numărul
8787878787878787 cu cheia 0E329232EA6D0D73 vom obţine 0000000000000000. La decriptare din 0000000000000000, folosind aceeaşi cheie, vom obţine tot
8787878787878787.

Atunci când lungimea textului iniţial nu este multiplu de 16, vom completa cu numărul de
0-uri necesar.

3. DES, descriere în detaliu

DES este un algoritm de criptare pe blocuri (block cipher) ce acţionează asupra grupurilor de 64 de biţi utilizând chei de 64 de biţi. Rezultatul este o permutaţie dintre cele 2^64 permutaţii posibile.

A. Paşi preliminari. Impărţirea în blocuri de 32 de biţi
Cei 64 de biţi iniţiali vor fi împărţiţi în 2 blocuri de 32 de biţi denumiţi L (left) and R (right), după exemplul de mai jos:

M = 0123456789ABCDEF
M = 0000 0001 0010 0011 0100 0101 0110 0111 1000 1001 1010 1011 1100 1101 1110 1111
L = 0000 0001 0010 0011 0100 0101 0110 0111
R = 1000 1001 1010 1011 1100 1101 1110 1111

B. Paşi preliminari. Alegerea cheii

Fie K cheia hexazecimală
K = 133457799BBCDFF1
Ce va duce la reprezentarea ei binară:
K = 00010011 00110100 01010111 01111001 10011011 10111100 11011111 11110001

Pasul 1 – Crearea a 16 sub-chei de 48 de biţi fiecare

P1.1
Cheia originală este permutată conform tabelei PC-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

Adică, bitul 57 din cheia originală va deveni bitul 1 în cheia K+, bitul 49 din cheia originală va deveni bitul 2 din cheia K+, ş.a.m.d.

Ex:
K = 00010011 00110100 01010111 01111001 10011011 10111100 11011111 11110001 (64b)
K+ = 1111000 0110011 0010101 0101111 0101010 1011001 1001111 0001111 (56b)
Observaţie: K+ conţine grupuri de 7 biţi, nu de 8

P1.2
Vom împărţi K+ în cele două jumătăţi ale sale pe care le vom denumi C0 şi D0:

K+ = 1111000 0110011 0010101 0101111 0101010 1011001 1001111 0001111 (56b)
C_0 = 1111000 0110011 0010101 0101111 (! Sunt grupuri de 7b, nu de 8)
D_0 = 0101010 1011001 1001111 0001111 (! Sunt grupuri de 7b, nu de 8)

P1.3
Folosind C0 şi D0, vom crea 16 blocuri Cn, Dn n=1,16. Fiecare Cn, Dn va fi creat din C_{n-1}, D_{n-1} folosind schema de mai jos în care vom face rotaţii la stînga, funcţie de numărul din cea de-a doua coloană:

Numărul Iteraţiei	Numărul de rotiri stânga
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

Exemplu: C3,D3 vor fi obţinute din C2,D2 rotite la stânga cu 2 biţi

Din C0, D0 originali, vom obţine:

C_0 = 11110000110011001010101011111
D_0 = 01010101011001100111100011111

C_1 = 11100001100110010101010111111
D_1 = 10101010110011001111000111110

C_2 = 11000011001100101010101111111
D_2 = 01010101100110011110001111101

C_3 = 00001100110010101010111111111
D_3 = 01010110011001111000111101101

C_4 = 00110011001010101011111111100
D_4 = 01011001100111100011110101101

C_5 = 11001100101010101111111110000
D_5 = 01100110011110001111010101101

C_6 = 00110010101010111111111000011
D_6 = 10011001111000111101010101101

C_7 = 11001010101011111111100001100
D_7 = 01100111100011110101010110110

C_8 = 00101010101111111110000110011
D_8 = 10011110001111010101011001101

C_9 = 01010101011111111100001100110
D_9 = 00111100011110101010110011011

C_{10} = 01010101111111100001100110011
D_{10} = 11110001111010101010110011001100

C_{11} = 01010111111110000110011001011
D_{11} = 11000111101010101011001100111

C_{12} = 01011111111000011001100101011
D_{12} = 00011110101010101100110011111

C_{13} = 01111111100001100110010101011
D_{13} = 01111010101010110011001111001

C_{14} = 11111110000110011001010101011
D_{14} = 11101010101011001100111100011

C_{15} = 11111000011001100101010101111
D_{15} = 10101010101100110011110001111

C_{16} = 11110000110011001010101011111
D_{16} = 01010101011001100111100011111

P1.4

Vom calcula sub-cheile Kn n=1,16 aplicând următoarea permutare celor 16 numere binare obţinute prin concatenarea CnDn:

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

Observaţie: Din nou, primul bit din Kn va fi cel de-al 14-lea din sirul concatenat CnDn, al doilea bit din Kn va fi cel de-al 17-lea din CnDn, ş.a.m.d.

Ex:

C_1D_1 = 1110000 1100110 0101010 1011111 1010101 0110011 0011110 0011110

K_1 = 000110 110000 001011 101111 111111 000111 000001 110010

Celelalte chei vor fi în ordine:

K_2 = 011110 011010 111011 011001 110110 111100 100111 100101 (6 *8 = 48b)

K_3 = 010101 011111 110010 001010 010000 101100 111110 011001

K_4 = 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

Temă de laborator:

Implementaţi funcţiile:

- DES_Key_Permut_PC1();
- DES_ Key_Divide_C0_D0() ;
- DES_ Key_Divide_Ci_Di() ;
- DES_ Key_Permut_PC2();
- DES_Key_CreateSubKeys() ce va apela toate funcţiile anterioare.