Ministerul Educaţiei și Cercetării al Republicii Moldova

Universitatea Tehnică a Moldovei

Facultatea Calculatoare, Informatică şi Microelectronică

**RAPORT**

Lucrare de laborator Nr.4

*la Criptografie*

Tema: Cifruri bloc. Algoritmul DES

A efectuat: st. gr. SI-212 Șeremet Alexandru

A verificat: lect. asist. Victor Andronatiev

Chişinău 2022

**Sarcina:**

Sarcina 1. Studiați materiale didactice plasate pe ELSE:

• C4. Cifruri bloc

• DES-FIPS-46

• DES-FIPS-46-3

• Theory of Data Encryption Standard (DES)

• ASCII Character Set

• DES eng

• DES RO

Sarcina 2. De elaborat un program în unul din limbajele de programare

preferate pentru implementarea unui element al algoritmului DES. Sarcina se

va alege în conformitate cu numărul n de ordine al studentului din lista grupei,

în conformitate cu formula: nr\_sarcina = n mod 11. Pentru fiecare sarcină să

fie afișate la ecran tabelele utilizate și toți pașii intermediari. Datele de intrare

să fie posibil de introdus de utilizator sau de generat în mod aleatoriu.

Atenție! La susținerea lucrării vor fi puse întrebări despre lucrul întregului

algoritm!!!

V 12 (12 mod 11 == 1)

2.1. Fiind dată cheia algoritmului DES (8 simboluri), de determinat K+.

**Rezumat:**

**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 hexa 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 cănd lungimea textului iniţial nu este multiplu de 16, vom completa cu numărul de 0-uri necesar.

**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)**

**(! Sunt grupuri de 7b, nu de 8)**

**Codul deplin al programului:**

from random import randint

PC1 = [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]

def hexToBin(K1):

K1 = int(K1, 16)

K1 = bin(K1).replace('0b', '')

while len(K1) % 8 != 0:

K1 = '0' + K1

return(K1)

def printPC1():

for i in range(len(PC1)):

if (i + 1) % 7 == 0:

print(PC1[i])

else:

print(PC1[i], end=' ')

def printBin(K):

for i in range(len(K)):

if (i + 1) % 8 == 0:

print(K[i], end=' ')

else:

print(K[i], end='')

def permutate(K1):

res = ''

for i in range(len(PC1)):

res += K1[PC1[i] - 1]

print('nr. ordine:', PC1[i], 'bit:', K1[PC1[i] - 1])

print('\n')

print('Rezultatul concatenarii bitilor, care si sunt cheia K+:')

printBin(res)

print('\n')

print('lungime:', len(res), 'biti')

def main():

print('\n')

choice = input(

'doriti sa introduceti cheia sau sa o generati aleatoriu?(1/2) ')

print('\n')

if choice == '1':

K1 = input('introduceti cheia hexadecimala: ').replace(' ', '')

elif choice == '2':

K1 = hex(randint(1000000000000000000, 9999999999999999999)

).upper().replace('0X', '')

print('Cheia generata:')

print(K1)

print('\n')

print('Se transforma din hexadecimal in binar:')

K1 = hexToBin(K1)

printBin(K1)

print('\n')

print('Dupa aia, cu ajutorul tabelului PC1:')

printPC1()

print('\n')

print('Se efectueaza operatiile de permutare:')

permutate(K1)

main()

**Execuția programului:**

1. Cheie introdusă de utilizator:









2. Cheie aleatorie:









**Concluzii:**

* Algoritmul DES este primul standard în criptografie, având un algoritm relativ complex la bază, ceea ce a și permis rezistența sa în fața timpului prin iterații ale lui, precum AES și 3-DES.
* Criptarea prin DES fără ajutorul automării computerului este extrem de ineficientă, și deci se poate spune că DES este un rezultat al digitalizării lumii și al începutului veacului informației.