Ministerul Educatiei, Culturii şi Cercetarii al Republicii Moldova
Universitatea Tehnică a Moldovei
Facultatea Calculatoare, Informatică şi Microelectronică
Departamentul Ingineria Software şi Automatica

Raport
pentru lucrarea de laborator Nr. 1
la cursul „Metode criptografice de protecție a informației”

A efectuat:
Cătălin MÎŢU, gr. SI-191
A verificat:
Aureliu ZGUREANU

Chişinău – 2022

Subiectul: Cifrul lui Cesar

Sarcini:

1. De implementat algoritmul Cezar pentru alfabetul limbii engleze în unul din limbajele de programare. Utilizaţi doar codificarea literelor cum este arătat în tabelul 1 (nu se permite de folosit codificările specificate în limbajul de programare, de ex. ASCII sau Unicode). Valorile cheii vor fi cuprinse între 1 şi 25 inclusiv şi nu se permit alte valori. Valorile caracterelor textului sunt cuprinse între 'A' şi 'Z', 'a' şi 'z' şi nu sunt premise alte valori. În cazul în care utilizatorul introduce alte valori - i se va sugera diapazonul corect. Înainte de criptare textul va fi transformat în majuscule şi vor fi eliminate spaţiile. Utilizatorul va putea alege operaţia - criptare sau decriptare, va putea introduce cheia, mesajul sau criptograma şi va obţine respectiv criptograma sau mesajul decriptat.

2. De implementat algoritmul Cezar cu 2 chei, cu păstrarea condiţiilor exprimate în Sarcina 1. În plus, cheia 2 trebuie să conţină doar litere ale alfabetului latin, şi să aibă o lungime nu mai mică de 7.

Cifrul lui Cesar

În acest cifru fiecare literă a textului clar este înlocuită cu o nouă literă obţinută printr-o deplasare alfabetică. Cheia secretă k, care este aceeaşi la criptare cât şi la decriptare, constă în numărul care indică deplasarea alfabetică, adică $k \in \{1, 2, 3, \ldots, n-1\}$, unde n este lungimea alfabetului. Criptarea şi decriptarea mesajului cu cifrul Cezar poate fi definită de formulele

$$c = e_k(x) = x + k \pmod{n},$$
$$m = d_k(y) = y - k \pmod{n},$$

unde x şi y sunt reprezentarea numerică a caracterului respectiv al textului clar. Funcţia numită *Modulo* (a mod b) returnează restul împărţirii numărului întreg a la numărul întreg b. De exemplu, pentru k = 3 avem (fig 1):

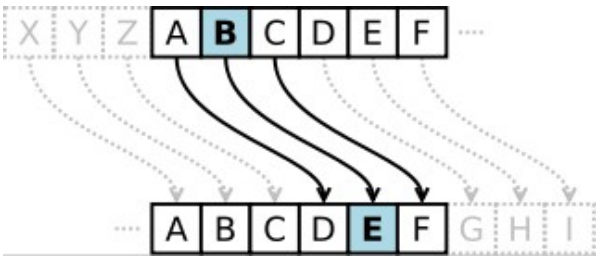

Figura 1 – Exemplu de deplasare alfabetica

Pentru a spori criptorezistența cifrului Cezar se poate de aplicat o permutare a alfabetului prin aplicarea unui cuvânt-cheie (a nu se confunda cu cheia de bază a cifrului). Această cheie poate fi orice consecutivitate de litere a alfabetului - fie un cuvânt din vocabular, fie unul fără sens.

Fie cheia a doua este k_2 =cryptography. Aplicăm această cheie asupra alfabetului

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

și obținem:

C R Y P T O G A H B D E F I J K M L N Q S U V W X Z

Această ordine nouă am obținut-o prin plasarea literelor lui k_2 la început, apoi urmează celelalte litere ale alfabetului în ordinea lor naturală. Vom ține cont de faptul că literele nu se vor repeta, adică dacă litera se întâlnește de câteva ori, ea se plasează doar o singură dată.

Rezultatul efectuarii sarcinilor:

Initial, dacă utilizatorul nu introduce nici un argument, va fi afisat mesajul cu instrucțiuni (figura 2).

```
catalin@UFO-L:~/UTM/Sem6/MCPI/Lab1/SourceCode/cmake-build-debug$ ./CaesarCipher
Introduceti argumentele in formatul urmator
./CaesarCipher actiunea mesajul cheia [cheia secundara]
Unde:
        actiunea - 'e' pentru criptare si 'd' pentru decriptare
        mesajul  - un sir de caractere ce contine litere ale alfabetului latin a-z si optional
 spatii
        cheia    - un numar cuprins intre 1 si 25, ambele inclusiv
        cheia secundara - un sir de caractere alcatuit din literele alfabetului latin cu lungi
mea cuprinsa intre 7 si 26 caractere
```

Figura 2 – Mesajul informational

Pentru a cripta un mesaj se introduce acţiunea „e”, mesajul, cheia de deplasare şi optional cheia secundara. Un exemplu de criptare cu o cheie este afisat în figura 3.

```
catalin@UFO-L:~/UTM/Sem6/MCPI/Lab1/SourceCode/cmake-build-debug$ ./CaesarCipher
 e "mesajul ce urmeaza a fi criptat" 13

--- inceput rezultat ---
ZRFNWHYPRHEZRNMNNSVPEVCGNG
--- sfarsit rezultat ---
```

Figura 3 – Criptarea mesajului cu o cheie

Un exemplu de criptare cu doua chei este afisat în figura 4.

```
catalin@UFO-L:~/UTM/Sem6/MCPI/Lab1/SourceCode/cmake-build-debug$ ./CaesarCipher
 e "mesajul ce urmeaza a fi criptat" 13 cheiesecundara
S-a primit advanced alphabet: CHEISUNDARBFGJKLMOPQTVWXYZ

--- inceput rezultat ---
ILOVCPEJLPWILVGVVYMJWMUDVD
--- sfarsit rezultat ---
```

Figura 4 – Critarea mesajului cu doua chei

Pentru a decripta un mesaj, e nevoie de introdus acţiunea „d”, mesajul şi cheia de decriptare şi la necesitate cheia secundara. Un exemplu de decriptare cu o cheie este afisat în figura 5.

```
catalin@UFO-L:~/UTM/Sem6/MCPI/Lab1/SourceCode/cmake-build-debug$ ./CaesarCipher
 d ZRFNWHYPRHEZRNMNNSVPEVCGNG 13

--- inceput rezultat ---
MESAJULCEURMEAZAAFICRIPTAT
--- sfarsit rezultat ---
```

Figura 5 – Decriptarea cu o cheie

Un exemplu de decriptare cu doua chei este afisat în figura 6.

```
catalin@UFO-L:~/UTM/Sem6/MCPI/Lab1/SourceCode/cmake-build-debug$ ./CaesarCipher
 d ILOVCPEJLPWILVGVVYMJWMUDVD 13 cheiesecundara
S-a primit advanced alphabet: CHEISUNDARBFGJKLMOPQTVWXYZ

--- inceput rezultat ---
MESAJULCEURMEAZAAFICRIPTAT
--- sfarsit rezultat ---
```

Figura 6 – Decriptarea cu doua chei

Concluzie: La aceasta lucrare de laborator am învăţat principiul de funcţionare a algoritmului de criptare a lui Cezar. Am efectuat aplicaţie care utilizeaza cifrul lui Cezar cu una şi cu doua chei pentru a cripta şi decripteaza mesaje. Chiar dacă versiunea cu doua chei nu pare la prima vedere atât de complicata, însă ea scade semnificativ şansele de a sparge acest cifru, în comparaţie cu cel cu doar o cheie. Daca am utiliza metoda exhaustiva vom avea nevoie de a verifica 26! * 25 opţiuni pentru a decodifica mesajul.