MINISTERUL EDUCAȚIEI ȘI CERCETĂRII AL REPUBLICII MOLDOVA

Universitatea Tehnică a Moldovei

Facultatea Calculatoare, Informatică şi Microelectronică

Departamentul Inginerie Software și Automatică

RAPORT

Lucrare de laborator nr. 3

la disciplina „***Criptografie***”

Tema: Cifruri polialfabetice

A efectuat : **St. gr. SI-253, Borta Adrian**

A verificat: **Asis.univ. Zaica Maia**

Chișinău 2025

INTRODUCERE

Substituţiile monoalfabetice au vulnerabilitatea că frecvenţa literelor din textul cifrat reflectă aproape fidel frecvenţa literelor în limba originală, ceea ce oferă indici unui criptanalist. Cifrele polialfabetice sunt mai sigure, pentru că ele schimbă regula de substituţie pe parcursul textului, de exemplu în funcţie de poziţie sau de context, astfel încât acelaşi caracter din textul clar nu este întotdeauna înlocuit cu acelaşi caracter în textul cifrat. Aceasta conduce la mult mai multe chei posibile şi la o distribuţie a literelor în textul cifrat mult mai uniformă, făcând atacurile bazate pe analiza frecvenţelor mult mai dificile. Se consideră că primul sistem de criptare de acest tip a fost realizat de Leon Battista Alberti în 1568. Chiar şi astăzi, unele sisteme moderne mai folosesc cifruri polialfabetice pentru anumite componente.

1.1. Algoritmul de criptare *Playfair*

1.1.1. Scurt istoric

Cu toate că poartă numele baronului Lyon Playfair, algoritmul a fost inventat de prietenul acestuia, Charles Wheatstone și descris pentru întâia dată într-un document la 26 martie 1854. La început a fost respins de "British Foreign Office" deoarece a fost considerat foarte greu de înțeles. Atunci când Wheatstone s-a oferit să demonstreze că în 15 minute va învăța să folosească algoritmul 3 băieți din 4 din școala aflată în apropiere, secretarul biroului de externe i-a răspuns: „Da, este foarte posibil, însă nu îi vei putea învăța să fie buni diplomați".

După crearea algoritmului, baronul Playfair a convins guvernul britanic să adopte acest algoritm pentru uz oficial și de aceea poartă numele său și nu al creatorului, Wheatstone. Algoritmul a fost utilizat de către armata britanică în războiul cu burii din Africa de Sud, iar versiuni modificate au fost folosite tot de britanici în primul război mondial cât și de armata australiană în cel de-al doilea război mondial.

Din punct de vedere al criptografiei moderne, algoritmul de criptare Playfair este unul învechit, chiar primitiv. Orice calculator personal modern poate găsi (sparge) cheia și descifra mesajul într-un interval de timp de câteva secunde sau chiar sutimi de secunde, folosind software-ul potrivit. Unii dintre cei mai iscusiți criptanaliști sau chiar unii experți în cuvinte încrucișate pot sparge mesajul criptat în câteva minute folosind doar un creion și o foaie de hârtie.

Cu toate că este un algoritm depășit din toate punctele de vedere, algoritmul Playfair este unul dintre primii algoritmi ce folosește principiile moderne ale cifrurilor bloc. Studierea acestui algoritm vă poate oferi o mai bună înțelegere intuitivă a criptografiei moderne fără a folosi cunoștințe complexe de matematică sau teoria numerelor.

1.1.2. Descrierea generală a algoritmului Playfair

Criptarea Playfair implică parcurgerea următorilor pași:

a) pregătirea textului ce urmează a fi criptat;
b) construirea matricei de criptare;
c) construirea mesajului criptat.

2. SARCINA

Sarcină 3.1. De implementat algoritmul Playfair în unul din limbajele de programare pentru mesaje în limba română (31 de litere). Valorile caracterelor textului sunt cuprinse între ‘A’ și ’Z’, ’a’ și ’z’ și nu sunt premise alte valori. În cazul în care utilizatorul introduce alte valori - i se va sugera diapazonul corect al caracterelor. Lungimea cheii nu trebuie să fie mai mică de 7. Utilizatorul va putea alege operația - *criptare* sau *decriptare*, va putea introduce *cheia, mesajul* sau *criptograma* și va obține *criptograma* sau *mesajul decriptat*. Faza finală de adăugare a spațiilor noi, în funcție de limba folosită și de logica mesajului – se va face manual.

3.MERSUL LUCRĂRII

Codul sarcinii este pe linkul:https://gist.github.com/MrAditzu/f7c4f597e5026d5a8e298d189363ad51

```
Alegeti c pentru Criptare sau d pentru Decriptare: c
Mesajul:
salut lume
Introduceti cheia alfabetica(minim 7 litere):
playfair
Matricea Playfair 5x6:
P L A Y F I
R Ă Â B C D
E G H J K M
N O Q S Ș T
Ț U V W X Z


Mesajul criptat: QYĂLOIZGKȚ
```

Figura 1. Execuție a programului

```python
def creare_matrice_playfair(cheie: str):
    cheie = cheie.upper()
    # eliminăm caractere nevalide și duplicate
    cheie_curata = ""
    for letter in cheie:
        if letter in alfabet and letter not in cheie_curata:
            cheie_curata += letter
    # completăm cu literele alfabetului român care nu sunt în cheie
    for letter in alfabet:
        if letter not in cheie_curata:
            cheie_curata += letter
    # construim matricea 5x6
    matrice = [[cheie_curata[i * matrix_cols + j] for j in range(matrix_c
    return matrice
```

Figura 2. Funcția de creare a matricei

Funcția prezentată creeeaza matricea 5x6 prin eliminarea caracterelor duplicate si completarea matricei cu alfabetul român.

```python
# afișăm matricea
def afisare_matrice(matrice):
    print("Matricea Playfair 5x6:")
    for row in matrice:
        print(" ".join(row))
    print("\n")
# găsim poziția unei litere în matrice
def gasire_locatie(matrice, char):
    for i in range(matrix_rows):
        for j in range(matrix_cols):
            if matrice[i][j] == char:
                return i, j
    return -1, -1
```

Figura 3. Funcțiile de afișare a matricei si găsire a pozitiilor

Funcția prezentată afișeaza matricea Playfair de 5x6 ,pe urmă caută litera in matrice și întoarce coordonatele ei,adica rândul(i) si coloana (j)

```python
# pregătim mesajul pentru criptare
def pregatire_mesaj(mesaj):
    msg = "".join(letter for letter in mesaj.upper() if letter in alfabet)
    i = 0
    rezultat = ""
    while i < len(msg):
        a = msg[i]
        if i + 1 < len(msg):
            b = msg[i + 1]
            if a == b:
                rezultat += a + inserare
                i += 1
            else:
                rezultat += a + b
                i += 2
        else:
            rezultat += a + inserare
            i += 1
    return rezultat
```

Figura 4.Funcția de pregătire a mesajului pentru a fi criptat sau decriptat

```python
# criptăm o pereche de litere în funcție de regulile Playfair
def criptare_pereche(matrice, m1, m2):
    r1, c1 = gasire_locatie(matrice, m1)
    r2, c2 = gasire_locatie(matrice, m2)

    if r1 == r2:
        # dacă sunt pe aceeași linie, mutăm la dreapta
        c1_new = (c1 + 1) % matrix_cols
        c2_new = (c2 + 1) % matrix_cols
        return matrice[r1][c1_new], matrice[r2][c2_new]

    elif c1 == c2:
        # dacă sunt pe aceeași coloană, mutăm în jos
        r1_new = (r1 + 1) % matrix_rows
        r2_new = (r2 + 1) % matrix_rows
        return matrice[r1_new][c1], matrice[r2_new][c2]

    else:
        # dacă sunt pe rânduri și coloane diferite, formăm un dreptunghi
        return matrice[r1][c2], matrice[r2][c1]
```

Figura 5. Functia criptare_pereche

Functia criptează fiecare pereche de literă conform regulilor standard Playfair în funcție de coloana și rândul pe care se află pereche de litere

```python
# decriptăm o pereche de litere în funcție de regulile Playfair
def decriptare_pereche(matrice, c1, c2):
    r1, col1 = gasire_locatie(matrice, c1)
    r2, col2 = gasire_locatie(matrice, c2)
    if r1 == r2:
        # Dacă sunt pe aceeași linie, mutăm la stânga
        return matrice[r1][(col1 - 1) % matrix_cols], matrice[r2][(col2 - 1) % matrix_cols]

    elif col1 == col2:
        # Dacă sunt pe aceeași coloană, mutăm în sus
        return matrice[(r1 - 1) % matrix_rows][col1], matrice[(r2 - 1) % matrix_rows][col2]
    else:
        # Dacă sunt pe rânduri și coloane diferite, formăm un dreptunghi
        return matrice[r1][col2], matrice[r2][col1]
```

Figura 6. Functia decriptare_pereche

Decriptarea are loc ca și criptarea doar ca este inversată direcția.

```python
# funcția principală pentru criptare și decriptare
def playfair_criptare(mesaj: str, cheie: str) -> str:
    matrice = creare_matrice_playfair(cheie)
    afisare_matrice(matrice)  # afișăm matricea
    mesaj = pregatire_mesaj(mesaj)
    criptograma = ""
    for i in range(0, len(mesaj), 2):
        a, b = mesaj[i], mesaj[i+1]
        c1, c2 = criptare_pereche(matrice, a, b)
        criptograma += c1 + c2
    return criptograma
def playfair_decriptare(criptograma: str, cheie: str) -> str:
    matrice = creare_matrice_playfair(cheie)
    afisare_matrice(matrice)
    mesaj = ""
    for i in range(0, len(criptograma), 2):
        a, b = criptograma[i], criptograma[i+1]
        m1, m2 = decriptare_pereche(matrice, a, b)
        mesaj += m1 + m2

    i = 0
    while i < len(mesaj) - 2:
        if mesaj[i] == mesaj[i+2] and mesaj[i+1] == inserare:
            mesaj = mesaj[:i+1] + mesaj[i+2:]
        i += 1

    if len(mesaj) > 0 and mesaj[-1] == inserare:
        mesaj = mesaj[:-1]

    return mesaj
```

Figura 7. Functiile principale de criptare si decriptare

La criptare se creează matricea ,se pregătește mesajul și il cripteaza pereche cu pereche. La decriptare creeaza aceeași matrice și decriptează fiecare pereche și elimină literele înserate intre duble.

CONCLUZIE

Din punct de vedere al criptografiei moderne, algoritmul de criptare Playfair este unul învechit, chiar primitiv. Orice calculator personal modern poate găsi (sparge) cheia și descifra mesajul într-un interval de timp de câteva secunde sau chiar sutimi de secunde, folosind software-ul potrivit. Cu toate că este un algoritm depășit din toate punctele de vedere, algoritmul Playfair este unul dintre primii algoritmi ce folosește principiile moderne ale cifrurilor bloc. Studierea acestui algoritm mi-a oferit o mai bună înțelegere intuitivă a criptografiei moderne fără a folosi cunoștințe complexe de matematică sau teoria numerelor.