

Отчёт

по лабораторной работе 2

Кочетов Андрей Владимирович

Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выводы	9

List of Figures

3.1	рис.1. Маршрутное шифрование	7
3.2	рис.2. Шифрование с помощью решеток	8
3.3	рис.3. Шифр Виженера	8

List of Tables

1 Цель работы

Реализовать маршрутное шифрование, шифрование с помощью решеток, шифр Виженера.

2 Задание

Лабораторная работа подразумевает написание программ на языке python, которые реализуют маршрутное шифрование, шифрование с помощью решеток, шифр Виженера.

3 Выполнение лабораторной работы

1. Реализовал маршрутное шифрование.

```
In [16]: import numpy as np
import random
import string

In [7]: def mar_shifr():
    m = 5
    n = 6
    text = "РЭИ лучший университет в России"
    text1 = text.upper()
    result = list(text1)

    for symbol in result:
        if (symbol==" ") or (symbol==".") or (symbol==",") or (symbol=="=") or (symbol=="?") or (symbol=="'") or (symbol=="'");
            index = result.index(symbol)
            element = result.pop(index)
            result1=result.copy()
            result2=[]

            for i in range (0,len(result1),n):
                result2.append(list(result1[i:n]))

            while (len(result2[m-1])<n):
                result2[m-1].append('A')

    text2="пароль"
    text2=text2.upper()
    password=list(text2)

    result3=list(result2)
    result3.append(password)
    alphabet="ABCFGDEKHIJKLMNOPQRSTUVWXYZ4567890"
    indices=[]

    for pas in password:
        for letter in alphabet:
            if pas==letter:
                ind=alphabet.find(letter)
                indices.append(ind)

    result4=list(result3)
    result4.append(indices)
    result5=np.array(result4)
    result6=result5[:,np.argsort(result5[-1,:])]
    result7=list(result6)

    del (result7[-1])
    return result7
```

Figure 3.1: рис.1. Маршрутное шифрование

2. Реализовал шифрование с помощью решеток.

```

In [15]: def resh():
    k = 2

    osnova=np.linspace(1,k**2,k**2)
    result = []
    for i in range(0,len(osnova),k):
        result.append(list(osnova[i:i+k]))

    def rot90(matrix):
        return [list(reversed(col)) for col in zip(*matrix)]

    matrix = np.full((2*k,2*k),0)

    matrix[:k,:k]=result

    result2=rot90(result)
    matrix[k:2*k,:k]=result2

    result3=rot90(result2)
    matrix[k:2*k,k:2*k]=result3

    result4=rot90(result3)
    matrix[k:2*k,k:2*k]=result4

    holes=[]
    for i in range (1,k**2+1):
        indexes=[]
        for m in range(0,2*k):
            for j in range(0,2*k):
                if matrix[m][j]==i:
                    coords=tuple([m,j])
                    indexes.append(coords)
            find=random.randint(0,1)
            holes.append(indexes[find])

    template = np.full((2*k,2*k),0)
    for d in range (k**2):
        template[holes[d][0],holes[d][1]] = 1

    template1 = rot90(template)
    template2 = rot90(template1)
    template3 = rot90(template2)
    text = "ДОГОВОРОПДПКАЛН"

    indexes1=[]
    for m1 in range (0,2*k):
        for j1 in range (0,2*k):
            if template1[m1][j1]==1:
                coords1=tuple([m1,j1])

```

Figure 3.2: рис.2. Шифрование с помощью решеток

3. Реализовал шифр Виженера.

```

In [17]: def vig(plaintext: str, keyword: str) -> str:
    letters = string.ascii_letters
    abc = letters[:len(letters)//2]
    ABC = letters[len(letters)//2:]

    while len(plaintext) > len(keyword):
        keyword += keyword
        keyword = keyword[:len(plaintext)].upper()

    ciphertext = ""
    for i in range(len(plaintext)):
        n = ABC.find(keyword[i])
        chiper_letters = abc[n:] + abc[:n] + ABC[n:] + ABC[:n]
        if plaintext[i] in letters:
            ciphertext += chiper_letters[letters.find(plaintext[i])]
        else:
            ciphertext += plaintext[i]
    return ciphertext

In [18]: vig("ATTACK", "LEMON")
Out[18]: 'LXFOPV'

In [ ]:

```

Figure 3.3: рис.3. Шифр Виженера

4 Выводы

Я написал программный код, который реализует маршрутное шифрование, шифрование с помощью решеток, шифр Виженера.