Лабораторная работа №2

Шифры перестановки

Румянцева Александра Сергеевна

30 сентября, 2022

Цели и задание на лабораторную работу

Цель работы: ознакомление с тремя методами шифрования: маршрутным шифрованием, шифрованием с помощью решеток, таблицей Виженера, а так же их реализация на произвольном языке программирования.

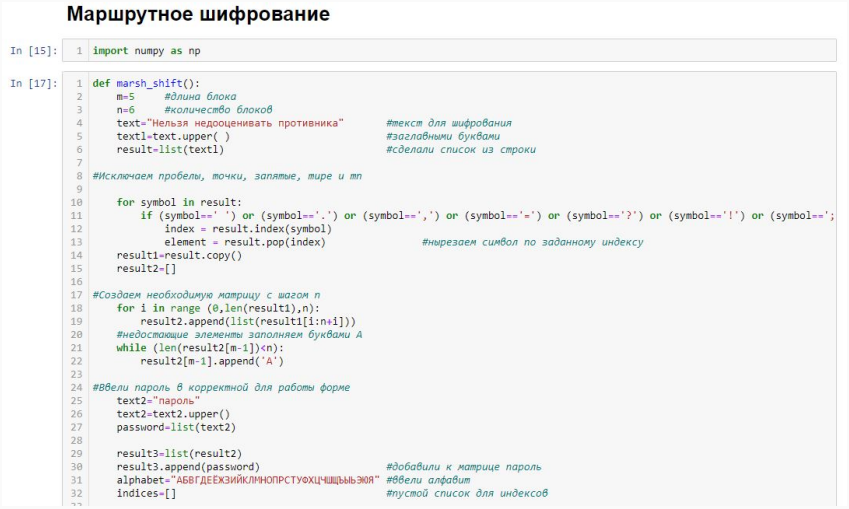
Задание:

1. Реализовать метод маршрутного шифрования.
2. Реализовать метод шифрования с помощью решеток.
3.

Выполнение лабораторной работы

1. Изучила теорию и указание к лабораторной работе.
2. Реализация кода для маршрутного шифрования.

Мною был написан код для шифрования текста методом маршрутного шифрования (рис. 1-2).



Маршрутное шифрование

In [15]:
```
import numpy as np
```

In [17]:
```
def marsh_shift():
    m=5      #длина блока
    n=6      #количество блоков
    text="Нельзя недооценивать противника"        #текст для шифрования
    text1=text.upper( )                           #заглавными буквами
    result=list(text1)                            #сделали список из строки

#Исключаем пробелы, точки, запятые, тире и тп

    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)               #вырезаем символ по заданному индексу
    result1=result.copy()
    result2=[]

#Создаем необходимую матрицу с шагом n
    for i in range (0,len(result1),n):
        result2.append(list(result1[i:n+i]))
    #недостающие элементы заполняем буквами А
    while (len(result2[m-1])<n):
        result2[m-1].append('А')

#Ввели пароль в корректной для работы форме
    text2="пароль"
    text2=text2.upper()
    password=list(text2)

    result3=list(result2)
    result3.append(password)                      #добавили к матрице пароль
    alphabet="АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ" #ввели алфавит
    indices=[]                                    #пустой список для индексов
```

Figure 1: рис. 1. Шифрование текста методом маршрутного шифрования 1.

```
#Смотрим на индексы пароля в алфавите
    for pas in password:
        for letter in alphabet:
            if pas==letter:
                ind=alphabet.find(letter)
                indices.append(ind)
    result4=list(result3)
    result4.append(indices)                  #добавили индексы в матрицу
    result5=np.array(result4)
    result6=result5[:,np.argsort(result5[-1,:]) ]#сортировка
    result7=list(result6)

#убрали две последние строки в матрице
    del (result7[-1] )
    del (result7[-1])
    result8=np.array(result7)
    result9=result8.transpose() #транспонирует для того чтобы выписать шифр
    result10=[]

#Начали работу над выписыванием шифра
    for i in range (n):
        result10.extend(result9[i])
    print("".join(result10))#выписали строку шифра

marsh_shift( )
```

```
ЕЕНПНЗОАТАЬОВОКННЕЬВЛДИРИЯЦТИА
```

Figure 2: рис. 2. Шифрование текста методом маршрутного шифрования 2.

3. Реализация кода для шифрования методом решеток.

Мною был написан код для шифрования текста методом решеток (рис. 3-6).

Метод решеток

In [18]:
```
import numpy as np
import random
```

In [23]:
```
def turning_grille():
    k=2       #вводим k

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

    #вводим функцию для поворота матрицы
    def rot90(matrix):
        return [list(reversed(col)) for col in zip(*matrix)]

    matrix=np.full((2*k,2*k),0)        #создали и заполнили нулями матрицу 2k x 2k

    #заполняем матрицу matrix по четвертям
        #1 четверть
    matrix[:k,:k]=result
        #2 четверть
    result2=rot90(result)
    matrix[:k,k:2*k]=result2
        #3 четверть
    result3=rot90(result2)
    matrix[k:2*k,k:2*k]=result3
        #4 четверть
    result4=rot90(result3)
    matrix[k:2*k,:k]=result4
```

Figure 3: рис. 3. Шифрование текста методом решеток 1.

В качестве параметров системы были взяты данные из описательной части лабораторной работы портала ТУИС.

```
#работа с отверстиями (определение координат)
holes=[]
for i in range (1,k**2+1):#прогонка по отдельному числу, например, по единичкам
    indexes=[ ]
    for m in range(0,2*k):#прогонка по строкам
        for j in range(0,2*k):#прогонка по столбцам
            if matrix[m][j]==i:
                coords=tuple([m, j])
                indexes.append(coords)
    find=random.randint(0,3)#выбираем 1 из 4 координат
    holes.append(indexes[find])

#работа с отверстиями (продолжение) визуализация поворотов и случаев размещений отверстий
template=np.full((2*k,2*k),0)
for d in range (k**2):
    template[holes[d][0],holes[d] [1]] =1
#1 поворот
template1=rot90(template)
#2 поворот
template2=rot90(template1)
#3 поворот
template3=rot90(template2)
text="ДОГОВОРПОДПИСАЛИ"

```

Figure 4: рис. 4. Шифрование текста методом решеток 2.

```
    #прогоняем templates для нахождения координат для заполнения буквами
    #1 поворот
    indexes1=[]
    for m1 in range (0,2*k):
        for j1 in range (0,2*k):
            if template1[m1][j1]==1:
                coords1=tuple([m1,j1])
                indexes1.append(coords1)
    #2 поворот
    indexes2=[]
    for m2 in range (0,2*k):
        for j2 in range (0,2*k):
            if template2[m2][j2]==1:
                coords2=tuple([m2,j2])
                indexes2.append(coords2)
    #3 поворот
    indexes3=[]
    for m3 in range (0,2*k):
        for j3 in range (0,2*k):
            if template3[m3][j3]==1:
                coords3=tuple([m3,j3])
                indexes3.append(coords3)

    #Переходим к образованию матрицы с буквами
    letters_matrix=np.full((2*k,2*k),'O')
    #0
    for d in range (k**2):
        letters_matrix[holes[d][0], holes[d][1]]=text[d]
    #1
    for d in range (k**2):
        letters_matrix[indexes1[d][0],indexes1[d][1]]=text[d+k**2]
    #2
    for d in range (k**2):
        letters_matrix[indexes2[d][0], indexes2[d][1]]=text[d+2*(k**2)]
    #3
    for d in range (k**2):
        letters_matrix[indexes3[d][0],indexes3[d][1]]=text[d+3*(k**2)]
#####
```

Figure 5: рис. 5. Шифрование текста методом решеток 3.

```
#####
    letter_matrix=list(letters_matrix)
    text2="шифр"
    text2=text2.upper()
    password=list(text2)
    letter_matrix.append(password)
    alphabet="АБВГДЕЁЖЗИЙКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ"
    indices=[]

    #Смотрим на индексы пароля в алфавите
    for pas in password:
        for letter in alphabet:
            if pas==letter:
                ind=alphabet.find(letter)
                indices.append(ind)
    letter_matrix.append(indices)
    letter_matrix=np.array(letter_matrix)
    letter_matrix=letter_matrix[:,np.argsort(letter_matrix[-1,:])]#упорядочили
    letter_matrix=list(letter_matrix)
    del (letter_matrix[-1])#убрали строку с индексами букв из пароля в алфавите
    del (letter_matrix[-1])#убрали строку с индексами букв из пароля в алфавите
    letter_matrix=np.array(letter_matrix)
    letter_matrix=letter_matrix.transpose()
    letter_matrix=list(letter_matrix)

#выводим ответ в виде строки
    result1=[]
    for i in range (2*k):
        result1.extend(letter_matrix[i])
    print("".join(result1))

turning_grille()
```

```
ОВОРДЛГПАПИОСДОИ
```

Figure 6: рис. 6. Шифрование текста методом решеток 4.

3. Реализация кода для шифрования методом Виженера.

Написан код для шифрования текста методом Виженера (рис. 7).

Шифр Виженера

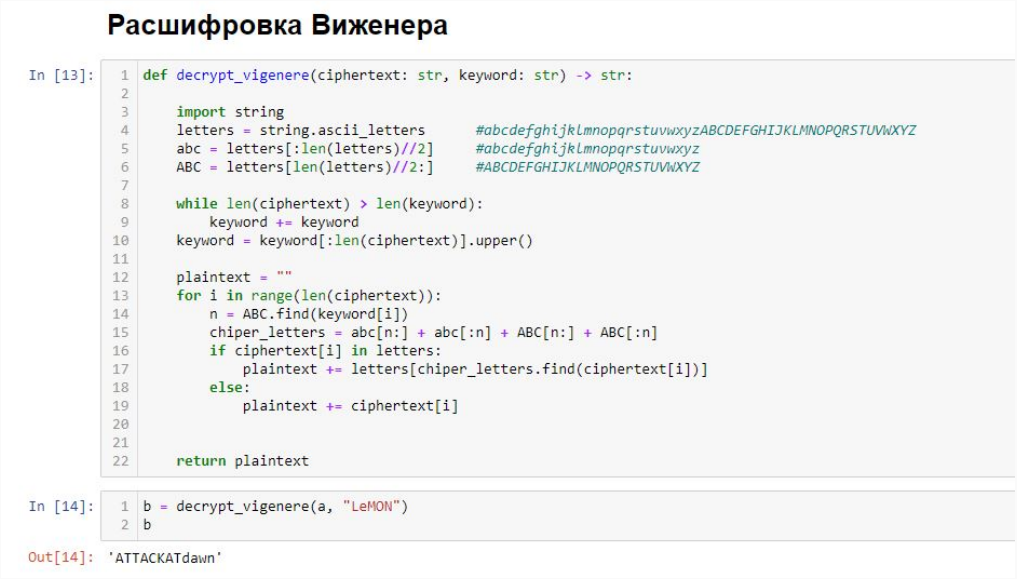
In [11]:
```
import string
def encrypt_vigenere(plaintext: str, keyword: str) -> str:
    letters = string.ascii_letters      #abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ
    abc = letters[:len(letters)//2]     #abcdefghijklmnopqrstuvwxyz
    ABC = letters[len(letters)//2:]     #ABCDEFGHIJKLMNOPQRSTUVWXYZ

    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 [12]:
```
a = encrypt_vigenere( "ATTACKATdawn", "LeMON")
a
```

Out[12]: 'LXFOPVEFrnhr'

Figure 7: рис. 7. Шифрование текста шифром Виженера.

Написан код для дешифрования текста методом Виженера (рис. 8).



Расшифровка Виженера

In [13]:
```
def decrypt_vigenere(ciphertext: str, keyword: str) -> str:

    import string
    letters = string.ascii_letters      #abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ
    abc = letters[:len(letters)//2]     #abcdefghijklmnopqrstuvwxyz
    ABC = letters[len(letters)//2:]     #ABCDEFGHIJKLMNOPQRSTUVWXYZ

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

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


    return plaintext
```

In [14]:
```
b = decrypt_vigenere(a, "LeMON")
b
```

Out[14]: 'ATTACKATdawn'

Figure 8: рис. 8. Деифрование рашиврованного шифром Виженера текста.

Выводы

Таким образом, была достигнута цель, поставленная в начале лабораторной работы: я ознакомилась с тремя методами шифрования – маршрутным шифрованием, шифрованием с помощью решеток, таблицей Виженера, а так же мне удалось реализовать их на языке программирования Python.