Лабораторная работа №1

Цель работы

Освоить на практике написание шифров перестановки. Использовать методы маршрутного шифрования, таблицу Виженера и шифрование решеток.

Задание

1. Реализовать маршрутное шифрование.
2. Реализовать шифрование решеток.
3. Реализовать Таблицу Виженера.

Теоретическое введение

Маршрутное шифрование — это метод симметричного **шифрования**, в котором элементы исходного открытого текста меняют местами. Элементами текста могут быть отдельные символы (самый распространённый случай), пары букв, тройки букв, комбинирование этих случаев и так далее. Типичными примерами перестановки являются анаграммы.. [1].

Шифровальная **решётка** — трафарет с прорезями-ячейками (из бумаги, картона или аналогичного материала), использовавшийся для шифрования открытого текста. Текст наносился на лист бумаги через такой трафарет по определённым правилам, и расшифровка текста была возможна только при наличии такого же трафарета.[3]

Шифр Виженера (фр. Chiffre de Vigenère) — метод полиалфавитного шифрования буквенного текста с использованием ключевого слова [2].

Оборудование

Лабораторная работа выполнялась дома со следующими характеристиками техники:

– Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHz 2.81GHz
– ОС Майкрософт Windows 10
– VirtualBox верс. 6.1.26

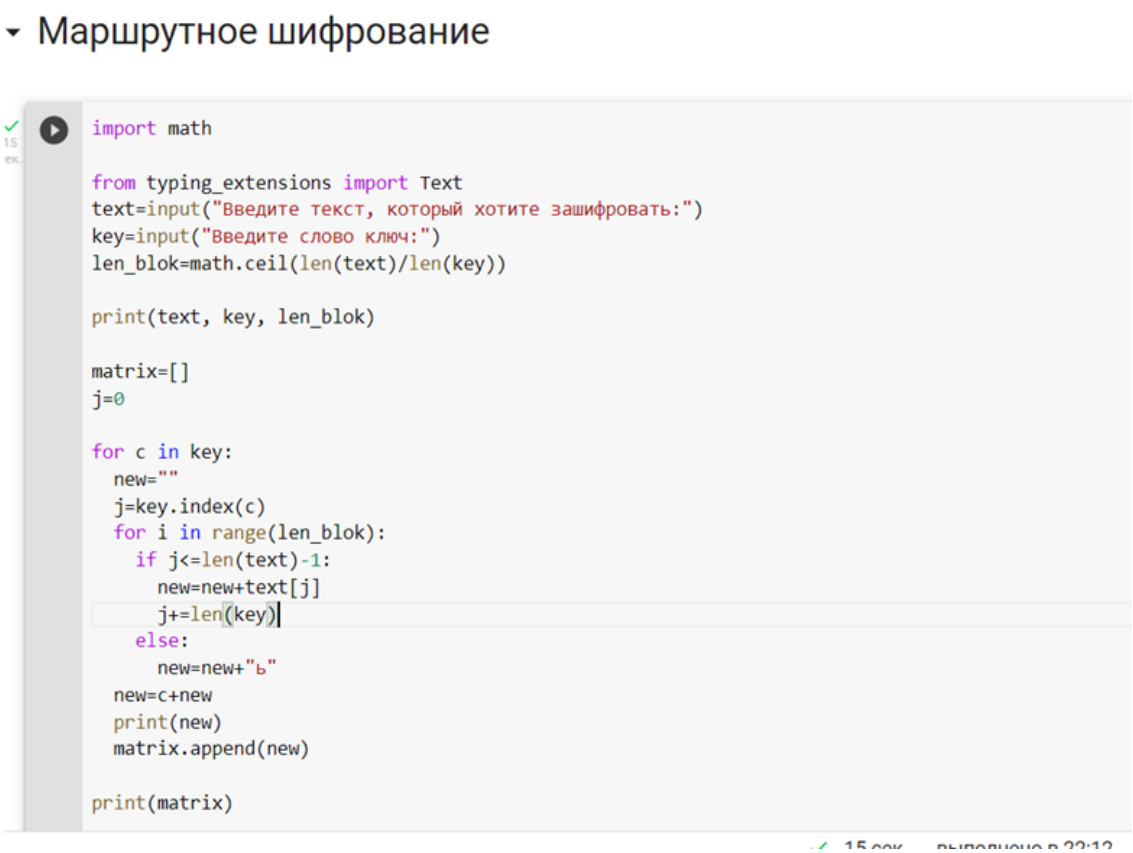
Код был написан на языке Python2.

Демонстрация работы кода проводилась в продукте Google Colaboratory.

Выполнение лабораторной работы

1. Реализовала маршрутное шифрование.

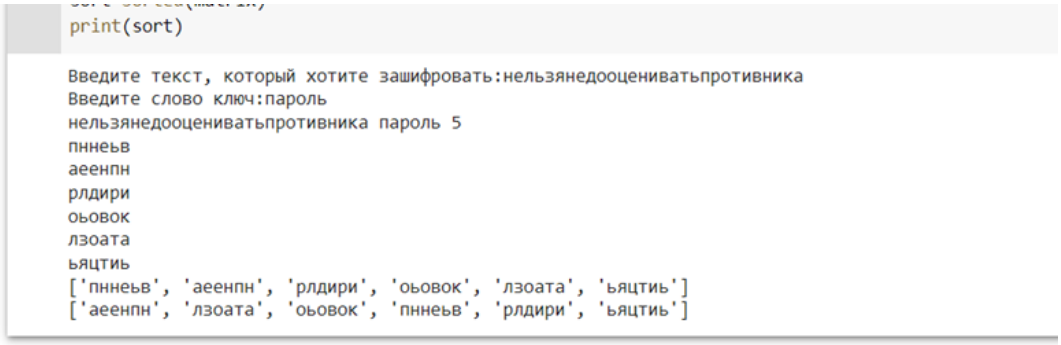
(рис. -@fig:001)



У нас есть текст: нельзя недооценивать противника, и ключ- "пароль". Текст мы используем без пробелов.

2. Показала работу шифра.

(рис. -@fig:002)



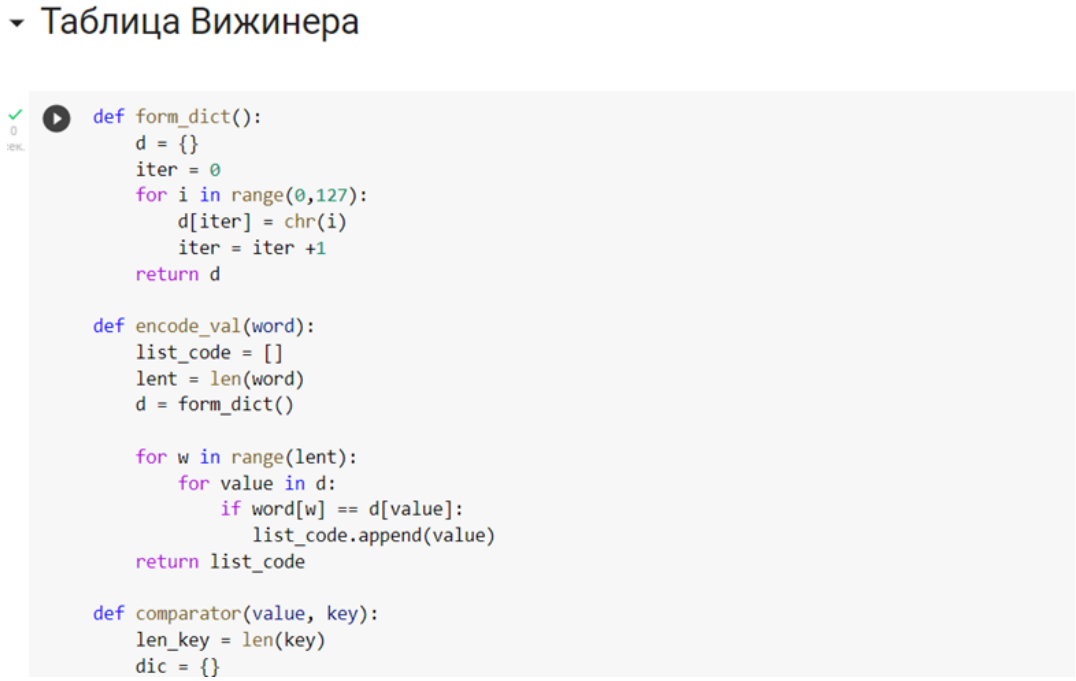
Если кратко пересказывать работу данного шифра, то текст разбивается на количество блоков = длине слова пароля. В данном случае это 6 блоков длиной в 5 символов. Чтобы все блоки были одинаковой длинны, мы заполняем оставшееся место мягким знаком. Далее, в первый блок мы записываем 0 символ текста и + каждый 6 символ. Во второй блок 1 и +6 символ и тд. К началу каждого блока приписываем одну букву слова пароля.

И в конце мы сортируем блоки по алфавиту, использую для этого первую букву блока - букву слова пароля.

На картинке видно, как в конце, после всех преобразований, мы получили шифротекст методом маршрутное шифрование.

3. Реализовала Шифр Виженера.

(рис. -@fig:003)



В этом случае наш текс - Hello world и ключ - paralel.

Также нам нужно создать словарь, для того чтобы шифровать символы. На скрине изображены 2 первые функции, с помощью которых можно представить тестовый символ английской раскладки в виде цифр (юникод раскладка поэтому 127 символов).

2. Зашифровала символы с помощью шифра Виженера.

(рис. -@fig:004)

```
def comparator(value, key):
    len_key = len(key)
    dic = {}
    iter = 0
    full = 0

    for i in value:
        dic[full] = [i,key[iter]]
        full = full + 1
        iter = iter +1
        if (iter >= len_key):
            iter = 0
    return dic

def full_encode(value, key):
    dic = comparator(value, key)
    print ('Compare full encode', dic)
    lis = []
    d = form_dict()

    for v in dic:
        go = (dic[v][0]+dic[v][1]) % len(d)
        lis.append(go)
    return lis
```

Далее мы к каждому символу и текста и слово-пароля подставляем цифру из нашего словаря. Комбинируем в новый словарь цифры шифра и цифры текста, повторяя заново цифры слово-пароля.

Далее мы суммируем цифру текста 72 с цифрой слова пароля 112 (72+112=184 % 127= 57), и делим на общее количество символов в словаря, оставляя остаток в качестве шифро-текста (57).

Проделываем данную операцию для всех символов.

2. Дешифровала символы.

(рис. -@fig:005)

```
    return lis

def decode_val(list_in):
    list_code = []
    lent = len(list_in)
    d = form_dict()

    for i in range(lent):
        for value in d:
            if list_in[i] == value:
                list_code.append(d[value])
    return list_code

def full_decode(value, key):
    dic = comparator(value, key)
    print ('Deshifre=', dic)
    d = form_dict()
    lis =[]

    for v in dic:
        go = (dic[v][0]-dic[v][1]+len(d)) % len(d)
        lis.append(go)
    return lis
```

Используя примерно те же действия для дешифровки символов, только цифры шифротекста - цифры слова пароля +127. Тем самым получаем первоначальный текст.

(рис. -@fig:006)

```
key_encoded = encode_val(key)
value_encoded = encode_val(word)

print ('Value= ',value_encoded)
print ('Key= ', key_encoded)

shifre = full_encode(value_encoded, key_encoded)
print ('Шифр=', ''.join(decode_val(shifre)))

decoded = full_decode(shifre, key_encoded)
print ('Decode list=', decoded)
decode_word_list = decode_val(decoded)
print ('Word=',''.join(decode_word_list))
```

```
Слово: Hello world
Ключ: paralel
Value=  [72, 101, 108, 108, 111, 32, 119, 111, 114, 108, 100]
Key=  [112, 97, 114, 97, 108, 101, 108]
Compare full encode {0: [72, 112], 1: [101, 97], 2: [108, 114], 3: [108, 97], 4: [111, 108], 5: [32, 101], 6: [119, 108], 7: [111, 112], 8:
Шифр= 9G_N\Bd`T_F
Deshifre= {0: [57, 112], 1: [71, 97], 2: [95, 114], 3: [78, 97], 4: [92, 108], 5: [6, 101], 6: [100, 108], 7: [96, 112], 8: [84, 97], 9: [95
Decode list= [72, 101, 108, 108, 111, 32, 119, 111, 114, 108, 100]
Word= Hello world
```

Выводы

В ходе данной лабораторной работы, были реализованы разные виды шифров перестановки.

Список литературы

1. Шифры перестановки// Хабр URL: https://habr.com/ru/post/583616/ (дата обращения: 22.09.2022).
2. Лабораторная работа 2. Шифры перестановки. // Туис URL: https://esystem.rudn.ru/pluginfile.php/1198312/mod_resource/content/2/007-lab_crypto-gamma.pdf (дата обращения:

28.09.2022).
3. Простейшие методы шифрования с симметричным ключом// НОУ ИНТУТ URL: https://intuit.ru/studies/courses/691/547/lecture/12373?page=4 (дата обращения: 29.09.2022).