# Отчёт по лабораторной работе №2

**Шифры перестановки**

Бакундукизе Эжид Принц НФИмд-01-21

# Содержание

**1 Цель работы** **4**

**2 Теоретические сведения** **5**
2.1 Шифры перестановки . . . . . 5
2.2 Маршрутное шифрование . . . . . 5
2.3 Шифрование с помощью решеток . . . . . 6
2.4 Шифр Виженера . . . . . 7

**3 Выполнение работы** **8**
3.1 Реализация шифра маршрутной перестановки на языке Python . 8
3.2 Реализация шифра решеткой на языке Python . . . . . 9
3.3 Реализация шифра Виженера на языке Python . . . . . 13
3.4 Контрольный пример . . . . . 17

**4 Выводы** **19**

**Список литературы** **20**

# List of Figures

3.1 Работа алгоритма маршрутной перестановки . . . . . 17
3.2 Работа алгоритма решетки . . . . . 18
3.3 Работа алгоритма Виженера . . . . . 18

# 1 Цель работы

Изучение алгоритмов шифров перестановки: маршрутное шифрование, шифрование с помощью решеток и таблица Виженера

# 2 Теоретические сведения

## 2.1 Шифры перестановки

Шифр перестано́вки — 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].

Как правило, при шифровании и дешифровании шифра простой перестановки используется таблица перестановок:

| | | | | |
|---|---|---|---|---|
| $1$ | $2$ | $3$ | ... | $n$ |
| $I_1$ | $I_2$ | $I_3$ | ... | $I_n$ |

Первая строка — позиция символа в открытом тексте, вторая строка — позиция в шифрограмме. Таким образом, при длине сообщения n символов существует ровно n! ключей.

## 2.2 Маршрутное шифрование

Наибольшее распространение получили маршрутные шифры перестановки, основанные на прямоугольниках (таблицах). Например, можно записать сообщение в прямоугольную таблицу по маршруту: по горизонтали, начиная с верхнего левого угла, поочередно слева направо. Сообщение будем списывать по маршруту: по вертикалям, начиная с верхнего правого угла, поочередно сверху вниз [1].

Шифрование перестановкой заключается в том, что текста переставляются по определенному правилу. Простейшим примеров перестановочного шифра являются так называемые «маршрутные перестановки», использующие некоторую геометрическую фигуру (плоскую или объемную). Шифрование заключается в том, что текст записывается в такую фигуру по некоторой траектории, а выписывается по другой траектории. Пример — маршрутные шифры перестановки, основанные на прямоугольниках (таблицах). Шифруемое сообщение в этом случае записывается в прямоугольную таблицу по маршруту: по горизонтали, начиная с верхнего левого угла, поочередно слева направо.

## 2.3 Шифрование с помощью решеток

Решётка Кардано — инструмент кодирования и декодирования, представляющий собой специальную прямоугольную (в частном случае — квадратную) таблицу-карточку, четверть ячеек которой вырезана.

Таблица накладывается на носитель, и в вырезанные ячейки вписываются буквы, составляющие сообщение. После переворачивания таблицы вдоль вертикальной оси, процесс вписывания букв повторяется. Затем то же самое происходит после переворачивания вдоль горизонтальной и снова вдоль вертикальной осей.

В частном случае квадратной таблицы, для получения новых позиций для вписывания букв, можно поворачивать квадрат на четверть оборота.

Чтобы прочитать закодированное сообщение, необходимо наложить решётку Кардано нужное число раз на закодированный текст и прочитать буквы, расположенные в вырезанных ячейках.

Такой способ шифрования сообщения был предложен математиком Джеролламо Кардано в 1550 году, за что и получил своё название [2].

## 2.4 Шифр Виженера

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

Этот метод является простой формой многоалфавитной замены. Шифр Виженера изобретался многократно. Впервые этот метод описал Джован Баттиста Беллазо (итал. Giovan Battista Bellaso) в книге La cifra del. Sig. Giovan Battista Bellaso в 1553 году, однако в XIX веке получил имя Блеза Виженера, французского дипломата. Метод прост для понимания и реализации, он является недоступным для простых методов криптоанализа [3].

В шифре Цезаря каждая буква алфавита сдвигается на несколько строк; например в шифре Цезаря при сдвиге +3, A стало бы D, B стало бы E и так далее. Шифр Виженера состоит из последовательности нескольких шифров Цезаря с различными значениями сдвига. Для зашифровывания может использоваться таблица алфавитов, называемая tabula recta или квадрат (таблица) Виженера. Применительно к латинскому алфавиту таблица Виженера составляется из строк по 26 символов, причём каждая следующая строка сдвигается на несколько позиций. Таким образом, в таблице получается 26 различных шифров Цезаря. На каждом этапе шифрования используются различные алфавиты, выбираемые в зависимости от символа ключевого слова.

# 3 Выполнение работы

## 3.1 Реализация шифра маршрутной перестановки на языке Python

```
def marhsrutshifr():
# задание 1. Маршрутное шифрование.
    text = input("Input anything").replace(' ', '')
# вводим текст и очищаем от пробелов
    n = int(input("Введите число n"))
# на сколько блоков надо разбить текст
    m = int(input("Введите число m"))
# на сколько блоков надо разбить текст
    parol = input("Введите слово-пароль")
# слово-пароль
    # мы не предусмотрели проверку на соотвествие
    lists = [['a' for i in range(0, n)] for j in range(m)]
#создаем матрицу nxm
    it = 0 #итератор
    for i in range(m):
        for j in range(n):
            if it < len(text):
                lists[i][j] = text[it]
            #заполняем текстом
```

```
            it += 1
    lis = list()
    for i in range(n):
        lis.append(parol[i])
    # добавляем пароль в список
    lists.append(lis)
    # а список в матрицу
    prrint(lists)
    result = ""
    # сюда будем записывать результат
    spisok = sorted(lists[len(lists) - 1])
    # сортируем по буквам пароль
    for i in spisok:
    # и согласно сортировке выписывает столбцами результат
        print(i, " = ", lists[len(lists)-1].index(i))
        for j in range(len(lists)):
            if j==len(lists)-1:
                continue
            result += lists[j][lists[len(lists)-1].index(i)]
    print(result)
```

## 3.2 Реализация шифра решеткой на языке Python

```
# функция для поворота матрицы. нужен для 2рого задания
def rot90(matrix):
    return[list(reversed(col)) for col in zip(*matrix)]

# функция удаления чисел из матрицы нужен для 2 рого задания
def udalenie(largelist, inn, k):
```

```
    for i in range(k * 2):
        for j in range(k * 2):
            if largelist[i][j] == inn:
                largelist[i][j] = " "
                return


def cardangrille(): # второе задания
    k = int(input("Введите число k")) # вводим наше число  k
    s=1
    lists = [[i for i in range(k)] for i in range(k)]
    # строим матрицу этого размера
    for i in range(k):
        for j in range(k):
            lists[i][j] = s # заполняем матрицу числами
            s += 1
    print(lists)
    lists1 = rot90(lists)
# заранее делаем повороты и сохраняем эти матрицы, чтоб потом приклеить друг к др
    lists2 = rot90(lists1)
    lists3 = rot90(lists2)
    largelist = [[1 for i in range(2*k)] for i in range(2*k)]
    # создаем большую матрицу, сюда будем клеить
    for i in range(k):
    # тут уже каждый цикл клеет все: 1 верхний кубик, 2 правый верхний кубик и тд
        for j in range(k):
            largelist[i][j] = lists[i][j]
    i1 = 0
    j1 = 0
```

```
for i in range(0, k):
    for j in range(k, k*2):
        largelist[i][j] = lists1[i1][j1]
        j1 += 1
    j1 = 0
    i1 += 1
i1 = 0
j1 = 0
for i in range(k, k*2):
    for j in range(k, k * 2):
        largelist[i][j] = lists2[i1][j1]
        j1 += 1
    j1 = 0
    i1 += 1
i1 = 0
j1 = 0
for i in range(k, k * 2):
    for j in range(0, k):
        largelist[i][j] = lists3[i1][j1]
        j1 += 1
    j1 = 0
    i1 += 1
prrint(largelist)
text = "договорподписали"
# текст который мы шифруем. его можно менять и даже нужно
largelist_a = [[" " for i in range(2*k)] for i in range(2*k)]
# тут вторая матрица из букв. Сюда будем вписывать буквы
s = 0
li = [i for i in range(1,k**2+1)]
```

```
# список из чисел, которые надо удалить
for inn in li:
    udalenie(largelist, inn, k)
# удаляем по очереди. Да, согласен алгоритм удаления такое себе))
ind = 0
# а тут уже "выписываем" буквы. Если текста еще есть то делаем повороты и тд
for i in range(k * 2):
    for j in range(k * 2):
        if largelist[i][j] == largelist_a[i][j] and len(text) > 0:
            largelist_a[i][j] = text[0]
            text = text[1:]
largelist = rot90(largelist)
for i in range(k * 2):
    for j in range(k * 2):
        if largelist[i][j] == largelist_a[i][j] and len(text) > 0:
            largelist_a[i][j] = text[0]
            text = text[1:]
if len(text) > 0:
    largelist = rot90(largelist)
    for i in range(k * 2):
        for j in range(k * 2):
            if largelist[i][j] == largelist_a[i][j] and len(text) > 0:
                largelist_a[i][j] = text[0]
                text = text[1:]
if len(text) > 0:
    largelist = rot90(largelist)
    for i in range(k * 2):
        for j in range(k * 2):
            if largelist[i][j] == largelist_a[i][j] and len(text) > 0:
```

```
                largelist_a[i][j] = text[0]
                text = text[1:]
    prrint(largelist_a)
    stri = input("Введите пароль")
    # тут дописываем пароль или удаляем чтоб длина слова была норм и потом прибав
    if len(stri) > k*2:
        stri = stri[:k*2]
    elif len(stri) < k*2:
        while len(stri) != k*2:
            stri += "z"
    largelist_a.append(list(stri))
    prrint(largelist_a)
    result = ""
    #фактически эта часть кода, такая же как в первом задании.
    spisok = sorted(largelist_a[len(largelist_a) - 1])
    for i in spisok:
        print(i, " = ", largelist_a[len(largelist_a) - 1].index(i))
        for j in range(len(largelist_a)):
            if j==len(largelist_a)-1:
                continue
            result += largelist_a[j][largelist_a[len(largelist_a) - 1].index(i)]
    print(result.replace(" ", ""))
```

## 3.3 Реализация шифра Виженера на языке Python

```
# Вижинер
# https://habr.com/ru/post/140820/

def form_dict():
```

```
    d = {}
    iter = 0
    for i in range(0,127):
        d[iter] = chr(i)
        iter = iter +1
    return d

def encode_val(word):
    list_code = []
    lent = len(word)
    d = form_dict()

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

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


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

def vijer():
    word = "Hello world"
    key = "key"
    sys.stdout.write(word)
    sys.stdout.write(key)
    key_encoded = encode_val(key)
    value_encoded = encode_val(word)
    sys.stdout.write(str(key_encoded))
    sys.stdout.write(str(value_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))
```

## 3.4 Контрольный пример

```
Введите текст: секретный текст
Введите число n: 3
Введите число m: 3
Введите слово-пароль: пароль
с е к
р е т
н ы й
п а р
а  =  1
п  =  0
р  =  2
ееысрнктй
```

Figure 3.1: Работа алгоритма маршрутной перестановки

```
Введите число k: 3
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
1 2 3 7 4 1
4 5 6 8 5 2
7 8 9 9 6 3
3 6 9 9 8 7
2 5 8 6 5 4
1 4 7 3 2 1
с е к р е т
  т н ы е к
    й с т



Введите пароль: пароль
с е к р е т
  т н ы е к
    й с т



п а р о л ь
а  =  1
л  =  4
о  =  3
п  =  0
р  =  2
ь  =  5
етеетрысскнйтк
```

Figure 3.2: Работа алгоритма решетки

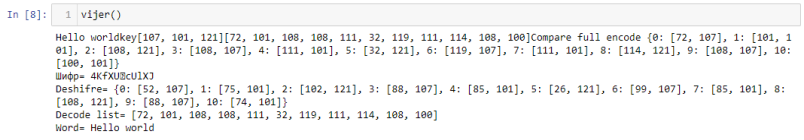


Figure 3.3: Работа алгоритма Виженера

# 4 Выводы

В ходе выполнения данной лабораторной работы мы изучили алгоритмы шифров перестановки: маршрутное шифрование, шифрование с помощью решеток и таблица Виженер. Реализовали данные методы шифрования программно и продемонстрировали результат.

# Список литературы


1. А. П. Алферов, А. Ю. Зубов, А. С. Кузьмин, А. В. Черёмушкин. Основы криптографии. — Гелиос АРВ, 2002.
2. Шифр Решетками
3. Шифр Виженера