

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ  
Факультет физико-математических и естественных наук  
Кафедра прикладной информатики и теории вероятностей

## Отчёт по лабораторной работе №2 Шифры перестановки

*Дисциплина: Математические основы защиты  
информации и информационной безопасности*

Студент: Леонова Алина Дмитриевна, 1032212306

Группа: НФИмд-01-21

Преподаватель: Кулябов Дмитрий Сергеевич,  
д-р.ф.-м.н., проф.

Москва 2021

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>7</b>
3.1	1. Маршрутное шифрование (Маршрутная перестановка) . . . . .	7
3.2	2. Шифрование с помощью решеток . . . . .	9
3.3	3. Таблица Виженера . . . . .	10
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>12</b>
4.1	0. Импортирование библиотек и промежуточные функции . . . . .	12
4.2	1. Маршрутное шифрование (Маршрутная перестановка) . . . . .	13
4.3	2. Шифрование с помощью решеток . . . . .	15
4.4	3. Таблица Виженера . . . . .	21
<b>5</b>	<b>Выводы</b>	<b>24</b>
	<b>Список литературы</b>	<b>25</b>

# List of Figures

3.1	Пример шифр перестановки (1)	8
3.2	Пример шифр перестановки (2)	8
3.3	Пример шифрования с помощью решетки (1)	9
3.4	Пример шифрования с помощью решетки (2)	9
3.5	Пример использования таблицы Виженера	11
4.1	Результат выполнения (1)	15
4.2	Результат выполнения (2)	20
4.3	Результат выполнения (3)	23

## List of Tables

# 1 Цель работы

Целью данной работы является ознакомление с шифрами перестановки и их реализация на выбранном языке программирования.

## 2 Задание

Реализовать все рассмотренные шифры программно.

## **3 Теоретическое введение**

Шифры перестановки преобразуют открытый текст в криптограмму путем перестановки его символов. Способ, каким при шифровании переставляются буквы открытого текста, и является ключем шифра.

### **3.1 1. Маршрутное шифрование (Маршрутная перестановка)**

Маршрутная перестановка — это шифр вертикальной перестановки, метод симметричного шифрования, в котором элементы исходного открытого текста меняют местами [1].

ОТКРЫТЫЙ ТЕКСТ: пример маршрутной перестановки  
 КЛЮЧ: (3, 1, 4, 2, 5)

3	1	4	2	5
п	р	и	м	е
р	м	а	р	ш
р	у	т	н	о
й	п	е	р	е
с	т	а	н	о
в	к	и		

КРИПТОГРАММА: рмупткмрнрнпррйсвиатеаиешоео

Figure 3.1: Пример шифр перестановки (1)

н	е	л	ь	з	я
н	е	д	о	о	ц
е	н	и	в	а	т
ь	п	р	о	т	и
в	н	и	к	а	а
<hr/>					
п	а	р	о	л	ь

Figure 3.2: Пример шифр перестановки (2)

В этом шифре также используется прямоугольная таблица, в которую сообщение записывается по строкам слева направо. Выписывается шифрограмма по вертикалям, при этом столбцы выбираются в порядке, определяемом ключом



(см. рис. 3.1). Также возможна вариация, когда ключём служит пароль и есть договорённость как его использовать, например, в алфавитном порядке (см. рис. 3.2).

## 3.2 2. Шифрование с помощью решеток

Выбирается натуральное число  $k > 1$ , и квадрат размерности  $k \times k$  построчно заполняется числами  $1, 2, \dots, k$ . Для примера возьмем  $k = 2$ .

1	2	3	1
3	4	4	2
2	4	4	3
1	3	2	1

Figure 3.3: Пример шифрования с помощью решетки (1)

Квадрат поворачивается по часовой стрелке на  $90^\circ$  и размещается вплотную к предыдущему квадрату. Аналогичные действия совершаются еще два раза, так чтобы в результате из четырех малых квадратов образовался один большой с длиной стороны  $2k$  (см. рис. 3.3).

			Д				Д		с	о	а	д
					в				д	в	п	л
	о		г	о	о		г	о	о	и	г	
		о			р	о	п	и	р	о	п	
									ш	и	ф	р

Figure 3.4: Пример шифрования с помощью решетки (2)

Далее из большого квадрата вырезаются клетки с числами от 1 до  $k^2$ , для каж-

дого числа одна клетка. Процесс шифрования происходит следующим образом. Сделанная решетка (квадрат с прорезями) накладывается на чистый квадрат  $2k \times 2k$  и в прорези по строчкам (т.е. слева направо и сверху вниз) вписываются первые буквы открытого текста. Затем решетка поворачивается на  $90^\circ$  по часовой стрелке и накладывается на частично заполненный квадрат, вписывание продолжается.

После третьего поворота, наложения и вписывания все клетки квадрата будут заполнены. Например, с использованием изображенной выше решетки и пароля ш и ф р открытый текст договор подписали переводится в криптограмму за пять шагов, итоговая криптограмма: ОВОРДЛГПАПИОСДОИ (см. рис. 3.4) [2].

### **3.3 3. Таблица Виженера**

Шифр Виженера — метод полиалфавитного шифрования буквенного текста с использованием ключевого слова. Этот метод является простой формой многоалфавитной замены [3].

м	а	т	е	м	а	т	и	к	а	м	а	т	е	м	а	т	и	к	а	м	а	т	е	м	а
к	р	и	п	т	о	г	р	а	ф	и	я	с	е	р	ь	е	з	н	а	я	н	а	у	к	а

А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Э	Ю	Я
Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Э	Ю	Я	А
В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Э	Ю	Я	А	Б
Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Э	Ю	Я	А	Б	В
Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Э	Ю	Я	А	Б	В	Г
Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Э	Ю	Я	А	Б	В	Г	Д
Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Э	Ю	Я	А	Б	В	Г	Д	Е
З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж
И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З
Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И
К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й
Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К
М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л
Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М
О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н
П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О
Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П
С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р
Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С
У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т
Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У
Х	Ц	Ч	Ш	Щ	Ъ	Ы	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф
Ц	Ч	Ш	Щ	Ъ	Ы	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х
Ч	Ш	Щ	Ъ	Ы	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц
Ш	Щ	Ъ	Ы	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч
Щ	Ъ	Ы	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш
Ъ	Ы	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ
Ы	Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ
Э	Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы
Ю	Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Э
Я	А	Б	В	Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Э	Ю

В горизонтальном алфавите находим букву «к», а в вертикальном – букву «м». На пересечении столбца и строки в таблице расположена буква «ц». Далее переходим к буквам «р» и «а» соответственно. В итоге получается следующая криптограмма: ЦРЬФЯОХШКФЯДКЭЪЧПЧАЛНТШЩА.

Figure 3.5: Пример использования таблицы Виженера

Пароль записывается с повторениями пока не поровняется по длине с текстом, который требуется зашифровать. Для шифрования и дешифрования используется таблица Виженера, она содержит все возможные смещения интересующего алфавита (см. рис. 3.5).

## 4 Выполнение лабораторной работы

### 4.1 0. Импортирование библиотек и промежуточные функции

Для первого и второго шифров в конце требуется делать одно и то же действие:

```
import numpy as np
import math

def grid_reading(key, table):
    _key = sorted(key)
    order = [key.index(i) for i in _key]
    print(list(key))
    #print(_key)
    print('Порядок использования столбцов:', order)
    m = table.shape[0]
    res = ''
    for j in order:
        for i in range(m):
            res += table[i][j]
    return res

ru = [chr(i) for i in range(ord('a'), ord('я')+1)]
```

```
print('ru: ',ru)
```

## 4.2 1. Маршрутное шифрование (Маршрутная перестановка)

```
print('#####')
# 1. Маршрутное шифрование
print('# 1.')
def task1_rout(text, key):
    print('Текст: ', text)
    text = text.replace(' ', '').lower()

    print('Ключ:', key)
    key = key.lower()

    n = len(key)
    t_len = len(text)
    m = math.ceil(t_len/n)
    A = np.full((m,n), '')
    print('Длина текста:', t_len)
    print('n = ', n, '\nm = ', m)

    for k in range(n*m - t_len):
        text += text[-1]
    #print(text)

    k = 0
    for i in range(m):
        for j in range(n):
```

```

        A[i][j] = text[k]
        k += 1
    print(A)

    res = grid_reading(key, A)

    print('Криптограмма: ', res)

text = 'Нельзя недооценивать противника'
key = 'пароль'
task1_rout(text, key)

print('-----')
text = 'Live long and prosper'
key = 'Spock'
task1_rout(text, key)

```

```

ru: ['а', 'б', 'в', 'г', 'д', 'е', 'ж', 'з', 'и', 'й', 'к', 'л',
     'м', 'н', 'о', 'п', 'р', 'с', 'т', 'у', 'ф', 'х', 'ц', 'ч', 'ш', 'щ',
     'ъ', 'ы', 'ь', 'э', 'ю', 'я']
#####
# 1.
Текст: Нельзя недооценивать противника
Ключ: пароль
Длина текста: 29
n = 6
m = 5
[['н', 'е', 'л', 'ь', 'з', 'я']
 ['н', 'е', 'д', 'о', 'о', 'ц']
 ['е', 'н', 'и', 'в', 'а', 'т']
 ['ь', 'п', 'р', 'о', 'т', 'и']
 ['в', 'н', 'и', 'к', 'а', 'а']]
['п', 'а', 'р', 'о', 'л', 'ь']
Порядок использования столбцов: [1, 4, 3, 0, 2, 5]
Криптограмма: еенпнзоатаьовокннееьвдиряцтиа
-----
Текст: Live long and prosper
Ключ: Spock
Длина текста: 18
n = 5
m = 4
[['l', 'i', 'v', 'e', 'l']
 ['o', 'n', 'g', 'a', 'n']
 ['d', 'p', 'r', 'o', 's']
 ['p', 'e', 'r', 'r', 'r']]
['s', 'p', 'o', 'c', 'k']
Порядок использования столбцов: [3, 4, 2, 1, 0]
Криптограмма: eaorlnsrvgrrinpelodp
#####

```

Figure 4.1: Результат выполнения (1)

Результат выполнения этого фрагмента кода, реализации маршрутного шифрования и проверкой (см. рис. 4.1).

## 4.3 2. Шифрование с помощью решеток

```

print('#####')
# 2. Шифрование с помощью решеток

```

```

def rotate_90r(A):
    x = A.shape[0]
    y = A.shape[1]
    res = np.empty((y,x))
    for i in range(x):
        for j in range(y):
            res[j, x-1-i] = A[i,j]
    return res

def generate_key(lenth):
    key = ''
    while (len(key) != lenth):
        _key = np.random.randint(len(ru))
        if (key.count(ru[_key]) == 0):
            key += ru[_key]
    return key

print('# 2.')
def task2_grid(text):
    print('Текст: ', text)
    text = text.replace(' ', '').lower()

    t_len = len(text)
    print('Длина текста:', t_len)

    size = math.ceil(np.sqrt(t_len))    # количество чисел в маленькой таблице
    while size % np.sqrt(size) != 0:

```



```

    size += 1

for k in range(size*size - t_len):
    text += text[-1]

t_s = int(np.sqrt(size))    # размер маленькой таблицы
t_s2 = t_s * 2             # размер большой таблицы
t_el = np.arange(1,size+1)  # массив значений size

key = generate_key(t_s2)
#key = 'шифр'
print('Ключ:', key)
key = key.lower()

A0 = np.empty((t_s,t_s))
n = 0
for i in range(t_s):
    for j in range(t_s):
        A0[i][j] = t_el[n]
        n += 1

A1 = np.concatenate((A0,rotate_90r(A0)),axis=1)
A2 = np.concatenate((A1,rotate_90r(rotate_90r(A1))),axis=0)
#print(A0)
#print(A1)
print('Квадрат цифр:\n', A2)

R = np.zeros((t_s2,t_s2))

```

```

tmp = t_el.copy()
for k in range (size):
    r = np.random.randint(4)
    tmp_count = 0
    for i in range (t_s2):
        for j in range (t_s2):
            if (A2[i][j] == k + 1):
                if(tmp_count == r):
                    R[i][j] = A2[i][j]
                    tmp_count = -100
                    tmp = tmp[tmp != k+1]
                else:
                    tmp_count += 1
print('Сгенерированная схема:\n', R)

n = 0
answer = np.full((t_s2, t_s2), '')

for k in range(4):
    for i in range(t_s2):
        for j in range(t_s2):
            if(R[i][j] != 0):
                answer[i][j] = text[n]
                n += 1

#print(k)
#print(R)
#print(answer)

R = rotate_90r(R)

```

```
print('Схема из букв:\n', answer)

res = grid_reading(key, answer)
print('Криптограмма: ', res)

text = 'договор подписали'
task2_grid(text)

print('-----')

text = 'за что мне все эти страдания'
task2_grid(text)
```



Результат выполнения этого фрагмента кода, реализации шифрования с помощью решеток с проверкой (см. рис. 4.2).

## 4.4 3. Таблица Виженера

```
print('#####')
letters = {ru[i]:i for i in range(len(ru))}
print('Словарь букв ru: ', letters)

# 3. Таблица Виженера
print('# 3.')

vigenere_table = np.array(ru)
for i in range(1, len(ru)):
    row = np.roll(ru, -i)
    vigenere_table = np.vstack((vigenere_table, row))
print('Таблица Виженера: \n',vigenere_table)

def task3_vigenere(text, key):
    print('Текст: ', text)
    text = text.replace(' ', '').lower()

    t_len = len(text)
    print('Длина текста:', t_len)

    print('Ключ:', key)
    key = key.lower()

    n = len(key)
```

```

_key = key

while len(_key) < t_len:
    _key += _key[len(_key) - n]
print('-----')
print(text)
print(_key)
print('-----')

res = ''
for i in range(t_len):
    x = letters[_key[i]]    # номера букв ключа
    y = letters[text[i]]   # номера букв текста

    res += vigenere_table[x][y]
print('Криптограмма: ', res)

text = 'криптография серьезная наука'
key = 'математика'
task3_vigenere(text, key)

print('-----')
text = 'ты не пройдешь'
key = 'Гендальф'
task3_vigenere(text, key)

```

```
#####
Словарь букв ru: {'а': 0, 'б': 1, 'в': 2, 'г': 3, 'д': 4, 'е': 5,
'ж': 6, 'з': 7, 'и': 8, 'й': 9, 'к': 10, 'л': 11, 'м': 12, 'н': 13,
'о': 14, 'п': 15, 'р': 16, 'с': 17, 'т': 18, 'у': 19, 'ф': 20, 'х':
21, 'ц': 22, 'ч': 23, 'ш': 24, 'щ': 25, 'ъ': 26, 'ы': 27, 'ь': 28,
'э': 29, 'ю': 30, 'я': 31}
# 3.
Таблица Виженера:
[['а' 'б' 'в' ... 'э' 'ю' 'я']
 ['б' 'в' 'г' ... 'ю' 'я' 'а']
 ['в' 'г' 'д' ... 'я' 'а' 'б']
 ...
 ['э' 'ю' 'я' ... 'ъ' 'ы' 'ь']
 ['ю' 'я' 'а' ... 'ы' 'ь' 'э']
 ['я' 'а' 'б' ... 'ь' 'э' 'ю']]
Текст: криптография серьезная наука
Длина текста: 26
Ключ: математика
-----
криптографиясерьезнаянаука
математикаматематикаматема
-----
Криптограмма: црѣфюхшкфягкьъчпчалнтшца
-----
Текст: ты не пройдешь
Длина текста: 12
Ключ: Гендальф
-----
тынепройдешь
гендальфгенд
-----
Криптограмма: хаъйпыкэзкеа
```

Figure 4.3: Результат выполнения (3)

Результат выполнения этого фрагмента кода, реализации использования таблицы Виженера с проверкой (см. рис. 4.3).

## 5 Выводы

Цель лабораторной работы была достигнута, три данные шифра перестановки были изучены и реализованы на языке программирования Python.



## Список литературы

1. NeverWalkAloner. Перестановочный шифр [Электронный ресурс]. Википедия, 2021. URL: [https://ru.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D0%B5%D1%81%D1%82%D0%B0%D0%BD%D0%BE%D0%B2%D0%BE%D1%87%D0%BD%D1%8B%D0%B9\\_%D1%88%D0%B8%D1%84%D1%80](https://ru.wikipedia.org/wiki/%D0%9F%D0%B5%D1%80%D0%B5%D1%81%D1%82%D0%B0%D0%BD%D0%BE%D0%B2%D0%BE%D1%87%D0%BD%D1%8B%D0%B9_%D1%88%D0%B8%D1%84%D1%80).
2. Перестановочные шифры. [Электронный ресурс]. ИТ1406: Информационная безопасность, 2021. URL: <https://it.rfei.ru/course/~k017/~7mdCpor7/~c5kOtaHY>.
3. Шифр Виженера [Электронный ресурс]. Википедия, 2021. URL: [https://ru.wikipedia.org/wiki/%D0%A8%D0%B8%D1%84%D1%80\\_%D0%92%D0%B8%D0%B6%D0%B5%D0%BD%D0%B5%D1%80%D0%B0](https://ru.wikipedia.org/wiki/%D0%A8%D0%B8%D1%84%D1%80_%D0%92%D0%B8%D0%B6%D0%B5%D0%BD%D0%B5%D1%80%D0%B0).