

# **Отчет по лабораторной работе №2**

**Дисциплина: Математические основы защиты информации и  
информационной безопасности**

Живцова Анна

# 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>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>9</b>
4.1	Маршрутное шифрование . . . . .	9
4.2	Шифрование с помощью решеток . . . . .	10
4.3	Шифр Вижнера. Тип 1 . . . . .	12
4.4	Шифр Вижнера. Тип 2 . . . . .	13
<b>5</b>	<b>Выводы</b>	<b>14</b>
	<b>Список литературы</b>	<b>15</b>

## Список иллюстраций

4.1	Тестирование маршрутного шифрования . . . . .	9
4.2	Тестирование шифрования с помощью решеток . . . . .	11
4.3	Тестирование шифрования с помощью решеток . . . . .	12
4.4	Тестирование шифрования Виженера типа 1 . . . . .	13
4.5	Тестирование шифрования Виженера типа 2 . . . . .	13

## **Список таблиц**

# 1 Цель работы

Изучить алгоритмы шифрования с помощью перестановки. Реализовать шифрование с помощью маршрутов, шифрование с помощью решеток, и шифрование Виженера.

## 2 Задание

Реализовать шифрование с помощью маршрутов, шифрование с помощью решеток, и шифрование Виженера.

### 3 Теоретическое введение

Шифры перестановки преобразуют открытый текст в криптограмму путем перестановки его символов. Подробнее в источниках [1,2].

Маршрутное шифрование реализуется следующим образом. Исходный текст построчно записывается в таблицу размера  $n \times m$ . При недостатке символов исходного текста свободные места в таблице заполняются произвольными символами. В последнюю  $n + 1$  строчку таблицы записывается пароль. Символы из таблицы считываются по столбцам, отсортированным по последнему элементу (букве пароля), формируя зашифрованный текст.

Шифрование с помощью решеток. Таблица размера  $k \times k$  заполняется различными числами от 1 до  $k^2$ . Далее эту таблицу три раза поворачивают на 90 градусов и из полученных четырех таблиц, отличающихся только поворотом, формируют таблицу размером  $2k \times 2k$ . Из этой таблицы случайно выбирается  $k^2$  различных чисел. Исходный текст записывается в пустую таблицу в ячейки, которые соответствуют выбранным  $k^2$  числам. Далее таблица  $2k \times 2k$  переворачивается. Операция повторяется еще три раза. К полученной заполненной таблице в последней строке приписывается пароль. Символы из таблицы считываются по столбцам, отсортированным по последнему элементу (букве пароля), формируя зашифрованный текст.

Шифр Виженера. Тип 1. Исходный текст разбивается на блоки длины  $n$  (если символов не хватает, то дописываются произвольные). Каждый блок циклически сдвигается по алфавиту на величину  $a_i$ .

Шифр Виженера. Тип 2. Задана таблица в которой каждая строчка  $i$  представ-

ляет собой алфавит циклически сдвинутый на  $i$  позиций. Посредством циклического повторения пароля формируется кодовое слово  $key$ , по длине равное исходному тексту  $text$ . Символ под номером  $i$  исходного текста шифруется с помощью символа таблицы, стоящего в столбце, начинающемся на символ  $key[i]$ , и строке, начинающейся на символ  $text[i]$ .



## 4 Выполнение лабораторной работы

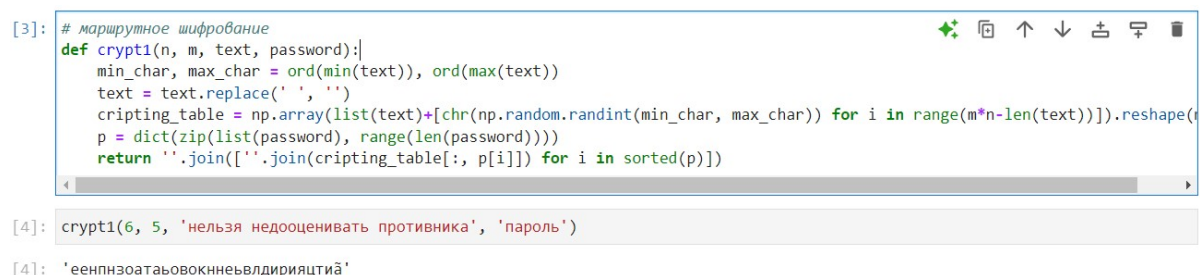
### 4.1 Маршрутное шифрование

Для реализации маршрутного шифрования на языке Python была написана следующая функция.

```
def crypt1(n, m, text, password):  
    min_char, max_char = ord(min(text)), ord(max(text))  
    text = text.replace(' ', '')  
    cripting_table = np.array(list(text)+[chr(np.random.randint(min_char, max_char)) for i in range(m*n-len(text))]).reshape((n, m))  
    p = dict(zip(list(password), range(len(password))))  
    return ''.join([''.join(cripting_table[:, p[i]]) for i in sorted(p)])
```

Тут  $n$  и  $m$  – размеры кодовой таблицы. Переменная *password* отвечает за пароль, а переменная *text* за исходный текст.

Функциональность данной функции была протестирована в среде jupyter notebook (см. рис. 4.1).



```
[3]: # маршрутное шифрование  
def crypt1(n, m, text, password):  
    min_char, max_char = ord(min(text)), ord(max(text))  
    text = text.replace(' ', '')  
    cripting_table = np.array(list(text)+[chr(np.random.randint(min_char, max_char)) for i in range(m*n-len(text))]).reshape((n, m))  
    p = dict(zip(list(password), range(len(password))))  
    return ''.join([''.join(cripting_table[:, p[i]]) for i in sorted(p)])  
  
[4]: crypt1(6, 5, 'нельзя недооценивать противника', 'пароль')  
  
[4]: 'ееппнзоатъаьовкннеъввдиряцтиâ'
```

Рис. 4.1: Тестирование маршрутного шифрования

## 4.2 Шифрование с помощью решеток

Реализовано с помощью функции

```
def crypt2(k, text, password):
    text = text.replace(' ', '')
    text = np.array(list(text)+[chr(np.random.randint(min_char, max_char)) for i in range(2*k)])
    base_square = np.arange(1, k**2 + 1).reshape(k, k)
    square = base_square.copy()
    for i in range(3):
        base_square = np.array([i[::-1] for i in base_square.transpose()])
        square = np.concatenate([square, base_square], axis=1)
    square = np.concatenate([square[:, :2*k], np.concatenate([square[:, 3*k:], square[:, 4*k:]])], axis=1)
    print('Числовой блок из 4 поворотов', '\n', square)
    bool_matrix = np.zeros((2*k, 2*k))
    for i in range(1, 2*k+1):
        ind = np.random.randint(0, 3)
        bool_matrix[np.where(square == i)[0][ind], np.where(square == i)[1][ind]] = 1
    print('Выбираем k различных чисел', '\n', bool_matrix)
    text_matrix = np.array([[ '0' for j in range(2*k)] for i in range(2*k)])
    counter = 0
    for i in range(4):
        for j in range(2*k):
            for m in range(2*k):
                if bool_matrix[j, m]:
                    text_matrix[j][m] = text[counter]
                    counter += 1
    print('записываем текст через решето, итерация ', i, '\n', text_matrix)
    bool_matrix = np.array([m[::-1] for m in bool_matrix.transpose()])
    p = dict(zip(list(password[:2*k]), range(len(password[:2*k]))))
```

```
return ''.join([''.join(text_matrix[:, p[i]]) for i in sorted(p)])
```

Функция была протестирована функцией (см. рис. 4.2).

```
[41]: # шифрование с помощью решеток
def crypt2(k, text, password):
    text = text.replace(' ', '')
    text = np.array(list(text) + [chr(np.random.randint(min_char, max_char)) for i in range(k*k-len(text))])
    base_square = np.arange(1, k**2 + 1).reshape(k, k)
    square = base_square.copy()
    for i in range(3):
        base_square = np.array([i[::-1] for i in base_square.transpose()])
        square = np.concatenate([square, base_square], axis=1)
    square = np.concatenate([square[:, :2*k], np.concatenate([square[:, 3*k:], square[:, 2*k:3*k]], axis = 1)])
    print('Числовой блок из 4 поворотов', '\n', square)
    bool_matrix = np.zeros((2*k, 2*k))
    for i in range(1, 2*k+1):
        ind = np.random.randint(0, 3)
        bool_matrix[np.where(square == i)[0][ind], np.where(square == i)[1][ind]] = 1
    print('Выбираем k различных чисел', '\n', bool_matrix)
    text_matrix = np.array(['0' for j in range(2*k)] for i in range(2*k))
    counter = 0
    for i in range(4):
        for j in range(2*k):
            for m in range(2*k):
                if bool_matrix[j, m]:
                    text_matrix[j][m] = text[counter]
                    counter += 1
        print('записываем текст через решето, итерация ', i, '\n', text_matrix)
        bool_matrix = np.array([m[::-1] for m in bool_matrix.transpose()])
    p = dict(zip(list(password[:2*k]), range(len(password[:2*k]))))
    return ''.join([''.join(text_matrix[:, p[i]]) for i in sorted(p)])
```

Рис. 4.2: Тестирование шифрования с помощью решеток

```

crypt2(2, 'договор подписали', 'шифр')
Числовой блок из 4 поворотов
[[1 2 3 1]
 [3 4 4 2]
 [2 4 4 3]
 [1 3 2 1]]
Выбираем k различных чисел
[[1. 0. 0. 0.]
 [1. 1. 0. 0.]
 [1. 0. 0. 0.]
 [0. 0. 0. 0.]]
записываем текст через решето, итерация 0
[['д' '0' '0' '0']
 ['о' 'г' '0' '0']
 ['о' '0' '0' '0']
 ['0' '0' '0' '0']]
записываем текст через решето, итерация 1
[['д' 'в' 'о' 'р']
 ['о' 'г' 'п' '0']
 ['о' '0' '0' '0']
 ['0' '0' '0' '0']]
записываем текст через решето, итерация 2
[['д' 'в' 'о' 'р']
 ['о' 'г' 'п' 'о']
 ['о' '0' 'д' 'п']
 ['0' '0' '0' 'и']]
записываем текст через решето, итерация 3
[['д' 'в' 'о' 'р']
 ['о' 'г' 'п' 'о']
 ['о' 'с' 'д' 'п']
 ['а' 'л' 'и' 'и']]
'вгслропиопдидооа'

```

Рис. 4.3: Тестирование шифрования с помощью решеток

## 4.3 Шифр Вижнера. Тип 1

Реализован с помощью функции

```

a1 = 'a b c d e f g h i j k l m n o p q r s t u v w x y z'.split(' ')
def crypt3(text, password, n):
    min_char, max_char = ord(min(text)), ord(max(text))
    text = text.replace(' ', '')
    text = np.array(list(text)+[chr(np.random.randint(min_char, max_char))

```

```

        for i in range(n*(len(text)//n+1*bool(n%len(text)))):
            return ''.join([''.join([al[(ord(j)+ord(k))%len(al)]
                                     for j in t]) for t, k in zip(text, password)])

```

Функция была протестирована функцией (см. рис. 4.4).

```

# шифр вижнера 1
al = 'a b c d e f g h i j k l m n o p q r s t u v w x y z'.split(' ')
def crypt3(text, password, n):
    min_char, max_char = ord(min(text)), ord(max(text))
    text = text.replace(' ', '')
    text = np.array(list(text)+[chr(np.random.randint(min_char, max_char))
                                for i in range(n*(len(text)//n+1*bool(n%len(text)))-len(text))]).reshape(-1, n)
    return ''.join([''.join([al[(ord(j)+ord(k))%len(al)]
                              for j in t]) for t, k in zip(text, password)])

crypt3('how to check this code', 'passw', 5)

'ipxupotqowxlmwqshisg'

```

Рис. 4.4: Тестирование шифрования Вижнера типа 1

## 4.4 Шифр Вижнера. Тип 2

Реализован с помощью функции

```

al = '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'.split(' ')
def crypt4(text, password):
    password = [password[i%len(password)].lower() for i in range(len(text))]
    return ''.join([al[(al.index(i)+al.index(j))%len(al)] for i, j in zip(text.lower(), password)])

```

Функция была протестирована функцией (см. рис. 4.5).

```

# шифр вижнера 2
al = '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'.split(' ')
def crypt4(text, password):
    password = [password[i%len(password)].lower() for i in range(len(text))]
    return ''.join([al[(al.index(i)+al.index(j))%len(al)] for i, j in zip(text.lower().replace(' ', ''), password)])

crypt4('криптография серьезная наука', 'математика')

'црѣфюохшкфѣгкьѣчпчалнтщца'

```

Рис. 4.5: Тестирование шифрования Вижнера типа 2

## **5 Выводы**

В данной работе я изучила алгоритмы шифрования с помощью перестановки. Реализовала и протестировала шифрование с помощью маршрутов, шифрование с помощью решеток, и шифрование Виженера.

## Список литературы

1. Kulyabov D., Korolkova A., Gevorgyan M. Информационная безопасность компьютерных сетей: лабораторные работы. 2015.
2. Самуйлов К.Е. и др. Сети и телекоммуникации : Учебник и практикум. Издательство Юрайт, 2019. С. 1–363.