

# Лабораторная работа №2

## Математические основы защиты информации и информационной безопасности

Роман Сергей Михайлович

### Содержание

Цель работы .....	1
Задание .....	1
Выполнение лабораторной работы .....	1
Выводы.....	5
Список литературы.....	5

### Цель работы

Изучить шифры перестановки, реализовать программным путём маршрутное шифрование, шифрование с помощью решёток и таблицу Виженера.

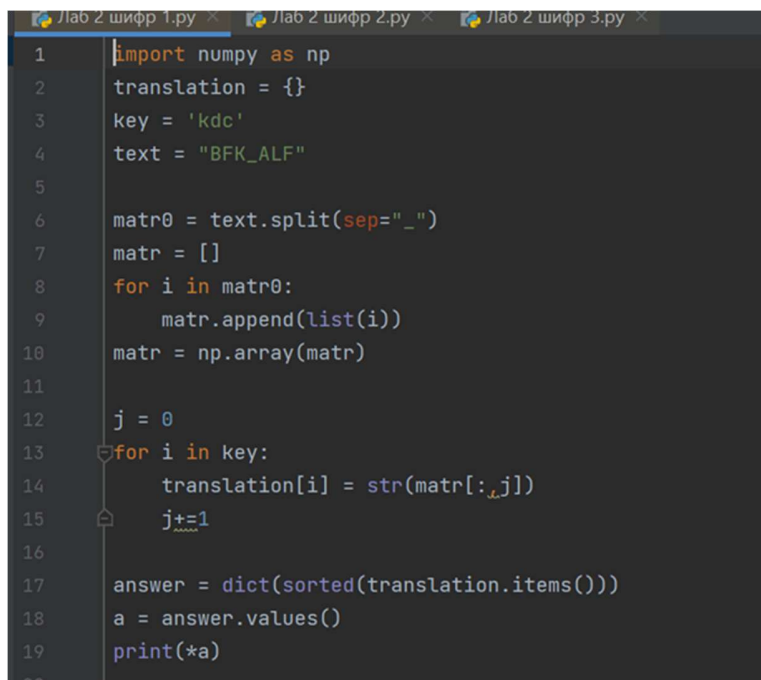
### Задание

1. Реализовать маршрутное шифрование
2. Реализовать шифрование с помощью решёток
3. Реализовать шифрование с помощью таблицы Виженера

### Выполнение лабораторной работы

Так как я не изучал язык Julia на бакалавриате, лабораторную работу я реализовал на python. При дальнейшем изучении я перейду на новый язык.

Для реализации маршрутного шифрования мной была написана следующая программа (рис. 1 ) :



```

1 import numpy as np
2 translation = {}
3 key = 'kdc'
4 text = "BFK_ALF"
5
6 matr0 = text.split(sep="_")
7 matr = []
8 for i in matr0:
9     matr.append(list(i))
10 matr = np.array(matr)
11
12 j = 0
13 for i in key:
14     translation[i] = str(matr[:,j])
15     j+=1
16
17 answer = dict(sorted(translation.items()))
18 a = answer.values()
19 print(*a)

```

Figure 1: Программа реализации первого шифра

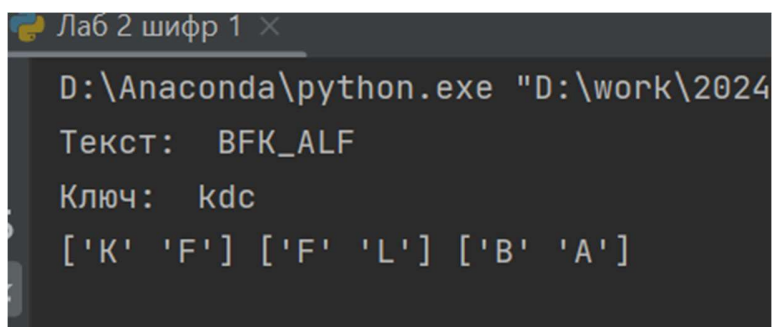
В данной программе:

2-3 строки: инициализация используемых переменных

6-10 строка: деление заданного текста на части и запись в матрицу

12-15: заполнение “переводчика”, каждой букве пароля присваивается соответствующий столбец матрицы

Далее представлен результат работы программы (рис. 2 )



```

D:\Anaconda\python.exe "D:\work\2024
Текст:  BFK_ALF
Ключ:   kdc
['K' 'F'] ['F' 'L'] ['B' 'A']

```

Figure 2: Вывод программы

Как видно, программа работает верно

Теперь представим программу шифрования с помощью решёток (рис. 3 ) (рис. 4 )

```

1 def compress(key, val): # similar for itertools.compress
2     key = list(''.join(key)) # make all in one list like ['.....']
3     val = list(''.join(val))
4     return ''.join(v for v, k in zip(val, key) if k == 'X')
5
6 def right_rotate(array):
7     return tuple(map(lambda a: ''.join(reversed(a)), zip(*array)))
8
9 key = ('X...',
10       '..X.',
11       'X..X',
12       '....')
13
14 value = ('itdf',
15          'gdce',
16          'aton',
17          'qndi')

```

Figure 3: Реализация второго шифра 1

```

18
19 (k, v) = (key, value)
20 l = ''
21 for i in range(4):
22     l += compress(k, v)
23     k = right_rotate(k)
24
25 print(l)

```

Figure 4: Реализация второго шифра 2

В данной программе:

1-4 строки: реализация “прикладывания” решётки к имеющейся матрице

6-7 строка: “переворот” матрицы на 90 градусов

9-12: заданная “решётка”

14-17: матрица букв

19: сокращённая запись

21-23: запуск функций

Посмотрим на результат работы программы, на ней представлены все повороты решёток и конечный шифр (рис. 5 )

```

n: Лаб 2 шифр 2
D:\Anaconda\python.exe "D:\work\2024-2025\Мат основы защиты информации
ican
icantfor
icantforgeti
icantforgetiddqd

```

Figure 5: Вывод программы

Программа работает верно.

Реализуем последний шифр - таблицу Виженера (рис. 6 ) (рис. 7 ) (рис. 8 )

```
1 def form_dict():
2     d = {}
3     iter = 0
4     for i in range(0,127):
5         d[iter] = chr(i)
6         iter = iter + 1
7     return d
8
9 def encode_val(word):
10    list_code = []
11    d = form_dict()
12    for i in range(len(word)):
13        for value in d:
14            if word[i] == d[value]:
15                list_code.append(value)
16    return list_code
```

Figure 6: Реализация третьего шифра 1

```
18 def comparator(value, key):
19     len_key = len(key)
20     dic = {}
21     iter = 0
22     full = 0
23
24     for i in value:
25         dic[full] = [i, key[iter]]
26         full = full + 1
27         iter = iter + 1
28         if (iter >= len_key):
29             iter = 0
30     return dic
```

Figure 7: Реализация третьего шифра 2

```
31
32 def full_encode(value, key):
33     dic = comparator(value, key)
34
35     lis = []
36     d = form_dict()
37
38     for v in dic:
39         go = (dic[v][0]+dic[v][1]) % len(d)
40         lis.append(go)
41     return lis
```

Figure 8: Реализация третьего шифра 3

```
42
43 word = 'Hello'
44 key = 'key'
45
46 print('Текст: ', word)
47
48 print('Ключ: ', key)
49
50 key_encoded = encode_val(key)
51 value_encoded = encode_val(word)
52
53 shifre = full_encode(value_encoded, key_encoded)
54 print('Шифр:', ''.join(decode_val(shifre)))
55
```

Figure 9: Реализация третьего шифра 4

В данной программе:

1-7: функция задающая индексы значений, которые мы будем использовать

9-16: функция, сопоставляющая значения индексов в тексте

18-22: сдвиг по значениям индексов

Последнее изображение (рис. 9 ) задаёт текст, ключ и запускает функции программы.

Результаты работы программы (рис. 10 )



```
Лаб 2 шифр 3 x
D:\Anaconda\python.exe "D:\work\2024-2025\Мат основы защиты и...
Текст: Hellow
Ключ: key
Шифр= 4KfXUq
Process finished with exit code 0
```

Figure 10: Вывод программы

## Выводы

Изучил шифры перестановки, реализовал программным путём маршрутное шифрование, шифрование с помощью решёток и таблицу Виженера.

## Список литературы

Лабораторная работа №2

Шифры перестановки [Электронный ресурс]. URL:  
<https://esystem.rudn.ru/mod/folder/view.php?id=1150970>