

Математические основы защиты информации и информационной безопасности

Отче по лабораторной работе № 2

Меньшов Иван Сергеевич НПМмд-02-21

Содержание

1	Цель работы	4
2	Теоретические сведения	5
2.1	Шифр маршрутной перестановки	5
2.2	Шифр Кардано	5
2.3	Шифр Виженера	6
3	Выполнение работы	7
3.1	Реализация шифра маршрутной перестановки на языке Python .	7
3.2	Реализация шифра решеткой на языке Python	8
3.3	Реализация шифра Виженера на языке Python	10
3.4	Контрольный пример	11
4	Выводы	13
	Список литературы	14

List of Figures

3.1	Работа алгоритма маршрутной перестановки	11
3.2	Работа алгоритма решетки	12
3.3	Работа алгоритма Виженера	12

1 Цель работы

Изучение алгоритмов маршрутной перестановки, решеток и Виженера

2 Теоретические сведения

2.1 Шифр маршрутной перестановки

Широкое распространение получили шифры перестановки, использующие некоторую геометрическую фигуру. Преобразования из этого шифра состоят в том, что в фигуру исходный текст вписывается по ходу одного “маршрута”, а затем по ходу другого выписывается с нее. Такой шифр называют маршрутной перестановкой. Например, можно вписывать исходное сообщение в прямоугольную таблицу, выбрав такой маршрут: по горизонтали, начиная с левого верхнего угла поочередно слева направо и справа налево. Выписывать же сообщение будем по другому маршруту: по вертикали, начиная с верхнего правого угла и двигаясь поочередно сверху вниз и снизу вверх.

2.2 Шифр Кардано

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 Джероламо Кардано в 1550 году, за что и получил своё название.

2.3 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.

В шифре Цезаря каждая буква алфавита сдвигается на несколько строк; например в шифре Цезаря при сдвиге +3, А стало бы D, В стало бы Е 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

```
# задаем пароль
password = str(input('Введите пароль:'))
# задаем фразу для шифрования
my_word = ''.join(str(input('Введите фразу:')).split())
# находим n
n = len(password)
# дополняем слово мусором по необходимости
if len(my_word)%n != 0:
    my_word += 'a'*(n-len(my_word) % n)
# находим порядок выписывания слов по паролю
password_sort = ''.join(sorted(password))
index_list = []
for i in range (len(password)):
    f_index = password.find(password_sort[i])
    index_list.append(f_index)
# записываем зашифрованное слово
new_word = ''
for i in index_list:
    for j in range(len(my_word)//n):
```

```

        new_word += my_word[j*n+i]
print(new_word)

```

3.2 Реализация шифра решеткой на языке Python

```

import numpy as np
import random

# вводим пароль слово для шифрования, и находим k
my_word = ''.join(str(input('Введите фразу:')).split())
# делаем k целым
while (len(my_word) / 4) ** 0.5 % 1 != 0:
    my_word = my_word + 'a'
k = int((len(my_word) / 4) ** 0.5)
print(my_word)
print('k = ', k)
# создаем матрицу размером 2k
matrix_1 = np.reshape(np.arange(1, k ** 2 + 1), (k, k))
matrix_2 = np.rot90(matrix_1, -1)
matrix_4 = np.rot90(matrix_2, -1)
matrix_3 = np.rot90(matrix_4, -1)
mat_1 = np.concatenate((matrix_1, matrix_2), axis=1)
mat_2 = np.concatenate((matrix_3, matrix_4), axis=1)
mat = np.concatenate((mat_1, mat_2), axis=0)
print(mat)
# случайно выбираем позиции в матрице 2k
str_mat = mat.astype('|S1').tobytes().decode('utf-8')
dic = {}
for i in range(1, k ** 2 + 1):

```



```

index = []
for j in range(len(str_mat)):
    if str(i) == str_mat[j]:
        index.append(j)
dic[i] = index
print(dic)
chouse_pos = []
for i in dic:
    value = dic[i]
    chouse_val = random.choice(value)
    i_index = chouse_val // (k * 2)
    j_index = chouse_val % (k * 2)
    chouse_pos.append([i_index, j_index])
print(chouse_pos)
# процесс шифрования с ключевой матрицей и кодовым словом
key_matrix = np.zeros((2 * k, 2 * k), dtype=int)
val = 1
for i, j in chouse_pos:
    key_matrix[i][j] = val
    val += 1
print(key_matrix)
matrix_end = np.copy(key_matrix)
for i in range(3):
    key_matrix = np.rot90(key_matrix, -1)
    for j in range(2 * k):
        for q in range(2 * k):
            if key_matrix[j][q] != 0:
                key_matrix[j][q] += k ** 2
print(f'после {i} шага')

```

```

        matrix_end = matrix_end + key_matrix
print(matrix_end)
while True:
    password = str(input(f'Введите пароль длиной {2 * k}: '))
    if len(password) == 2 * k:
        break
    else:
        print('Не выполнены условия ввода пароля')
password_sort = ''.join(sorted(password))
index_list = []
for i in range (len(password)):
    f_index = password.find(password_sort[i])
    index_list.append(f_index)
new_word = []
for i in index_list:
    for j in range(matrix_end.shape[0]):
        new_word.append(matrix_end[j][i])
print(new_word)
kod_word = ''
for i in range(len(new_word)):
    kod_word += my_word[new_word[i]-1]
print('Зашифрованное сообщение: ',kod_word)

```

3.3 Реализация шифра Виженера на языке Python

```

# задаем 1 шифр цезаря
slovr = 'абвгдеёжзийклмнопрстуфхцчщъыьэя'
# задаем пароль
pasword = str(input('Введите пароль: ')).lower()

```

```

# задаем фразу для шифрования
word = str(input('Введите фразу для шифрования: ')).lower()
# растягиваем пароль
k = (len(word) % len(password))
password_len = '' + password * (len(word) // len(password)) + password[:k]
print(word, password_len, sep='\n')
# создаем квадрат вижинера
slovr_visinera = []
slovr_i = 'абвгдеёжзийклмнопрстуфхцчшщъыьэюя'
for i in range(len(slovr)):
    slovr_visinera.append(slovr_i)
    new = slovr_i[1:] + slovr_i[0]
    slovr_i = new
print("Квадрат вижинера:", slovr_visinera)
# шифруем сообщение
message = ''
for i in range(len(word)):
    f_index1 = slovr.find(word[i])
    f_index2 = slovr.find(password_len[i])
    message += slovr_visinera[f_index1][f_index2]
print(f'Зашифрованное сообщение: {message}')

```

3.4 Контрольный пример

```

C:\Users\xslix\PycharmProjects\Rudn\venv\Scripts\python.exe "C:/Users/xslix/PycharmProjects/Rudn/Основы шифрования/2 lab/1 task.py"
Введите пароль: 1234
Введите фразу: likehotrlikeetoa
likehotrlikeetoa
Process finished with exit code 0

```

Figure 3.1: Работа алгоритма маршрутной перестановки

```

C:\Users\xslix\PycharmProjects\Ruoh\venv\Scripts\python.exe "C:/Users/xslix/PycharmProjects/Ruoh/Основы шифрования/2 lab/2 task.py"
Введите фразу:
Helloaaaaaa
n = 2
[[1, 2, 3, 1],
 [3, 4, 4, 2],
 [2, 4, 4, 1],
 [1, 3, 2, 1]]
[1: [0, 3, 12, 10], 2: [1, 7, 8, 14], 3: [2, 4, 11, 13], 4: [5, 6, 9, 10]]
[[3, 0], [0, 1], [2, 3], [2, 1]]
[[0, 2, 0, 0],
 [0, 0, 0, 0],
 [0, 4, 0, 1],
 [1, 0, 0, 0]]
после 0 шага
после 1 шага
после 2 шага
[[ 5, 2, 15, 9],
 [11, 0, 12, 6],
 [14, 4, 10, 3],
 [ 1, 7, 10, 13]]
Введите пароль длиной 4: 1234
[15, 12, 16, 10, 5, 11, 14, 1, 9, 6, 3, 13, 2, 8, 4, 7]
Зашифрованное сообщение:  aaazaaazaaLaazaa
Process finished with exit code 0
|

```

Figure 3.2: Работа алгоритма решетки

```

C:\Users\xslix\PycharmProjects\Ruoh\venv\Scripts\python.exe "C:/Users/xslix/PycharmProjects/Ruoh/Основы шифрования/2 lab/3 task.py"
Введите пароль:
Введите фразу для шифрования: 123456789
туктококо
туктококо
Квадрат вивинера: ['абгдейзийклмнопрстуфхцщъыьэяа', 'бгдейзийклмнопрстуфхцщъыьэяа', 'гдейзийклмнопрстуфхцщъыьэяа', 'ддейзийклмнопрстуфхцщъыьэяа', 'едейзийклмнопрстуфхцщъыьэяа', 'фдейзийклмнопрстуфхцщъыьэяа', 'хдейзийклмнопрстуфхцщъыьэяа', 'цдейзийклмнопрстуфхцщъыьэяа', 'щдейзийклмнопрстуфхцщъыьэяа', 'ъдейзийклмнопрстуфхцщъыьэяа', 'ыдейзийклмнопрстуфхцщъыьэяа', 'эдейзийклмнопрстуфхцщъыьэяа', 'ядейзийклмнопрстуфхцщъыьэяа', 'адейзийклмнопрстуфхцщъыьэяа', 'бдейзийклмнопрстуфхцщъыьэяа', 'гдейзийклмнопрстуфхцщъыьэяа', 'ддейзийклмнопрстуфхцщъыьэяа', 'едейзийклмнопрстуфхцщъыьэяа', 'фдейзийклмнопрстуфхцщъыьэяа', 'хдейзийклмнопрстуфхцщъыьэяа', 'цдейзийклмнопрстуфхцщъыьэяа', 'щдейзийклмнопрстуфхцщъыьэяа', 'ъдейзийклмнопрстуфхцщъыьэяа', 'ыдейзийклмнопрстуфхцщъыьэяа', 'эдейзийклмнопрстуфхцщъыьэяа', 'ядейзийклмнопрстуфхцщъыьэяа']
Зашифрованное сообщение: ехехехехе
Process finished with exit code 0

```

Figure 3.3: Работа алгоритма Виженера

4 Выводы

Изучили алгоритмы шифрования с помощью перестановок

Список литературы

1. Шифр маршрутной перестановки
2. Шифр Кардано
3. Шифр Виженера