Отчёт по лабораторной работе №2

Студент: Майорова О.А., НФИмд-02-21

Преподаватель: д.ф.-м.н. Кулябов Д.С.

Москва 2021

Содержание

# 1 Цель работы

Цель: Ознакомиться с шифрами перестановки на примере маршрутного шифрования, шифрования с помощью решёток и шифра Виженера.

# 2 Задание

Программно реализовать маршрутное шифрование, шифрование с помощью решёток и шифр Виженера.

# 3 Теоретическое введение

Шифр перестановки — класс методов шифрования, в котором элементы исходного открытого текста меняют местами по определённому правилу. Элементами текста могут быть отдельные символы (самый распространённый случай), пары букв, тройки букв, комбинирование этих случаев и так далее. Типичными примерами перестановки являются анаграммы. В классической криптографии шифры перестановки можно разделить на два класса:

* Шифры одинарной (простой) перестановки — при шифровании символы открытого текста перемещаются с исходных позиций в новые один раз;
* Шифры множественной (сложной) перестановки — при шифровании символы открытого текста перемещаются с исходных позиций в новые несколько раз.

Далее мы рассмотрим маршрутное шифрование, шифрование с помощью решёток, которые являются шифрами простой перестановки и шифр Виженера, который на самом деле является шифром замены [1,2].

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. Шифрованное сообщение записывается, например, по маршруту: по вертикалям, начиная с верхнего левого угла, поочередно сверху вниз [3].

Шифрования с помощью решёток - метод шифрования, использующий для шифрования открытого текста трафарет с прорезями-ячейками. Самый ранний из известных таких инструментов — решётка Кардано, датируемая 1550 годом, в ней использовался прямоугольный трафарет, позволяющий писать отдельные буквы, слога или слова, а затем читать их через специальные прорези-ячейки. Письменные фрагменты открытого текста дополнительно маскировались тем, что промежутки между шифруемыми фрагментами заполнялись ничего не означающими словами или буквами. Этот вариант является также примером стеганографии [4]. Решетка Кардано (поворотная решетка) — это прямоугольная или квадратная карточка с четным числом строк и столбцов. В ней проделаны отверстия таким образом, что при последовательном отражении или поворачивании и заполнении открытых клеток карточки постепенно будут заполнены все клетки листа. Карточку сначала отражают относительно вертикальной оси симметрии, затем — относительно горизонтальной оси, и снова — относительно вертикальной. Если решетка Кардано — квадратная, то возможен и другой вариант ее преобразований — поворот на 90°. Получатель должен знать трафарет и наложить его в той же последовательности, что и при шифровании. Ключом является выбранный тип перемещения решетки (отражение или поворот) и трафарет — расположение отверстий [5].

Шифр Виженера — метод полиалфавитного шифрования буквенного текста с использованием ключевого слова. Шифр Виженера состоит из последовательности нескольких шифров Цезаря с различными значениями сдвига. Для зашифровывания может использоваться таблица алфавитов, называемая квадрат или 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 (табл. 1).

Table 1: Таблица Виженера

| A | B | C | D | E | F | G | H | I | J | … | Q | R | S | T | U | V | W | X | Y | Z |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| B | C | D | E | F | G | H | I | J | K | … | R | S | T | U | V | W | X | Y | Z | A |
| C | D | E | F | G | H | I | J | K | L | … | S | T | U | V | W | X | Y | Z | A | B |
| D | E | F | G | H | I | J | K | L | M | … | T | U | V | W | X | Y | Z | A | B | C |
| … | … | … | … | … | … | … | … | … | … | … | … | … | … | … | … | … | … | … | … | … |
| Z | Y | X | W | V | U | T | S | R | Q | … | J | I | H | G | F | E | D | C | B | A |

Если - количество букв в алфавите, - — номер буквы открытого текста, — номер буквы ключа в алфавите, то шифрование Виженера можно записать следующим образом:

И расшифровывание:

Проще говоря, расшифровывание производится следующим образом: находим в таблице Виженера строку, соответствующую первому символу ключевого слова, и в данной строке находим первый символ зашифрованного текста. Столбец, в котором находится данный символ, соответствует первому символу исходного текста. Следующие символы зашифрованного текста расшифровываются подобным образом [2].

# 4 Выполнение лабораторной работы

Для выполнения лабораторной работы был выбрат язык Python. Перед началом работы подключим библиотеку numpy:

import numpy as np

## 4.1 Маршрутное шифрование

Сначала реализуем метод маршрутного шифрования для русского алфавита нижнего регистра:

# C - исходный текст  
# p - пароль  
  
def route(C, p):  
 C = C.replace(' ', '')  
 n = len(p)  
 m = int(np.ceil(len(C) / n))  
   
 for i in range(m\*n - len(C)):  
 C += chr(np.random.randint(ord('а'), ord('я') + 1))  
   
 tbl = np.array(list(C)).reshape(m, n)  
   
 r = p  
 cypher = ''  
 for i in range(n):  
 idx = p.index(min(r))  
 r = r.replace(min(r), '')  
 cypher += ''.join(tbl[:, idx])  
   
 return cypher

Результатом запуска функции для примера из задания к лабораторной работе будет рис. 1.

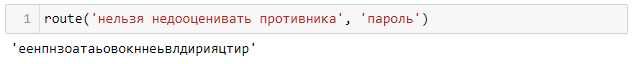


Figure 1: Проверка функции маршрутного шифрования

Можно видеть, что полученное зашифрованное сообщение длиннее приведённого в задании, однако, можно заметить, что длина всё же совпадает с длиной исходного текста. Таким образом, шифрование функцией было произведено корректно.

## 4.2 Шифрование с помощью решёток

Теперь реализуем шифрование с помощью решёток для заданных исходного текста и пароля в виде следующей функции:

# C - исходный текст  
# p - пароль  
  
def grid(C, p):  
 C = C.replace(' ', '')  
 k = int(np.ceil(len(p) / 2))  
   
 sq1 = np.arange(k\*\*2).reshape((k, k)) + 1  
 sq2 = np.rot90(sq1, axes=(1, 0))  
 sq3 = np.rot90(sq1, k=2, axes=(1, 0))  
 sq4 = np.rot90(sq1, k=3, axes=(1, 0))  
   
 tbl = np.vstack((np.hstack((sq1, sq2)), np.hstack((sq4, sq3))))  
   
 for s in range(k\*\*2):  
 shot = np.random.randint(0, 4)  
 idx, jdx = np.argwhere(tbl == s+1)[shot]  
 tbl[idx, jdx] = 0  
   
 cyph\_tbl = np.empty([2\*k, 2\*k], dtype=str)  
 for i in range(4):  
 for j in range(k\*\*2):  
 idx, jdx = np.argwhere(tbl == 0)[j]  
 cyph\_tbl[idx, jdx] = C[j]  
   
 C = C[k\*\*2:]  
 tbl = np.rot90(tbl, axes=(1, 0))  
   
 r = p  
 cypher = ''  
 for i in range(len(p)):  
 idx = p.index(min(r))  
 r = r.replace(min(r), '')  
 cypher += ''.join(cyph\_tbl[:, idx])  
   
 return cypher

Результатом запуска функции для примера из задания к лабораторной работе будет рис. 2.

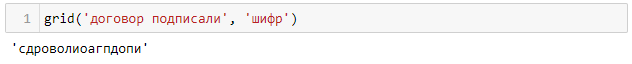


Figure 2: Проверка функции шифрования с помощью решёток 1

Можно видеть, что полученное зашифрованное сообщение отличается от приведённого в задании, однако, это объясняется тем, что задание “отверстий” в трафарете осуществляется на основе случайности. На рис. 3 можно видеть результат ещё одного запуска функции для тех же вводных данных.

Figure 3: Проверка функции шифрования с помощью решёток 2

Figure 3: Проверка функции шифрования с помощью решёток 2

Таким образом, шифрование функцией было произведено корректно.

## 4.3 Шифр Виженера

Наконец, реализуем шифр Виженера. Сперва модифицируем функцию из предыдущей лабораторной работы, реализующую шифр Цезаря, для русского алфавита нижнего регистра:

# c - буква для шифрования  
# k - ключ  
  
def Caesar(c, k):  
 n = [(i + k) % 33 for i in range(33)][ord(c) - ord('а')]  
 return chr(ord('а') + n)

Далее напишем новую функцию, реализующую непосредственно шифр Виженера, которая будет использовать функцию Caesar:

# C - исходный текст  
# p - пароль  
  
def Vigenеre(C, p):  
 C = C.replace(' ', '')  
 P = ''  
 for i in range(int(np.floor(len(C)/len(p)))):  
 P += ''.join(p)  
 P += ''.join(p[:len(C)%len(p)])  
   
 cypher = ''  
 for i in range(len(C)):  
 k = ord(P[i]) - ord('а')  
 cypher += ''.join(Caesar(C[i], k))  
   
 return cypher

Проверим работу функции на примере из задания к лабораторной работе (рис. 4).

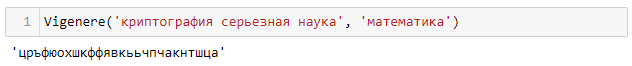


Figure 4: Проверка функции шифра Виженера

Можно видеть, что полученное зашифрованное сообщение отличается от приведённого в задании, однако, это объясняется тем, что исходные алфавиты, использованные в тексте задания и при реализации функции, отличаются на одну букву - “ъ”. В задании к лабораторной работе “ъ” отсутствует. Таким образом, шифрование функцией было произведено корректно.

# 5 Выводы

Таким образом, была достигнута цель, поставленная в начале лабораторной работы. Было осуществлено знакомство с шифрами перестановки на примере маршрутного шифрования, шифрования с помощью решёток и шифра Виженера. Также были получены реализации данных шифров на языке Python:

* реализация шифрования с помощью решёток для заданного пароля;
* реализации маршрутного шифрования и шифра Виженера для русского алфавита нижнего регистра.

# Список литературы

1. Перестановочный шифр [Электронный ресурс]. Wikipedia, 2021. URL: <https://ru.wikipedia.org/w/index.php?title=Перестановочный_шифр&oldid=117349815>.

2. Шифр Виженера [Электронный ресурс]. Wikipedia, 2021. URL: <https://ru.wikipedia.org/w/index.php?title=Шифр_Виженера&oldid=117916678>.

3. Простейшие методы шифрования текста. Часть 2 «Перестановочные шифры» [Электронный ресурс]. Мир информатики, 2020. URL: <https://infojournal.ru/wp-content/uploads/2020/04/mir_info-4-2020.pdf>.

4. Шифровальная решётка [Электронный ресурс]. Wikipedia, 2020. URL: <https://ru.wikipedia.org/w/index.php?title=Шифровальная_решётка&oldid=110003845>.

5. Классические шифры перестановки [Электронный ресурс]. Студми. Учебные материалы для студентов, 2021. URL: <https://studme.org/239548/informatika/klassicheskie_shifry_perestanovki>.