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: Агеева Анастасия Сергеевна, 1032212304
Группа: НФИмд-02-21
Преподаватель: Кулябов Дмитрий Сергеевич,
д-р.ф.-м.н., проф.

Москва 2021

Содержание

List of Figures

1 Цель работы

Цель данной лабораторной работы изучение реализация трех алгоритмов шифрования: маршрутное шифрование, шифрование с помощью решеток и шифрование при помощи таблицы Виженера.

2 Задание

1. Реализовать программно маршрутное шифрование;
2. Реализовать программно шифрование с помощью решеток;
3. Реализовать программно шифрование при помощи таблицы Виженера.

3 Теоретическое введение

3.1 Шифры перестановки

Шифр перестано́вки [1] — это метод симметричного шифрования, в котором элементы исходного открытого текста меняют местами. 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.

В качестве альтернативы шифрам перестановки можно рассматривать **подстановочные шифры**. В них элементы текста не меняют свою последовательность, а изменяются сами.

3.2 Маршрутное шифрование

Преобразования состоят в том, что отрезок открытого текста записывается в такую фигуру по некоторой траектории, а выписывается по другой траектории [2].

Наибольшее распространение получили маршрутные шифры перестановки, основанные на прямоугольниках (таблицах). Например, можно записать сооб-

щение в прямоугольную таблицу по маршруту: по горизонтали, начиная с верхнего левого угла, поочередно слева направо. Сообщение будем списывать по маршруту: по вертикалям, начиная с верхнего правого угла, поочередно сверху вниз.

Пример [1]:

ОТКРЫТЫЙ ТЕКСТ: *пример маршрутной перестановки*

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: *ешоеомрнрниатеаирмупткпррйсв*

3.3 Шифрование с помощью решеток

В 1550 году итальянский математик Джероламо Кардано (1501—1576) в книге «О тонкостях» предложил новую технику шифрования сообщений — **решётку**. Изначально решётка Кардано представляла собой *трафарет с отверстиями*, в которые записывали буквы, слоги или слова сообщения. Затем трафарет убирали, а свободное место заполняли более или менее осмысленным текстом. Такой метод сокрытия информации относится к стеганографии. Позднее был предложен **шифр «поворотная решётка»** — первый транспозиционный (геометрический) шифр. Несмотря на то, что существует большая разница между изначальным предложением Кардано и шифром «поворотная решётка», методы шифрования, основанные на трафаретах, принято называть «решётками Кардано» [1].

Для шифрования и дешифрования с помощью данного шифра изготовляется трафарет с вырезанными ячейками. При наложении трафарета на таблицу того же размера четырьмя возможными способами, его вырезы полностью должны покрывать все клетки таблицы ровно по одному разу. При шифровании трафарет накладывают на таблицу. В видимые ячейки по определённому маршруту вписывают буквы открытого текста. Далее трафарет переворачивают три раза, каждый раз проделывая операцию заполнения. Шифрограмму выписывают из получившейся таблицы по определённому маршруту. Ключом являются трафа-

рет, маршрут вписывания и порядок поворотов.

Шифрование с помощью решеток в первой половине 1917 года германская армия использовала на Восточном (против России) фронте. В 1982 годе его применяли британские войска в вооруженном конфликте с Аргентиной за Фолклендские острова [3].

3.4 Таблица 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 Джовани Баттиста Белласо (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. Хотя шифр легко понять и реализовать, на протяжении трех столетий он противостоял всем попыткам его сломать; чем и заработал имя le chiffre indéchiffrable (фр. неразгаданный шифр). Многие люди пытались реализовать схемы шифрования, которые по сути являлись шифрами Виженера.

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 [4].

4 Выполнение лабораторной работы

1. Подключение библиотек.

Figure 4.1: Подключение библиотеки

2. **Реализация маршрутного шифрования**

    1. В качестве начальных значений берется ключ “пароль”. Алфавитом может быть любая строка неповторяющихся символов. Я использую кириллицу. Также задаю строку сообщение, которое будет шифроваться.

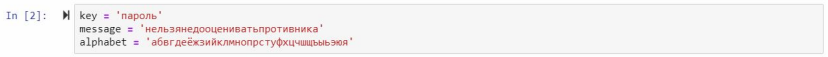

Figure 4.2: Начальные значения для маршрутного шифрования

    2. Задам функцию *path()*, в качестве параметров передаются заданные начальные данные. Внутри функции ключ, алфавит и сообщения преобразую в массив. В функции расчитываются значения n и m. Затем заполняется матрица размером n•m случайными символами из алфавита. Потом в эту же матрицу записывается сообщение и приписывается ключ. Значения выписываются в алфавитном порядке символов в ключе.

```
In [3]:
def path(alp, k, mes):
    alp = list(alp)
    k = list(k)
    mes = list(mes)
    n = len(k)
    m = len(mes)//n
    if (len(mes)//n != 0):
        m+=1
    matr = [[np.random.choice(alp) for i in range(0, n)] for j in range (m)]
    c = 0
    for i in range(m):
        for j in range(n):
            if c < len(mes):
                matr[i][j] = mes[c]
                c += 1
    matr.append(k)
    way = sorted(matr[len(matr) - 1])
    new_mes = []
    for i in way:
        for j in range(len(matr)):
            if j == len(matr) - 1:
                continue
            new_mes += matr[j][matr[len(matr) - 1].index(i)]
    new_mes = ''.join(new_mes)
    return new_mes
```

Figure 4.3: Функция шифрования path()

3. Выведу результат работы программы для заданных начальных значений. Зашифрованное сообщение получается на несколько символом длиннее (в данном случае на один), поскольку в матрицу дописываются недостающие элементы. Дефект можно устранить, но в примере к лабораторной работе символ в конце присутствует (там “а”, но у меня матрица заполняется случайным образом, поэтому окончание зашифрованного сообщения отличается).

```
In [4]:
new_message = path(alphabet, key, message)

In [5]:
print(message, "- сообщение")
print(new_message, "- зашифрованное сообщение")

нельзянедооцениватьпротивника - сообщение
еенпнзоатаьовокннеьвлдириятив - зашифрованное сообщение
```

Figure 4.4: Результат выполнения программы

3. **Реализация шифрования с помощью решеток**

1. В качестве начальных значений берется ключ “шифр”. Алфавитом может быть любая строка неповторяющихся символов, аналогично использую кириллицу. Также задаю строку сообщение, которое будет шифроваться.

```
In [6]:
key = 'шифр'
message = 'договорподписали'
alphabet = 'абвгдеёжзийклмнопрстуфхцчшщьыьэюя'
```

Figure 4.5: Начальные значения для шифрования с помощью решеток

2. В функции *delete()* удаляются в некотором порядке элементы матрицы.

```
In [7]:
def delete(b_m, i_1, k_1):
    for i in range(2*k_1):
        for j in range(2*k_1 - i):
            if b_m[j][i] == i_1:
                b_m[j][i] = ' '
                return
```

Figure 4.6: Вспомогательная функция delete()

3. Задам функцию *lattice()*, в которой будет реализован алгоритм шифрования с помощью решеток.

```
In [8]:
def lattice(alp, key, mes):
    key = list(key)
    mes = list(mes)
    alp = list(alp)
    k = int(np.sqrt(len(key)))
    matr1 = [[i for i in range(k)] for i in range(k)]
    c = 1
    for i in range(k):
        for j in range(k):
            matr1[i][j] = c
            c += 1
    matr2 = np.rot90(matr1, k = 1, axes = (1, 0))
    matr3 = np.rot90(matr2, k = 1, axes = (1, 0))
    matr4 = np.rot90(matr3, k = 1, axes = (1, 0))
    bigmatr_n = [[0 for i in range(2*k)] for i in range(2*k)]
    for i in range(k):
        for j in range(k):
            bigmatr_n[i][j] = matr1[i][j]
            bigmatr_n[i][j + k] = matr2[i][j]
            bigmatr_n[i + k][j + k] = matr3[i][j]
            bigmatr_n[i + k][j] = matr4[i][j]
    bigmatr_l = [[' ' for i in range(2*k)] for i in range(2*k)]
    print(bigmatr_n)
    list1 = [i for i in range(1, k**2+1)]
    for i in list1:
        delete(bigmatr_n, i, k)
    print(bigmatr_n)
```

Figure 4.7: Функция шифрования lattice() (1)

```
    list1 = [i for i in range(1, k**2+1)]
    for i in list1:
        delete(bigmatr_n, i, k)
    print(bigmatr_n)
    for i in range(4):
        for i in range(k*2):
            for j in range(k*2):
                if bigmatr_n[i][j] == bigmatr_l[i][j] and len(mes) > 0:
                    bigmatr_l[i][j] = mes[0]
                    mes = mes[1:]
        bigmatr_n = np.rot90(bigmatr_n, k = 1, axes = (1, 0))
    bigmatr_l.append(key)
    way = sorted(bigmatr_l[len(bigmatr_l) - 1])
    new_mes = []
    for i in way:
        for j in range(len(bigmatr_l)):
            if j == len(bigmatr_l) - 1:
                continue
            new_mes += bigmatr_l[j][bigmatr_l[len(bigmatr_l) - 1].index(i)]
    print(bigmatr_l)
    new_mes = ''.join(new_mes)
    return new_mes
```

Figure 4.8: Функция шифрования lattice() (2)

4. Выведу результат работы программы для заданных начальных значений. Сравнивать его с результатом, представленным в задании к лабораторной работе, поскольку символы для решета там убирались случайным образом. Но при проверке результата по промежуточным итогам можно увидеть, что результат корректный.

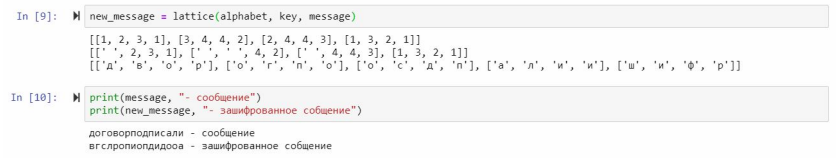

Figure 4.9: Результат выполнения программы

4. **Реализация таблицы Виженера**

1. В качестве начальных значений берется ключ “шифр”. Алфавитом может быть любая строка неповторяющихся символов, аналогично использую кириллицу. Также задаю строку сообщение, которое будет шифроваться.

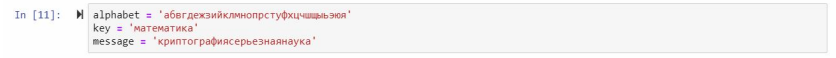

Figure 4.10: Начальные значения для

2. Задам функцию *vigenere()*, которая основываясь на алгоритме построения таблицы Виженера будет шифровать переданное сообщение. Входящие параметры: ключ, сообщение, алфавит.

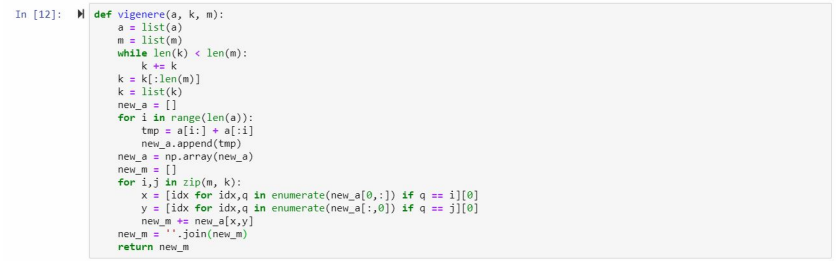

Figure 4.11: Функция шифрования vigenere()

3. Выведу результат работы программы для заданных начальных значений. Результат не совпадает с представленным примером в задании, поскольку там отсутствуют буквы “ё” и “ъ”.

```
In [13]: new_message = vigenere(alphabet, key, message)

In [14]: print(message, "- сообщение")
         print(new_message, "- зашифрованное собщение")

         криптографиясерьезнаянаука - сообщение
         црыфяохшкффядкэьчпчалнтшца - зашифрованное собщение
```

Figure 4.12: Результат выполнения программы

5 Выводы

В ходе данной лабораторной работы я реализовала три алгоритмов шифрования: маршрутное шифрование, шифрование с помощью решеток и шифрование при помощи таблицы Виженера.

Список литературы

1. Перестановочный Шифр [Электронный ресурс]. Википедия, 2019. URL: https://ru.wikipedia.org/wiki/Перестановочный_шифр.
2. Шифр табличной маршрутной перестановки [Электронный ресурс]. Студопедия, 2015. URL: https://studopedia.ru/9_184418_shifr-tablichnoy-marshrutnoy-perestanovki.html.
3. Салий В.Н. Криптографические методы и средства защиты информации. Саратовский государсвенный университет, 2012. 42 с.
4. Шифр Веженера [Электронный ресурс]. Википедия, 2021. URL: https://ru.wikipedia.org/wiki/Шифр_Виженера.