Отчёт по лабораторной работе №2.

Шифры перестановки

Коне Сирики

21 сентября 2024

Российский университет дружбы народов, Москва, Россия

Объединённый институт ядерных исследований, Дубна, Россия

Информация

Докладчик

- Коне Сирики
- Студент физмат
- профессор кафедры прикладной информатики и теории вероятностей
- Российский университет дружбы народов
- konesirisil@yandex.ru
- https://github.com/skone19

Цели и задачи работы

Целью данной лабораторной работы является ознакомление с одним методом полиалфавитного шифрования – *шифром Виженера* – и двумя широко известными шифрами перестановки – *маршрутным шифрованием* и *шифрованием с помощью решёток*, – а также их последующая программная реализация.

Задачи рассмотреть и реализовать на языке программирования Python:

1. Шифрование методом столбцовой перестановки;
2. Шифрование с помощью поворотных решёток;
3. Шифр Виженера.

Теоретическое введение

Шифр перестановки

Шифр, преобразования из которого изменяют только порядок следования символов исходного текста, но не изменяют их самих, называется *шифром перестановки*.

Подстановка

Таблица, в первой строке которой указывается позиция символа в исходном сообщении, а во второй – его позиция в шифрограмме, называется *подстановкой* степени n.

$$\begin{pmatrix} 1 & 2 & \cdots & n \\ i_1 & i_2 & \cdots & i_n \end{pmatrix}$$

Маршрутное шифрование. Столбцовая перестановка

Маршрутная перестановка

Шифр, преобразования которого состоят в том, что в некоторую геометрическую фигуру исходный текст вписывается по ходу одного ”маршрута”, а затем по ходу другого выписывается с нее, называют *маршрутной перестановкой*.

Столбцовая перестановка

Маршрутная перестановка на основе прямоугольной таблицы, вписывание в которую осуществляется по строкам слева-направо, а выписывание – по столбцам сверху-вниз в порядке, определяемым некоторым ключом, называют *столбцовой перестановкой*.

Шифрование с помощью решёток

- Решётка Кардано представляла собой трафарет с прорезанными в нем отверстиями. При шифровании трафарет накладывался на таблицу, и в её видимые ячейки выписывались буквы исходного текста. Пустые ячейки в таблице затем заполняются “мусором”.
- Поворотная решётка подразумевает повороты трафарета и последовательное выписывание символов сообщения в таблицу блоками до её заполнения. Шифрограмму выписывают из итоговой таблицы по определенному маршруту.

Поворотная решётка. Подготовка трафарета

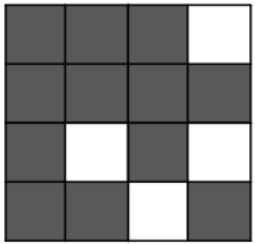
1	2
3	4

1	2	3	1
3	4	4	2

1	2	3	1
3	4	4	2
		4	3
		2	1

1	2	3	1
3	4	4	2
2	4	4	3
1	3	2	1

1	2	3	1
3	4	4	2
2	4	4	3
1	3	2	1



Шифр Виженера

Шифр Виженера – это полиалфавитный шифр подстановки, представляющий собой последовательность из нескольких шифров Цезаря с различными значениями сдвига, задаваемыми некоторым ключом. Так, если n – количество букв в алфавите, m_j – номер буквы открытого текста, k_j – номер буквы ключа, c_j – номер буквы шифротекста, то:

$$c_j = (m_j + k_j) \; mod \; n$$

Таблица Виженера

	А	Б	В	Г	Д	Е	Ё	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
А	А	Б	В	Г	Д	Е	Ё	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
Б	Б	В	Г	Д	Е	Ё	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	А
В	В	Г	Д	Е	Ё	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	А	Б
Г	Г	Д	Е	Ё	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	А	Б	В
Д	Д	Е	Ё	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	А	Б	В	Г
Е	Е	Ё	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	А	Б	В	Г	Д
Ё	Ё	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	А	Б	В	Г	Д	Е
Ж	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	А	Б	В	Г	Д	Е	Ё
З	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	А	Б	В	Г	Д	Е	Ё	Ж
И	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ё	Ж	З
Й	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ё	Ж	З	И
К	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й
Л	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К
М	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л
Н	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М
О	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н
П	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О
Р	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П
С	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П	Р
Т	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С
У	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т
Ф	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У
Х	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф
Ц	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х
Ч	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц
Ш	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч
Щ	Щ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш
Ъ	Ъ	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ
Ы	Ы	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ
Ь	Ь	Э	Ю	Я	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы
Э	Э	Ю	Я	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь
Ю	Ю	Я	А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э
Я	Я	А	Б	В	Г	Д	Е	Ё	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

Ход выполнения и результаты

```
import math
import numpy as np
import string
# русский алфавит
abc  = [chr(code) for code in range(ord('а'), ord('я') + 1)]

# словарь вида {буква : порядковый номер}
letter2number = {abc[i] : i for i in range(len(abc))}

mes = message.lower().replace(" ", "")
mes = mes.translate(str.maketrans('', '', string.punctuation))
```

Столбцовая перестановка. Фрагменты кода

```
table = np.full((m, n), 'a')
for i in range(m):
    for j in range(n):
        if i * n + j < len(mes):
            table[i][j] = mes[i * n + j]
        else:
            break

nums = sorted([letter2number[letter] for letter in key])
route_order = [abc[number] for number in nums]
route_order = [key.index(letter) for letter in route_order]

for j in route_order: # проходим по столбцам в заданном порядке
    for i in range(m): # проходим по всем строкам
        message_encrypted += table[i][j]
```

Шифр Виженера. Фрагменты кода

```
vigenere_table = np.array(abc)
for i in range(1, len(abc)):
    row = np.roll(abc, -i)
    vigenere_table = np.vstack((vigenere_table, row))

long_key = key # удлинним ключ так, чтобы он покрывал всё сообщение
n = len(key)
while len(long_key) < len(mes):
    m = len(long_key)
    long_key = long_key + long_key[m - n]

for i in range(len(mes)):
    column = letter2number[mes[i]]
    row = letter2number[long_key[i]]
    message_encrypted += vigenere_table[row][column]
```

```
    print(columnar_cipher("Нельзя недооценивать противника", "пароль"))
    print(columnar_cipher("Стремясь к лучшему, мы часто портим хорошее", "корольлир"))
[3] ✓ 0.2s
... еенпнзоатаьовокннеьвлдирияцтиа
ьмреслчимеормеортуамтуамрчсхрчсхямпо
```

Рис. 3: Пример шифрования методом столбцовой перестановки

```
    print(vigenere_cipher("криптография серьезная наука", "математика", vigenere_table))
    print(vigenere_cipher("Мир - сцена, где всякий свою роль играть обязан", "венецианскийкупец", vigenere_table))
[6] ✓ 0.4s
... цръфюохшкффягкььчпчалнтшца
онэцмнннфонлытщняузыгжцлйщншйьпэжхйеь
```

Рис. 4: Пример шифрования с помощью таблицы Виженера

Шифрование с помощью решёток. Фрагменты кода (1)

```
def rotare_cell(cell, k):
    cell_r = cell.T # транспонируем исходную матрицу
    result = np.full((k, k), 'a') # результирующая решетка
    for i in range(k):
        for j in range(k):
            result[i][j] = cell_r[i][k - j - 1] <...>

def get_holes(cell, k):
    cell_nums = np.random.randint(0, 4, k ** 2)
    intervals = { 0 : [[0, k], [0, k]] <...> }

    for i in range(k ** 2): <...>
        for j in range(interval[0][0], interval[0][1]):
            for l in range(interval[1][0], interval[1][1]):
                if cell[j][l] == number:
```

Шифрование с помощью решёток. Фрагменты кода (2)

```
n = len(mes)
k = math.ceil(math.ceil(np.sqrt(n)) / 2)
while len(mes) < (2 * k) ** 2:
    mes += 'a'

cell_1 = np.full((k, k), 0)
for i in range(k):
    for j in range(k):
        cell_1[i][j] = str(i * k + j + 1)
cell_2 = rotare_cell(cell_1, k) <...>

cell = np.full((2 * k, 2 * k), '0')
cell[:k, :k] = cell_1 <...>

holes = sorted(get_holes(cell, k), key = lambda x : (x[0], x[1]))
```

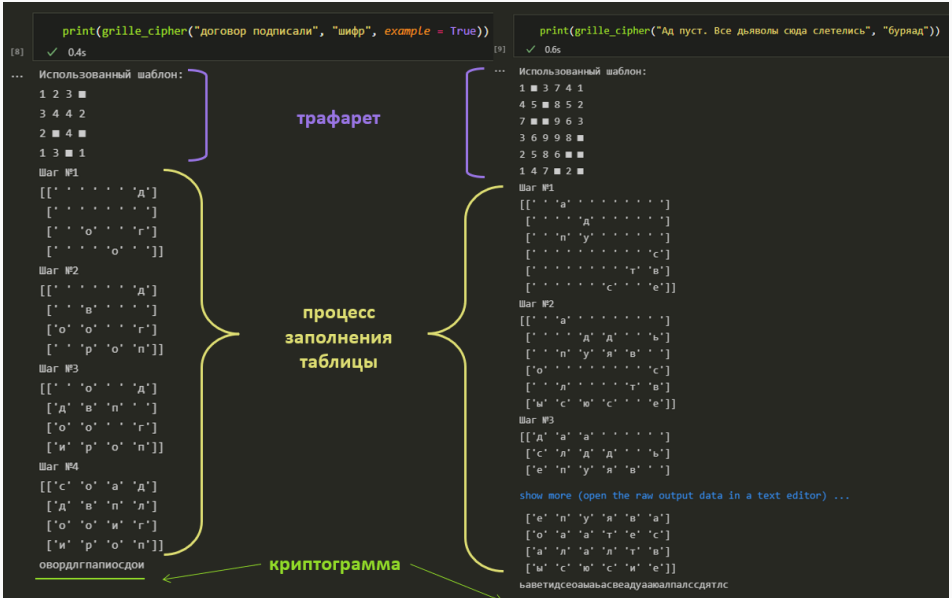
Шифрование с помощью решёток. Фрагменты кода (3)

```
table = np.full((2 * k, 2 * k), ' ') # таблица
template = np.full((2 * k, 2 * k), '0') # трафарет

for i in range(2 * k): # заполняем трафарет
    for j in range(2 * k):
        if (i, j) in holes:
            template[i][j] = '1'

for i in range(4):
    for j in range(k ** 2):
        table[holes[j][0]][holes[j][1]] = mes[i * (k ** 2) + j]
    template = rotare_cell(template, 2 * k) # поворачиваем трафарет
    holes = [(hole[0], hole[1])
             for hole in np.array(np.where(template == '1')).T]
```

Шифрование с помощью решёток. Результаты



Заключение

Таким образом, была достигнула цель, поставленная в начале лабораторной работы:

- Было проведено знакомство с шифром Виженера, а также с шифрами перестановки на примере маршрутного шифрования и шифрования с помощью решёток;
- Были реализованы шифрование методом столбцовой перестановки, шифрование с помощью поворотных решёток и шифр Виженера для русского алфавита.

Спасибо за внимание