Министерство науки и высшего образования РФ

Федеральное государственное автономное образовательное

учреждения высшего образования

Университет науки и технологий МИСИС

«МИСиС»

Отчет по практической работе №3 на тему:

«Маршрутная перестановка»

по дисциплине:

«Методы и средства защиты компьютерной информации»

Направление подготовки: 09.03.03

Семестр 5

|  |  |
| --- | --- |
| Выполнил:    Бобров Михаил Алексеевич  БПИ-20-2  (группа)  22.12.2022  (дата)  Подпись: \_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Проверил:  доцент, к. т. н.  Костин Виталий Николаевич  \_\_\_\_\_\_\_\_\_\_  (дата)  \_\_\_\_\_\_\_\_\_\_  (оценка)  Подпись: \_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

Москва 2022

ОГЛАВЛЕНИЕ

[ВВЕДЕНИЕ 3](#_Toc122629444)

[Определение входных и выходных данных 4](#_Toc122629445)

[1.1 Входные данные 4](#_Toc122629446)

[1.2 Выходные данные 4](#_Toc122629447)

[Алгоритм МАРШРУТНОЙ ПЕРЕстановки 5](#_Toc122629448)

[2.1 Алгоритм шифрования 5](#_Toc122629449)

[2.2 Алгоритм расшифрования 5](#_Toc122629450)

[Блок-схемы алгоритма 6](#_Toc122629451)

[3.1 Блок-схема всего алгоритма 6](#_Toc122629452)

[3.2 Алгоритм шифрования и расшифрования 7](#_Toc122629453)

[Листинг программы 8](#_Toc122629454)

[Тестирование программы 9](#_Toc122629455)

ВВЕДЕНИЕ

Отчёт составлен по результатам выполнения практической работы на тему «Маршрутная перестановка» по дисциплине «Методы и средства защиты компьютерной информации».

Целью работы является шифрование данных методом маршрутной перестановки, расшифровка шифрованных данных, и его реализация на языке программирования Python.

Достижение указанной цели предполагает решение следующих задач:

1. Изучение Маршрутной перестановки;
2. Рассмотрение возможности его применения для данного случая;
3. Составление блок-схем, описывающих работу алгоритма;
4. Проверка корректности работы алгоритма путем его тестирования.

Определение входных и выходных данных

1.1 Входные данные

Ввод данных реализован посредством консоли.

Требования: ввод текста, который читается как строчный формат. Максимальная длина строки - 231 – 1 = 2147483647 значений.

1.2 Выходные данные

Вывод данных реализован посредством консоли.

Выходными данными, согласно требованиям, являются шифрованная и расшифрованная строка.

Алгоритм МАРШРУТНОЙ ПЕРЕстановки

2.1 Алгоритм шифрования

Шифрование маршрутной перестановкой происходит следующим образом: сначала вычисляется количество столбцов и строк по корню длины вводной строки. Если Квадрат корня длины меньше длины строки, то значение корня длины увеличивается на 1, пока квадрат корня длины не будет больше длины строки.

Далее в квадратную таблицу с количеством столбцов и строк равными значению корня длины шифруются значения из вводной строки. Значения берутся по формуле: S \* I + R, где S – корень длины строки, I – множитель, возрастающий с 0 до S - 1, R – номер строки. Зашифрованный текст в виде таблицы можно читать с верхнего левого края сверху вниз. Далее таблица построчно переносится в одну зашифрованную строку.

2.2 Алгоритм расшифрования

Происходит такое же вычисление количества строк и столбцов, что и в пункте 2.1 алгоритм шифрования. Также происходит такая же запись зашифрованной строки в таблицу. При этом таблицу уже можно читать как обычный текст, с верхнего левого края слева направо, построчно. Далее таблица построчно переносится в расшифрованную строку.

Блок-схемы алгоритма

3.1 Блок-схема всего алгоритма

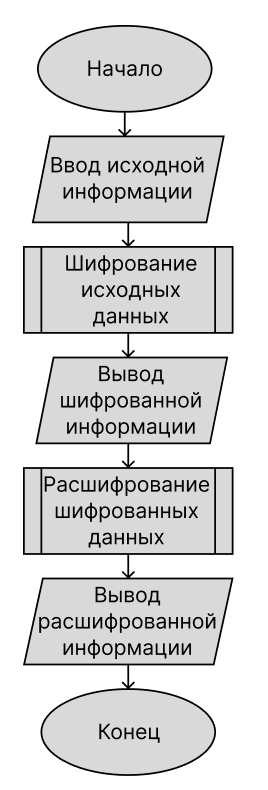


Рисунок 3.1 – блок-схема всего алгоритма

3.2 Алгоритм шифрования и расшифрования

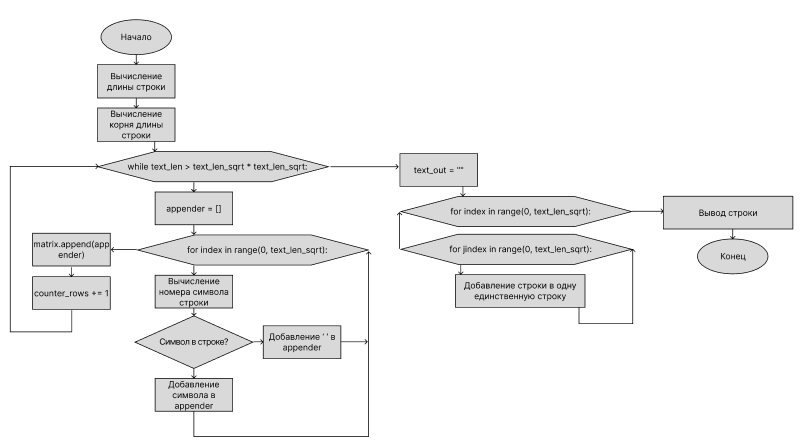


Рисунок 3.2 – алгоритм шифрования и разшифрования

Листинг программы

Ниже приведен листинг программы на языке программирования Python.

import math

def script(text):

    text\_len = len(text)

    text\_len\_sqrt = int(math.sqrt(text\_len))

    while text\_len > text\_len\_sqrt \* text\_len\_sqrt:

        text\_len\_sqrt+=1

    matrix = []

    counter\_rows = 0

    while counter\_rows < text\_len\_sqrt:

        appender = []

        for index in range(0, text\_len\_sqrt):

            number = text\_len\_sqrt \* index + counter\_rows

            if number < text\_len:

                appender.append(text[number])

            else:

                appender.append(' ')

        matrix.append(appender)

        counter\_rows += 1

    text\_out = ""

    for index in range(0, text\_len\_sqrt):

     for jindex in range(0, text\_len\_sqrt):

            text\_out+=str(matrix[index][jindex])

    return text\_out

#Шифровка

user\_text = input("Введите текст: ")

output\_s = script(user\_text)

print(output\_s)

#Расшифровка

output\_r = script(output\_s)

print(output\_r)

Тестирование программы

Входные данные: “Багровый и белый отброшен и скомкан, в зеленый бросали горстями дукаты, а черным ладоням сбежавшихся окон раздали горящие желтые карты”.

Выходные данные:

1) Зашифрованная строка: Бен ряаоизетал вом нхд ыгыи сичясаж.рй за емяле о селдр ил воклиунсо т ытое кыбкгы йбмнгамеоое ркыот жнр иоайрыла як шн с,аврща бе,бт дшаир

2) Расшифрованная строка: Багровый и белый отброшен и скомкан, в зеленый бросали горстями дукаты, а черным ладоням сбежавшихся окон раздали горящие желтые карты.

Изображение выглядит как текст, внутренний, снимок экрана, закрыть

Автоматически созданное описаниеРисунок 5.1 – тестирование программы.