Лабораторная работа 2

Шифры перестановки

Пологов Владислав Александрович

2022 Москва

RUDN University, Moscow, Russian Federation

Цель работы

Цель работы

Реализовать метод маршрутного шифрования.

Реализовать метод шифрования с помощью решёток.

Реализовать метод шифрования с помощью таблицы Виженера.

Описание реализации

Описание реализации

Для реализации алгоритмов использовались средства языка Python.

Каждый метод был реализован отдельной функцией с соответствующим названием. (рис. 1)

Каждая функция принимала в качестве входных параметров строку и пароль для её шифрования. Помимо этого в методе машрутного шифрования необходимо было сообщать функции о длине блока n. А в методе шифрования с помощью решёток на вход ещё подавались сторона изначального квадрата k и координаты прорезей решётки.

Описание реализации

```
def route_encryption(orig_string, pswd, n):···

def lattice_encryption(orig_string, pswd, k, xys):···

def vigenere_encryption(orig_string, pswd):···

print(route_encryption('нельзя недооценивать противника', 'пароль', 6))

print(lattice_encryption('договор подписали', 'шифр', 2, [(0,3), (3,2), (2,3), (2,1)]))

print(vigenere_encryption('криптография серьезная наука', 'математика'))
```

Figure 1: Функции методов шифрования

Реализация

Маршрутное шифрование

Данный способ шифрования разработал фанцунзский математик Франсуа Виет. Его суть заключалась в записи исходного текста в некоторую геометрическую форму(обычно прямоугольник) по некоторому пути, а затем, выписывая символы по другому пути, можно было получить шифртекст. (рис.-fig. 2)

Маршрутное шифрование

В данном способе можно обойтись без составления матрицы. Доступ к строке и её шифрования осуществлялось следующим образом:

$$j * n + nums.index(i)$$

где j — номер строки, n — длина одного модуля или количество столбцов, $nums.index(i)$ — функция, вовзращающая индекс буквы пароля в соответствие с алфавитным порядком.

ШМаршрутное шифрование

```python
def route_encryption(orig_string, pswd, n):
    string = orig_string.replace(' ','')
    m = len(string)//n+bool(len(string)%n)
    string += 'a'*(m*n-len(string))
    nums = [sorted(pswd).index(c) for c in pswd]
    print(nums)
    result = ''
    for i in range(n):
        for j in range(m):
            result += string[j*n + nums.index(i)].upper()
    return result
```

Figure 2: Код маршрутного шифрования

Шифрование с помощью решёток

Данный способ шифрования предложил австрийский криптограф Эдуард Флейснер в 1881 году. Суть заключается в том, что строится матрица размерности 2*k. Далее в матрице вырезаются клетки, содержащие числа от 1 до k^2. Получается решето, которое мы накладываем на наш исходный текст. Алгоритм является иттерационнным и каждая иттерация сопровождается поворотом решётки на 90 градусов. Число k выбирается в соответствии с количеством букв исходного текста. (рис. 3)

Шифрование с помощью решёток

```
def lattice_encryption(orig_string, pswd, k, xys):
    string = orig_string.replace(' ','')
    matr = []
    for i in range(k*2):
        matr.append(['.']*(k*2))
    u = 0
    for i in range(4):
        for x, y in xys:
            matr[x][y] = string[u]
            u+=1
        xys = [(y, 2*k-1-x) for x, y in xys]
    res = ''
    for i in range(k*2):
        for c in matr[i]:
            res += c
    return route_encryption(res, pswd, 2*k)
```

Шифрование с помощью таблицы Виженера

Суть этого метода заключается в том, что имеется таблица, составленная при помощи циклического сдвига букв русского алфавита на одну позицию влево. Для составления шифра используется пароль, который продлевается до конца исходной строки. После, в соответствии с паролем, находится буква в таблице Виженера для шифрования соответствующей буквы исходного текста. (рис. 4)

Шифрование с помощью таблицы Виженера

```python
def vigenere_encryption(orig_string, pswd):
    string = orig_string.replace(' ','')
    pswd = (pswd*(len(string)//len(pswd)+1))[:len(string)]
    alphabet = [chr(c) for c in range(ord('а'), ord('я') + 1)]
    matr = []
    res = ''
    for i in range(32):
        matr.append(alphabet)
        alphabet = alphabet[1:] + [alphabet[0]]
        #print(matr[i])
    for p,s in zip(pswd,string):
        res += matr[ord(s)-ord('а')][ord(p)-ord('а')]
    return res.upper()
```

Figure 4: Код шифрования с помощью таблицы Виженера

Вывод

Вывод

Ознакомились с шифрами перестановок.

Реализовали метод маршрутного шифрования.

Реализовали метод шифрования с помощтю решёток.

Реализовали метод шифрования с помощью таблицы Виженера.