Лабораторная работа №2

Математические основы защиты информации и информационной безопасности

Леонтьева К. А., НПМмд-02-23

27 сентября 2023

Российский университет дружбы народов

Москва, Россия

Цель лабораторной работы

1) Реализовать на языке программирования маршрутное шифрование, шифрование с помощью решеток и таблицу Виженера

Маршрутное шифрование разработал французский математик Франсуа Виет. Открытый текст записывают в некоторую геометрическую фигуру, например, прямоугольник, разбив предварительно текст на блоки, длина которых равна количеству букв в пароле. Блоки располагаются друг под другом. Затем выписывают столбцы в порядке следования букв в пароле по алфавиту.

Шифрование с помощью решеток предложил австрийский криптограф Эдуард Флейснер в 1881 году. Формируется решетка, заполненная цифрами от 1 до 4, из которой удаляются ячейки с разными цифрами. Затем полученная решетка накладывается на аналогичную, но пустую, и, когда заполняются все прорези буквами исходного текста по порядку их следования, решетка поворачивается на 90 градусов и вписывание букв продолжается. Далее подбирается подходящий пароль, выписываются буквы по столбцами. Очередность столбцов определяется алфавитным порядком букв пароля как в маршрутном шифровании.

Шифр Виженера опубликовал в 1585 году французский криптограф Блез Виженер в “Трактате о шифрах”. Он считался нераскрываемым до 1863 года, когда австриец Фридрих Казиски взломал его. Формируется таблица, где в строчках записаны буквы русского алфавита. При переходе от одной строке к другой происходит циклический сдвиг на одну позицию. Пароль записывается с повторениями над буквами сообщения. Далее в горизонтальном алфавите находим букву из исходного текста, в вертикальном - из пароля. На пересечении столбца и строки в таблице распологается нужная буква.

Ход выполнения лабораторной работы

- Реализуем маршрутное шифрование

```
phrase = 'нельзя недооценивать противника'
key = 'пароль'
phrase = list(phrase.replace(" ", ""))
key = list(key)
```

```
m = len(phrase)
n = len(key)
l = m % n

if l < n:
    for i in range(n-l):
        phrase.append('а')

blocks = [phrase[i:i+n] for i in range(0, len(phrase), n)]
blocks
```

```
[['н', 'е', 'л', 'ь', 'з', 'я'],
 ['н', 'е', 'д', 'о', 'о', 'ц'],
 ['е', 'н', 'и', 'в', 'а', 'т'],
 ['ь', 'п', 'р', 'о', 'т', 'и'],
 ['в', 'н', 'и', 'к', 'а', 'а']]
```

```
alphabet = []

for j in range(n):
    alphabet.append(ord(key[j]))

new_alphabet = sorted(range(len(alphabet)), key=alphabet.__getitem__)
```

```
word = []

for g in range(n):
    for h in range(int(len(phrase)/len(key))):
        word.append(blocks[h][new_alphabet[g]])
```

```
print(''.join(word))
```

```
еенпнзоатььовокннеьвлдирияцтиа
```

Figure 1: Рис.1: Маршрутное шифрование

Ход выполнения лабораторной работы

- Реализуем шифрование с помощью решеток (часть 1)

```
import numpy as np
```

```
a_1 = np.array([[1,2],[3,4]])
a_2 = np.rot90(a_1, 3)
a_3 = np.rot90(a_2, 6)
a_4 = np.rot90(a_3, 1)
a_12 = np.concatenate ((a_1, a_2), axis = 1 )
a_34 = np.concatenate ((a_3, a_4), axis = 1 )
a = np.concatenate ((a_12, a_34), axis = 0 )
aa = np.concatenate((a[0], a[1], a[2], a[3]), axis = 0 )

for i in range(4):
    exec(f"i{i+1} = [j for j in range(0, len(aa)) if aa[j]==i+1]")
    exec(f"ind{i+1} = np.random.randint(0,4)")
    exec(f"aa[i{i+1}[ind{i+1}]] = -1")

for i in range(len(aa)):
    if aa[i] != -1:
        aa[i] = 0
aaa = aa.tolist()
```

```
m = [aaa[i:i+4] for i in range(0, len(aa), 4)]
k = [[0,0,0,0],[0,0,0,0], [0,0,0,0], [0,0,0,0]]
phrase = list('договорподписали')
key = list('шифр')

indices = []

m
```

```
[[-1, 0, 0, 0], [-1, 0, 0, -1], [0, -1, 0, 0], [0, 0, 0, 0]]
```

Figure 2: Рис.2: Шифрование с помощью решеток

Ход выполнения лабораторной работы

- Реализуем шифрование с помощью решеток (часть 2)

```
while any(0 in s for s in k) == True:
    for i in range(len(m)):
        for j in range(len(m)):
            if m[i][j] == -1:
                indices.append([i, j])

    for i in range(4):
        k[indices[i][0]][indices[i][1]] = phrase[0]
        phrase = phrase[1:]

    m = np.rot90(m, 3)

    indices = []
```

```
k
```

```
[['д', 'с', 'в', 'о'],
 ['о', 'р', 'о', 'г'],
 ['д', 'о', 'а', 'п'],
 ['л', 'и', 'п', 'и']]
```

```
alph = []

for j in range(len(key)):
    alph.append(ord(key[j]))

new_alph = sorted(range(len(alph)), key=alph.__getitem__)
```

```
word = []

for g in range(len(key)):
    for h in range(len(m)):
        word.append(k[h][new_alph[g]])
```

```
print(''.join(word))
```

```
сроиогпивоапдодл
```

Figure 3: Рис.3: Шифрование с помощью решеток

Ход выполнения лабораторной работы

- Реализуем таблицу Виженера

```
import math
```

```
phrase = 'криптография серьезная наука'
key = 'математика'
phrase = list(phrase.replace(" ", ""))
key = list(key)
```

```
alphabet = []
for i in range(1072,1104):
    alphabet.append(chr(i))

table = list()
for i in range(len(alphabet)):
    table.append(alphabet[i:] + alphabet[:i])
```

```
k = math.floor(len(phrase)/len(key))
key_list = []
for j in range(k):
    key_list.append(key)
m = len(phrase) % len(key)
part_key = key[:m]
key_list.append(part_key)
key_list = sum(key_list, [])
```

```
cipher = []
for g in range(len(phrase)):
    cipher.append( table[alphabet.index(phrase[g])][alphabet.index(key_list[g])] )
```

```
print(''.join(cipher))
```

```
црьфюохшкффягкььчпчалнтшца
```


Figure 4: Рис.4: Таблица Виженера

Вывод

- В ходе выполнения данной лабораторной работы были реализованы маршрутное шифрование, шифрование с помощью решеток и таблица Виженера