Лабораторная работа №2

Шифры перестановки

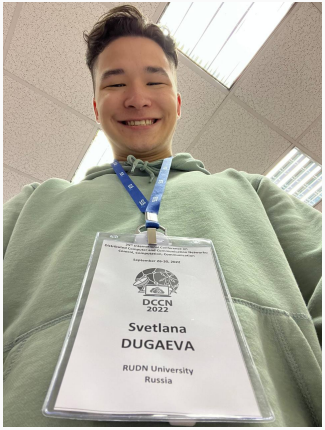
Ли Т.А.

1 октября 2022

Российский университет дружбы народов, Москва, Россия

Докладчик

- Ли Тимофей Александрович
- студент группы НФИмд-02-22, студ. билет 1132223452
- Российский университет дружбы народов
- 1132223452@rudn.ru



Цель работы

Цель данной работы — изучить и программно реализовать шифры перестановки.

Шифры перестановки преобразуют открытый текст в криптограмму путём перестановки его символов. Способ, каким при шифровании переставляются буквы открытого текста, и является ключом шифра. Важным требованием является равенство длин ключа исходного текста.

Выполнение лабораторной работы

Для реализации шифров мы будем использовать Python, так как его синтаксис позволяет быстро реализовать необходимые нам алгоритмы.

Реализация маршрутного шифрования

Код маршрутного шифрования реализуем в виде функции следующего вида:

In [1]:
```
rus='абвгдеёжзиклмнопрстуфхцчшщъыьэюя'
def marsh(text,key,m,n):
    global rus
    textws=text.replace(' ','')
    if len(textws)<m*n:
        textws+=rus[:m*n-len(textws)]
    t=iter(textws)
    matrix=[[next(t) for y in range(m)] for x in range (n)]
    ps=[rus.index(x) for x in key]
    pss=sorted(ps)
    output=''
    for letter in pss:
        for x in range(n):
            output+=matrix[x][ps.index(letter)]
    return output
```

In [7]:
```
print(marsh('нельзя недооценивать противника','пароль',6,5))
```

```
еенпнзоатаьовокннеьвлдирияцтиа
```

Рис. 1: код1

Реализация шифрования с помощью решеток

Шифрование с помощью решеток реализуем в виде функции следующего вида:

```
import numpy as np
k=2
k_2=[x+1 for x in range(k**2)]
matrix=[[0 for x in range(2*k)]for y in range(2*k)]
matrix=np.array(matrix)
for x in range(k**2):
    c=0
    for x in range(k):
        for y in range(k):
            matrix[x][y]=k_2[c]
            c+=1
    matrix=np.rot90(matrix)
ds={k: 0 for k in k_2}
dss={1:2,2:4,3:3,4:3}
for x in range(k**2):
    for y in range(k**2):
        ds[matrix[x][y]]+=1
        if ds[matrix[x][y]]!=dss[matrix[x][y]]:
            matrix[x][y]=-1
        else:
            matrix[x][y]=0
text='договорподписали'
key='шифр'
```

```
ct=0
t=iter(text)
matrixt=[['O' for y in range(k**2)] for x in range(k**2)]
for d in range(4):
    for x in range(k**2):
        for y in range(k**2):
            if matrix[x][y]==0:
                matrixt[x][y]=text[ct]
                ct+=1
    matrix=np.rot90(matrix,-1)
ps=[rus.index(x) for x in key]
pss=sorted(ps)
output=''
for letter in pss:
    for x in range(k**2):
        output+=matrixt[x][ps.index(letter)]
print(output)
```

```
оворддлгпапиосдои
```

Рис. 2: код2

Реализация таблицы Виженера

Таблицу Виженера реализуем в виде функций следующего вида:

In [17]:
```
def genkey(m,key):
    key.replace(' ','')
    m.replace(' ','')
    key=list(key)
    if len(m)==len(key):
        return(key)
    else:
        for i in range(len(m)-len(key)):
            key.append(key[i%len(key)])
    return(''.join(key))
def vig(m,key):
    ct=[]
    m.replace(' ','')
    for i in range(len(m)):
        x=(ord(m[i])+ord(key[i]))%26
        x+=ord('A')
        ct.append(chr(x))
    return(''.join(ct))
```

```
m='letsss go first try'
key='key'
print(vig(m,genkey(m,key)))
```

```
HUDOICJWYJVSNIDJJBU
```

Рис. 3: код3

Выводы

Лабораторная работа выполнена.