Отчёта по лабораторной работе №2

Шифры перестановки

Шутенко Виктория Михайловна

Содержание

Список иллюстраций

1 Цель работы

Приобрести практические навыки работы с шифрами перестановки.

2 Ход работы

1. В первом задании рассматривалось маршрутное шифрование. Для написания использовался высокоуровневый язык программирования python. Для реализации работы шифра создана функция `marsh(text, key, m, n)`, которая принимает на входе текст, ключ и простые числа.

```
rus = 'АБВГДЕЁЖЗИКЛМНОПРСТУФХЦЧШЩЪЫЬЭЮЯ'
def marsh(text, key, m, n):
    global russian
    textws=text.replace(' ','')
    if len(textws)<m*n:
        textws+=rus[:m*n-len(textws)]
    t=iter(textws)
    matrix=[[next(t) for j in range (m)] for i in range (n)]
    ps=[rus.index(i) for i in key]
    pss=sorted(ps)
    output=''
    for l in pss:
        for i in range(n):
            output+=matrix[i][ps.index(l)]
    return output
print(marsh('нельзя недооценивать противника', 'пароль', 6, 5))
```

[27]:
```
rus='АБВГДЕЁЖЗИКЛМНОПРСТУФХЦЧШЩЪЫЬЭЯЮ'
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 l in pss:
        for x in range(n):
            output+=matrix[x][ps.index(l)]
    return output
```

[28]:
```
print(marsh('НЕЛЬЗЯ НЕДООЦЕНИВАТЬ ПРОТИВНИКА', 'ПАРОЛЬ', 6, 5))
```

```
ЕЕНПНЗОАТАЬОВОКННЕЬВЛДИРИЯЦТИА
```

Рис. 2.1: Маршрутное шифрование

2. Для второго задания осуществлялась реализация шифрования с помощью решеток.

Для реализации работы шифра понадобилось библитека numpy:

```
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=[['0' for y in range(2*k)] for x in range(2*k)]
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=[russ.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)
```

[93]:
```
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=[['0' for y in range(2*k)] for x in range(2*k)]
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=[russian.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)
```

```
овордлгпапиосдои
```

Рис. 2.2: Шифрования с помощью решеток

3. В третьем задании использывалось шифрование методом Таблицы Вижинера. Здесь для реализации созданы 2 функции `def genkey(m, key)`, создающая ключ и `def vig(m, key)` для перехода между кодировками.

```
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='letss goo sleep'
key='key'
print(vig(m,genkey(m,key)))
```

[94]:
```
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='letss goo sleep'
key='key'
print(vig(m,genkey(m,key)))
```

```
HUDOIXCEYJIVAUZ
```

Рис. 2.3: Таблица Виженера