

# Защита лабораторной работы №2. Шифры перестановки

---

Асеинова Елизавета

2023 Sep 23th

RUDN University, Moscow, Russian Federation

## Результат выполнения лабораторной работы №2

---

## Цель выполнения лабораторной работы

Цель данной работы - ознакомиться с шифрами перестановки: маршрутное шифрование, шифрование с помощью решеток и таблица Вижинера, а также научиться применять их на практике.

Прописан код для маршрутного шифрования на языке программирования Python.

```
▶ alphabet = 'абвгдеёжзиклмнопрстуфхцчщъыьэюя'  
def shifr(text, key, m, n):  
    global alphabet  
    text_n = text.replace(' ', '')  
    if len(text_n) < (m*n):  
        text_n += alphabet[:m*n - len(text_n)]  
    ch = iter(text_n)  
    matr = [[next(ch) for y in range(m)] for x in range(n)]  
    passw = [alphabet.index(x) for x in key]  
    passw_sort = sorted(passw)  
    result = ''  
    for char in passw_sort:  
        for x in range(n):  
            result += matr[x][passw.index(char)]  
    return result
```

Figure 1: Маршрутное шифрование

Выводим на экран результат применения шифра

```
✓ [8] print(shifr('нельзя недооценивать противника', 'пароль', 6, 5))  
0  
нелпнзоатаьовокннеьвдиряцтиа
```

Figure 2: Результат применения

## Результат выполнения лабораторной работы

Прописан код для шифрования с помощью решеток на языке программирования Python.

```
[10] import numpy as np
k = 2
k_2 = [x+1 for x in range(k**2)]
matr = [[0 for x in range(2*k)] for y in range(2*k)]
matr = np.array(matr)
for x in range(k**2):
    c=0
    for x in range(k):
        for y in range(k):
            matr[x][y] = k_2[c]
            c+=1
    matr = np.rot90(matr)
mv = { k: 0 for k in k_2}
mv_2 = {1:2, 2:4, 3:3, 4:3}
for x in range(k**2):
    for y in range(k**2):
        mv[matr[x][y]]+=1
        if mv[matr[x][y]]!= mv_2[matr[x][y]]:
            matr[x][y] = -1
        else:
            matr[x][y] = 0
```

Figure 3: Шифрование с помощью решеток

Выводим на экран результат применения.

```
[13] text = 'договорподписали'
     key = 'шифр'

     ct = 0
     t = iter(text)
     matr2 = [['0' for y in range(k**2)] for x in range(k**2)]
     for v in range(4):
         for x in range(k**2):
             for y in range(k**2):
                 if matr[x][y]==0:
                     matr2[x][y] = text[ct]
                     ct+=1
     matr = np.rot90(matr, -1)
     passw = [alphabet.index(x) for x in key]
     passw_sort = sorted(passw)
     result = ''
     for char in passw_sort:
         for x in range(k**2):
             result+=matr2[x][passw.index(char)]
     print(result)
```

овордлгпапиосдои

Figure 4: Результат применения

## Результат выполнения лабораторной работы

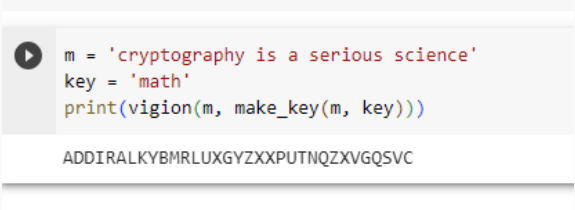
Прописан код для использования таблицы Вижинера на языке программирования Python.

```
▶ def make_key(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 vigion(m, key):  
    v = []  
    m.replace(' ', '')  
    for i in range(len(m)):  
        x = (ord(m[i]) + ord(key[i])) % 26  
        x += ord('A')  
        v.append(chr(x))  
    return (''.join(v))
```

Figure 5: Таблица Вижинера



Выводим на экран результат применения.



```
m = 'cryptography is a serious science'  
key = 'math'  
print(vigion(m, make_key(m, key)))
```

The image shows a code execution environment. On the left, there is a play button icon. To its right, a code block contains three lines of Python code. The first line assigns the string 'cryptography is a serious science' to the variable 'm'. The second line assigns the string 'math' to the variable 'key'. The third line calls the function 'vigion' with arguments 'm' and 'make\_key(m, key)', and the result is printed. Below the code block, the output of the program is displayed as a single line of text.

ADDIRALKYBMRLUXGYZXXPUTNQZXVGQ SVC

Figure 6: Результат применения

В рамках данной лабораторной работы было произведено ознакомление с маршрутным шифрованием, шифрованием с помощью решеток и таблицей Вижинера. Шифры были реализованы на языке программирования Python.