

# Лабораторная работа №8

---

“Низамова Альфия Айдаровна. НФИбд-01-20”<sup>1</sup>

23 сентября, 2023, Москва, Россия

<sup>1</sup>Российский Университет Дружбы Народов

# Цель работы

---

## Цель работы:

Освоить на практике применение режима однократного гаммирования на примере кодирования различных исходных текстов одним ключом.

## Задание

---

## Задание:

Задание:

Два текста кодируются одним ключом (однократное гаммирование). Требуется не зная ключа и не стремясь его определить, прочесть оба текста. Необходимо разработать приложение, позволяющее шифровать и дешифровать тексты P1 и P2 в режиме однократного гаммирования. Приложение должно определить вид шифротекстов C1 и C2 обоих текстов P1 и P2 при известном ключе; Необходимо определить и выразить аналитически способ, при котором злоумышленник может прочесть оба текста, не зная ключа и не стремясь его определить.

## **Ход лабораторной работы.**

---

Для написания программы использовала язык Python.  
Для начала импортировала необходимые библиотеки и  
задала две исходные строки равной длины(рис.1)

```
In [1]: ► import numpy as np  
import operator as op  
import sys
```

```
In [5]: ► p1 = "текст первый"  
print(len(p1))  
p2 = "второй текст"  
print(len(p2))
```

```
12  
12
```

**Рис. 1: Рис. 1**

Написала функцию, которая определяет вид шифротекстов обеих строк при известном ключе(рис.2)

```
def encrypt(t1, t2):
    print("text 1:", t1)
    newtext1 = []
    for i in t1:
        newtext1.append(i.encode("cp1251").hex())
    print("text 1 in 16:", newtext1)
    print("text 2:", t2)
    newtext2 = []
    for i in t2:
        newtext2.append(i.encode("cp1251").hex())
    print("text 2 in 16:", newtext2)
    r = np.random.randint(0,255,len(t1))
    key = [hex(i)[2:] for i in r]
    newkey = []
    for i in key:
        newkey.append(i.encode("cp1251").hex().upper())
    print("key in 16:", newkey)
    xortext1 = []
    for i in range(len(newtext1)):
        xortext1.append("{:02x}".format(int(key[i], 16) ^ int(newtext1[i], 16)))
    print("cypher text1 in 16:", xortext1)
    en_t1 = bytearray.fromhex("".join(xortext1)).decode("cp1251")
    print("cypher text1:", en_t1)
    xortext2 = []
    for i in range(len(newtext2)):
        xortext2.append("{:02x}".format(int(key[i], 16) ^ int(newtext2[i], 16)))
```



## Вывод первой функции(рис.3)

```
k, t1, et1, t2, et2 = encrypt(p1, p2)

text 1: текст первый
text 1 in 16: ['f2', 'e5', 'ea', 'f1', 'f2', '20', 'ef', 'e5', 'f0', 'e
2', 'fb', 'e9']
text 2: второй текст
text 2 in 16: ['e2', 'f2', 'ee', 'f0', 'ee', 'e9', '20', 'f2', 'e5', 'e
a', 'f1', 'f2']
key in 16: ['3939', '6534', '6131', '3535', '6465', '6438', '3462', '33
34', '6161', '6132', '6134', '3734']
cypher text1 in 16: ['6b', '01', '4b', 'a4', '2c', 'f8', 'a4', 'd1', '5
a', '40', '5f', '9d']
cypher text1: k███,ш██CZ@_ќ
cypher text2 in 16: ['7b', '16', '4f', 'a5', '30', '31', '6b', 'c6', '4
f', '48', '55', '86']
cypher text2: {00Г01kЖОНУ†
```

Рис. 3: Рис. 3

Написала функцию, которая при известных двух шифротекстах и одном открытом тексте находит вид второго открытого текста без ключа (рис.4)

```
def dc(c1, c2, p1):  
    print("cypher text1:", c1)  
    newc1 = []  
    for i in c1:  
        newc1.append(i.encode("cp1251").hex())  
    print("cypher text1 in 16:", newc1)  
  
    print("cypher text2:", c2)  
    newc2 = []  
    for i in c2:  
        newc2.append(i.encode("cp1251").hex())  
    print("cypher text2 in 16:", newc2)  
  
    print("open text1:", p1)  
    newp1 = []  
    for i in p1:  
        newp1.append(i.encode("cp1251").hex())  
    print("open text1 in 16:", newp1)  
  
    xortmp = []  
    sp2 = []  
  
    for i in range(len(p1)):
```

## Вывод второй функции(рис.5)

```
dc(et1, et2,p1)
```

```
cypher text1: k@K#ш#CZ@_k
```

```
cypher text1 in 16: ['6b', '01', '4b', 'a4', '2c', 'f8', 'a4', 'd1', '5a', '40', '5f', '9d']
```

```
cypher text2: {@0f01kжонu†
```

```
cypher text2 in 16: ['7b', '16', '4f', 'a5', '30', '31', '6b', 'c6', '4f', '48', '55', '86']
```

```
open text1: текст первый
```

```
open text1 in 16: ['f2', 'e5', 'ea', 'f1', 'f2', '20', 'ef', 'e5', 'f0', 'e2', 'fb', 'e9']
```

```
open text2 in 16: ['e2', 'f2', 'ee', 'f0', 'ee', 'e9', '20', 'f2', 'e5', 'ea', 'f1', 'f2']
```

```
open text2: второй текст
```

```
('текст первый', 'второй текст')
```

**Рис. 5:** Рис. 5

## **Выводы**

---

В ходе лабораторной работы мне удалось освоить на практике применение режима однократного гаммирования на примере кодирования различных исходных текстов одним ключом.