

# **Лабораторная работа №7**

**Основы информационной безопасности**

Колчева Юлия Вячеславовна

# Содержание

1	Цель работы	5
2	Теоритическое введение	6
3	Выполнение лабораторной работы	7
4	Выводы	9
5	Список литературы	10

## Список иллюстраций

3.1	Код программы . . . . .	7
3.2	Результат работы . . . . .	8

## Список таблиц

# **1 Цель работы**

Освоить на практике применение режима однократного гаммирования.

## 2 Теоритическое введение

Гаммирование - наложение (снятие) на открытые (зашифрованные) данные последовательности элементов других данных, полученной с помощью некоторого криптографического алгоритма, для получения зашифрованных (открытых) данных.

Основная формула, необходимая для реализации однократного гаммирования:

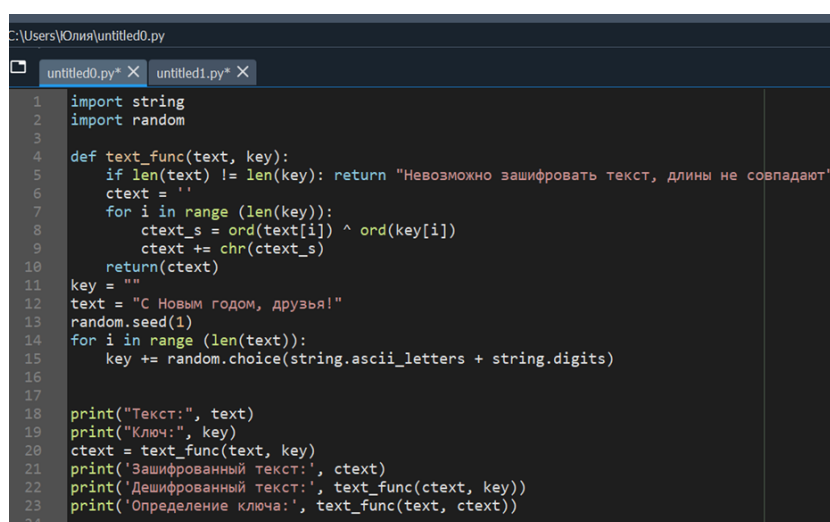
$C_i = P_i \text{ XOR } K_i$ , где  $C_i$  -  $i$ -й символ зашифрованного текста,  $P_i$  -  $i$ -й символ открытого текста,  $K_i$  -  $i$ -й символ ключа. Аналогичным образом можно найти ключ:  $K_i = C_i \text{ XOR } P_i$ .

Необходимые и достаточные условия абсолютной стойкости шифра:

- длина открытого текста равна длине ключа
- ключ должен использоваться однократно
- ключ должен быть полностью случаен

### 3 Выполнение лабораторной работы

В ходе выполнения данной лабораторной работы я освоила на практике применение режима однократного гаммирования.(рис. [3.2])



```
1 import string
2 import random
3
4 def text_func(text, key):
5     if len(text) != len(key): return "Невозможно зашифровать текст, длины не совпадают"
6     ctext = ''
7     for i in range (len(key)):
8         ctext_s = ord(text[i]) ^ ord(key[i])
9         ctext += chr(ctext_s)
10    return(ctext)
11
12 key = ""
13 text = "С Новым годом, друзья!"
14 random.seed(1)
15 for i in range (len(text)):
16     key += random.choice(string.ascii_letters + string.digits)
17
18 print("Текст:", text)
19 print("Ключ:", key)
20 ctext = text_func(text, key)
21 print('Зашифрованный текст:', ctext)
22 print('Дешифрованный текст:', text_func(ctext, key))
23 print('Определение ключа:', text_func(text, ctext))
24
```

Рис. 3.1: Код программы

1-2 строки: импорт необходимых библиотек

4-10 строки: функция, реализующая сложение по модулю два двух строк

12: строка открытый/исходный текст

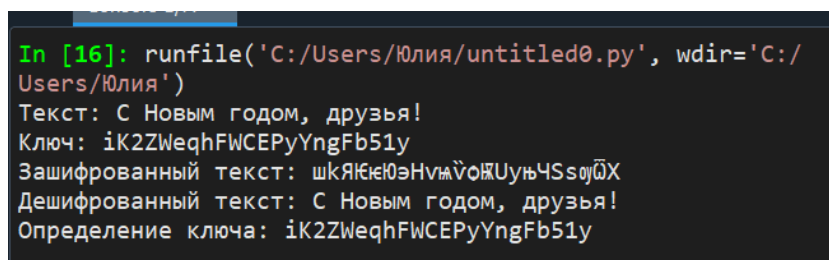
14-15: создание ключа той же длины, что и открытый текст

20: получение шифротекста с помощью функции, созданной ранее, при условии, что известны открытый текст и ключ

22: получение открытого текста с помощью функции, созданной ранее, при условии, что известны шифротекст и ключ

23: получение ключа с помощью функции, созданной ранее, при условии, что известны открытый текст и шифротекст

Результат работы программы можно увидеть на следующем скриншоте (рис. [??])



```
In [16]: runfile('C:/Users/Юлия/untitled0.py', wdir='C:/
Users/Юлия')
Текст: С Новым годом, друзья!
Ключ: iK2ZWeqhFWCEPyYngFb51y
Зашифрованный текст: шкЯКёКёЮэHvм\`oЖUуъ4Ss0jQX
Дешифрованный текст: С Новым годом, друзья!
Определение ключа: iK2ZWeqhFWCEPyYngFb51y
```

Рис. 3.2: Результат работы



## **4 Выводы**

В ходе выполнения данной лабораторной работы я освоила на практике применение режима однократного гаммирования.

## 5 Список литературы

Лабораторная работа №7

Однократное гаммирование [Электронный ресурс]. URL: [https://esystem.rudn.ru/pluginfile.php/1651639/mod\\_resource/content/2/007-lab\\_cryptogamma.pdf](https://esystem.rudn.ru/pluginfile.php/1651639/mod_resource/content/2/007-lab_cryptogamma.pdf).