

# Отчет по лабораторной работе №8

---

Перельгин Сергей Викторович

## Цель работы

---

## Цель лабораторной работы

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

# **Выполнение лабораторной работы**

---

## Задачи лабораторной работы №8

1. Не зная ключа и не стремясь его определить, прочитать оба исходных текста
2. Разработать приложение, позволяющее шифровать и дешифровать тексты в режиме однократного гаммирования
3. Определить и выразить аналитически способ, при котором злоумышленник может прочитать оба текста, не зная ключа и не стремясь его определить.

Гаммирование представляет собой наложение (снятие) на открытые (зашифрованные) данные последовательности элементов других данных, полученной с помощью некоторого криптографического алгоритма, для получения зашифрованных (открытых) данных. Иными словами, наложение гаммы — это сложение её элементов с элементами открытого (закрытого) текста по некоторому фиксированному модулю, значение которого представляет собой известную часть алгоритма шифрования.

В соответствии с теорией криптоанализа, если в методе шифрования используется однократная вероятностная гамма (однократное гаммирование) той же длины, что и подлежащий сокрытию текст, то текст нельзя раскрыть. Даже при раскрытии части последовательности гаммы нельзя получить информацию о всём скрываемом тексте. Наложение гаммы по сути представляет собой выполнение операции сложения по модулю 2 (XOR) (обозначаемая знаком  $\boxplus$ ) между элементами гаммы и элементами подлежащего сокрытию текста.

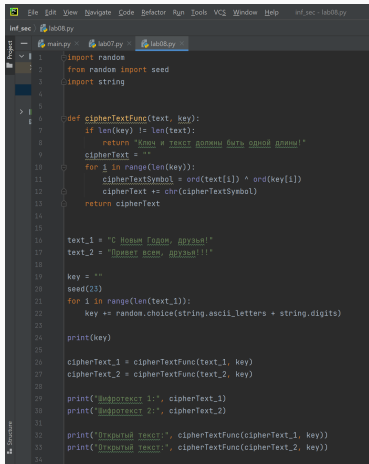
Создал функцию, которая осуществляет однократное гаммирование посредством побитового XOR.



Создал две строки одинаковой длины. Далее я создал ключ той же длины, что и открытые тексты, получил шифротексты с помощью функции, созданной ранее, при условии, что известны открытые тексты и ключ. После этого получил открытый текст с помощью функции, созданной ранее, при условии, что известны шифротекст и ключ. Затем я получил открытые тексты с помощью функции, созданной ранее, при условии, что известны шифротексты и ключ. Сложил по модулю два два шифротекста и получил открытые тексты с помощью функции, созданной ранее

После этого я получил части первого открытого текста (срез), а также часть второго текста (на тех позициях, на которых расположены символы части первого открытого текста) с помощью функции, созданной ранее, при условии, что известны оба шифротекста и часть первого открытого текста.

# Код программы (часть 1)



```
inf sec | lab08.py
main.py | lab07.py | lab08.py
1 import random
2 from random import seed
3 import string
4
5
6 def cipherTextFunc(text, key):
7     if len(key) != len(text):
8         return "Ключ и текст должны быть одной длины!"
9     cipherText = ""
10    for i in range(len(key)):
11        cipherTextSymbol = ord(text[i]) ^ ord(key[i])
12        cipherText += chr(cipherTextSymbol)
13    return cipherText
14
15
16 text_1 = "С Новым годом, друзья!"
17 text_2 = "Привет всем, друзья!!!"
18
19 key = ""
20 seed(23)
21 for i in range(len(text_1)):
22     key += random.choice(string.ascii_letters + string.digits)
23
24 print(key)
25
26 cipherText_1 = cipherTextFunc(text_1, key)
27 cipherText_2 = cipherTextFunc(text_2, key)
28
29 print("Зашифрованный текст 1:", cipherText_1)
30 print("Зашифрованный текст 2:", cipherText_2)
31
32 print("Открытый текст:", cipherTextFunc(cipherText_1, key))
33 print("Открытый текст:", cipherTextFunc(cipherText_2, key))
34
```

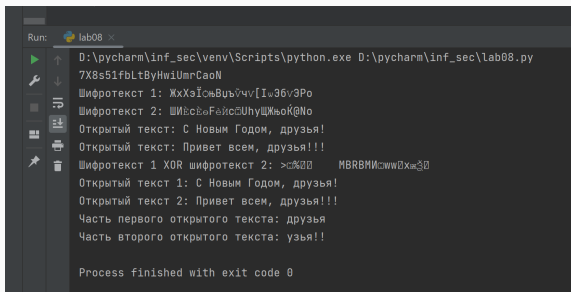
Рис. 1: Код (часть 1)

## Код программы (часть 2)

```
34
35 cipherText_XOR = cipherTextFunc(cipherText_1, cipherText_2)
36 print("Шифротекст 1 XOR шифротекст 2:", cipherText_XOR)
37
38 print("Открытый текст 1:", cipherTextFunc(cipherText_XOR, text_2))
39 print("Открытый текст 2:", cipherTextFunc(cipherText_XOR, text_1))
40
41 txt1 = text_1[15:21]
42 print("Часть первого открытого текста:", txt1)
43
44 cipherTxt2 = cipherTextFunc(cipherText_1[15:21], cipherText_2[15:21])
45 print("Часть второго открытого текста:", cipherTextFunc(cipherTxt2, txt1))
46
47
```

Рис. 2: Код (часть 2)

# Результат работы программы



```
Run: lab08 <
D:\pycharm\inf_sec\venv\Scripts\python.exe D:\pycharm\inf_sec\lab08.py
7X8s51fbLtByHwiUmrCaoN
Шифротекст 1: ЖхХэЇсѣВуѣѣчv[Iu36v3Po
Шифротекст 2: ШИёсёёFeйсёUhyЩЖьёй@No
Открытый текст: С Новым Годом, друзья!
Открытый текст: Привет всем, друзья!!!
Шифротекст 1 XOR шифротекст 2: >с%ёё MBRBMHсwwёXмёё
Открытый текст 1: С Новым Годом, друзья!
Открытый текст 2: Привет всем, друзья!!!
Часть первого открытого текста: друзья
Часть второго открытого текста: узя!!

Process finished with exit code 0
```

Рис. 3: Вывод программы

## Выводы

---

В ходе выполнения данной лабораторной работы я освоил на практике применение режима однократного гаммирования на примере кодирования различных исходных текстов одним ключом.

Спасибо за внимание!