

# **Отчёт по лабораторной работе №7**

**Элементы криптографии. Однократное гаммирование**

Ханина Ирина Владимировна, НБИбд-02-18

# Содержание

1	Цель работы	5
2	Задание	6
3	Теоретическое введение	7
4	Выполнение лабораторной работы	8
5	Выводы	9
	Список литературы	10

## List of Tables

# List of Figures

4.1 Рис 1. Код приложения . . . . . 8

# **1 Цель работы**

Освоить на практике применение режима однократного гаммирования.

## 2 Задание

Нужно подобрать ключ, чтобы получить сообщение «С Новым Годом, друзья!». Требуется разработать приложение, позволяющее шифровать и дешифровать данные в режиме однократного гаммирования. Приложение должно:

1. Определить вид шифротекста при известном ключе и известном открытом тексте.
2. Определить ключ, с помощью которого шифротекст может быть преобразован в некоторый фрагмент текста, представляющий собой один из возможных вариантов прочтения открытого текста.

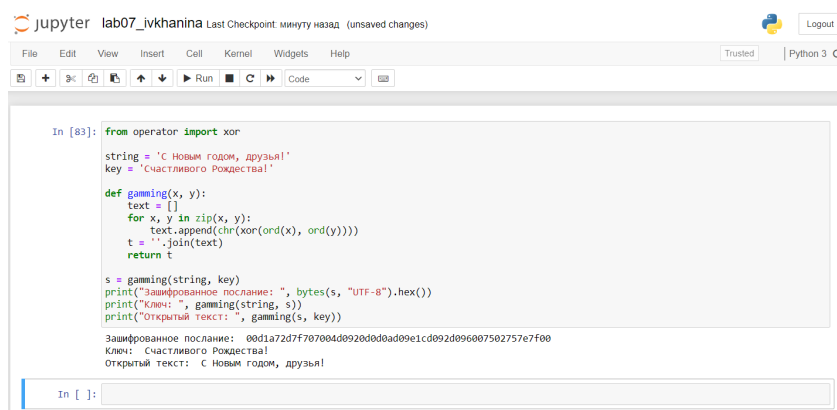
### 3 Теоретическое введение

Гаммирование или Шифр XOR - это наложение или снятие на открытые или зашифрованные данные последовательности элементов других данных, полученной с помощью некоторого криптографического алгоритма, для получения зашифрованных или открытых данных. С точки зрения теории криптоанализа метод шифрования однократной вероятностной гаммой (однократного гаммирования) той же длины, что и открытый текст, является невскрываемым. Наложение гаммы по сути представляет собой выполнение операции сложения по модулю 2 (XOR) между элементами гаммы и элементами подлежащего сокрытию текста. Данный метод шифрования является симметричным, так как двойное прибавление одной и той же величины по модулю 2 восстанавливает исходное значение, а шифрование и расшифрование выполняется одной и той же программой.

Необходимые и достаточные условия абсолютной стойкости шифра: - полная случайность ключа; - равенство длин ключа и открытого текста; - однократное использование ключа. [1]

## 4 Выполнение лабораторной работы

Я разработала приложение на языке Python. Запуск программы производился в Jupiter Notepad. Ключ “Счастливого Рождества!” был подобран случайно. Длина ключа и открытого текста равны. Для операции сложения по модулю 2 я использовала функцию `xor()` из модуля `operator`. (рис. 1)



```
In [83]: from operator import xor

string = 'С Новым годом, друзья!'
key = 'Счастливого Рождества!'

def gamming(x, y):
    text = []
    for x, y in zip(x, y):
        text.append(chr(xor(ord(x), ord(y))))
    t = ''.join(text)
    return t

s = gamming(string, key)
print("Зашифрованное послание: ", bytes(s, "UTF-8").hex())
print("Ключ: ", gamming(string, s))
print("Открытый текст: ", gamming(s, key))

Зашифрованное послание:  00d1a72d7f707004d0920d0d0ad09e1cd092d096007502757e7f00
Ключ:  Счастливого Рождества!
Открытый текст:  С Новым годом, друзья!
```

Figure 4.1: Рис 1. Код приложения



## **5 Выводы**

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

# Список литературы

1. Программные средства защиты информации