

# Презентация по лабораторной работе №7

Элементы криптографии. Однократное гаммирование

---

Ибатулина Д.Э.

10 мая 2024

Российский университет дружбы народов, Москва, Россия

## Информация

---

- Ибатулина Дарья Эдуардовна
- студентка группы НКАбд-01-22
- факультет физико-математических и естественных наук
- Российский университет дружбы народов
- [deibatulina.github.io](https://deibatulina.github.io)
- <https://github.com/deibatulina>

## Вводная часть

---

Решение задач шифрования является очень важным умением для специалиста по информационной безопасности.

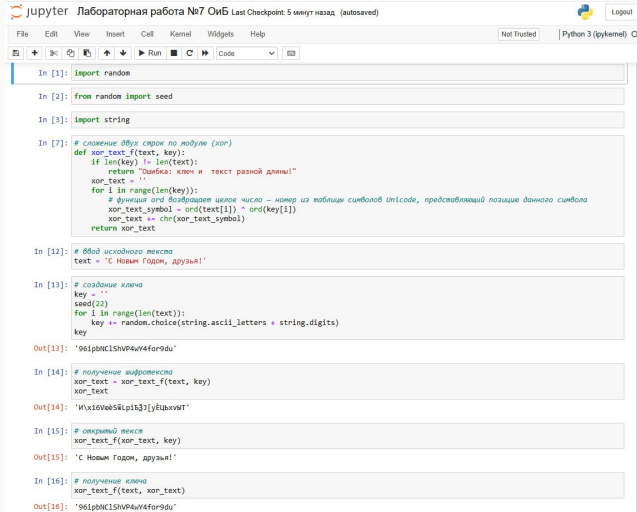
Освоить на практике применение режима однократного гаммирования.

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 и возможных вариантов прочтения открытого текста.

## Выполнение





The image shows a JupyterLab interface with a single notebook. The top bar indicates the notebook is titled "Лабораторная работа №7 ОИБ" and shows the last checkpoint was 5 minutes ago. The interface includes a menu bar (File, Edit, View, Insert, Cell, Kernel, Widgets, Help) and a toolbar with icons for file operations, cell navigation, and execution. The code is written in Python 3 (ipykernel). The script implements a simple XOR cipher. It starts by importing the 'random' module. Then, it defines a function 'xor\_text\_f' that takes a text and a key as input. The function checks if the lengths of the text and key are equal; if not, it returns an error message. If they are equal, it iterates over each character in the text, XORs it with the corresponding character in the key, and builds a new string. The main part of the script demonstrates the process: it creates a key, seeds the random number generator, and generates a key of the same length as the text "С Новым Годом, друзья!". It then shows the original text, the XORed ciphertext, and finally the decrypted text, which matches the original.

```
In [1]: import random

In [2]: from random import seed

In [3]: import string

In [7]: # сложение двух строк по модулю (xor)
def xor_text_f(text, key):
    if len(key) != len(text):
        return "Ошибка: ключ и текст разной длины!"
    xor_text = ''
    for i in range(len(key)):
        # функция ord возвращает целое число - номер из таблицы символов Unicode, представляющий позицию данного символа
        xor_text_symbol = ord(text[i]) ^ ord(key[i])
        xor_text += chr(xor_text_symbol)
    return xor_text

In [12]: # ввод исходного текста
text = 'С Новым Годом, друзья!'

In [13]: # создание ключа
key = ''
seed(22)
for i in range(len(text)):
    key += random.choice(string.ascii_letters + string.digits)
key

Out[13]: '961pbNC1ShVP4wY4for9du'

In [14]: # получение шифротекста
xor_text = xor_text_f(text, key)
xor_text

Out[14]: '\x16Veb5\Rp1t33[yEjxvWT'

In [15]: # открытый текст
xor_text_f(xor_text, key)

Out[15]: 'С Новым Годом, друзья!'

In [16]: # получение ключа
xor_text_f(text, xor_text)

Out[16]: '961pbNC1ShVP4wY4for9du'
```

Рис. 1: Программный код



В ходе выполнения лабораторной работы я научилась зашифровывать и дешифровывать сообщения путем применения однократного гаммирования, познакомилась с этим способом в криптографии.