

# Презентация по лабораторной работе №8

Элементы криптографии. Шифрование (кодирование) различных исходных текстов одним ключом

---

Ибатулина Д.Э.

17 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.

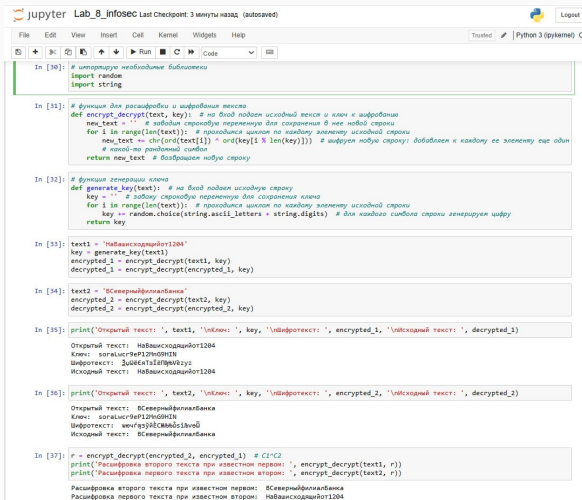
Освоить на практике применение режима однократного гаммирования на примере кодирования различных исходных текстов одним ключом.

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  $P_1$  и  $P_2$  в режиме однократного гаммирования. Приложение должно определить вид шифротекстов  $C_1$  и  $C_2$  обоих текстов  $P_1$  и  $P_2$  при известном ключе. Необходимо определить и выразить аналитически способ, при котором злоумышленник может прочесть оба текста, не зная ключа и не стремясь его определить.

## Выполнение

---





```
Jupyter Lab_8_infosec Last Checkpoint: 3 минуты назад (autosaved)
File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (pykernel)
In [30]: # импортируем необходимые библиотеки
import random
import string

In [31]: # функция для расшифровки и шифрования текста
def encrypt_decrypt(text, key): # на вход подаем исходный текст и ключ к шифрованию
    new_text = '' # задаем строковую переменную для сохранения в нее новой строки
    for i in range(len(text)): # проходимся циклом по каждому элементу исходной строки
        new_text += chr(ord(text[i]) + ord(key[i % len(key)])) # шифруем новую строку: добавляем к каждому ее элементу еще один
        # какой-то случайный символ
    return new_text # возвращаем новую строку

In [32]: # функция генерации ключа
def generate_key(text): # на вход подаем исходную строку
    key = '' # задаем строковую переменную для сохранения ключа
    for i in range(len(text)): # проходимся циклом по каждому элементу исходной строки
        key += random.choice(string.ascii_letters + string.digits) # для каждого символа строки генерируем цифру
    return key

In [33]: text1 = 'Наваинскодщайот1204'
key = generate_key(text1)
encrypted_1 = encrypt_decrypt(text1, key)
decrypted_1 = encrypt_decrypt(encrypted_1, key)

In [34]: text2 = 'ВСеверныйФилиалБанка'
encrypted_2 = encrypt_decrypt(text2, key)
decrypted_2 = encrypt_decrypt(encrypted_2, key)

In [35]: print('Открытый текст: ', text1, '\nКлюч: ', key, '\nШифротекст: ', encrypted_1, '\nИсходный текст: ', decrypted_1)

Открытый текст: Наваинскодщайот1204
Ключ: soqaluc9eP12fmg0HIN
Шифротекст: 3u00EAt1d8PwV8zuz
Исходный текст: Наваинскодщайот1204

In [36]: print('Открытый текст: ', text2, '\nКлюч: ', key, '\nШифротекст: ', encrypted_2, '\nИсходный текст: ', decrypted_2)

Открытый текст: ВСеверныйФилиалБанка
Ключ: soqaluc9eP12fmg0HIN
Шифротекст: wcnf859dECM8d0s19v00
Исходный текст: ВСеверныйФилиалБанка

In [37]: r = encrypt_decrypt(encrypted_2, encrypted_1) # C1^C2
print('Расшифровка второго текста при известном первом: ', encrypt_decrypt(text1, r))
print('Расшифровка первого текста при известном втором: ', encrypt_decrypt(text2, r))

Расшифровка второго текста при известном первом: ВСеверныйФилиалБанка
Расшифровка первого текста при известном втором: Наваинскодщайот1204
```

Рис. 1: Программный код на ЯП Python

В ходе выполнения лабораторной работы я научилась шифровать различные исходные тексты одним ключом.