

# Презентация по лабораторной работе № 8

Информационная безопасность

---

Адоле Фейт

19.10.2023

Российский университет дружбы народов, Москва, Россия

## Информация

---

- Адоле Фейт Эне
- студент группы НПМбд-02-20
- Факультет физико-математических и естественных наук
- Российский университет дружбы народов

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



Я начала с импорта необходимых библиотек. Затем я реализовала функцию для сложения по модулю два двух строк. Открытые или исходные тексты имели одинаковую длину. После этого я создала ключ такой же длины, что и открытые тексты. С использованием ранее созданной функции я получила шифротексты, предполагая знание как открытых текстов, так и ключа. Точно так же я извлекла открытые тексты с использованием ранее созданной функции, предполагая знание как шифротекстов, так и ключа.

Кроме того, я выполнила сложение по модулю два двух шифротекстов с использованием ранее определенной функции. Кроме того, я получила открытые тексты при условии знания как обоих шифротекстов, так и одного из открытых текстов. Также я извлекла сегмент первого открытого текста с помощью среза. Наконец, я получила сегмент второго текста, расположенный на позициях символов сегмента первого открытого текста, с использованием ранее созданной функции, предполагая знание как обоих шифротекстов, так и части первого открытого текста.

# Выполнение лабораторной работы 3

```
File Edit View Insert Cell Kernel Help
In [1]: import random
        from random import seed
        import string

In [7]: def cipher_text_function(text, key):
        if len(key) != len(text):
            return "Ключ и текст должны быть одной длины!"
        cipher_text = ""
        for i in range(len(key)):
            cipher_text_symbol = ord(text[i]) ^ ord(key[i])
            cipher_text += chr(cipher_text_symbol)
        return cipher_text

In [8]: text = "С Новым годом, друзья!"

In [9]: key = ""
        seed(23)
        for i in range(len(text)):
            key += random.choice(string.ascii_letters + string.digits)
        print(key)
        7X8s51fbtSyHdUmrCao

In [11]: cipher_text = cipher_text_function(text, key)
         print("Шифротекст:", cipher_text)
         Шифротекст: K0x010u8RvWv[1w8V3P

In [12]: print("Открытый текст:", cipher_text_function(cipher_text, key))
         Открытый текст: С Новым годом, друзья

In [13]: print("Ключ:", cipher_text_function(text, cipher_text))
         Ключ: 7X8s51fbtSyHdUmrCao
```

Рис. 1: Приложение, реализующее режим однократного гаммирования для двух текстов одним ключом, Часть 1

```
File Edit View Insert Cell Kernel Help
In [1]: import random
        from random import seed
        import string

In [7]: def cipher_text_function(text, key):
        if len(key) != len(text):
            return "Ключ и текст должны быть одной длины!"
        cipher_text = ""
        for i in range(len(key)):
            cipher_text_symbol = ord(text[i]) ^ ord(key[i])
            cipher_text += chr(cipher_text_symbol)
        return cipher_text

In [8]: text = "С Новым годом, друзья!"

In [9]: key = ""
        seed(23)
        for i in range(len(text)):
            key += random.choice(string.ascii_letters + string.digits)
        print(key)
        7X8s51fbtSyHdUmrCao

In [11]: cipher_text = cipher_text_function(text, key)
         print("Шифротекст:", cipher_text)
         Шифротекст: K0x010u8RvWv[1w8V3P

In [12]: print("Открытый текст:", cipher_text_function(cipher_text, key))
         Открытый текст: С Новым годом, друзья

In [13]: print("Ключ:", cipher_text_function(text, cipher_text))
         Ключ: 7X8s51fbtSyHdUmrCao
```

Рис. 2: Код



Рис. 3: Приложение, реализующее режим однократного гаммирования для двух текстов одним ключом, Часть 2

```
print("Первый открытый текст:", cipher_text_function(cipher_text_2, key))

Первый открытый текст: С Новым годом, друзья!
Второй открытый текст: Поздравления с 8 марта!

In [11]: cipher_text_xor = cipher_text_function(cipher_text_1, cipher_text_2)
print("Первый шифротекст XOR со Вторым шифротекстом:", cipher_text_xor)

Первый шифротекст XOR со Вторым шифротекстом: XP
r(0|0|0|0|0|0)

In [12]: print("Первый открытый текст:", cipher_text_function(cipher_text_xor, test_2))
Первый открытый текст: С Новым годом, друзья!
Второй открытый текст: Поздравления с 8 марта!

Первый открытый текст: С Новым годом, друзья!
Второй открытый текст: Поздравления с 8 марта!

In [13]: test_1 = test_1[:16]
print("Часть первого открытого текста:", test_1)

Часть первого открытого текста: 000

In [14]: cipher_text_xor_2 = cipher_text_function(cipher_text_1[:16], cipher_text_1[:16])
print("Часть первого открытого текста:", cipher_text_function(cipher_text_xor_2, test_1))

Часть первого открытого текста: 000
```

8/9

## Вывод

---

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