

**МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКИЙ УНИВЕРСИТЕТ
ДРУЖБЫ НАРОДОВ**

Факультет физико-математических и естественных наук

Кафедра информационных технологий

ОТЧЕТ по лабораторной работе 8

**ТЕМА «Элементы криптографии. Шифрование (кодирование) различных исходных
текстов одним ключом»**

по дисциплине «Информационная безопасность»

Выполнил:

Студент группы НПИбд-02-21

Студенческий билет № 1032205421

Стелина Петрити

Список содержания

[Список содержания.](#)

[Список изображений](#)

[Цель работы.](#)

[Последовательность выполнения работы](#)

[Выводы](#)

Список изображений

[рис. 1 Код для шифрования и дешифрования сообщений](#)

Цель работы

Ознакомиться с практическим использованием метода одноразового шифрования с помощью гаммирования, закодировав несколько исходных текстов с использованием одного ключа.

Последовательность выполнения работы

1. 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 P1 и P2 в режиме однократного гаммирования. Приложение должно определить вид шифротекстов C1 и C2 обоих текстов P1 и P2 при известном ключе ; Необходимо определить и выразить аналитически способ, при котором злоумышленник может прочесть оба текста, не зная ключа и не стремясь его определить. Определить вид шифротекста при известном ключе и известном открытом тексте.

Задача состоит в том, чтобы разработать приложение, которое будет шифровать и дешифровать тексты с использованием однократного гаммирования, а затем попытаться прочесть тексты без знания ключа.

```
def xorbytes(data, key):
    extend_key = (key * (len(data)//len(key)+1))[0:len(data)]
    return bytes([b1^b2 for b1, b2, in zip(data, extend_key)])

def decryption(cipher_txt, key):
    return xorbytes(cipher_txt, key).decode('utf-8')

key=bytes([0x05, 0x0C, 0x17, 0x0E, 0x4E, 0x37, 0xD2, 0x94, 0x10, 0x09,
          0x2E, 0x22, 0x57, 0xFF, 0xC8, 0x00, 0x82, 0x70, 0x54])
P1 = "НаВашисходящийот1204".encode('utf-8')
P2 = "ВСеверныйфилиалБанкав".encode('utf-8')
c1 = xorbytes(P1, key)
c2 = xorbytes(P2, key)
print("c1 = ", c1)
print("c2 = ", c2)
P1_xor_P2 = xorbytes(c1,c2)
print("P1 ⊕ P2:", P1_xor_P2)
P1decryption = decryption(c1, key)
P2decryption = decryption(c2, key)
print("P1 decrypted= ", P1decryption)
print("P2 decrypted= ", P2decryption)

c1 = b'%\xdc\x8a\xde\xfe\xe7\x0d\x08\xa6\xf2\xef.I\xda7\xa0\xea\xd5\xb8\xc6\x81\x9f\xbe\x02,\xc0\xb0\xfe\x9c\x86}\xf99\x82D'
c2 = b'\xd5\x9e\x7\xaf\x9e\x82\x02&\xc0\xbc\xff\xa2\x19\x80b\xc9\x85\xdc\xaf\xde\xf5\xe7jd\xa0\xd9\x95\xf2\x06/x\xdb\x0f\xa0\xee\xd5\xbc'
P1 ⊕ P2: b'\xf0Bmq`e8b`dYPhIPZUioTdI_jYhh`ikn@R\x81\xe2\x8d\xe4'
P1 decrypted=  НаВашисходящийот1204
P2 decrypted=  ВСеверныйфилиалБанкав
```

рис. 1 Код для шифрования и дешифрования сообщений

Функция xorbytes:

Выполняет операцию XOR для каждого байта текста и ключа. Если текст длиннее ключа, ключ повторяется.

Функция decryption:

Использует ту же операцию XOR для расшифровки сообщения с известным ключом.

Шифрование:

Для двух исходных сообщений P1 и P2, шифротексты c1 и c2 вычисляются через XOR с ключом.

Атака:

Вычисляется $P1 \oplus P2$, чтобы показать, как можно восстановить одно сообщение, если известно другое.

Вывод

В ходе выполнения работы я приобрела практические навыки использования режима одноразового гаммирования с применением одного ключа для кодирования двух сообщений.