

# **Лабораторная работа №7**

**Элементы криптографии. Однократное гаммирование**

Ильин Андрей Владимирович

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>4</b>
<b>2</b>	<b>Задачи</b>	<b>5</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>6</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>7</b>
<b>5</b>	<b>Анализ результатов</b>	<b>12</b>
<b>6</b>	<b>Выводы</b>	<b>13</b>
	<b>Список литературы</b>	<b>14</b>

## Список иллюстраций

4.1	Класс Gumming . . . . .	9
4.2	Центр и Мюллер . . . . .	10
4.3	Ключ для С Новым Годом, друзья! . . . . .	11

# 1 Цель работы

Освоить на практике применение режима однократного гаммирования<sup>1</sup>

## 2 Задачи

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 представляет собой наложение (снятие) на открытые (зашифрованные) данные последовательности элементов других данных, полученной с помощью некоторого криптографического алгоритма, для получения зашифрованных (открытых) данных. Иными словами, наложение гаммы — это сложение её элементов с элементами открытого (закрытого) текста по некоторому фиксированному модулю, значение которого представляет собой известную часть алгоритма шифрования. [1]

В соответствии с теорией криптоанализа, если в методе шифрования используется однократная вероятностная гамма (однократное гаммирование) той же длины, что и подлежащий сокрытию текст, то текст нельзя раскрыть. Даже при раскрытии части последовательности гаммы нельзя получить информацию о всём скрываемом тексте.

Наложение гаммы по сути представляет собой выполнение операции сложения по модулю 2 (XOR).

## 4 Выполнение лабораторной работы

1. Для выполнения лабораторной работы воспользуемся открытым ресурсом Google Colab. Создадим новый ноутбук - в нем будем выполнять лабораторную работу. Реализуем класс `Gumming`, в нем будут следующие методы: `xor` и `__xor` - выполняет операцию XOR к двум строкам шестнадцатиричного кода (разделенного пробелами), `to_hex` - метод конвертации обычной строки к строке шестнадцатиричного кода, `from_hex` - метод для конвертации строки шестнадцатиричного кода к обычной строке. (рис. 4.1)

```
class Gumming:
    def xor(self, hex_seq1, hex_seq2):
        hex1 = hex_seq1.split()
        hex2 = hex_seq2.split()
        return ' '.join([self.__xor(hex1, hex2) for hex1, hex2 in zip(hex1, hex2)])

    def to_hex(self, msg):
        msg_hex = []
        for char in msg:
            char_cp1251 = char.encode('cp1251')
            char_code = int.from_bytes(char_cp1251, 'little')
            char_hex = hex(char_code)[-2:].upper()
            msg_hex.append(char_hex)
        return ' '.join(msg_hex)
```

```

def from_hex(self, msg_hex):
    msg = ''
    for char_hex in msg_hex.split():
        char_code = int(char_hex, 16)
        char_cp1251 = char_code.to_bytes(1, 'little')
        char = char_cp1251.decode('cp1251')
        msg += char
    return msg

def __xor(self, sym1, sym2):
    xor = lambda x, y: bytes(a^b for a, b in zip(x, y))
    b_sym1 = bytes.fromhex(sym1)
    b_sym2 = bytes.fromhex(sym2)
    r_result = xor(b_sym1, b_sym2)
    result = r_result.hex().upper()
    return result

```

```
gumming = Gumming()
```



```

class Gumming:
    def xor(self, hex_seq1, hex_seq2):
        hex1 = hex_seq1.split()
        hex2 = hex_seq2.split()
        return ' '.join([self.__xor(hex1, hex2) for hex1, hex2 in zip(hex1, hex2)])

    def to_hex(self, msg):
        msg_hex = []
        for char in msg:
            char_cp1251 = char.encode('cp1251')
            char_code = int.from_bytes(char_cp1251, 'little')
            char_hex = hex(char_code)[-2:].upper()
            msg_hex.append(char_hex)
        return ' '.join(msg_hex)

    def from_hex(self, msg_hex):
        msg = ''
        for char_hex in msg_hex.split():
            char_code = int(char_hex, 16)
            char_cp1251 = char_code.to_bytes(1, 'little')
            char = char_cp1251.decode('cp1251')
            msg += char
        return msg

    def __xor(self, sym1, sym2):
        xor = lambda x, y: bytes(a^b for a, b in zip(x, y))
        b_sym1 = bytes.fromhex(sym1)
        b_sym2 = bytes.fromhex(sym2)
        r_result = xor(b_sym1, b_sym2)
        result = r_result.hex().upper()
        return result

gumming = Gumming()

```

Рис. 4.1: Класс Gumming

2. Проведем эксперимент “Центр и Мюллер”, приведенный в указаниях к лабораторной работе, используя написанный класс. (рис. 4.2)

```

src_msg = 'Штирлиц - Вы Герой!!'
hex_msg = gumming.to_hex(src_msg)
src_key = '05 0C 17 7F 0E 4E 37 D2 94 10 09 2E 22 57 FF C8 0B B2 70 54'
enc_msg = gumming.xor(hex_msg, src_key)

mul_key = '05 0C 17 7F 0E 4E 37 D2 94 10 09 2E 22 55 F4 D3 07 BB BC 54'
mul_res = gumming.xor(enc_msg, mul_key)
mul_msg = gumming.from_hex(mul_res)

print(src_msg, '<-- Сообщение Центра')
print(hex_msg, '<-- Сообщение Центра (16)')
print(src_key, '<-- Ключ Центра')
print(enc_msg, '<-- Закодированное сообщение Центра')

```

```

print(mul_key, '<-- Ключ Мюллера')
print(mul_res, '<-- Сообщение Мюллера (16)')
print(mul_msg, '<-- Сообщение Мюллера')

```

```

src_msg = 'Штирлиц - Вы Герой!!'
hex_msg = gumming.to_hex(src_msg)
src_key = '05 0C 17 7F 0E 4E 37 D2 94 10 09 2E 22 57 FF C8 0B 82 70 54'
enc_msg = gumming.xor(hex_msg, src_key)

mul_key = '05 0C 17 7F 0E 4E 37 D2 94 10 09 2E 22 55 F4 D3 07 BB BC 54'
mul_res = gumming.xor(enc_msg, mul_key)
mul_msg = gumming.from_hex(mul_res)

print(src_msg, '<-- Сообщение Центра')
print(hex_msg, '<-- Сообщение Центра (16)')
print(src_key, '<-- Ключ Центра')
print(enc_msg, '<-- Закодированное сообщение Центра')
print(mul_key, '<-- Ключ Мюллера')
print(mul_res, '<-- Сообщение Мюллера (16)')
print(mul_msg, '<-- Сообщение Мюллера')

```

```

Штирлиц - Вы Герой!! <-- Сообщение Центра
D8 F2 E8 F0 EB E8 F6 20 96 20 C2 FB 20 C3 E5 F0 EE E9 21 21 <-- Сообщение Центра (16)
05 0C 17 7F 0E 4E 37 D2 94 10 09 2E 22 57 FF C8 0B 82 70 54 <-- Ключ Центра
D0 FE FF 8F E5 A6 C1 F2 02 30 CB D5 02 04 1A 38 E5 5B 51 75 <-- Закодированное сообщение Центра
05 0C 17 7F 0E 4E 37 D2 94 10 09 2E 22 55 F4 D3 07 BB BC 54 <-- Ключ Мюллера
D8 F2 E8 F0 EB E8 F6 20 96 20 C2 FB 20 C1 EE EB E2 E0 ED 21 <-- Сообщение Мюллера (16)
Штирлиц - Вы Болван! <-- Сообщение Мюллера

```

Рис. 4.2: Центр и Мюллер

3. Определим ключ, с помощью которого шифротекст может быть преобразован в некоторый фрагмент текста (С Новым Годом, друзья!), (рис. 4.3)

```

ng_msg = 'С Новым Годом, друзья!'
ng_hex = gumming.to_hex(ng_msg)
ng_key = gumming.xor(enc_msg, ng_hex)
ng_res = gumming.from_hex(gumming.xor(enc_msg, ng_key))

print(ng_msg, '<-- Необходимое сообщение')
print(ng_hex, '<-- Необходимое сообщение (16)')
print(enc_msg, '<-- Закодированное сообщение Центра')
print(ng_key, '<-- Искомый ключ')
print(ng_res, '<-- Необходимое сообщение из сообщения Центра')

```

```
ng_msg = 'С Новым Годом, друзья!'
ng_hex = gumming.to_hex(ng_msg)
ng_key = gumming.xor(enc_msg, ng_hex)
ng_res = gumming.from_hex(gumming.xor(enc_msg, ng_key))

print(ng_msg, '<-- Необходимое сообщение')
print(ng_hex, '<-- Необходимое сообщение (16)')
print(enc_msg, '<-- Закодированное сообщение Центра')
print(ng_key, '<-- Искомый ключ')
print(ng_res, '<-- Необходимое сообщение из сообщения Центра')
```

С Новым Годом, друзья! <-- Необходимое сообщение  
D1 20 CD EE E2 FB EC 20 C3 EE E4 EE EC 2C E4 F0 F3 E7 FC FF 21 <-- Необходимое сообщение (16)  
DD FE FF 8F E5 A6 C1 F2 02 30 CB D5 02 94 1A 38 E5 5B 51 75 <-- Закодированное сообщение Центра  
0C DE 32 61 07 5D 2D D2 C1 DE 2F 3B EE B8 FE C8 16 BC AD 8A <-- Искомый ключ  
С Новым Годом, друзья <-- Необходимое сообщение из сообщения Центра

Рис. 4.3: Ключ для С Новым Годом, друзья!

## **5 Анализ результатов**

Работа выполнена без непредвиденных проблем в соответствии с руководством. Ошибок и сбоев не произошло.

## 6 Выводы

Нам удалось освоить на практике применение режима однократного гаммирования, в дополнение закрепили навыки владения языками программирования, в частности языком программирования - python.

## Список литературы

1. Материалы по лабораторной работе [Электронный ресурс]. RUDN. URL: [https://esystem.rudn.ru/pluginfile.php/2090284/mod\\_resource/content/2/007-lab\\_crypto-gamma.pdf](https://esystem.rudn.ru/pluginfile.php/2090284/mod_resource/content/2/007-lab_crypto-gamma.pdf).