

# **Отчёт по лабораторной работе №8**

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

Голощапова Ирина Борисовна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Теоретическое введение</b>	<b>6</b>
<b>3</b>	<b>Постановка задачи</b>	<b>7</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>8</b>
<b>5</b>	<b>Выводы</b>	<b>11</b>
<b>6</b>	<b>Библиография</b>	<b>12</b>

## Список иллюстраций

4.1	Импорт библиотек и задание исходных данных . . . . .	8
4.2	Функция 1 (шифрование) . . . . .	9
4.3	Работа функции шифрования . . . . .	9
4.4	Функция 2 (расшифровка) . . . . .	10
4.5	Работа функции расшифрования . . . . .	10

## Список таблиц

# 1 Цель работы

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

## 2 Теоретическое введение

Предложенная Г. С. Вернамом так называемая «схема однократного использования (гаммирования)» является простой, но надёжной схемой шифрования данных. 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.

### 3 Постановка задачи

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.

## 4 Выполнение лабораторной работы

1. Для написания программы использовала язык Python. Для начала импортировала необходимые библиотеки и задавала две исходные строки равной длины (рис. 4.1)

```
In [1]: import numpy as np
import operator as op
import sys

In [2]: p1 = "НаВашисходящийот1204"
print(len(p1))

p2 = "ВСеверныйфилиалБанка"
print(len(p2))

20
20
```

Рис. 4.1: Импорт библиотек и задание исходных данных

2. Написала функцию, которая определяет вид шифротекстов обеих строк при известном ключе(рис. 4.2)

Функция получает на вход две символьные строки, переводит их в 16-ый формат и генерирует 16-ый ключ. Далее с помощью данного ключа вычисляются шифротексты в 16-ой СС и переводятся в строковый формат.



```

In [3]: def encrypt(text1, text2):
        print("text1:", text1)
        newtext1=[]
        for i in text1:
            newtext1.append(i.encode("cp1251").hex())
        print("text1 in 16: ", newtext1)
        print("text2: ", text2)
        newtext2 = []

        for i in text2:
            newtext2.append(i.encode("cp1251").hex())
        print("text1 in 16: ", newtext2)

        r=np.random.randint(0,255,len(text1))
        key=[hex(i)[2:] for i in r]
        newkey=[]
        for i in key:
            newkey.append(i.encode("cp1251").hex().upper())
        print("key in 16:", key)
        xortext1=[]
        for i in range(len(newtext1)):
            xortext1.append("{:02x}".format(int(key[i],16) ^ int(newtext1[i], 16)))
        print("cypher text1 in 16: ", xortext1)
        en_text1=bytearray.fromhex("".join(xortext1)).decode("cp1251")
        print("cypher text1: ", en_text1)

        xortext2=[]
        for i in range(len(newtext2)):
            xortext2.append("{:02x}".format(int(key[i],16) ^ int(newtext2[i], 16)))
        print("cypher text1 in 16: ", xortext2)
        en_text2=bytearray.fromhex("".join(xortext2)).decode("cp1251")
        print("cypher text1: ", en_text2)

        return key, xortext1, en_text1, xortext2, en_text2

```

Рис. 4.2: Функция 1 (шифрование)

### 3. Проверила работу первой функции (рис. 4.3)

```

In [4]: k, t1, et1, t2, et2=encrypt(p1, p2)

text1: НаВашеисходящийот1204
text1 in 16: ['cd', 'e0', 'c2', 'e0', 'f8', 'e8', 'f1', 'f5', 'ee', 'e4', 'ff', 'f9', 'e8', 'e9', 'ee', 'f2', '31', '32', '30', '34']
text2: ВСеверныйФилиалБанка
text1 in 16: ['c2', 'd1', 'e5', 'e2', 'e5', 'f0', 'ed', 'fb', 'e9', 'f4', 'e8', 'eb', 'e8', 'e0', 'eb', 'c1', 'e0', 'ed', 'ea', 'e0']
key in 16: ['45', '92', 'd7', '23', 'f4', '8b', '72', '39', '4c', '1f', '65', 'd7', '86', '16', 'c6', '7f', '41', '8e', '10', 'e1']
cypher text1 in 16: ['88', '72', '15', 'c3', '0c', '63', '83', 'cc', 'a2', 'fb', '9a', '2e', '6e', 'ff', '28', '8d', '70', 'bc', '20', 'd5']
cypher text1: €r0ГсfМумм.на(Крj X
cypher text1 in 16: ['87', '43', '32', 'c1', '11', '7b', '9f', 'c2', 'a5', 'eb', '8d', '3c', '6e', 'f6', '2d', 'be', 'a1', '63', 'fa', '01']
cypher text1: ‡C26@uBГnK<nc-s9cш

```

Рис. 4.3: Работа функции шифрования

### 4. Написала функцию, которая при известных двух шифротекстах и одном открытом тексте находит вид второго открытого текста без ключа (рис. 4.4)

Функция получает на вход две зашифрованные строки и один открытый текст. Далее шифротексты переводятся в 16-ый формат и применяется принцип однократного гаммирования. Находится вид второго сообщения без использования ключа и возвращается расшифрованное сообщение.

```

In [5]: def decrypt(c1, c2, p1):
    print("cypher text1: ", c1)
    newc1=[]
    for i in c1:
        newc1.append(i.encode("cp1251").hex())
    print("cypher text1 in 16: ", newc1)
    print("cypher text2: ", c2)
    newc2=[]
    for i in c2:
        newc2.append(i.encode("cp1251").hex())
    print("cypher text2 in 16: ", newc2)
    print("open text1: ", p1)

    newp1=[]
    for i in p1:
        newp1.append(i.encode("cp1251").hex())
    print("open text1 in 16: ", newp1)
    xortmp=[]
    sp2=[]
    for i in range(len(p1)):
        xortmp.append("{:02x}".format(int(newc1[i],16)^int(newc2[i],16)))
        sp2.append("{:02x}".format(int(xortmp[i],16)^int(newp1[i],16)))
    print("open text2 in 16: ", sp2)
    p2 = bytearray.fromhex("".join(sp2)).decode("cp1251")
    print("open text2: ", p2)
    return p1, p2

```

Рис. 4.4: Функция 2 (расшифровка)

5. Проверила работу функции расшифрования (рис. 4.5)

```
In [6]: decrypt(et1, et2, p1)

cypher text1: 6r$0cfm0yus.na(kp) X
cypher text1 in 16: ['88', '72', '15', 'c3', '0c', '63', '83', 'cc', 'a2', 'fb', '9a', '2e', '6e', 'ff', '28', '8d', '70', 'b', 'c', '20', 'd5']
cypher text2: 2C2B{uBf0knc-s9csB
cypher text2 in 16: ['87', '43', '32', 'c1', '11', '7b', '9f', 'c2', 'a5', 'eb', '8d', '3c', '6e', 'f6', '2d', 'be', 'a1', '6', '3', 'fa', '01']
open text1: НаВашисходящийот1204
open text1 in 16: ['cd', 'e0', 'c2', 'e0', 'f8', 'e8', 'f1', 'f5', 'ee', 'e4', 'ff', 'f9', 'e8', 'e9', 'ee', 'f2', '31', '32', '30', '34']
open text2 in 16: ['c2', 'd1', 'e5', 'e2', 'e5', 'f0', 'ed', 'fb', 'e9', 'f4', 'e8', 'eb', 'e8', 'e0', 'eb', 'c1', 'e0', 'ed', 'ea', 'e0']
open text2: ВСеверныйфилиалБанка

Out[6]: ('НаВашисходящийот1204', 'ВСеверныйфилиалБанка')
```

Рис. 4.5: Работа функции расшифрования

## 5 Выводы

В ходе лабораторной работы мне удалось освоить на практике применение режима однократного гаммирования на примере кодирования различных исходных текстов одним ключом.

## 6 Библиография

1. Git - система контроля версий
2. Rocky Linux
3. Элементы криптографии
4. Однократное гаммирование