

Отчёт по лабораторной работе №7

Элементы криптографии. Однократное гаммирование

Акопян Изабелла Арменовна

Содержание

1	Цель работы	4
2	Задание	5
3	Теоретическое введение	6
4	Выполнение лаборатной работы	7
5	Выводы	10
	Список литературы	11

List of Figures

4.1	Программа шифрования 1/2	8
4.2	Программа шифрования 2/2	9

1 Цель работы

Освоить на практике применение режима однократного гаммирования.

2 Задание

Требуется разработать приложение, позволяющее шифровать и дешифровать данные в режиме однократного гаммирования.

3 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.

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.

4 Выполнение лабораторной работы

1. Создана программа, шифрующая сообщение по ключу.
2. Создана программа, определяющая ключ по шифру и исходному тексту.
3. Программа проверена на данных из пособия

Написала код (рис. fig. 4.1, fig. 4.2), который определяет вид шифротекста при известном ключе и известном открытом тексте, а также определяет ключ.

```
[49] import numpy as np

def code(text):
    array = []
    for i in text:
        array.append(i.encode('cp1251').hex())
    print('text 16: ', *array)

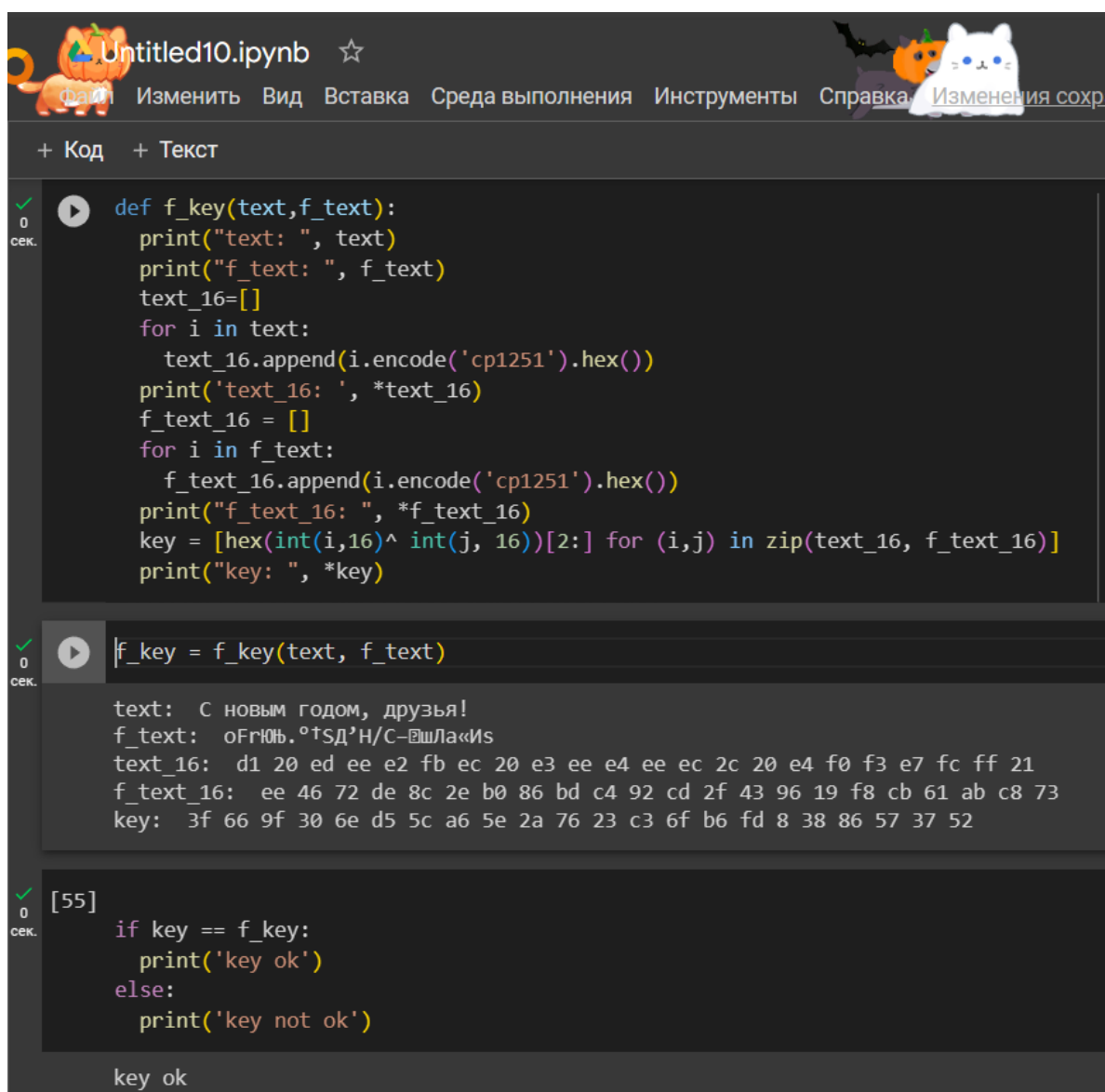
    key = np.random.randint(0, 255, len(text))
    key_16 = [hex(i)[2:] for i in key]
    print("key 16: ", *key_16)
    cr = []
    for i in range(len(array)):
        cr.append("{:02x}".format(int(array[i], 16) ^ int(key_16[i], 16)))
    print("text in 16: ", *cr)

    f_text = bytearray.fromhex(''.join(cr)).decode('cp1251')
    print('text: ', f_text)
    return key_16, f_text

text = "С НОВЫМ ГОДОМ, ДРУЗЬЯ!"
key, f_text = code(text)

text 16:  d1 20 ed ee e2 fb ec 20 e3 ee e4 ee ec 2c 20 e4 f0 f3 e7 fc ff 21
key 16:  3f 66 9f 30 6e d5 5c a6 5e 2a 76 23 c3 6f b6 fd 8 38 86 57 37 52
text in 16:  ee 46 72 de 8c 2e b0 86 bd c4 92 cd 2f 43 96 19 f8 cb 61 ab c8 73
text:  oFrЮЬ.°†SD'Н/С-Щла«Is
```

Figure 4.1: Программа шифрования 1/2



```
def f_key(text,f_text):
    print("text: ", text)
    print("f_text: ", f_text)
    text_16=[]
    for i in text:
        text_16.append(i.encode('cp1251').hex())
    print('text_16: ', *text_16)
    f_text_16 = []
    for i in f_text:
        f_text_16.append(i.encode('cp1251').hex())
    print("f_text_16: ", *f_text_16)
    key = [hex(int(i,16)^ int(j, 16))[2:] for (i,j) in zip(text_16, f_text_16)]
    print("key: ", *key)
```

```
f_key = f_key(text, f_text)
```

```
text: С новым годом, друзья!
f_text: оFrЮЬ.°†SД'Н/С-Шла«Is
text_16: d1 20 ed ee e2 fb ec 20 e3 ee e4 ee ec 2c 20 e4 f0 f3 e7 fc ff 21
f_text_16: ee 46 72 de 8c 2e b0 86 bd c4 92 cd 2f 43 96 19 f8 cb 61 ab c8 73
key: 3f 66 9f 30 6e d5 5c a6 5e 2a 76 23 c3 6f b6 fd 8 38 86 57 37 52
```

```
[55]
if key == f_key:
    print('key ok')
else:
    print('key not ok')
```

key ok

Figure 4.2: Программа шифрования 2/2

5 Выводы

Я освоила на практике применение режима однократного гаммирования.

Список литературы

Лабораторная работа № 7. Элементы криптографии. Однократное гаммирование
Однократное гаммирование