Лабораторная работа №8.

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

Силкина Мария Александровна

Содержание

List of Figures

List of Tables

1 Цель работы

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

2 Задачи

1. Написать программу, которая должна определять вид шифротекстов при известных открытых текстах и при известном ключе. Также эта программа должна определить вид одного из текстов, зная вид другого открытого текста и зашифрованный вид обоих текстов без использования ключа.
2. Ответить на контрольные вопросы

3 Выполнение лабораторной работы

##Теоретическая справка

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

3.1 Выполнение задач

Первым шагом написала функцию шифрования, которая определяет вид шифротекстов при известном ключе и известном открытом тексте. В выводе я получила наши изначальные тексты, их представление в шестнадцатеричной системе, рандомный ключ и зашифрованные тексты. (рис - @fig:001)

```python
def shifr(p1, p2):
    print ('Сообщение P1 - ', p1)
    text_arrayp1 = []
    for i in p1:
        text_arrayp1.append(i.encode('cp1251').hex())
    print('\nНаше сообщение P1 в 16ричной системе - ', *text_arrayp1)
    print ('Сообщение P2 - ', p2)
    text_arrayp2 = []
    for i in p2:
        text_arrayp2.append(i.encode('cp1251').hex())
    print('\nНаше сообщение P2 в 16ричной системе - ', *text_arrayp2)



    key_int = np.random.randint(0, 255, len(p1))
    key_hex = [hex(i)[2:] for i in key_int]
    print('\nШифр - ', *key_hex)


    text_cryptp1 = []
    for i in range(len(text_arrayp1)):
        text_cryptp1.append('{:02x}'.format(int(text_arrayp1[i], 16)^ int(key_hex[i], 16)))
    print('\nНаше зашифрованное сообщение C1 в 16ричной системе- ', *text_cryptp1)
    text_cryptp2 = []
    for i in range(len(text_arrayp2)):
        text_cryptp2.append('{:02x}'.format(int(text_arrayp2[i], 16)^ int(key_hex[i], 16)))
    print('\nНаше зашифрованное сообщение C2 в 16ричной системе- ', *text_cryptp2)

    finalp1 = bytearray.fromhex(''.join(text_cryptp1)).decode('cp1251')
    print ('\nЗашифрованное сообщение C1 - ', finalp1)
    finalp2 = bytearray.fromhex(''.join(text_cryptp2)).decode('cp1251')
    print ('\nЗашифрованное сообщение C2 - ', finalp2)

    return key_hex, finalp1, finalp2
```

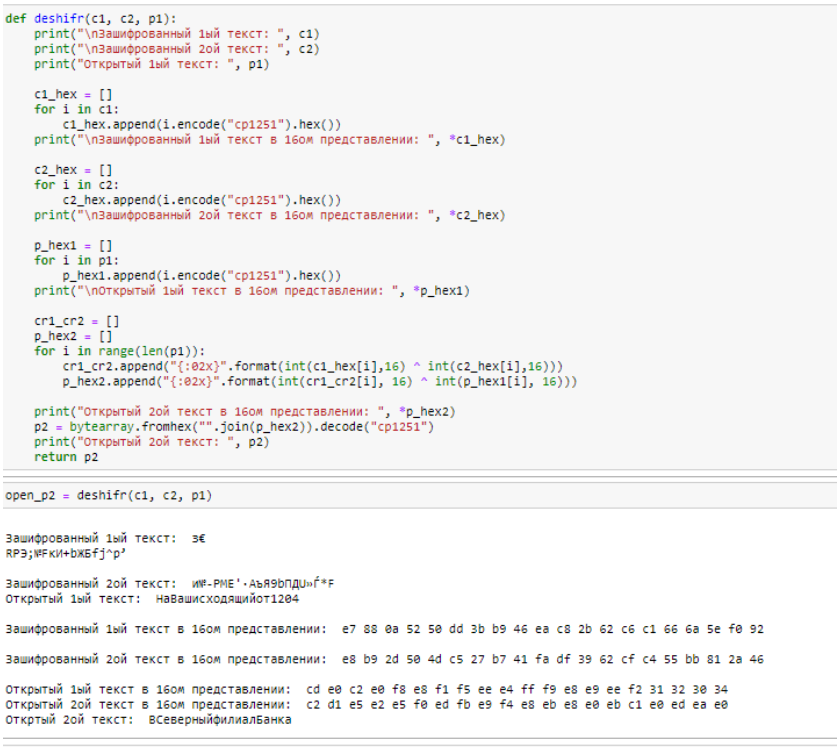
```python
p1 = "НаВашисходящийот1204"
p2 = "ВСеверныйфилиалБанка"
```

```python
key, c1, c2 = shifr(p1, p2)
```

```
Сообщение P1 -  НаВашисходящийот1204

Наше сообщение P1 в 16ричной системе -  cd e0 c2 e0 f8 e8 f1 f5 ee e4 ff f9 e8 e9 ee f2 31 32 30 34
Сообщение P2 -  ВСеверныйфилиалБанка

Наше сообщение P2 в 16ричной системе -  c2 d1 e5 e2 e5 f0 ed fb e9 f4 e8 eb e8 e0 eb c1 e0 ed ea e0

Шифр -  2a 68 c8 b2 a8 35 ca 4c a8 e 37 d2 8a 2f 2f 94 5b 6c c0 a6

Наше зашифрованное сообщение C1 в 16ричной системе-  e7 88 0a 52 50 dd 3b b9 46 ea c8 2b 62 c6 c1 66 6a 5e f0 92

Наше зашифрованное сообщение C2 в 16ричной системе-  e8 b9 2d 50 4d c5 27 b7 41 fa df 39 62 cf c4 55 bb 81 2a 46

Зашифрованное сообщение C1 -  з€
RPЭ;¥FкИ+bЖБfj^p’

Зашифрованное сообщение C2 -  иЎ-РМЕ'·АьЯ9bПДU»ѓ*F
```

Figure 3.1: Функция, шифрующая данные и ее выполнение

Далее я создала функцию для дешифрования, которая при знании зашифрованных текстов и иодного изначального, может найти второй (неизвестный) текст. (рис - @fig:002)



```
def deshifr(c1, c2, p1):
    print("\nЗашифрованный 1ый текст: ", c1)
    print("\nЗашифрованный 2ой текст: ", c2)
    print("Открытый 1ый текст: ", p1)

    c1_hex = []
    for i in c1:
        c1_hex.append(i.encode("cp1251").hex())
    print("\nЗашифрованный 1ый текст в 16ом представлении: ", *c1_hex)

    c2_hex = []
    for i in c2:
        c2_hex.append(i.encode("cp1251").hex())
    print("\nЗашифрованный 2ой текст в 16ом представлении: ", *c2_hex)

    p_hex1 = []
    for i in p1:
        p_hex1.append(i.encode("cp1251").hex())
    print("\nОткрытый 1ый текст в 16ом представлении: ", *p_hex1)

    cr1_cr2 = []
    p_hex2 = []
    for i in range(len(p1)):
        cr1_cr2.append("{:02x}".format(int(c1_hex[i],16) ^ int(c2_hex[i],16)))
        p_hex2.append("{:02x}".format(int(cr1_cr2[i], 16) ^ int(p_hex1[i], 16)))

    print("Открытый 2ой текст в 16ом представлении: ", *p_hex2)
    p2 = bytearray.fromhex("".join(p_hex2)).decode("cp1251")
    print("Открытый 2ой текст: ", p2)
    return p2
```

```
open_p2 = deshifr(c1, c2, p1)
```

```
Зашифрованный 1ый текст:  з€
RPЭ;№FкИ+bЖБfj^p’

Зашифрованный 2ой текст:  И№-РМЕ'·АьЯ9bПДЦ»ѓ*F
Открытый 1ый текст:  НаВашисходящийот1204

Зашифрованный 1ый текст в 16ом представлении:  e7 88 0a 52 50 dd 3b b9 46 ea c8 2b 62 c6 c1 66 6a 5e f0 92

Зашифрованный 2ой текст в 16ом представлении:  e8 b9 2d 50 4d c5 27 b7 41 fa df 39 62 cf c4 55 bb 81 2a 46

Открытый 1ый текст в 16ом представлении:  cd e0 c2 e0 f8 e8 f1 f5 ee e4 ff f9 e8 e9 ee f2 31 32 30 34
Открытый 2ой текст в 16ом представлении:  c2 d1 e5 e2 e5 f0 ed fb e9 f4 e8 eb e8 e0 eb c1 e0 ed ea e0
Откртый 2ой текст:  ВСеверныйфилиалБанка
```

Figure 3.2: Функция, дешифрующая данные и ее выполнение

##Контрольные вопросы

1. Зная один из текстов, мы можем определить другой, вопспользовавшись следующей формулой: $C_1 \oplus C_2 \oplus + P_1 = P_1 \oplus P_2 \oplus + P_1 = P_2$, где C_1 и C_2 - шифротексты. Т.е. ключ в данной формуле не используется.

2. При повторном использовании ключа при шифровании текста получим исходное сообщение.

3. Режим шифрования однократного гаммирования одним ключом двух открытых текстов реализуется по следующей формуле:

$$C_1 = P_1 \oplus + K$$

$$C_2 = P_2 \oplus + K,$$

где C_i - шифротексты, P_i - открытые тексты, K - единый ключ шифровки

4. Недостатки шифрования одним ключом двух открытых текстов:

- Имея на руках одно из сообщений в открытом виде и оба шифротекста, злоумышленник способен расшифровать каждое сообщение, не зная ключа.

- Зная шаблон сообщений, злоумышленник получает возможность определить те символы сообщения P_2, которые находятся на позициях известного шаблона сообщения P_1.

5. Преимущество шифрования одним ключом двух текстов заключается в том, что такой подход помогает упростить процесс шифрования и дешифровки. Также, при отправке сообщений между двумя компьютерами, удобнее пользоваться одним общим ключом для передаваемых данных

4 Выводы

Освоила использования однократного гаммирования для шифрования и дешифрования данных.

5 Библиография

1. Кулябов Д. С., Королькова А. В., Геворкян М. Н. Информационная безопасность компьютерных сетей. Лабораторная работа № 8. Элементы криптографии. Шифрование (кодирование) различных исходных текстов одним ключом.