Цель работы

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

Выполнение лабораторной работы

Импортирую библиотеки:

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

Подаю на вход 2 строки:

```
[2]  s1 = "С Новым Годом, друзья!"
     len(s1)

     22
```

```
[3]  s2 = "Лабораторная работа №8"
     len(s2)

     22
```

1. Напишу функцию, определяющую вид шифротекстов C1 и C2 обеих строк при известном ключе. Функция получает на вход две символьные строки, которые затем переводятся в 16-ую систему. Далее генерируется рандомный ключ, при помощи которого определяются соответствующие шифротексты в 16-й системе. Зачем шифротекст переводится в строковый формат. Функция возвращает ключ, оба шифротекста в 16-ой системе и строковом формате.

```python
[4] # функция шифрования
    def encryption(text1, text2):
        # работа с первым текстом
        print("Открытый текст 1: ", text1)
        new_text1 = []
        for i in text1:
            new_text1.append(i.encode("cp1251").hex())
        print("\nОткрытый текст 1 в 16-ой системе: ", new_text1)

        # работа со вторым текстом
        print("\nОткрытый текст 2: ", text2)
        new_text2 = []
        for i in text2:
            new_text2.append(i.encode("cp1251").hex())
        print("\nОткрытый текст 2 в 16-ой системе: ", new_text2)

        # генерация ключа
        r = np.random.randint(0, 255, len(text1))
        key = [hex(i)[2:] for i in r]
        new_key = []
        for i in key:
                new_key.append(i.encode("cp1251").hex().upper())
        print("\nКлюч в 16-ой системе: ", key)

        # получение зашифрованного сообщения из 1 текста
        xor_text1 = []
        for i in range(len(new_text1)):
            xor_text1.append("{:02x}".format(int(key[i], 16) ^ int(new_text1[i], 16)))
        print("\nШифротекст 1 в 16-ой системе: ", xor_text1)
        # переведу зашифрованное сообщение 1 в строку
        en_text1 = bytearray.fromhex("".join(xor_text1)).decode("cp1251")
        print("\nШифротекст 1: ", en_text1)

        # получение зашифрованного сообщения из 2 текста
        xor_text2 = []
        for i in range(len(new_text2)):
            xor_text2.append("{:02x}".format(int(key[i], 16) ^ int(new_text2[i], 16)))
        print("\nШифротекст 2 в 16-ой системе: ", xor_text2)
        # переведу зашифрованное сообщение 2 в строку
        en_text2 = bytearray.fromhex("".join(xor_text2)).decode("cp1251")
        print("\nШифротекст 2: ", en_text2)

        return key, xor_text1, en_text1, xor_text2, en_text2
```

Результат работы функции:

```
k, t1, et1, t2, et2 = encryption(s1, s2)

Открытый текст 1:  С Новым Годом, друзья!

Открытый текст 1 в 16-ой системе:  ['d1', '20', 'cd', 'ee', 'e2', 'fb', 'ec', '20', 'c3', 'ee', 'e4', 'ee', 'ec', '2c', '20', 'e4', 'f0', 'f3', 'e7', 'fc', 'ff', '21']

Открытый текст 2:  Лабораторная работа №8

Открытый текст 2 в 16-ой системе:  ['cb', 'e0', 'e1', 'ee', 'f0', 'e0', 'f2', 'ee', 'f0', 'ed', 'e0', 'ff', '20', 'f0', 'e0', 'e1', 'ee', 'f2', 'e0', '20', 'b9', '38']

Ключ в 16-ой системе:  ['28', '9c', '6e', '6b', '1', '1', 'd6', '33', '42', '9d', '39', '94', '60', '29', 'a3', 'dc', 'a8', 'e6', 'c1', '2d', 'fc', '4d']

Шифротекст 1 в 16-ой системе:  ['f9', 'bc', 'a3', '85', 'e3', 'fa', '3a', '13', '81', '73', 'dd', '7a', '8c', '05', '83', '38', '58', '15', '26', 'd1', '03', '6c']

Шифротекст 1:  щjЈ…гъ:ЃsЭzЊЃѓ8X&Сl

Шифротекст 2 в 16-ой системе:  ['e3', '7c', '8f', '85', 'f1', 'e1', '24', 'dd', 'b2', '70', 'd9', '6b', '40', 'd9', '43', '3d', '46', '14', '21', '0d', '45', '75']

Eu
```

2. Напишу функцию, которая при известных двух шифротекстах и одном открытом тексте находит вид второго открытого текста без ключа.
   Функция получает на вход два шифротекста и один открытый в строковом формате, затем переводит их в 16-ю систему. Затем применяя принцип однократного гаммирования, находит вид второго открытого сообщения без использования ключа шифрования. Возвращает функция второе расшифрованное сообщение в строковом формате и 16-ой системе.

```python
[6] # c1, c2 - шифротексты
    # p1 - открытый текст сообщения
    def decryption(c1, c2, p1):
       # работа с первым шифротекстом
        print("Шифротекст 1: ", c1)
        new_c1 = []
        for i in c1:
            new_c1.append(i.encode("cp1251").hex())
        print("\nШифротекст 1 в 16-ой системе: ", new_c1)

        # работа со вторым шифротекстом
        print("\nШифротекст 2: ", c2)
        new_c2 = []
        for i in c2:
            new_c2.append(i.encode("cp1251").hex())
        print("\nШифротекст 2 в 16-ой системе: ", new_c2)

        # работа с открытым текстом
        print("\nОткрытый текст 1: ", p1)
        new_p1 = []
        for i in p1:
            new_p1.append(i.encode("cp1251").hex())
        print("\nОткрытый текст 1 в 16-ой системе: ", new_p1)

        print("\nНахожу второй открытый текст...")

        # получение расшифрованного сообщения p2
        xor_tmp = []
        sp2 = []
        for i in range(len(p1)):
            xor_tmp.append("{:02x}".format(int(new_c1[i], 16) ^ int(new_c2[i], 16)))
            sp2.append("{:02x}".format(int(xor_tmp[i], 16) ^ int(new_p1[i], 16)))
        print("\nОткрытый текст 2 в 16-ой системе: ", sp2)

        # переведу расшифрованное сообщение 2 в строку
        p2 = bytearray.fromhex("".join(sp2)).decode("cp1251")
        print("\nОткрытый текст 2: ", p2)
        return p2, sp2
```

Результат работы функции:

```
s3 = decryption(et1, et2, s1)

Шифротекст 1:  щjЈ…гъ:Ѓs Эz Ђѓ8X&Сl

Шифротекст 1 в 16-ой системе:  ['f9', 'bc', 'a3', '85', 'e3', 'fa', '3a', '13', '81', '73', 'dd', '7a', '8c', '05', '83', '38', '58', '15', '26', 'd1', '03', '6c']

Eu

Шифротекст 2 в 16-ой системе:  ['e3', '7c', '8f', '85', 'f1', 'e1', '24', 'dd', 'b2', '70', 'd9', '6b', '40', 'd9', '43', '3d', '46', '14', '21', '0d', '45', '75']

Открытый текст 1:  С Новым Годом, друзья!

Открытый текст 1 в 16-ой системе:  ['d1', '20', 'cd', 'ee', 'e2', 'fb', 'ec', '20', 'c3', 'ee', 'e4', 'ee', 'ec', '2c', '20', 'e4', 'f0', 'f3', 'e7', 'fc', 'ff', '21']

Нахожу второй открытый текст...

Открытый текст 2 в 16-ой системе:  ['cb', 'e0', 'e1', 'ee', 'f0', 'e0', 'f2', 'ee', 'f0', 'ed', 'e0', 'ff', '20', 'f0', 'e0', 'e1', 'ee', 'f2', 'e0', '20', 'b9', '38']

Открытый текст 2:  Лабораторная работа №8
```

Ответы на контрольные вопросы

1. Как, зная один из текстов (P1 или P2), определить другой, не зная при этом ключа?
   Для этого надо воспользоваться формулой:

$$C_1 \oplus C_2 \oplus P_1 = P_1 \oplus P_2 \oplus P_1 = P_2,$$

где C1 и C2 – шифротексты. Как видно, ключ в данной формуле не используется.

2. Что будет при повторном использовании ключа при шифровании текста?
   В таком случае мы получим исходное сообщение.
3. Как реализуется режим шифрования однократного гаммирования одним ключом двух открытых текстов?
   Он реализуется по следующей формуле:

$$C_1 = P_1 \oplus K$$
$$C_2 = P_2 \oplus K,$$

   где Ci – шифротексты, Pi – открытые тексты, K – единый ключ шифрования.

4. Перечислите недостатки шифрования одним ключом двух открытых текстов.
   Во-первых, имея на руках одно из сообщений в открытом виде и оба шифротекста, злоумышленник способен расшифровать каждое сообщение, не зная ключа.
   Во-вторых, зная шаблон сообщений, злоумышленник получает возможность определить те символы сообщения P2, которые находятся на позициях известного шаблона сообщения P1. В соответствии с логикой сообщения P2, злоумышленник имеет реальный шанс узнать ещё некоторое количество

   символов сообщения P2. Таким образом, применяя формулу из п. 1, с подстановкой вместо P1 полученных на предыдущем шаге новых символов сообщения P2 злоумышленник если не прочитает оба сообщения, то значительно уменьшит пространство их поиска.
   Наконец, зная ключ, злоумышленник смоет расшифровать все сообщения, которые были закодированы при его помощи.

5. Перечислите преимущества шифрования одним ключом двух открытых текстов.
   Такой подход помогает упростить процесс шифрования и дешифровки. Также, при отправке сообщений между 2-я компьютерами, удобнее пользоваться одним общим ключом для передаваемых данных.

Вывод

В ходе данной лабораторной работы я освоила применение режима однократного гаммирования на примере кодирования различных исходных текстов одним ключом.

Список литературы

- Кулябов Д. С., Королькова А. В., Геворкян М. Н Лабораторная работа №8