Лабораторная работа 8

Цель работы

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

Выполнение лабораторной работы

Подключаю необходимые библиотеки и задаю две строки одинакого размера

In [2]:
```
import numpy as np
import operator as op
import sys
```

In [3]:
```
p1="text line one"

p2="text line two"
```

{ #fig:001 }

Задам функцию encrypt, которая определяет вид шифротекстов и возвращает два шифротекста в 16-ой системе и формате строки

In [4]:
```
def encrypt(text1,text2):

    print("Текст 1:", text1)
    newtext1=[]

    for i in text1:
        newtext1.append(i.encode("cp1251").hex())
    print("Текст в 16:", newtext1)

    print("Текст 2:", text2)
    newtext2=[]

    for i in text2:
        newtext2.append(i.encode("cp1251").hex())
    print("Текст 2 в 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 ("Ключ", key)
    xortext1=[]

    for i in range(len(newtext1)):
        xortext1.append("{:02x}".format(int(key[i],16)^int(newtext1[i],16)))
    print("Шифр текста 1 в 16:", xortext1)
    en_text1=bytearray.fromhex("".join(xortext1)).decode("cp1251")
    print("Шифр текста 1: ", en_text1)
    xortext2=[]

    for i in range(len(newtext2)):
        xortext2.append("{:02x}".format(int(key[i],16)^int(newtext2[i],16)))
    print("Шифр текста 2 в 16:", xortext2)
    en_text2=bytearray.fromhex("".join(xortext2)).decode("cp1251")
    print("Шифр текста 2:", en_text2)

    return key, xortext1, en_text1, xortext2, en_text2
```

{ #fig:002 }

```
In [5]: k, t1, et1, t2, et2=encrypt(p1, p2)

        Текст 1: text line one
        Текст в 16: ['74', '65', '78', '74', '20', '6c', '69', '6e', '65', '20', '6f', '6e', '65']
        Текст 2: text line two
        Текст 2 в 16: ['74', '65', '78', '74', '20', '6c', '69', '6e', '65', '20', '74', '77', '6f']
        Ключ ['d4', '14', 'fe', '9f', '16', 'e3', '22', 'a', '10', '4e', '5d', '4a', 'cc']
        Шифр текста 1 в 16: ['a0', '71', '86', 'eb', '36', '8f', '4b', '64', '75', '6e', '32', '24', 'a9']
        Шифр текста 1:   qтл6ЦKdun2$©
        Шифр текста 2 в 16: ['a0', '71', '86', 'eb', '36', '8f', '4b', '64', '75', '6e', '29', '3d', 'a3']
        Шифр текста 2:  qтл6ЦKdun)=J
```

{ #fig:003 }

Задала функцию decrypt, которая находит вид второго открытого текста без ключа и возвращает расшифрованное сообщение в форматах строки и 16-ой системе

```
In [6]: def decrypt(c1,c2,p1):

            print("Шифр текста 1:",c1)
            newc1=[]

            for i in c1:
                newc1.append(i.encode("cp1251").hex())
            print("Шифрт текста 1 в 16:",newc1)
            print("Шифрт текста 2:",c2)
            newc2=[]

            for i in c2:
                newc2.append(i.encode("cp1251").hex())
            print("Шифрт текста 2 в 16:",newc2)
            print("Открытый текст 1:",p1)
            newp1=[]

            for i in p1:
                newp1.append(i.encode("cp1251").hex())
            print("Откртый текст 1 в 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("Открытый текст 2 в 16: ", sp2)
            p2=bytearray.fromhex("".join(sp2)).decode("cp1251")
            print("Открытый текст 2: ", p2)

            return p1, p2
```

{ #fig:004 }

```
In [7]: decrypt(et1, et2, p1)

        Шифр текста 1:  qтл6ЦKdun2$©
        Шифрт текста 1 в 16: ['a0', '71', '86', 'eb', '36', '8f', '4b', '64', '75', '6e', '32', '24', 'a9']
        Шифрт текста 2:  qтл6ЦKdun)=J
        Шифрт текста 2 в 16: ['a0', '71', '86', 'eb', '36', '8f', '4b', '64', '75', '6e', '29', '3d', 'a3']
        Открытый текст 1: text line one
        Откртый текст 1 в 16 ['74', '65', '78', '74', '20', '6c', '69', '6e', '65', '20', '6f', '6e', '65']
        Открытый текст 2 в 16:  ['74', '65', '78', '74', '20', '6c', '69', '6e', '65', '20', '74', '77', '6f']
        Открытый текст 2:  text line two

Out[7]: ('text line one', 'text line two')

In [8]: decrypt(et2, et1, p2)

        Шифр текста 1:  qтл6ЦKdun)=J
        Шифрт текста 1 в 16: ['a0', '71', '86', 'eb', '36', '8f', '4b', '64', '75', '6e', '29', '3d', 'a3']
        Шифрт текста 2:  qтл6ЦKdun2$©
        Шифрт текста 2 в 16: ['a0', '71', '86', 'eb', '36', '8f', '4b', '64', '75', '6e', '32', '24', 'a9']
        Открытый текст 1: text line two
        Откртый текст 1 в 16 ['74', '65', '78', '74', '20', '6c', '69', '6e', '65', '20', '74', '77', '6f']
        Открытый текст 2 в 16:  ['74', '65', '78', '74', '20', '6c', '69', '6e', '65', '20', '6f', '6e', '65']
        Открытый текст 2:  text line one

Out[8]: ('text line two', 'text line one')
```

{ #fig:005 }

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:

C1(+)C2(+)P1 = P1(+)P2(+)P1 = P2,

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

1. Что будет при повторном использовании ключа при шифровании текста?

В таком случае мы получим исходное сообщение.

1. Как реализуется режим шифрования однократного гаммирования одним ключом двух открытых текстов?

Он реализуется по следующей формуле:

C1=P1(+)K

C2=P2(+)K,

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

1. 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.

1. 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 результате проделаной работы я освоила на практике применение режима однократного гаммирования на примере кодирования различных исходных текстов одним ключом.