Цель работы

Освоить на практике применение режима однократного гаммирования

Выполнение лабораторной работы

В ходе выполнения лабораторной работы была написана программа, которая может создавать зашифрованный текст, получая на вход исходный текст и ключ шифрования, расшифровывать текст с помощью зашифрованного текста и ключа, а также получать ключ, чтобы превратить заданный зашифрованный текст в расшифрованный текст.

Код программы:

```
def encoder(text, key):

    encodedText = ''

    if (len(text) == len(key)):

        if(type(key[0]) is int):
            for i in range(len(text)):
                encodedText += chr(ord(text[i]) ^ key[i])
        else:
            for i in range(len(text)):
                encodedText += chr(ord(text[i]) ^ ord(key[i]))

        return encodedText

    else:
        return

def decoder(encodedText, key):

    decodedText = ''

    if (len(encodedText) == len(key)):

        if(type(key[0]) is int):

            for i in range(len(encodedText)):

                decodedText += chr(ord(encodedText[i]) ^ key[i])

        elif(type(encodedText[0]) is int):

            for i in range(len(encodedText)):

                decodedText += chr(encodedText[i] ^ ord(key[i]))
```

```
        elif((type(encodedText[0]) is int) & (type(key[0]) is int)):
            for i in range(len(encodedText)):
                decodedText += chr(encodedText[i] ^ key[i])
        else:
            for i in range(len(encodedText)):
                decodedText += chr(ord(encodedText[i]) ^ ord(key[i]))
        return decodedText
    else:
        return
def keygen(text, encodedText, astype):
    if(astype == 'int'):
        key = []
    else:
        key = ''
    if(len(text) == len(encodedText)):
        if (astype == 'int'):
            for i in range(len(encodedText)):
                key.append(hex(ord(text[i]) ^ ord(encodedText[i])))
        else:
            for i in range(len(encodedText)):
                key += chr(ord(text[i]) ^ ord(encodedText[i]))
        return key
    else:
        return
txt = input()
key = input()
key = key.split(" ")

keyInInt = []
```

```
for element in key:
    keyInInt.append(int(element, 16))

encoded = encoder(txt, keyInInt)
print(encoded)

txt2 = input()
key2 = input()
key2 = key.split(" ")

keyInInt2 = []

for element in key2:
    keyInInt2.append(int(element, 16))

print(decoder(txt2, keyInInt2))

txt3 = input()
enctxt3 = input()

key3 = keygen(txt,enctxt,'int')

print(key3)
```

Результат выполнения команд (Рис. 1 - 3):

Генерация ключа:

```
txt3 = input()
enctxt3 = input()

key3 = keygen(txt3,enctxt3,'int')

print(key3)
```

```
Штирлиц - Вы Герой!!
С Новым Годом,друзья
['0x9', '0x462', '0x25', '0x7e', '0x9', '0x73', '0x7a', '0x0', '0x2400', '0x41e', '0x26', '0x75', '0x41c', '0x43f', '0x1', '0x
0', '0x7d', '0xe', '0x46d', '0x46e']
```

Получение закодированного сообщения из открытого:

```
txt = input()
key = input()
key = key.split(" ")

keyInInt = []

for element in key:
    keyInInt.append(int(element, 16))

encoded = encoder(txt, keyInInt)
print(encoded)
```

```
Штирлиц - Вы Герой!!
09 462 25 7E 09 73 7A 0 2400 41E 26 75 41C 43F 01 0 7D E 46D 46E
С Новым Годом,друзья
```

Декодирование текста по ключу:

```
txt2 = input()
key2 = input()
key2 = key2.split(" ")

keyInInt2 = []

for element in key2:
    keyInInt2.append(int(element, 16))

print(decoder(txt2, keyInInt2))
```

```
ЭюЯпеЎψò,0Лю▣фң◌ейQu
05 0C 17 7F 0E 4E 37 D2 94 10 09 2E 22 57 FF C8 0B B2 70 54
Штирлиц - Вы Герой!!
```