Лабораторная работа №7

Элементы криптографии. Однократное гаммирование

Монастырская Кристина Владимировна

Содержание

Список иллюстраций

Цель работы

Освоить на практике применение режима однократного гаммирования. [1]

Выполнение лабораторной работы

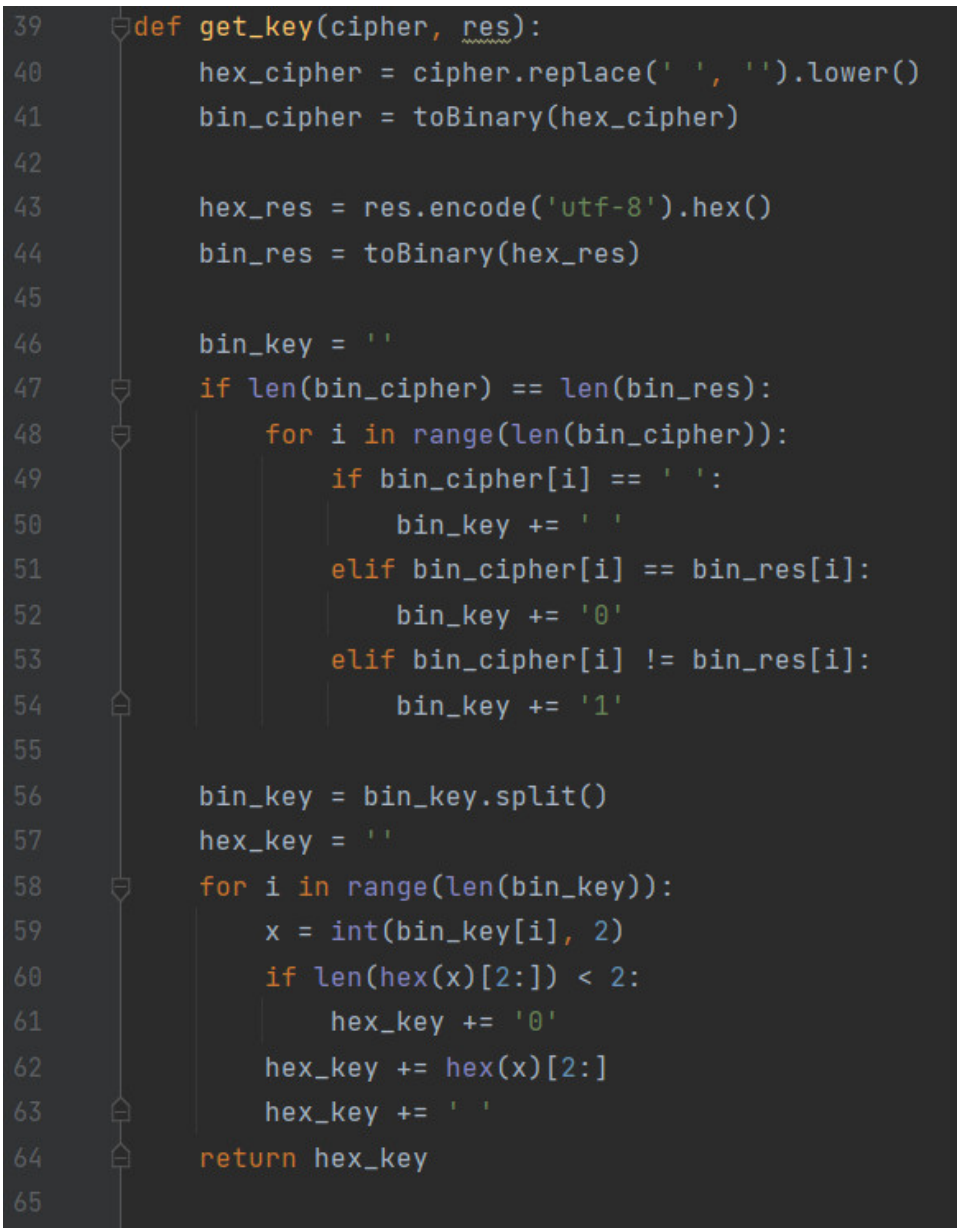
Разработала приложение, позволяющее шифровать и дешифровать данные в режиме однократного гаммирования.

Приложение может: 1. Определять вид шифротекста при известном ключе и известном открытом тексте. 2. Определить ключ, с помощью которого шифротекст может быть преобразован в некоторый фрагмент текста, представляющий собой один из возможных вариантов прочтения открытого текста.

Для реализации первого пункта реализовала функцию encryption ((рис. 1) В данной функции реализовано преобразование исходного текста и ключа в двоичный код в виде строки, а далее с этими строками происходит сложение по модулю 2. Благодаря этим действиям мы получили шифротекст в двоичном коде, который мы преобразовали в шестнадцатиричную строку.

```
def encryption(phrase, key):
    hex_phrase = phrase.encode('utf-8').hex()
    bin_phrase = toBinary(hex_phrase)

    hex_key = key.replace(' ', '').lower()
    bin_key = toBinary(hex_key)

    bin_cipher = ""
    if len(bin_phrase) == len(bin_key):
        for i in range(len(bin_phrase)):
            if bin_phrase[i] == ' ':
                bin_cipher += ' '
            elif bin_phrase[i] == bin_key[i]:
                bin_cipher += '0'
            elif bin_phrase[i] != bin_key[i]:
                bin_cipher += '1'

    bin_cipher = bin_cipher.split()
    cipher = ''
    for i in range(len(bin_cipher)):
        x = int(bin_cipher[i], 2)
        cipher += hex(x)[2:]
        cipher += ' '

    cipher = cipher.upper()
    return cipher

```

Для реализации второго пункта реализовала функцию get_key ((рис. 2), в которой нашла ключ преобразующий шифротекст в какой-либо заданный открытый текст. Функция реализована схожим образом: открытый итоговый текст и шифротекст преобразовываются в двоичный код, далее производится операция подбора значений ключа исходя из шифротекста и итогового текста. Получается аналогичное сложение по модулю 2. После этого ключ преобразуется в шестнадцатиричный формат.



```
def get_key(cipher, res):
    hex_cipher = cipher.replace(' ', '').lower()
    bin_cipher = toBinary(hex_cipher)

    hex_res = res.encode('utf-8').hex()
    bin_res = toBinary(hex_res)

    bin_key = ''
    if len(bin_cipher) == len(bin_res):
        for i in range(len(bin_cipher)):
            if bin_cipher[i] == ' ':
                bin_key += ' '
            elif bin_cipher[i] == bin_res[i]:
                bin_key += '0'
            elif bin_cipher[i] != bin_res[i]:
                bin_key += '1'

    bin_key = bin_key.split()
    hex_key = ''
    for i in range(len(bin_key)):
        x = int(bin_key[i], 2)
        if len(hex(x)[2:]) < 2:
            hex_key += '0'
        hex_key += hex(x)[2:]
        hex_key += ' '
    return hex_key

```

Рис. 1: Функция get_key

Результат работы программы ((рис. 3):

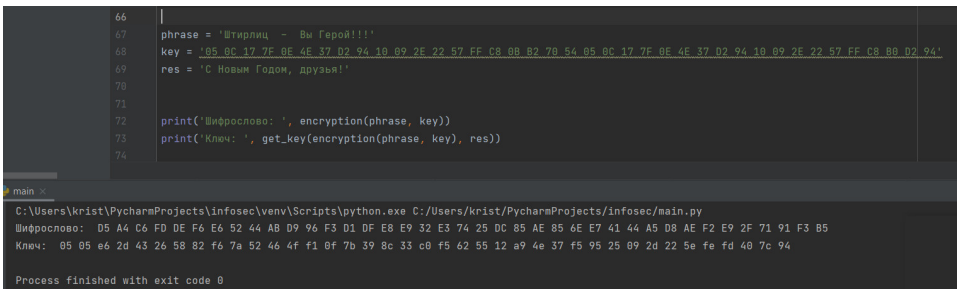


Рис. 2: Результат работы программы

Выводы

Освоить на практике применение режима однократного гаммирования.

Список литературы

1. Основы безопасности информационных систем : Учеб. пособие для студентов вузов, обучающихся по специальностям “Компьютер. безопасность” и “Комплекс. обеспечение информ. безопасности автоматизир. систем” / Д.А. Зегжда, А.М. Ивашко. - М. : Горячая линия - Телеком, 2000. - 449, [2] с. : ил., табл.; 21 см.; ISBN 5-93517-018-3.