Отчет по лабораторной работе №8

Элементы криптографии. Шифрование различных исходных текстов одним ключом

Евсеева Дарья Олеговна

29 октября, 2022

Содержание

Список иллюстраций

Цель работы

Целью данной работы является освоение на практике применения режима однократного гаммирования на примере кодирования различных исходных текстов одним ключом.

Задание

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

Приложение должно:

1. Определить вид шифротекстов двух открытых текстов при известном ключе.
2. Не зная ключа и не стремясь его определить прочитать оба текста.

Теоретическое введение

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.

В соответствии с теорией криптоанализа, если в методе шифрования используется однократная вероятностная гамма (однократное гаммирование) той же длины, что и подлежащий сокрытию текст, то текст нельзя раскрыть. Даже при раскрытии части последовательности гаммы нельзя получить информацию о всём скрываемом тексте.

Выполнение лабораторной работы

Выполнять работу будем в среде Jupyter Notebook на языке Python.

Для начала определим вспомогательные функции для дальнейшей работы.

```
def check_enc(array):
    arr = [i if len(i)==2 else "0"+i for i in array]
    return arr
```

```
def encode_str(string):
    s = []
    for i in string:
        s.append(i.encode('cp1251').hex())
    return s
```

Рис. 1: Вспомогательные функции

Далее определим основные функции. Первая функция ciphers_define будет использоваться для определения вида шифротекстов при известном ключе и двух открытых текстах. Сразу проверим результат работы функции на исходных данных.

```
def ciphers_define(text1, text2, key):
    t1 = encode_str(text1)
    t2 = encode_str(text2)
    c1 = []
    c2 = []
    for i in range(len(t1)):
        c1.append(hex(int(t1[i], 16) ^ int(key[i], 16))[2:])
        c2.append(hex(int(t2[i], 16) ^ int(key[i], 16))[2:])
    c1 = check_enc(c1)
    c2 = check_enc(c2)
    cipher1 = bytearray.fromhex("".join(c1)).decode('cp1251')
    cipher2 = bytearray.fromhex("".join(c2)).decode('cp1251')
    return cipher1, cipher2
```

```
p1 = "НаВашисходящийот1204"
p2 = "ВСеверныйфилиалБанка"
key = ['05', '0C', '17', '7F', '0E', '4E', '37', 'D2', '94', '10',
       '09', '2E', '22', '57', 'FF', 'C8', '0B', 'B2', '70', '54']
```

```
c1, c2 = ciphers_define(p1, p2, key)
print(c1)
print(c2)
```

```
ИмХцц¦Ж'zфцЧКs::Ђ@`
ЗЭтќnsb)}дбЕК·  л_љґ
```

Рис. 2: Функция определения шифротекстов

Как мы видим, мы получили шифротексты для двух открытых текстов с помощью одного ключа.

Далее реализуем определение неизвестного открытого текста по двум шифротекстам и второму открытому тексту. Напишем вспомогательную функцию для осуществления умножения по модулю два между тремя значениями, а также основную функцию text_define для определения неизвестного открытого текста. Проверим результат работы функции.

```python
def mod23(s1, s2, s3):
    res = []
    for i in range(len(s1)):
        res.append(hex(int(s1[i], 16) ^ int(s2[i], 16) ^ int(s3[i], 16))[2:])
    return res
```

```python
def text_define(cipher1, cipher2, text1, n):
    s1 = encode_str(cipher1)
    s2 = encode_str(cipher2)
    s3 = encode_str(text1)
    for i in range(n):
        txt = mod23(s1, s2, s3)
        txt = check_enc(txt)
        s1 = s3
        s2 = txt
    text2 = bytearray.fromhex("".join(txt)).decode('cp1251')
    return text2
```

```python
text = text_define(c1, c2, p1, 1)
print(text)
```

```
ВСеверныйфилиалБанка
```

Рис. 3: Функция определения открытого текста

Мы видим, что полученный открытый текст совпадает с исходными данными, а значит функцию успешно работает.

Выводы

В результате проделанной работы мы освоили на практике применение режима однократного гаммирования на примере кодирования различных исходных текстов одним ключом.

Список литературы

- Методические материалы к лабораторной работе, представленные на сайте “ТУИС РУДН” https://esystem.rudn.ru/