Лабораторная работа №7

Основы информационной безопасности

Кондрашова Анастасия Андреевна

Содержание

Список иллюстраций

Список таблиц

1 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. Наложение гаммы по сути представляет собой выполнение операции сложения по модулю 2 (XOR) (обозначаемая знаком ⊠) между элементами гаммы и элементами подлежащего сокрытию текста.

2 Цель работы

Освоить основы шифрования через однократное гаммирование.

3 Выполнение лабораторной работы

Лабораторная работа выполнена на языке Python 3 в среде Jupiter Notebook.

1. Создаём функцию, которая осуществляет однократное гаммирование посредством побитового XOR

```
def cript(text,key):
    if len(text) != len(key):
        return "Error; key must be the sme len as text"
    result = ''
    for i in range (len(key)):
        p= ord(text[i]) ^ ord(key[i])
        result += chr(p)
    return result
```

Рис. 3.1: Функция шифрования

2. Задаём текстовую строку и создаём случайный символьный ключ такой же длины

```
: text = "С новым годом друзья!"
```

Рис. 3.2: Исходные данные

```
from random import randint, seed
seed(31)
key = ''
for i in range (len(text)):
    key += chr(randint(0,5000))
print(key)
```

```
d࿐θஃςƮΨIꟸڈṼԿďǰŷݦꟺണ3ഉ ۮ
```

Рис. 3.3: Задание ключа

3. Запускаем функцию. В первом случае получаем зашифрованный текст. Далее, используя тот же самый ключ, осущвляем дешифровку текста. Так же, зная оригинальный текст и его шифорку, можем получить ключ. Все эти действия осуществляются через одну и ту же функцию.

```
cipher = cript(text,key)
print(cipher)
```

```
Xی³ؤفيجɈLւ؎CԳĭj̱ڿ□ل·ज़گ
```

```
(print(cript(cipher, key)))
```

```
С новым годом друзья!
```

Рис. 3.4: Результат работы программы - дешифровка текста

```
print(cript(text, cipher))
```

```
d࿐θஃςƮΨIꟸڈṼԿďǰŷݦꟺണ3ഉ ۮ
```

Рис. 3.5: Результат работы программы - ключ

4 Выводы

Я освоила основы шифрования через однократное гаммирование