Лабораторная работа №8

Основы информационной безопасности

Балакирева Д. С.

19 октября 2022

Российский университет дружбы народов, Москва, Россия

Элементы криптографии. Шифрование (кодирование) различных исходных текстов одним ключом

Цель лабораторной работы

- Освоить на практике применение однократного гаммирования при работе с различными текстами на одном ключе.

Задачи лабораторной работы

- Написать функцию, осуществляющую однократное гаммирование
- Защифровать два исходных текста
- Определить способ, при котором злоумышленник может получить данные, не зная ключа

Ход лабораторной работы

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.

Функция шифрования

Создаём функцию, которая осуществляет однократное гаммирование посредством побитового XOR

```
def cript(text,key):
    if len(text) != len(key):
        return " Ошибка: ключь должен быть той же длины, что и текст"
    result = ''
    for i in range(len(key)):
        n = ord(text[i] ^ ord(key[i]))
        result += chr(n)
        return result
```

Figure 1: Функция шифрования

Задаём две равные по длине текстовые строки и создаём случайный символьный ключ такой же длины

```python
text1 = "С Новым годом,друзья!!"
text2 = "Хорошего дня всем вам!"
```

```python
from random import randint, seed
seed(21)
key = ''
for i in range(len(text1)):
    key += chr(randint(0,5000))
print(key)
```

```
[illegible] Ŝ
```

Осуществляем шифрование двух текстов по ключу с помощью написанной функции

```
shifr1 = cript(text1, key)
shifr2 = cript(text2, key)
print(shifr1, shifr2, sep="\n")
```

Figure 3: Шифрование данных

Получение данных без ключа

Создаём переменную, которая, прогнав два шифрованных текста через побитовый XOR, поможет злоумышленнику получить один текст, зная другой, без ключа

```
: shifr12 = cript(shifr1, shifr2)

: print(cript(shifr12, text1))

  Хорошего дня всем вам!

: print(cript(shifr12, text2))

  С Новым годом,друзья!!
```

Получение части данных

Таким же способом можно получить часть данных

```
text2[9:17]
```

```
'дня всем'
```

```
shifr12_part = cript(shifr1[8:14],shifr2[8:14])
print(cript(shifr12_part, text2[8:14]))
```

```
годом,
```

Figure 5: Получение части данных

Выводы

- Освоено на практике применение режима однократного гаммирования
- Изучены недостатки однократного гаммирования