

Лабораторная работа №8

Основы информационной безопасности

Полиенко Анастасия Николаевна

19 октября 2022

Российский университет дружбы народов, Москва, Россия

НПМбд-01-19

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

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

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

Ход лабораторной работы

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 "Error: key must be the same lenght as text"  
    result = ''  
    for i in range(len(key)):  
        p = ord(text[i]) ^ ord(key[i])  
        result += chr(p)  
    return result
```

Figure 1: Функция шифрования

Исходные данные

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

```
text1 = "С Новым годом, друзья!"
text2 = "С днём рождения тебя!!"
```

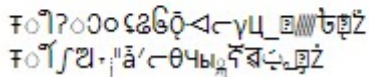
```
from random import randint, seed
seed(21)
key = ''
for i in range(len(text1)):
    key += chr(randint(0,5000))
print(key)
```

[illegible]

Figure 2: Исходные данные

Осуществляем шифрование двух текстов по ключу с помощью написанной функции

```
cipher1 = cript(text1, key)
cipher2 = cript(text2, key)
print(cipher1, cipher2, sep="\n")
```



Two lines of encrypted data, appearing as random characters and symbols.

Figure 3: Шифрование данных

Получение данных без ключа

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

```
zlo = cript(cipher1, cipher2)
```

```
print(cript(zlo, text1))
```

С днём рождения тебя!!

```
print(cript(zlo, text2))
```

С Новым годом, друзья!

Figure 4: Получение данных без ключа

Таким же способом можно получить часть данных

```
text2[7:15]
```

```
'рождения'
```

```
zlo_part = cript(cipher1[7:15], cipher2[7:15])  
print(cript(zlo_part, text2[7:15]))
```

```
годом,
```

Figure 5: Получение части данных

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