Лабораторная работа №8

Информационная безопасность

Банникова Екатерина Алексеевна

Содержание

Список иллюстраций

Список таблиц

1 Цель работы

Освоить на практике применение однократного гаммирования при работе с различными текстами на дном ключе

2 Теоретическое введение

3 Выполнение лабораторной работы

Создаем 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"
    res = ''
    for i in range(len(key)):
        p = ord(text[i]) ^ ord(key[i])
        res += chr(p)
    return res
```

Рис. 3.1: Функция шифрования

Задаем две равные по длине текстовые строки и создаем 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(text)):
    key += chr(randint(0,5000))
print(key)
```

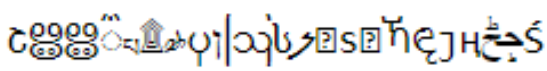

Рис. 3.2: Исходные данные

Осуществляем шифрование двух текстов по ключу с помощью написанной функции

```
cipher1 = cript(text1, key)
cipher2 = cript(text2, key)
print(cipher1)
print(cipher2)
```

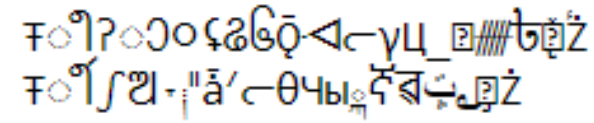

Рис. 3.3: Шифрование данных

Создаем переменную, которая, прогнав два шифрованных текста через побитый XOR, поможет злоумышлинику получить один текс, зная другой, без ключа

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

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

```
С днём рождения тебя!!
```

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

```
С Новым годом, друзья!
```

Рис. 3.4: 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,
```

Рис. 3.5: Получение части данных

4 Выводы

Я освоила на практике применение режима однократного гаммирования при работе с несколькими текстами.