

Отчёт по лабораторной работе №7

Дисциплина: Основы информационной безопасности

Полиенко Анастасия Николаевна, НПМбд-01-19

Содержание

1	Цель работы	4
2	Теоретическое введение	5
3	Выполнение лабораторной работы	6
4	Выводы	9
	Список литературы	10

Список иллюстраций

3.1	Функция шифрования	6
3.2	Исходные данные	7
3.3	Результат работы программы	8

1 Цель работы

Освоить основы шифрования через однократное гаммирование.

2 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 \boxplus) между элементами гаммы и элементами подлежащего сокрытию текста.

3 Выполнение лабораторной работы

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

Создаём функцию, которая осуществляет однократное гаммирование посредством побитового XOR (рис. 3.1)

```
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
```

Рис. 3.1: Функция шифрования

Задаём текстовую строку и создаём случайный символьный ключ такой же длины (рис. 3.2)

```
text = "С Новым годом, друзья!"
```

```
from random import randint, seed  
seed(42)  
key = ''  
for i in range(len(text)):  
    key += chr(randint(0,5000))  
print(key)
```

```
тiE_ΔV_—'HЕAô_ЖoI9U_OL.т||
```

Рис. 3.2: Исходные данные

Запускаем функцию. В первом случае получаем зашифрованный текст. Далее, используя тот же самый ключ, осуществляем дешифровку текста. Так же, зная оригинальный текст и его шифровку, можем получить ключ.

Все эти действия осуществляются через одну и ту же функцию. (рис. 3.3)

فمن ثم، فإنَّ

С Новым годом, друзья!

विष्णु- 'मम' शब्दोऽङ्गिभूतः।

8

4 Выводы

Я освоила на практике применение режима однократного гаммирования.

Список литературы