Лабораторная работа №7

Дисциплина: Основы информационной безопасности

Коновалова Татьяна Борисовна

Содержание

1 Цель работы 5

2 Теоретические данные 6

3 Задание 7

4 Выполнение лабораторной работы 8

5 Выводы 10

6 Библиография 11

Список иллюстраций

4.1 Функция шифрования 8
4.2 Исходная строка 8
4.3 Исходные данные 8
4.4 Результат работы программы 9

Список таблиц

1 Цель работы

Цель лабораторной работы — Освоить основы шифрования через однократное гаммирование.

2 Теоретические данные

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.

3 Задание

1.Подобрать ключ, чтобы получить сообщение «С Новым Годом, друзья!»; 2.Разработать приложение, позволяющее шифровать и дешифровать данные в режиме однократного гаммирования.

4 Выполнение лабораторной работы

Лабораторную работу выполнила на языке Pythin 3 в среде Jupiter Notebook.

1.Создала функцию, которая осуществляет однократное гаммирование посредством побитового XOR (рис. [4.1])

```
Ввод [6]:
def cript(text, key):
    if len(text) != len(key):
        return "Предупреждение: длины текста и ключа должны быть одинаковы!"
    result = ''
    for i in range(len(key)):
        p = ord(text[i]) ^ ord(key[i])
        result += chr(p)
    return result
```

Рис. 4.1: Функция шифрования

2.Задала текстовую строку и случайный символьный ключ такой же длины (рис. [4.2]) и (рис. [4.3]).

```
Ввод [7]:
text = 'С Новым годом, друзья!'
```

Рис. 4.2: Исходная строка

```
Ввод [8]:
from random import randint, seed
seed(20)
key = ''
for i in range(len(text)):
    key += chr(randint(0, 100))
print(key)
```

```
(<J9!VQ)I44
```

Рис. 4.3: Исходные данные

3.Запустила функцию. В первом случае получила зашифрованный текст. После этого, используя тот же самый ключ, осуществила дешифровку текста. Так же, зная оригинальный текст и его шифорку, с помощью кода могу получить ключ.

Все эти действия осуществляются через одну и ту же функцию. (рис. [4.4])

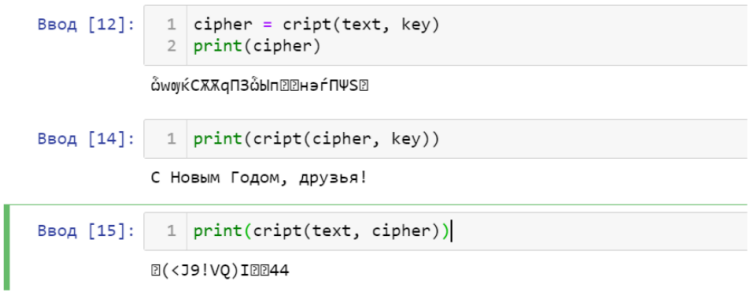

Рис. 4.4: Результат работы программы

5 Выводы

Освоила основы шифрования через однократное гаммирование

6 Библиография

СПИСОК ЛИТЕРАТУРЫ

1.Медведовский И.Д., Семьянов П.В., Платонов В.В. Атака через Internet. — НПО “Мир и семья-95”, 1997. — URL: http://bugtraq.ru/library/books/attack1/index.html

2.Теоретические знания, приведённые в Лабораторной работе №7 - https://esystem.rudn.ru/pluginfile.php/2090133/mod_resource/content/2/007-lab_crypto-gamma.pdf

3.Запечников С. В. и др. Информационн~пасность открытых систем. Том 1. — М.: Горячаая линия -Телеком, 2006.

СПИСОК ИНТЕРНЕТ-ИСТОЧНИКОВ

1.[Электронный ресурс] - доступ: https://codeby.school/blog/informacionnaya-bezopasnost/razgranichenie-dostupa-v-linux-znakomstvo-s-astra-linux

2.[Электронный ресурс] - доступ: https://debianinstall.ru/diskretsionnoe-razgranichenie-dostupa-linux/