Лабораторная работа №7

Дисциплина: Основы информационной безопасности

Дарижапов Тимур Андреевич

Содержание

List of Figures

List of Tables

1 Цель работы

Освоить на практике применение режима однократного гаммирования.

2 Теоретическое введение

Гаммирование - наложение (снятие) на открытые (зашифрованные) данные последовательности элементов других данных, полученной с помощью некоторого криптографического алгоритма, для получения зашифрованных (открытых) данных. Основная формула, необходимая для реализации однократного гаммирования: Ci = Pi XOR Ki, где Ci - i-й символ зашифрованного текста, Pi - i-й символ открытого текста, Ki - i-й символ ключа. Аналогичным образом можно найти ключ: Ki = Ci XOR Pi. Необходимые и достаточные условия абсолютной стойкости шифра: • длина открытого текста равна длине ключа • ключ должен использоваться однократно • ключ должен быть полностью случаен

Более подробно см. в [1].

3 Выполнение лабораторной работы

1)Код программы(Рисунок 3.1).

```
In [1]: import random
        from random import seed
        import string

In [2]: def cipher_text_function(text, key):
            if len(key) != len(text):
                return "Ключ и текст должны быть одной длины!"
            cipher_text = ''
            for i in range(len(key)):
                cipher_text_symbol = ord(text[i]) ^ ord(key[i])
                cipher_text += chr(cipher_text_symbol)
            return cipher_text

In [38]: text = "С новым годом, друзья!"

In [27]: key = ''
         seed(21)
         for i in range(len(text)):
             key += random.choice(string.ascii_letters + string.digits)
         print(key)

         kASAOsE1nYEZ9GlGHpYaax

In [30]: cipher_text = cipher_text_function(text, key)
         print('Шифротекст:', cipher_text)

         Шифротекст: ъаƷѾѽиɱ▯йѧψѤSkLɵJгƷЭЮY

In [48]: print('Открытый текст:', cipher_text_function(cipher_text, key))

         Открытый текст: С новым годом, друзья!

In [49]: print('Ключ:', cipher_text_function(text, cipher_text))

         Ключ: kASAOsE1nYEZ9GlGHpYaax
```

Figure 3.1: Приложение, реализующее режим однократного гаммирования

• In[21]: импорт необходимых библиотек • In[22]: функция, реализующая сложение по модулю два двух строк • In[23]: открытый/исходный текст • In[24]: создание ключа той же длины, что и открытый текст • In[25]: получение шифротекста с помощью функции, созданной ранее, при условии, что известны открытый текст и ключ • In[26]: получение открытого текста с помощью функции, созданной

ранее, при условии, что известны шифротекст и ключ • In[27]: получение ключа с помощью функции, созданной ранее, при условии, что известны открытый текст и шифротекст

4 Выводы

- В ходе выполнения данной лабораторной работы я освоил на практике применение режима однократного гаммирования.

5 Список литературы

- Однократное гаммирование [Электронный ресурс]. URL: https://esystem.rudn.ru/pluginfile.
  lab_cryptogamma.pdf.