Лабораторная работа №8

Информационная безопасность

Маметкадыров Ынтымак | НПМбд-02-20

Содержание

Список иллюстраций

1 Цель работы

Освоить на практике применение режима однократного гаммирования на примере кодирования различных исходных текстов одним ключом.

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.

В данном случае для двух шифротекстов будет две формулы: C1 = P1 xor K и C2 = P2 xor K, где индексы обозначают первый и второй шифротексты соответственно.

Если нам известны оба шифротекста и один открытый текст, то мы можем найти другой открытый текст, это следует из следующих формул: C1 xor C2 = P1 xor K xor P2 xor K = P1 xor P2, C1 xor C2 xor P1 = P1 xor P2 xor P1 = P2.

Более подробно см. в [1].

3 Выполнение лабораторной работы

Код программы (рис. 3.1).

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

```
[2]:
def cipher(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
```

```
[15]:
text_1 = 'С Новым годом, друзья!'
text_2 = 'Поздравляем с 8 марта!'
```

```
[6]:
key = ''
seed(10)
for i in range(len(text_1)):
    key += random.choice(string.ascii_letters + string.digits)
key
```

```
[6]:
'KcBEKanD0F0rPZkcHFuep8'
```

```
[8]:
cipher_text_1 = cipher(text_1, key)
cipher_text_2 = cipher(text_2, key)
print("Шифротекст 1: ", cipher_text_1)
print("Шифротекст 2: ", cipher_text_2)
```

```
Шифротекст 1:  ӜCџoѹђђdЃџЄьѪvKїJSтЩп
Шифротекст 2:  єйvψЋёќѿѿѳЌRБzSCѴѶeЧр
```

```
[9]:
print('Открытый текст 1: ', cipher(cipher_text_1, key))
print('Открытый текст 2: ', cipher(cipher_text_2, key))
```

```
Открытый текст 1:  С Новым годом, друзья!
Открытый текст 2:  Поздравляем с 8 марта!
```

Рис. 3.1: Приложение, реализующее режим однократного гаммирования для двух текстов одним ключом, Часть 1

- In[1]: импорт необходимых библиотек
- In[2]: функция, реализующая сложение по модулю два двух строк
- In[15]: открытые/исходные тексты (одинаковой длины)
- In[6]: создание ключа той же длины, что и открытые тексты

- In[8]: получение шифротекстов с помощью функции, созданной ранее, при условии, что известны открытые тексты и ключ
- In[9]: получение открытых текстов с помощью функции, созданной ранее, при условии, что известны шифротексты и ключ

```
In [9]: cipher_text_xor = cipher_text_function(cipher_text_1, cipher_text_2)
        print('Первый шифротекст XOR Второй шифротекст:', cipher_text_xor)

        Первый шифротекст XOR Второй шифротекст: >O*
        r{␂Л|O}␂Д|sw␂␂

In [10]: print('Первый открытый текст:', cipher_text_function(cipher_text_xor, text_2))
         print('Второй открытый текст:', cipher_text_function(cipher_text_xor, text_1))

         Первый открытый текст: С Новым годом, друзья!
         Второй открытый текст: Поздравляем с 8 марта!

In [12]: text_1_ = text_1[3:6]
         print('Часть первого открытого текста:', text_1_)

         Часть первого открытого текста: овы

In [14]: cipher_text_xor_ = cipher_text_function(cipher_text_1[3:6], cipher_text_2[3:6])
         print('Часть второго открытого текста:', cipher_text_function(cipher_text_xor_, text_1_))

         Часть второго открытого текста: дра
```

Рис. 3.2: Приложение, реализующее режим однократного гаммирования для двух текстов одним ключом, Часть 2

- In[11]: сложение по модулю два двух шифротекстов с помощию функции, созданной ранее
- In[12]: получение открытых текстов с помощью функции, созданной ранее, при условии, что известны оба шифротекста и один из открытых текстов
- In[16]: получение части первого открытого текста (срез)
- In[17]: получение части второго текста (на тех позициях, на которых расположены символы части первого открытого текста) с помощью функции, созданной ранее, при условии, что известны оба шифротекста и часть первого открытого текста

4 Выводы

В ходе выполнения данной лабораторной работы мы освоили на практике применение режима однократного гаммирования на примере кодирования различных исходных текстов одним ключом.

Список литературы

1. Однократное гаммирование [Электронный ресурс]. URL: https://esystem.rudn.ru/pluginfile.php/1651641/mod_resource/content/2/008-lab_crypto-key.pdf.