Лабораторная работа № 7

Элементы криптографии. Однократное гаммирование

Форис Анастасия Дмитриевна

Содержание

1 Цель работы

Освоить на практике применение режима однократного гаммирования.

2 Выполнение лабораторной работы

Код программы (рис.7).

File Edit View Insert Cell Kernel Help

Run Code

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

In [7]:
```
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 [8]:
```
text = "С Новым годом, друзья"
```

In [9]:
```
key = ''
seed(23)
for i in range(len (text)):
   key += random.choice(string.ascii_letters + string.digits)
print (key)
```

```
7X8s51fbLtByHwiUmrCao
```

In [11]:
```
cipher_text = cipher_text_function (text, key)
print( 'Шифротекст:', cipher_text)
```

```
Шифротекст: ЖхХэЇЮњВ№ъЎчV[IwЭбVЭР
```

In [12]:
```
print('Отрытый текст:', cipher_text_function (cipher_text, key))
```

```
Отрытый текст: С Новым годом, друзья
```

In [13]:
```
print('Ключ:', cipher_text_function(text, cipher_text))
```

```
Ключ: 7X8s51fbLtByHwiUmrCao
```

Рис. 7: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 помощию 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

3 Выводы

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