

Лабораторная работа

№ 8

Информационная безопасность

Цель работы

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

Выполнение лабораторной работы

Код программы

```
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 [11]: text_1 = "С новым годом, друзья!"
        text_2 = "Поздравляем с 8 марта!"
```

```
In [12]: key = ''
        seed(23)
        for i in range(len(text_1)):
            key += random.choice(string.ascii_letters + string.digits)
        print(key)

7X8s51fbLtByHwiUmrCaoN
```

- In[1]: импорт необходимых библиотек
- In[2]: функция, реализующая сложение по модулю два двух строк
- In[11]: открытые/исходные тексты (одинаковой длины)
- In[12]: создание ключа той же длины, что и открытые тексты

Выполнение лабораторной работы

```
In [13]: ► cipher_text_1 = cipher_text_function(text_1, key)
          cipher_text_2 = cipher_text_function(text_2, key)
          print('Первый шифротекст:', cipher_text_1)
          print('Второй шифротекст:', cipher_text_2)
```

Первый шифротекст: ЖхSэЇОњBѡъŸчV[IwЭ6VЭРо
Второй шифротекст: ШАЦчvЁельfсŸYЪWQuётfУцo

```
In [14]: ► print('Первый открытый текст:', cipher_text_function(cipher_text_1, key))
          print('Второй открытый текст:', cipher_text_function(cipher_text_2, key))
```

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

- In[13]: получение шифротекстов с помощью функции, созданной ранее, при условии, что известны открытые тексты и ключ
- In[14]: получение открытых текстов с помощью функции, созданной ранее, при условии, что известны шифротексты и ключ

Выполнение лабораторной работы

```
In [15]: cipher_text_xor = cipher_text_function(cipher_text_1, cipher_text_2)
print('Первый шифротекст XOR Второго шифротекста', cipher_text_xor)
```

Первый шифротекст XOR Второго шифротекста >0

r{[Л|0}Д|sw

```
In [16]: 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 [17]: text_1 = text_1[3:6]
print('Часть первого открытого текста:', text_1)
```

Часть первого открытого текста: овы

```
In [18]: 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

- In[15]: сложение по модулю два двух шифротекстов с помощью функции, созданной ранее
- In[16]: получение открытых текстов с помощью функции, созданной ранее, при условии, что известны оба шифротекста и один из открытых текстов
- In[17]: получение части первого открытого текста (срез)
- In[18]: 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

Выводы

В ходе выполнения данной лабораторной работы я освоила на практике применение режима однократного гаммирования на примере кодирования различных исходных текстов одним ключом