

# **Лабораторная работа №8**

**Дисциплина: Информационная безопасность**

Боровикова Карина Владимировна

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>5</b>
<b>2</b>	<b>Задание</b>	<b>6</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>7</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>8</b>
<b>5</b>	<b>Выводы</b>	<b>10</b>
	<b>Список литературы</b>	<b>11</b>

# Список иллюстраций

4.1	Первая часть кода . . . . .	8
4.2	Вторая часть кода . . . . .	9

## Список таблиц

# 1 Цель работы

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

## 2 Задание

- Написать программу для шифровки и дешифровки текстов
- Произвести работу по шифровке и дешифровке текстов

### 3 Теоретическое введение

**Шифрование** – это такое преобразование исходного сообщения, которое не позволит всяким нехорошим людям прочитать данные, если они это сообщение перехватят.

**Исходное сообщение** – это, собственно, то, что мы хотим зашифровать. Классический пример — текст.

**Шифрованное сообщение** – это сообщение, прошедшее процесс шифрования.

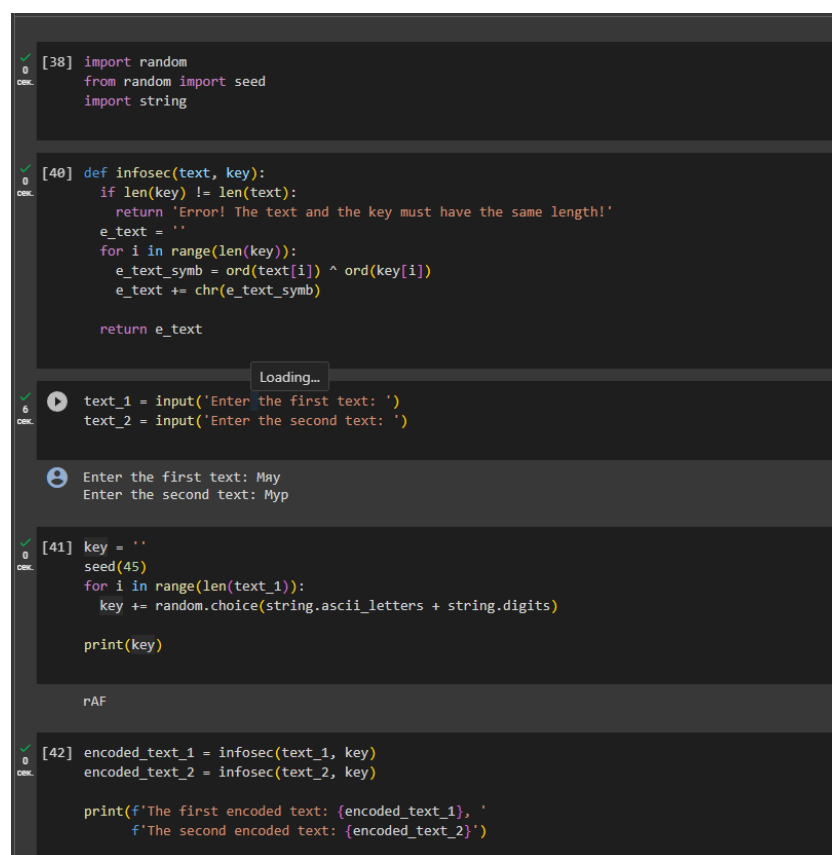
**Шифр** — это сам алгоритм, по которому мы преобразовываем сообщение.

**Ключ** — это компонент, на основе которого можно произвести шифрование или дешифрование.

**Алфавит** – это перечень всех возможных символов в исходном и зашифрованном сообщении. Включая цифры, знаки препинания, пробелы, отдельно строчные и заглавные буквы и т.д.

## 4 Выполнение лабораторной работы

1. Пишем код на языке Python: (рис. 4.1 - 4.2):



```
[38] import random
      from random import seed
      import string

[40] def infosec(text, key):
      if len(key) != len(text):
          return 'Error! The text and the key must have the same length!'
      e_text = ''
      for i in range(len(key)):
          e_text_symb = ord(text[i]) ^ ord(key[i])
          e_text += chr(e_text_symb)

      return e_text

Loading...

text_1 = input('Enter the first text: ')
text_2 = input('Enter the second text: ')

Enter the first text: May
Enter the second text: Myp

[41] key = ''
      seed(45)
      for i in range(len(text_1)):
          key += random.choice(string.ascii_letters + string.digits)

      print(key)

rAF

[42] encoded_text_1 = infosec(text_1, key)
      encoded_text_2 = infosec(text_2, key)

      print(f'The first encoded text: {encoded_text_1}, '
            f'The second encoded text: {encoded_text_2}')
```

Рис. 4.1: Первая часть кода

- 
- 
-



- 
- 
- 

```

[42] encoded_text_1 = infosec(text_1, key)
      encoded_text_2 = infosec(text_2, key)

      print(f'The first encoded text: {encoded_text_1}, '
            f'The second encoded text: {encoded_text_2}')

The first encoded text: 39S, The second encoded text: 3bI

[43] print(f'The first decoded text: {infosec(encoded_text_1, key)}, '
          f'The second decoded text: {infosec(encoded_text_2, key)}')

The first decoded text: May, The second decoded text: Myp

[44] e_text_xor = infosec(encoded_text_1, encoded_text_2)
      print(f'The first xor encoded text: {e_text_xor}')

The first xor encoded text: 43

[45] print(f'The first decoded text: {infosec(e_text_xor, text_2)}, '
          f'The second decoded text: {infosec(e_text_xor, text_1)}')

The first decoded text: May, The second decoded text: Myp

[46] text_1_slice = text_1[1:3]
      print(f'The known slice of the first text: {text_1_slice}')

The known slice of the first text: my

[47] e_text_xor_slice = infosec(encoded_text_1[1:3], encoded_text_2[1:3])
      print(f'Part of the second decoded text: {infosec(e_text_xor_slice, text_1_slice)}')

Part of the second decoded text: yp

```

Рис. 4.2: Вторая часть кода

- созданной ранее
- при условии, что известны оба шифротекста и один из открытых текстов
- 
- [47]: 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

## 5 Выводы

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

## **Список литературы**