

# **Лабораторная работа №3**

**Дисциплина: Математические основы защиты информации и  
информационной безопасности**

Аветисян Давид Артурович

# Содержание

1	Цель работы	5
2	Задание	6
3	Выполнение лабораторной работы	7
4	Выводы	9

## List of Tables

# List of Figures

3.1	Запрос текста и пароля у пользователя . . . . .	7
3.2	Шифрование гаммированием на языке Python . . . . .	8
3.3	Проверка метода шифрования гаммированием . . . . .	8

# 1 Цель работы

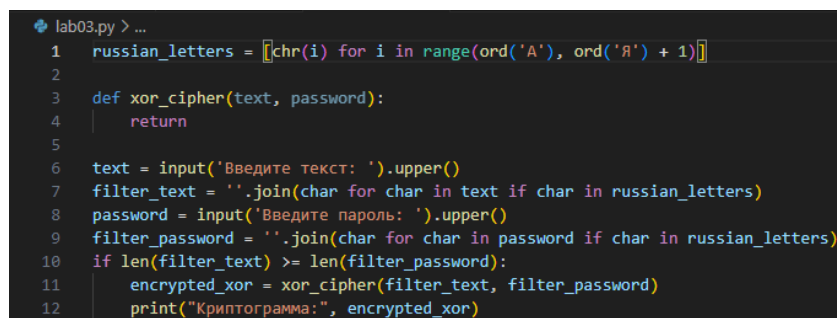
Познакомиться с шифрованием гаммированием.

## 2 Задание

1. Реализовать шифрование гаммированием.

### 3 Выполнение лабораторной работы

- 1) Я решил реализовать шифрование гаммированием на языке Python. Сначала я задал матрицу с русскими буквами, а также реализовал запрос текста и пароля у пользователя. Все буквы в тексте и пароля я сделал заглавными, а затем я отфильтровал текст и пароль, оставив в нём только русские буквы.



```
lab03.py > ...
1  russian_letters = [chr(i) for i in range(ord('А'), ord('Я') + 1)]
2
3  def xor_cipher(text, password):
4      return
5
6  text = input('Введите текст: ').upper()
7  filter_text = ''.join(char for char in text if char in russian_letters)
8  password = input('Введите пароль: ').upper()
9  filter_password = ''.join(char for char in password if char in russian_letters)
10 if len(filter_text) >= len(filter_password):
11     encrypted_xor = xor_cipher(filter_text, filter_password)
12     print("Криптограмма:", encrypted_xor)
```

Figure 3.1: Запрос текста и пароля у пользователя

- 2) Далее я реализовал само шифрование гаммированием. Я задал три матрицы: для текста, для гаммы и для результата. Пароль я увеличил повторением до длины текста, как указано в теории к лабораторной работе №3. Далее я, опираясь на матрицу с русскими буквами, заполнил первые две матрицы значениями, на которых стоят буквы из текста и пароля соответственно. А уже затем я заполнял третью матрицу складывая значения из первой и второй матрицы и находя остаток от деления. После чего я преобразовал каждое значение из третьей матрицы в букву, исходя из матрицы с русскими буквами, и вывел пользователю.

```

lab03.py > ...
1  russian_letters = [chr(i) for i in range(ord('А'), ord('Я') + 1)]
2
3  def xor_cipher(text, password):
4      data = []
5      for char in text:
6          data.append(russian_letters.index(char))
7      key = []
8      repeated_password = (password * (len(text) // len(password) + 1))[:len(text)]
9      for char in repeated_password:
10         key.append(russian_letters.index(char))
11     cryptogram = []
12     for i in range(len(text)):
13         index = (data[i]+key[i]+1) % len(russian_letters)
14         cryptogram.append(russian_letters[index])
15     result = ''.join(cryptogram)
16     return result
17
18 text = input('Введите текст: ').upper()
19 filter_text = ''.join(char for char in text if char in russian_letters)
20 password = input('Введите пароль: ').upper()
21 filter_password = ''.join(char for char in password if char in russian_letters)
22 if len(filter_text) >= len(filter_password):
23     encrypted_xor = xor_cipher(filter_text, filter_password)
24     print("Криптограмма:", encrypted_xor)

```

Figure 3.2: Шифрование гаммированием на языке Python

- 3) Далее я запустил два теста через командную строку. Один тест как в теории к лабораторной работе №3. Второй тест для дополнительной проверки. Шифрование совпало с тестом в лабораторной работе №3, и реализовано верно.

```

C:\Users\yaeda\OneDrive\Рабочий стол\RUDN\Магистратура\МОЗИИИБ>py lab03.py
Введите текст: приказ
Введите пароль: гамма
Криптограмма: УСХЧБЛ

C:\Users\yaeda\OneDrive\Рабочий стол\RUDN\Магистратура\МОЗИИИБ>py lab03.py
Введите текст: я хочу играть в доту
Введите пароль: с мужиками
Криптограмма: СВВЮЬУДЭЙДЙЦЛЧЭФ

```

Figure 3.3: Проверка метода шифрования гаммированием



## 4 Выводы

Я реализовал шифрование гаммированием.