

# **Лабораторна работа № 3**

**Шифрование Гаммированием**

Покрас Илья Михайлович

# Содержание

<b>1</b>	<b>Цель работы</b>	<b>4</b>
<b>2</b>	<b>Задание</b>	<b>5</b>
<b>3</b>	<b>Теоретическое введение</b>	<b>6</b>
<b>4</b>	<b>Выполнение лабораторной работы</b>	<b>7</b>
<b>5</b>	<b>Выводы</b>	<b>9</b>
	<b>Список Литературы</b>	<b>10</b>

## Список иллюстраций

4.1	Инициализация переменных и вызов функции . . . . .	7
4.2	Результат выполнения программного кода . . . . .	8

# 1 Цель работы

Изучить и реализовать алгоритм шифрования гаммированием конечной гаммой.

## 2 Задание

- Создать алгоритм шифрования гаммированием

### 3 Теоретическое введение

Шифрование гаммированием — это симметричный метод шифрования, при котором на открытый текст накладывается последовательность, сформированная из случайных чисел.

Процесс шифрования:

- Генерируется ключевой поток гаммы из непредсказуемых и независимых друг от друга случайных чисел.
- Каждый символ сообщения комбинируется с символом ключевого потока гаммы с помощью операции XOR (исключающее ИЛИ).

## 4 Выполнение лабораторной работы

Я создал функцию шифрования гаммированием, с текстом и ключом шифрования. Далее создается алфавит, содержащий строчные и заглавные буквы английского и русского алфавитов, и создается массив для зашифрованного текста. После идет функция расчета позиций букв в алфавите с использованием XOR и записываются в массив, который будет возвращаться как строковое значение с конвертированными позициями алфавита непосредственно в буквы (рис. ??).

```
function gamma_encrypt(text, key)
  arr = [string(i) for i in 'a':'z']; [push!(arr, string(i)) for i in 'A':'Z']; [push!(arr, string(i)) for i in 'А':'Я']; [push!(arr, string(i)) for i in 'а':'я']
  new_text = Vector{String}()
  index = 1

  for i in eachindex(text)
    if i > length(key)
      if isletter(text[i])
        c = findfirst(isequal(string(text[i])), arr) ⊕ findfirst(isequal(string(key[i-length(key)*index])), arr)
      else
        c = findfirst(isequal(string(key[i-length(key)*index])), arr)
      end
      index = i ÷ length(key) + 1
    else
      if isletter(text[i])
        c = findfirst(isequal(string(text[i])), arr) ⊕ findfirst(isequal(string(key[i])), arr)
      else
        c = findfirst(isequal(string(key[i])), arr)
      end
    end
    push!(new_text, string(c))
  end

  return join([arr[tryparse{Int, i}] for i in new_text])
end
```

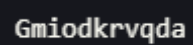
Далее я инициализировал переменные, которые содержат исходный текст и ключ шифрования, после чего использовал эти данные в вызове функции шифрования решеток (рис. 4.1).

```
text = "Hello world"
key = "checkkey"

println(gamma_encrypt(text, key))
```

Рис. 4.1: Инициализация переменных и вызов функции

И получил следующий результат (рис. 4.2):



Gmiodkrvqda

Рис. 4.2: Результат выполнения программного кода



## **5 Выводы**

Я изучил и реализовал алгоритм шифрования гаммированием конечной гаммой.

# Список Литературы

1. Julia - Control Flow
2. Julia - Mathematical Operations
3. Julia - Strings
4. Julia - Arrays
5. Julia - Collections and Data Structures