

# Лабораторная работа №3. Шифрование гаммированием

Выполнила: Лебедева Ольга Андреевна

РУДН, Москва, Россия

2025

Изучить и реализовать на языке Julia[1] метод симметричного шифрования гаммированием[2] с использованием операции сложения по модулю.

- 1 Ознакомиться с принципом работы метода гаммирования.
- 2 Реализовать алгоритм шифрования и дешифрования на языке Julia.
- 3 Провести тестирование алгоритма.

Объект исследования: симметричный метод шифрования гаммированием.

Предмет исследования: алгоритм шифрования гаммированием, его реализация средствами Julia.

Шифрование гаммированием — это метод симметричного шифрования, при котором каждый символ открытого текста складывается по модулю с символом ключевой последовательности (гаммы).

Гамма — псевдослучайная последовательность чисел или символов, используемая для наложения на исходный текст.

Модуль — число, определяющее диапазон возможных значений результата арифметической операции. В данной работе используется модуль 33, соответствующий количеству букв русского алфавита.

Ключ — исходное значение (фраза или набор чисел), из которого строится гамма и с помощью которого выполняется как шифрование, так и дешифрование.

Программное обеспечение:

- Язык программирования Julia.
- Среда разработки JupyterLab / VS Code.

Методы:

- Использование арифметики по модулю для операций над элементами алфавита.
- Преобразование текста в числовые последовательности и обратно.
- Работа с символьными строками и циклами в Julia.
- Реализация повторяющейся гаммы при шифровании длинных сообщений.

Метод гаммирования относится к симметричным криптографическим методам. Каждый символ открытого текста представляется числом по таблице алфавита и складывается с соответствующим числом гаммы по модулю мощности алфавита.

$$c_i = (p_i + g_i - 1) \bmod 33 + 1$$

$$p_i = (c_i - g_i - 1) \bmod 33 + 1$$

где

$p_i$  — код  $i$ -го символа исходного текста,

$g_i$  — код  $i$ -го символа гаммы,

$c_i$  — код  $i$ -го символа шифртекста.

Если длина гаммы меньше длины текста, гамма повторяется циклически.

Метод симметричен: операция шифрования и расшифрования описывается одинаковой формулой с противоположным знаком.

- 1 Реализовать алгоритм шифрования гаммированием на русском алфавите.
- 2 Написать функции преобразования текста в числовую форму и обратно.
- 3 Реализовать операции шифрования и расшифрования по приведённым формулам.
- 4 Проверить корректность работы программы на примере:



# Шифрование гаммированием

Выполним задание с помощью языка Julia:

```
function text_to_numbers(text)
    alphabet = ['A':'Я';]
    text = uppercase(text)
    return [findfirst(isequal(ch), alphabet) for ch in text]
end
```

```
function numbers_to_text(nums)
    alphabet = ['A':'Я';]
    return join([alphabet[n] for n in nums])
end
```

```
function encrypt_gammirovanie(text, gamma)
    p = text_to_numbers(text)
    g = text_to_numbers(gamma)
    m = 33
    c = [mod((p[i] + g[(i - 1) % length(g) + 1] -
1), m) + 1 for i in 1:length(p)]
    return c
end
```

```
function decrypt_gammirovanie(c, gamma)
    g = text_to_numbers(gamma)
    m = 33
    p = [mod((c[i] - g[(i - 1) % length(g) + 1] -
1), m) + 1 for i in 1:length(c)]
    return numbers_to_text(p)
end

maintext = "ПРИКАЗ"
gamma = "ГАММА"
println("Текст: ", maintext)
println("Гамма: ", gamma)
cipher_nums = encrypt_gammirovanie(plaintext, gamma)
println("Шифртекст (числа): ", cipher_nums)
decrypted_text = decrypt_gammirovanie(cipher_nums, gamma)
println("Расшифрованный текст: ", decrypted_text)
```

Проверим результат работы кода: См. рис. 1

```
»[30]: function text_to_numbers(text)
        alphabet = ['A','H'];
        text = uppercase(text)
        return [findFirst(isequal(ch), alphabet) for ch in text]
    end

    function numbers_to_text(nums)
        alphabet = ['A','H'];
        return join([alphabet[n] for n in nums])
    end

    function encrypt_gammirovaniye(text, gamma)
        p = text_to_numbers(text)
        g = text_to_numbers(gamma)
        m = 33
        c = [mod((p[i] + g[(i - 1) % length(g) + 1] - 1), m) + 1 for i in 1:length(p)]
        return c
    end

    function decrypt_gammirovaniye(c, gamma)
        g = text_to_numbers(gamma)
        m = 33
        p = [mod((c[i] - g[(i - 1) % length(g) + 1] - 1), m) + 1 for i in 1:length(c)]
        return numbers_to_text(p)
    end

    maintext = "ПРИКАЗ"
    gamma = "ГАММА"
    println("Текст: ", maintext)
    println("Гамма: ", gamma)
    cipher_nums = encrypt_gammirovaniye(plaintext, gamma)
    println("Шифртекст (числа): ", cipher_nums)
    decrypted_text = decrypt_gammirovaniye(cipher_nums, gamma)
    println("Расшифрованный текст: ", decrypted_text)

    Текст: ПРИКАЗ
    Гамма: ГАММА
    Шифртекст (числа): [20, 18, 22, 24, 2, 12]
    Расшифрованный текст: ПРИКАЗ
```

Рис. 1: Шифрование гаммированием

Принцип работы программы 1. Функция `text_to_numbers` преобразует буквы русского алфавита в числа от 1 до 33. 2. Функция `numbers_to_text` выполняет обратное преобразование - восстанавливает текст из чисел. 3. Функция `encrypt_gammirovanie` складывает значения текста и гаммы по модулю 33. Операция гарантирует, что намерация остается в пределах [1,33]. 4. Функция `decrypt_gammirovanie` производит обратное вычитание по модулю 33. Здесь также используется -1 перед `mod`, чтобы компенсировать смещение, связанное с нумерацией от 1. 5. После выполнения программы выводятся числовой шифртекст и восстановленный исходный текст.

Программа корректно реализует метод гаммирования на русском алфавите. При шифровании каждая буква заменяется суммой её позиции и позиции буквы гаммы по модулю. При расшифровке производится обратное вычитание, что полностью восстанавливает исходный текст. Результаты совпадают с приведенным примером из методического пособия.

[1] Julia: <https://ru.wikipedia.org/wiki/Julia>

[2] Шифрование гаммированием:

<https://ru.wikipedia.org/wiki/%D0%93%D0%B0%D0%BC%D0%BC%D0%B8%D1%80%D0%BE%D0>