

Лабораторная работа №3

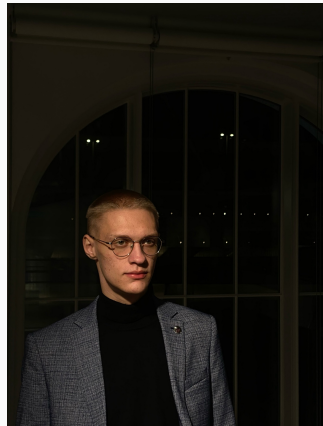
Математические основы защиты информации и информационной безопасности

Махорин И. С.

2025

Российский университет дружбы народов имени Патриса Лумумбы, Москва, Россия

- Махорин Иван Сергеевич
- Студент группы НФИмд-02-25
- Студ. билет 1032259380
- Российский университет дружбы народов имени Патриса Лумумбы



- Изучить шифрование гаммированием и научиться его реализовывать.

Выполнение лабораторной работы

Реализация шифрования гаммированием

```
# Шифрование гаммированием конечной гаммой

# Создаем массив символов русского алфавита для корректной работы
const RUSSIAN_ALPHABET = collect("абвгдезийклмнопрстуфхцщъыъэя")

# Функция для преобразования буквы в число (a=1, b=2, ..., я=33)
function char_to_num(c::Char)::Int
    idx = findfirst(isequal(c), RUSSIAN_ALPHABET)
    return idx == nothing ? 0 : idx
end

# Функция для преобразования числа в букву (1=a, 2=b, ..., 33=я)
function num_to_char(n::Int)::Char
    return RUSSIAN_ALPHABET[n]
end

# Функция шифрования гаммированием
function gamma_encrypt(text::String, gamma::String)::String
    # Приводим текст и гамму к нижнему регистру и удаляем пробелы
    text = replace(lowercase(text), " " => "")
    gamma = replace(lowercase(gamma), " " => "")

    # Преобразуем текст и гамму в числовые последовательности
    text_chars = collect(text)
    gamma_chars = collect(gamma)

    text_nums = [char_to_num(c) for c in text_chars]
    gamma_nums = [char_to_num(c) for c in gamma_chars]
```

Рис. 1: Реализация шифрования гаммированием

Реализация шифрования гаммированием

```
# Повторяем гамму до длины текста
repeated_gamma = Int[]
for i in 1:length(text_nums)
    push!(repeated_gamma, gamma_nums[(i-1) % length(gamma_nums) + 1])
end

# Шифруем с помощью сложения по модулю 33
cipher_nums = Int[]
for i in 1:length(text_nums)
    s = text_nums[i] + repeated_gamma[i]
    c = s % 33
    if c == 0
        c = 33
    end
    push!(cipher_nums, c)
end

# Преобразуем числа обратно в буквы
cipher_chars = Char[]
for n in cipher_nums
    push!(cipher_chars, num_to_char(n))
end

return String(cipher_chars)
end
```

Рис. 2: Реализация шифрования гаммированием

Реализация шифрования гаммированием

```
# Функция дешифрования гаммированием
function gamma_decrypt(cipher::String, gamma::String)::String
    # Приводим шифртекст и гамму к нижнему регистру
    cipher = lowercase(cipher)
    gamma = replace(lowercase(gamma), " " => "")

    # Преобразуем шифртекст и гамму в числовые последовательности
    cipher_chars = collect(cipher)
    gamma_chars = collect(gamma)

    cipher_nums = [char_to_num(c) for c in cipher_chars]
    gamma_nums = [char_to_num(c) for c in gamma_chars]

    # Повторяем гамму до длины шифртекста
    repeated_gamma = Int[]
    for i in 1:length(cipher_nums)
        push!(repeated_gamma, gamma_nums[(i-1) % length(gamma_nums) + 1])
    end

    # Дешифруем с помощью вычитания по модулю 33
    text_nums = Int[]
    for i in 1:length(cipher_nums)
        p = cipher_nums[i] - repeated_gamma[i]
        # Корректируем отрицательные значения и 0
        if p <= 0
            p += 33
        end
        push!(text_nums, p)
    end
end
```

Рис. 3: Реализация шифрования гаммированием

Реализация шифрования гаммированием

```
# Преобразуем числа обратно в буквы
text_chars = Char[]
for n in text_nums
    push!(text_chars, num_to_char(n))
end

return String(text_chars)
end

# Пример из лабораторной работы
text = "приказ"
gamma = "гамма"
println("Исходный текст: ", text)
println("Гамма: ", gamma)

encrypted = gamma_encrypt(text, gamma)
println("Зашифрованный текст: ", encrypted)

decrypted = gamma_decrypt(encrypted, gamma)
println("Расшифрованный текст: ", decrypted)
```

Рис. 4: Реализация шифрования гаммированием

Исходный текст: приказ

Гамма: гамма

WARNING: redefinition of constant Main.RUSSIAN_ALPHABET. This may fail, cause incorrect answers, or produce other errors.

Зашифрованный текст: усхчбл

Расшифрованный текст: приказ

Рис. 5: Проверка

Вывод

- В ходе выполнения лабораторной работы был изучен метод шифрования гаммированием, а также написан его алгоритм на языке Julia.

Список литературы. Библиография

[1] Julia: <https://docs.julialang.org/en/v1/>