

Лабораторная работа 3

Шифрование гаммированием конечной гаммы

Климин Никита Денисович

Содержание

1. Цель работы	3
2. Задание	4
3. Теоретическое введение	5
4. Выполнение лабораторной работы	6
5. Выводы	8
Список литературы	9

1. Цель работы

Реализовать алгоритм шифрования гаммированием конечной гаммы на языке julia. Проверить работу программы на примере расшифровки и зашифровки текста.

2. Задание

Реализовать алгоритм шифрования гаммированием конечной гаммы на языке julia. Проверить работу программы на примере расшифровки и зашифровки текста.

3. Теоретическое введение

Гаммирование --- это метод симметричного шифрования текста, при котором каждая буква исходного текста преобразуется с помощью ключа (гаммы) по модулю длины алфавита.

- Простейший вариант --- **одноразовое использование ключа**, где длина ключа равна длине текста.
- Шифрование выполняется операцией сложения по модулю 2 (XOR) для двоичных данных или по модулю N для букв (N --- длина алфавита).
- Дешифрование осуществляется повторным применением той же операции.

Формулы для шифрования и расшифровки букв русского алфавита ($N = 33$): $C_i = (P_i + K_i) \bmod 33$ # шифрование $P_i = (C_i - K_i) \bmod 33$ # расшифровка

- P_i --- порядковый номер буквы исходного текста
- K_i --- порядковый номер буквы ключа
- C_i --- порядковый номер зашифрованной буквы

4. Выполнение лабораторной работы

Программа была написана на Julia.

```
alphabet = ['A':'Я'...] #создаём массив русского алфавита
num(ch) = findfirst(==(ch), alphabet) # функция перевода букв в номер
lettr(n) = alphabet[n] # функция перевода номера обратно в букву

function gamma(text::String, key::String; decrypt=false) # функция
    t = uppercase(text) # переводим текст в верхний регистр
    k = replace(uppercase(key), " " => "") # переводим гамму в верхний
    tn = [ch for ch in t] # преобразуем текст в массив символов
    kn = [num.(collect(k))...] # преобразуем ключ в массив номеров букв
    kn = repeat(kn, ceil{Int}(length(tn)/length(kn)))[1:length(tn)]
    op = decrypt ? (-) : (+) # определяем операцию шифрования или рас
    r = [ # основной цикл шифрования и расшифрования
        ch == ' ' ? ' ' : lettr(mod1(op(num(ch), y), length(alphabet)
        for (ch, y) in zip(tn, kn) # иначе берём номер буквы и применяем
    ]
    return join(r)
end

println("введите текст")
text = readline()
println("введите гамму")
key = readline()
println("выберите действие, 1- шифрование, 2 - дешифрование")
```

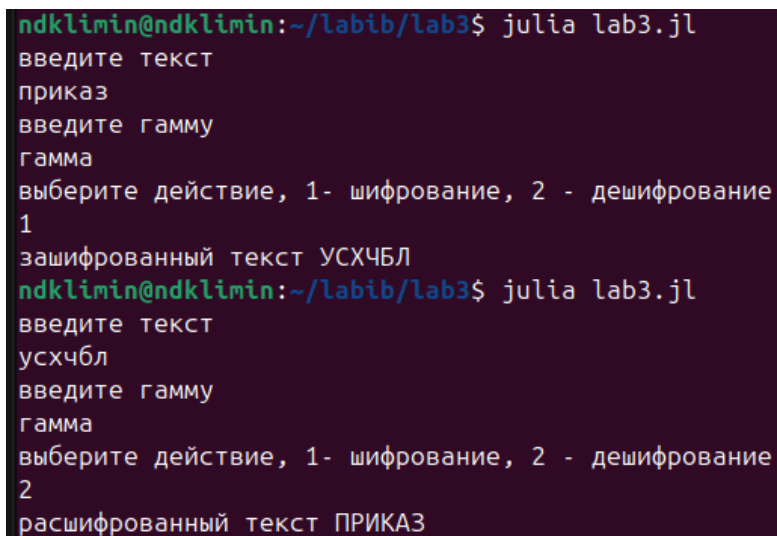
```

choice = readline()

if choice == "1"
    result = gamma(text, key)
    println("зашифрованный текст $result")
elseif choice == "2"
    result = gamma(text, key, decrypt=true)
    println("расшифрованный текст $result")
else
    println("введите 1 или 2")
end

```

Пример работы программы в терминале



```

ndklimin@ndklimin:~/labib/lab3$ julia lab3.jl
введите текст
приказ
введите гамму
гамма
выберите действие, 1- шифрование, 2 - дешифрование
1
зашифрованный текст УСХЧБЛ
ndklimin@ndklimin:~/labib/lab3$ julia lab3.jl
введите текст
усхчбл
введите гамму
гамма
выберите действие, 1- шифрование, 2 - дешифрование
2
расшифрованный текст ПРИКАЗ

```

Рис. 4.1.: Пример работы программы

5. Выводы

Реализован алгоритм шифрования гаммированием конечной гаммой. Программы корректно шифрует и расшифровывает текст на примере слова ПРИКАЗ

Список литературы