

Шифры перестановки

Лабораторная работа №2

Данилова А.С.

Изучить и реализовать шифры перестановки: маршрутное шифрование, шифрование с помощью решеток, таблица Виженера

Маршрутное шифрование

1. Открытый текст последовательно разбивается на части (блоки) с длиной, равной произведению m и n .
2. Блок вписывается построчно в таблицу размерности $m \times n$. Криптограмма получается выписыванием букв из таблицы в соответствии с некоторым маршрутом. Этот маршрут вместе с числами m и n составляет ключ шифра.

			д
	о		г
		о	

			д
	в		
о	о		г
	р	о	п

	о		д
а	в	п	
о	о		г
и	р	о	п

с	о	а	д
д	в	п	л
о	о	и	г
и	р	о	п
<i>ш</i>	<i>и</i>	<i>ф</i>	<i>р</i>

Рис. 1: Шифрование с помощью решеток

Шифрование с помощью *таблицы Виженера* основано на том, что каждая буква в исходном шифруемом тексте сдвигается по алфавиту не на фиксированное, а на переменное количество символов. Величина сдвига каждой буквы задаётся ключом.

Для шифрования используется так называемый «квадрат Виженера» — таблица, где в каждой строке алфавит сдвигается на одну позицию вправо.

```
1 function route_sh(message, key)
2   message = replace(uppercase(message), " " => "")
3   rows = ceil(Int, length(message)/length(key))
4   cols = length(key)
5
6   matrix = Matrix{Char}(undef, rows, cols)
7   index = 1
8   for j in 1:cols
9     for i in 1:rows
10      if index <= length(message)
11        matrix[i, j] = message[index]
12        index += 1
13      else
14        matrix[i, j] = " "
15      end
16    end
17  end
18  indkey = sortperm(collect(key))
19  newtext = ""
20  for j in indkey
21    for i in 1:rows
22      newtext *= string(matrix[i, j])
23    end
24  end
25
26  return newtext
27 end
28
29 print("Введите текст: ")
30 text = readline()
31 print("Ключ: ")
32 key = readline()
```

Рис. 2: Маршрутное шифрование

```
○ Activating project at `C:\Users\nastd\.julia\environments\v1.8`  
Введите текст: hello world  
Ключ: posts  
Зашифрованный текст: LLHEOWLDOR  
█
```

Рис. 3: Зашифрованный текст

```
function grid_sh(message::AbstractString, key::AbstractString)
    message = uppercase(message)
    key = uppercase(key)
    table_size = ceil{Int, sqrt(length(message))}
    message = message * " " ^ ((table_size * table_size) - length(message))

    key_inds = Dict{Char => Int} for (i, char) in enumerate(key)
    sorted_key = sort(collect(key))

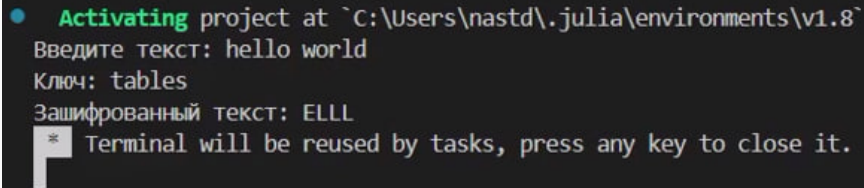
    table = reshape(collect(message), table_size, table_size)

    new_message = ""
    for char in sorted_key
        for j in 1:table_size
            for i in 1:table_size
                if table[i, j] == char
                    new_message *= char
                end
            end
        end
    end

    return new_message
end

print("Введите исходное сообщение: ")
text = readline()
print("Введите ключ: ")
key = readline()
new_message = grid_sh(text, key)
println("зашифрованное: $new_message")
```

Рис. 4: Шифрование с помощью решеток

A screenshot of a terminal window with a dark background and light-colored text. The text shows the activation of a project, input of 'hello world' and 'tables' as a key, and the resulting encrypted text 'ELLL'. A cursor is visible on the line 'Terminal will be reused by tasks, press any key to close it.'

```
• Activating project at `C:\Users\nastd\.julia\environments\v1.8`  
Введите текст: hello world  
Ключ: tables  
Зашифрованный текст: ELLL  
⌵ Terminal will be reused by tasks, press any key to close it.
```

Рис. 5: Зашифрованный текст

```
function vigenere(text::AbstractString, key::AbstractString)
    text = replace(uppercase(text), " " => "")
    key = uppercase(key)
    key_length = length(key)
    new_text = Char[]

    for (i, letter) in enumerate(text)
        shift = Int(key[mod1(i, key_length)]) - Int('A') + 1
        n_letter = shift_cipher(letter, shift)
        push!(new_text, n_letter)
    end
    return join(new_text)
end

function shift_cipher(letter::Char, shift::Int)
    if 'A' <= letter <= 'Z'
        encr = Char(((Int(letter) - Int('A') + shift) % 26) + Int('A'))
        return encr
    else
        return letter
    end
end

print("Введите текст: ")
text = readline()
print("Ключ: ")
key = readline()
newtext = vigenere(text, key)
println("Зашифрованный текст: $newtext")
```

Рис. 6: Шифрование с таблицей Виженера

```
• Activating project at `C:\Users\nastd\.julia\environments\v1.8`  
Введите текст: hello world  
Ключ: vignere  
Зашифрованный текст: DNSZTOTNUK  
* Terminal will be reused by tasks, press any key to close it.
```

Рис. 7: Зашифрованный текст

Мы изучили 3 шифра перестановки и реализовали их на языке программирования Julia.