Шифры перестановки

Тагиев Байрам Алтай оглы

Содержание

1 Цель работы

Целью данной работы является изучение алгоритмов шифрования перестановки, принцип его работы, реализация на Julia.

2 Выполнение лабораторной работы

2.1 Маршрутное шифрование

Реализация:

```
function route_encrypt(message, key, rows, cols)
        message = filter(!isspace, message)
        matrix = fill('_', rows, cols)
        index = 1
        new_message = ""
        for i = 1:rows
                for j = 1:cols
                        if index != rows * cols
                                matrix[i, j] = message[index]
                                index += 1
                        end
                end
        end
        for j in sort(collect(key))
                for i = 1:rows
                        new_message *= (matrix[i, (findfirst(j, key))])
                end
        end
        return new_message
```

```
end


message = "this is a test message!"
rows, cols = 4, 5
key = "water"
println(route_encrypt(message, key, rows, cols))
```

Выполнение:

```
$ julia route.jl
hamgses!iss_iteetsta
```

2.2 Шифрование с помощью решеток

Реализация:

```
function rails_encrypt(text, key, k)
        grid = fill(" ", 2 * k, 2 * k)
        matrix = fill(" ", k, k)
        index = 1
        new_message = ""
        text = replace(text, " " => "")
        for i in 1:k
                for j in 1:k
                        grid[i, j] = string(index)
                        matrix[i, j] = string(index)
                        index += 1
                end
        end
        for i = 1:(size(grid)[1])
                for j = (size(grid)[1]):-1:1
```

```
                if grid[i, j] == " "
                        matrix = rotr90(matrix)
                        grid[(i+k-1):-1:i, j:-1:(j-k+1)] = matrix[k:-1:1, k:-1
                end
        end
end

index = 1
arr = Vector{String}()

for r in text
        checker = false
        for i = 1:(size(grid)[1])
                for j = 1:(size(grid)[2])
                        if grid[i, j] == string(index) && checker == false
                                if ((string(i + 1, " ", j)  arr) && (string(i
                                        grid[i, j] = string(r)
                                        push!(arr, string(i, " ", j))
                                        checker = true
                                end
                        end
                end
                if checker == true
                        index += 1
                        if index > k^2
                                index = 1
                                empty!(arr)
                        end
                        break
```

```
                end
            end
        end

        for j in sort(collect(key))
            for i = 1:2k
                new_message *= (grid[i, (findfirst(j, key))])
                if tryparse(Float64, string(last(new_message))) != nothing
                    new_message = replace(new_message, last(new_message) =
                end
            end
        end
        return new_message

end

text = "Hello, New World!"
key = "keys"
k = 2
println(rails_encrypt(text, key, k))
```

Выполнение:

```
$ julia ./rails.jl
,lr!HNdwoeolle W
```

2.3 Таблица Вижинера

Реализация:

```
function vigenere_encrypt(text, key)
        alphabet = 'a':'z'
        output = ""
        key_index = 1

        for i in text
                if isletter(i)
                        offset = findfirst(isequal(key[key_index]), alphabet) - 1
                        index = findfirst(isequal(i), alphabet) + offset
                        index > 26 && (index -= 26)
                        output *= alphabet[index]
                        key_index += 1
                        key_index > length(key) && (key_index = 1)
                else
                        output *= i
                end
        end

        return output
end

text = "hello world"
key = "key"
println(vigenere_encrypt(text, key))
```

Выполнение:

```
$ julia vigener.jl
rijvs uyvjn
```

3 Выводы

В данной лабораторной работе были изучены три шифра перестановки, все алгоритмы были реализованы на языке Julia и работают корректно.