Лабораторная работа №2

Дисциплина: Математические основы защиты информации и информационной безопасности

Дарижапов Тимур Андреевич

Содержание

List of Tables

List of Figures

1 Цель работы

Изучить маршрутное шифрование, шифрование с помощью решёток и таблицу Виженера, научиться реализации данных шифров программным путём.

2 Задание

- Реализовать маршрутное шифрование
- Реализовать шифрование с помощью решёток
- Реализовать таблицу Виженера

3 Выполнение лабораторной работы

1)Напишем код для маршрутного шифрования с помощью языка Julia. На вход функции идёт текст, слово-шифр и длина стороны прямоугольника.

```
julia> function route_encryption(text, row_length, password)
           text = replace(text, " " => "")

           while length(text) % row_length != 0
               text *= 'a'
           end

           num_rows = div(length(text), row_length)
           table = [collect(text)[i * row_length + 1:(i + 1) * row_length] for i in 0:num_rows-1]
           password_indices = sortperm(collect(password))  # Преобразуем строку в массив символов

           encrypted_text = ""
           for col in password_indices
               for row in 1:num_rows
                   encrypted_text *= table[row][col]
               end
           end

           return encrypted_text
       end
route_encryption (generic function with 1 method)
```

Figure 3.1: Код функции для маршрутного шифрования

Далее мы вводим начальные данные. В нашем случае текст - нельзя недооценивать противника. Слово-шифр - пароль. Длина стороны - 5. Получаем вывод.

```
julia> text = "нельзя недооценивать противника"
"нельзя недооценивать противника"

julia> password = "пароль"
"пароль"

julia> row_length = 6
6

julia> println(route_encryption(text, row_length, password))
еенпнзоатаьовокннеьвлдирияцтиа
```

Figure 3.2: Начальные данные и результат

2)Для шифрования с помощью решёток нам понадобятся 2 функции. Первая функция будет разворачивать матрицу на 90 градусов по часовой стрелке.

```
julia> function rotate90(mat::Matrix{T}) where T
           n, m = size(mat)
           rotated = Matrix{T}(undef, m, n)  # Создаем новый пустой массив с перевернутыми размерами
           for i in 1:n
               for j in 1:m
                   rotated[m - j + 1, i] = mat[i, j]  # Заполняем новую матрицу
               end
           end
           return rotated
       end
rotate90 (generic function with 1 method)
```

Figure 3.3: Функция разворота матрицы

Вторая функция будет заполнять нашу большую матрицу символами из текста.

```
julia> function lattices(text::String, word::String)
           result = ""
           text = replace(text, " " => "")
           matrix = fill(' ', 4, 4)  # Создаем двумерный массив (Matrix{Char})

           # Правильная индексация для работы с многобайтовыми символами
           index = 1
           matrix[1, 4] = text[index]
           index = nextind(text, index)

           matrix[3, 2] = text[index]
           index = nextind(text, index)

           matrix[3, 4] = text[index]
           index = nextind(text, index)

           matrix[4, 3] = text[index]
           index = nextind(text, index)

           matrix = rotate90(matrix)

           matrix[3, 2] = text[index]
           index = nextind(text, index)

           matrix[4, 3] = text[index]
           index = nextind(text, index)

           matrix[3, 4] = text[index]
           index = nextind(text, index)

           matrix[1, 4] = text[index]
           index = nextind(text, index)

           matrix = rotate90(matrix)

           matrix[4, 3] = text[index]
           index = nextind(text, index)
```

Figure 3.4: Функция заполнения матрицы 1

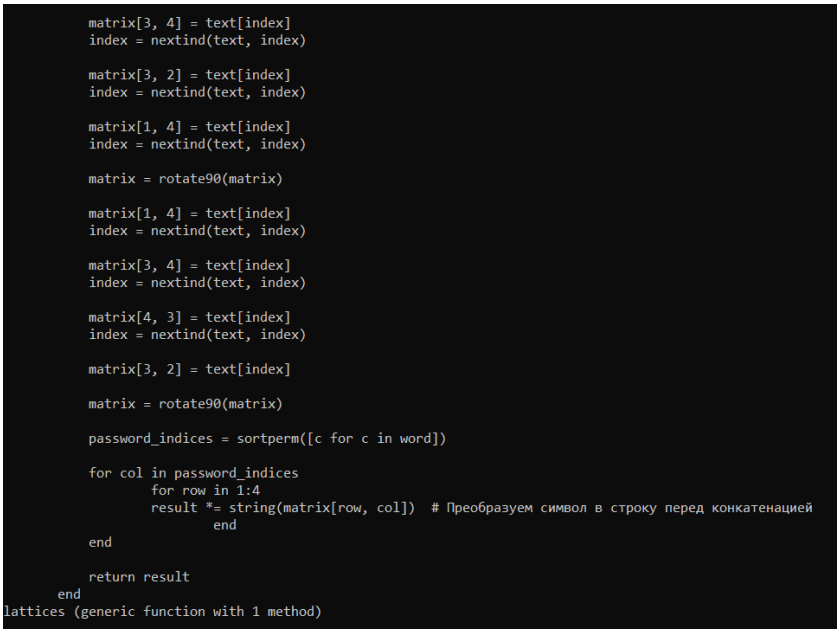

Figure 3.5: Функция заполнения матрицы 2

Вводим начальные данные. Текст - договор подписали. Слово-шифр - шифр. Получаем результат.

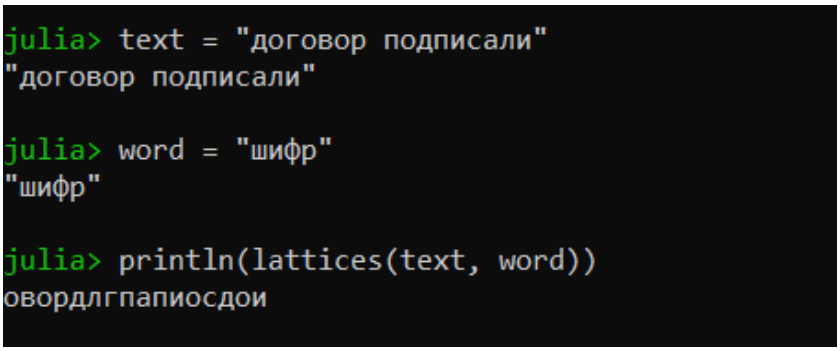

Figure 3.6: Начальные данные и результат

3)Таблицу Виженера на языке Julia сделать не получилось. Представляю код на языке Python.

```python
def vigenere_table(text, keyword):
    alphabet = 'АБВГДЕЖЗИЙКЛМНОПРСТУФХЦЧШЩЬЫЭЮЯ'
    keyword = keyword.upper()
    text = text.upper().replace(' ', '')
    result = ''

    keyword_repeated = ''
    while len(keyword_repeated) < len(text):
        keyword_repeated += keyword
    keyword_repeated = keyword_repeated[:len(text)]

    for i in range(len(text)):
        if text[i] in alphabet:
            p_index = alphabet.index(text[i])
            k_index = alphabet.index(keyword_repeated[i])
            c_index = (p_index + k_index) % len(alphabet)
            result += alphabet[c_index]
        else:
            text += text[i]
    return result

text = 'криптография серьезная наука'
keyword = 'математика'
print(vigenere_table(text, keyword))
```

Figure 3.7: Функция шифрования таблицей Виженера

Вводим начальные данные. Текст - криптография серьезная наука. Слово-шифр - математика. Получает результат.

```
C:\Users\tidaa\PycharmProjects\InfoBez\venv\Scripts\python.exe C:/Users/tidaa/Pycharı
ЦРЬФЯОХШКФФЯДКЭЬЧПЧАЛНТШЦА
```

Figure 3.8: Результат вывода

4 Выводы

- Я изучил маршрутное шифрование, шифрование с помощью решёток и таблицу Виженера, научился реализации данных шифров программным путём.