Лабораторная работа №2: Презентация.

Шифры простой замены.

Евдокимов Максим Михайлович. Группа - НФИмд-01-24.[1]

26 сентябрь, 2024, Москва, Россия

[1]Российский Университет Дружбы Народов

Цели и задачи работы

Цель лабораторной работы

Изучить способы шифрования методом перестановки разного типа.

Задание

1. Реализовать метод Маршрутного шифрования.
2. Реализовать метод шифрования с помощью решеток.
3. Реализовать метод шифрования через таблицу Виженера.

Теоретическое введение

Маршрутное шифрование

Маршрутное шифрование (также известное как маршрутное транспонирование) — это метод шифрования, при котором открытый текст записывается в матрицу (таблицу) по определенному маршруту, а затем считывается по другому маршруту для получения шифрованного текста. Основная идея заключается в изменении порядка символов в соответствии с заданным маршрутом.

Алгоритм действий маршрутного шифрования:

1. Выбор матрицы и маршрутов

- Размер матрицы: Определите размер матрицы n.
- Маршрут записи: Определите маршрут, по которому будут записываться символы открытого текста в матрицу.
- Маршрут считывания: Определите маршрут, по которому будут считываться символы из матрицы для получения шифрованного текста.

2. Запись открытого текста в матрицу

- Заполнение матрицы: Запишите символы открытого текста в матрицу по выбранному маршруту записи. Если текст короче, чем размер матрицы, можно добавить фиктивные символы (могут быть технически любым).

3. Считывание шифрованного текста

- Считывание матрицы: Считайте символы из матрицы по выбранному маршруту считывания. Результат будет шифрованным текстом.

4. Дешифрование

- Запись шифрованного текста в матрицу: Запишите символы шифрованного текста в матрицу по маршруту считывания.
- Считывание открытого текста: Считайте символы из матрицы по маршруту записи. Результат будет открытым текстом.

Шифрование с помощью решёток (также известное как “шифр Кардано”) - это метод криптографии, основанный на использовании специальной трафаретной маски, называемой “решёткой”.

Алгоритм действий Шифрование с помощью решеток

1. Создание решёток:

- Решётка для шифрования (Е-решётка):

Это квадратная матрица размером NxN клеток (обычно 4x4, 6x6 или 8x8). В некоторых клетках матрицы проделаны отверстия. Отверстия расположены таким образом, что при повороте решётки на 90 градусов, 180 градусов и 270 градусов, они не совпадают с предыдущими положениями. За один полный оборот решётки (360 градусов) отверстия проходят через все клетки матрицы.

- Решётка для расшифровки (D-решётка):

Это также квадратная матрица размером NxN клеток. Отверстия в D-решётке расположены таким образом, чтобы при наложении на E-решётку в определённом положении, отверстия совпадали с теми клетками, в которые были записаны символы.

2. Запись сообщения:

- Е-решётка накладывается на чистый лист бумаги.
- Сообщение записывается в отверстия Е-решётки.
- Затем Е-решётка поворачивается на 90 градусов и сообщение продолжает записываться в следующие отверстия.
- Процесс повторяется до тех пор, пока Е-решётка не будет повернута на 360 градусов.
- В результате на листе бумаги остаётся зашифрованное сообщение, состоящее из символов, расположенных в случайном порядке.

3. Расшифровка сообщения:

- D-решётка накладывается на зашифрованное сообщение в определённом положении. Символы, расположенные в отверстиях D-решётки, читаются и записываются.
- Затем D-решётка поворачивается на 90 градусов и процесс повторяется.
- Процесс повторяется до тех пор, пока D-решётка не будет повернута на 360 градусов.
- В результате будет прочитано исходное сообщение.

Шифрование через таблицу Виженера

Таблица Виженера (также известная как шифр Виженера) - это метод полиалфавитного шифрования, который использует ключевое слово для сдвига букв исходного текста на разные позиции в алфавите. Это делает шифр более стойким, чем простые моноалфавитные шифры, такие как шифр Цезаря.

Алгоритм действий для метода таблицу Виженера

1. Таблица Виженера:

- Таблица Виженера представляет собой квадратную матрицу, состоящую из 26 строк и 26 столбцов.
- Каждая строка соответствует сдвигу алфавита на определённое количество позиций. Например, первая строка - это обычный алфавит, вторая строка - алфавит со сдвигом на 1 позицию, третья строка - со сдвигом на 2 позиции и так далее.

2. Ключевое слово:

- Ключевое слово - это слово или фраза, которая используется для шифрования сообщения.
- Длина ключевого слова должна быть не меньше длины сообщения.
- Если ключевое слово короче, оно повторяется до тех пор, пока не достигнет нужной длины.

3. Шифрование:

- Для каждой буквы исходного текста находится соответствующая буква ключевого слова.
- В таблице Виженера находится пересечение строки, соответствующей букве исходного текста, и столбца, соответствующего букве ключевого слова.
- Буква на пересечении этих строки и столбца является зашифрованной буквой.

4. Расшифровка:

- Для расшифровки используется то же ключевое слово.
- Для каждой буквы зашифрованного текста находится соответствующая буква ключевого слова.
- В таблице Виженера находится строка, соответствующая букве ключевого слова.
- В этой строке находится буква, соответствующая зашифрованной букве. Эта буква является исходной буквой.

Ход работы

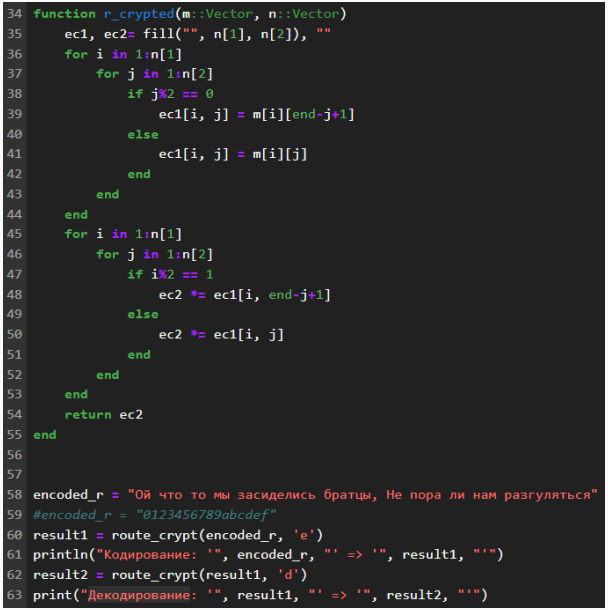
Создание кода для маршрутного шифрования:

```
function route_crypt(text::String, route::Char)
    matrix, n = creating_r(text)
    res = r_crypted(matrix, n)
    if route == 'd'
        res = replace(res, '_' => "")
    end
    return res
end

```

Рис. 1: Основная функция маршрутного

```julia
function creating_r(t::String)
    m, rez = split(t, ""), Vector{Vector{String}}(undef, 0)
    dividers, len, d = [5, 6, 7], length(t), 0
    while d == 0
        if len % dividers[1] == 0
            d = 5
        elseif len % dividers[2] == 0
            d = 6
        elseif len % dividers[3] == 0
            d = 7
        else
            push!(m, "_")
            len += 1
        end
    end
    for i in 1:len
        if i%d == 1
            push!(rez, [])
        end
        push!(rez[end], m[i])
    end
    return rez, [div(len, d), d]
end

```

Рис. 2: Создание матрицы маршрутного



```
function r_crypted(m::Vector, n::Vector)
    ec1, ec2= fill("", n[1], n[2]), ""
    for i in 1:n[1]
        for j in 1:n[2]
            if j%2 == 0
                ec1[i, j] = m[i][end-j+1]
            else
                ec1[i, j] = m[i][j]
            end
        end
    end
    for i in 1:n[1]
        for j in 1:n[2]
            if i%2 == 1
                ec2 *= ec1[i, end-j+1]
            else
                ec2 *= ec1[i, j]
            end
        end
    end
    return ec2
end


encoded_r = "Ой что то мы засиделись братцы, Не пора ли нам разгуляться"
#encoded_r = "0123456789abcdef"
result1 = route_crypt(encoded_r, 'e')
println("Кодирование: '", encoded_r, "' => '", result1, "'")
result2 = route_crypt(result1, 'd')
print("Декодирование: '", result1, "' => '", result2, "'")
```

Рис. 3: Вызов и вывод и функция маршрутного шифрования

Результат 1

```
Кодирование: 'Ой что то мы засиделись братцы, Не пора ли нам разгуляться' => 'тй чОоот  аы змседилбсь ирцтаы  Не,паро аи нлмар зтулягь_яс_'
Декодирование: 'тй чОоот  аы змседилбсь ирцтаы  Не,паро аи нлмар зтулягь_яс_' => 'Ой что то мы засиделись братцы, Не пора ли нам разгуляться'
```

Рис. 4: Результат Маршрутного шифра

Создание кода для шифра с помощью решеток:

```
function grid_crypt(text::String, route::Char, k::Vector)
    matrix, n = creating_g(text)
    cd = deepcopy(k)
    if route == 'd'
        cd = 4 .- cd
        res = g_crypted(matrix, n, cd)
        res = replace(res, '_' => "")
    else
        res = g_crypted(matrix, n, cd)
    end
    return res
end
```

```julia
function creating_g(t::String)
    m, rez = split(t, ""), Vector{Vector{String}}(undef, 0)
    dividers, len, d = [4, 6, 8, 10], length(t), 0
    while d == 0
        if len == dividers[1]^2
            d = dividers[1]
        elseif len == dividers[2]^2
            d = dividers[2]
        elseif len == dividers[3]^2
            d = dividers[3]
        elseif len == dividers[4]^2
            d = dividers[4]
        else
            push!(m, "_")
            len += 1
        end
    end
    for i in 1:len
        if i%d == 1
            push!(rez, [])
        end
        push!(rez[end], m[i])
    end
    return rez, d
end
```

Рис. 6: Создание матрицы решеток

```julia
function g_crypted(m::Vector, n::Int, moves::Vector)
    m_cube = fill("_", n, n)
    for i in 1:2:n
        for j in 1:2:n
            temp = circshift([m[i][j], m[i][j+1], m[i+1][j], m[i+1][j+1]], moves[end])
            pop!(moves)
            m_cube[i, j] = temp[1]
            m_cube[i, j+1] = temp[2]
            m_cube[i+1, j] = temp[3]
            m_cube[i+1, j+1] = temp[4]
        end
    end
    return join(join.(eachrow(m_cube)), "")
end

encoded_g = "Сдать лабу до 28 числа включительно"
#encoded_g = "Make a peace of."
key = [3, 2, 1, 1, 2, 2, 2, 3, 3, 2, 3, 2, 1, 0, 1, 3, 2, 1, 1, 3]
result3 = grid_crypt(encoded_g, 'e', key)
println("Кодирование: '", encoded_g, "' => '", result3, "'")
result4 = grid_crypt(result3, 'd', key)
println("Декодирование: '", result3, "' => '", result4, "'")
```

Рис. 7: Вызов и вывод и функция шифрования с помощью решеток

```
Кодирование: 'Сдать лабу до 28 числа включительно' => 'длуадьаСтб  ис8лв о а2ч клнюо_елчьит'
Декодирование: 'длуадьаСтб  ис8лв о а2ч клнюо_елчьит' => 'Сдать лабу до 28 числа включительно'
```

Рис. 8: Результат шифра Решётки

Создание кода для шифрования через таблицу Виженера:

```
function Vigener_crypt(text::String, route::Char, k)
    sw, temp = split(text, ""), split(k, "")
    nw, nk = length(sw), length(temp)
    sk = [temp[1+(j%nk)] for j in 0:nw]
    res = g_crypted(sw, sk, nw, route)
    return res
end

```

Рис. 9: Основная функция Виженера

```
function g_crypted(w1::Vector, w2::Vector, n::Int, r::Char)
    alfTable = [circshift([i for i in 'а':'я'], j) for j in 0:-1:-31]
    m = fill("_", n)
    for i in 1:n
        if !(first(w1[i]) in 'а':'я')
            m[i] = w1[i]
        elseif r == 'е'
            xn = Int(first(w1[i])) - Int('а') + 1
            yn = Int(first(w2[i])) - Int('а') + 1
            m[i] = string(alfTable[yn][xn])
        else
            yn = Int(first(w2[i])) - Int('а') + 1
            xn = 0
            for j in 1:32
                if first(w1[i]) == alfTable[yn][j]
                    xn = j
                end
            end
            m[i] = string(alfTable[1][xn])
        end
    end
    return join(m)
end
```

Рис. 10: функция шифрования через таблицу Виженера

```
33
34 encoded_v = "кабы не было зимы в городах и селах"
35 key = "сервер"
36 result5 = Vigener_crypt(encoded_v, 'e', key)
37 println("Кодирование: '", encoded_v, "' => '", result5, "'")
38 result6 = Vigener_crypt(result5, 'd', key)
39 println("Декодирование: '", result5, "' => '", result6, "'")
```

Рис. 11: Вызов и вывод шифра

```
Кодирование: 'кабы не было зимы в городах и селах' => 'ыесэ эц сэрю мшоа у урхюхее н вкывъ'
Декодирование: 'ыесэ эц сэрю мшоа у урхюхее н вкывъ' => 'кабы не было зимы в городах и селах'
```

Рис. 12: Результат шифра Виженера

Выводы по проделанной работе

Вывод

В ходе выполнения лабораторной работы я изучил представленные 3 вида шифра перестановки: Маршрутное шифрование, Шифрование с помощью решеток и таблица Вижинера. А также реализовал на языке программирования Julia методы шифрования и дешифрования для каждого шифра.