

Лабораторная работа №2: отчет.

Шифры перестановки

Евдокимов Максим Михайлович. Группа - НФИмд-01-24.

Содержание

Цели и задачи работы	4
Цель лабораторной работы	4
Задание	4
Теоретическое введение	5
Маршрутное шифрование	5
Алгоритм действий маршрутного шифрования:	5
Шифрование с помощью решеток	6
Алгоритм действий Шифрование с помощью решеток	6
Шифрование через таблицу Виженера	7
Алгоритм действий для метода таблицу Виженера	7
Ход работы	9
Задание 1	9
Результат 1	12
Задание 2	12
Результат 2	14
Задание 3	14
Результат 3	17
Выводы по проделанной работе	18
Вывод	18
Список литературы	19

Список иллюстраций

1	Основная функция маршрутного	9
2	Создание матрицы маршрутного	10
3	Вызов и вывод и функция маршрутного шифрования	11
4	Результат Маршрутного шифра	12
5	Основная функция решеток	12
6	Создание матрицы решеток	13
7	Вызов и вывод и функция шифрования с помощью решеток	14
8	Результат шифра Решётки	14
9	Основная функция Виженера	15
10	функция шифрования через таблицу Виженера	16
11	Вызов и вывод шифра	16
12	Результат шифра Виженера	17

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 решеток

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 $N \times N$ клеток (обычно 4×4 , 6×6 или 8×8). 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 $N \times N$ клеток. Отверстия в D-решётке расположены таким образом, чтобы при наложении на Е-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 таблицы 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

Задание 1

Создание кода для маршрутного шифрования:

```
1 function route_crypt(text::String, route::Char)
2     matrix, n = creating_r(text)
3     res = r_crypted(matrix, n)
4     if route == 'd'
5         res = replace(res, '_' => "")
6     end
7     return res
8 end
9
```

Рис. 1: Основная функция маршрутного

```

10 function creating_r(t::String)
11     m, rez = split(t, ""), Vector{Vector{String}}(undef, 0)
12     dividers, len, d = [5, 6, 7], length(t), 0
13     while d == 0
14         if len % dividers[1] == 0
15             d = 5
16         elseif len % dividers[2] == 0
17             d = 6
18         elseif len % dividers[3] == 0
19             d = 7
20         else
21             push!(m, "_")
22             len += 1
23         end
24     end
25     for i in 1:len
26         if i%d == 1
27             push!(rez, [])
28         end
29         push!(rez[end], m[i])
30     end
31     return rez, [div(len, d), d]
32 end
33

```

Рис. 2: Создание матрицы маршрутного

```

34 function r_crypted(m::Vector, n::Vector)
35     ec1, ec2= fill("", n[1], n[2]), ""
36     for i in 1:n[1]
37         for j in 1:n[2]
38             if j%2 == 0
39                 ec1[i, j] = m[i][end-j+1]
40             else
41                 ec1[i, j] = m[i][j]
42             end
43         end
44     end
45     for i in 1:n[1]
46         for j in 1:n[2]
47             if i%2 == 1
48                 ec2 *= ec1[i, end-j+1]
49             else
50                 ec2 *= ec1[i, j]
51             end
52         end
53     end
54     return ec2
55 end
56
57
58 encoded_r = "Ой что то мы засиделись братцы, Не пора ли нам разгуляться"
59 #encoded_r = "0123456789abcdef"
60 result1 = route_crypt(encoded_r, 'e')
61 println("Кодирование: '", encoded_r, "' => '", result1, "'")
62 result2 = route_crypt(result1, 'd')
63 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 ай эмседилбсь ирцтаы Не,паро ай нлмар этулягь_яс_'  
Декодирование: 'тй чооот ай эмседилбсь ирцтаы Не,паро ай нлмар этулягь_яс_' => 'Ой что то мы засиделись братцы, Не пора ли нам разгуляться'
```

Рис. 4: Результат Маршрутного шифра

Задание 2

Создание кода для шифра с помощью решеток:

```
1 function grid_crypt(text::String, route::Char, k::Vector)
2     matrix, n = creating_g(text)
3     cd = deepcopy(k)
4     if route == 'd'
5         cd = 4 .- cd
6         res = g_crypted(matrix, n, cd)
7         res = replace(res, '_' => "")
8     else
9         res = g_crypted(matrix, n, cd)
10    end
11    return res
12 end
13
```

Рис. 5: Основная функция решеток

```

14 function creating_g(t::String)
15     m, rez = split(t, ""), Vector{Vector{String}}(undef, 0)
16     dividers, len, d = [4, 6, 8, 10], length(t), 0
17     while d == 0
18         if len == dividers[1]^2
19             d = dividers[1]
20         elseif len == dividers[2]^2
21             d = dividers[2]
22         elseif len == dividers[3]^2
23             d = dividers[3]
24         elseif len == dividers[4]^2
25             d = dividers[4]
26         else
27             push!(m, "_")
28             len += 1
29         end
30     end
31     for i in 1:len
32         if i%d == 1
33             push!(rez, [])
34         end
35         push!(rez[end], m[i])
36     end
37     return rez, d
38 end
39

```

Рис. 6: Создание матрицы решеток

```

40 function g_crypted(m::Vector, n::Int, moves::Vector)
41     m_cube = fill("_", n, n)
42     for i in 1:2:n
43         for j in 1:2:n
44             temp = circshift([m[i][j], m[i][j+1], m[i+1][j], m[i+1][j+1]], moves[end])
45             pop!(moves)
46             m_cube[i, j] = temp[1]
47             m_cube[i, j+1] = temp[2]
48             m_cube[i+1, j] = temp[3]
49             m_cube[i+1, j+1] = temp[4]
50         end
51     end
52     return join(join.(eachrow(m_cube)), "")
53 end
54
55 encoded_g = "Сдать лабу до 28 числа включительно"
56 #encoded_g = "Make a peace of."
57 key = [3, 2, 1, 1, 2, 2, 2, 3, 3, 2, 3, 2, 1, 0, 1, 3, 2, 1, 1, 3]
58 result3 = grid_crypt(encoded_g, 'e', key)
59 println("Кодирование: '", encoded_g, "' => '", result3, "'")
60 result4 = grid_crypt(result3, 'd', key)
61 println("Декодирование: '", result3, "' => '", result4, "'")

```

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

Результат 2

```

Кодирование: 'Сдать лабу до 28 числа включительно' => 'длуадѡСтб ис8лв о а2ч клню_елчѣит'
Декодирование: 'длуадѡСтб ис8лв о а2ч клню_елчѣит' => 'Сдать лабу до 28 числа включительно'

```

Рис. 8: Результат шифра Решётки

Задание 3

Создание кода для шифрования через таблицу Виженера:

```
1 function Vigenere_encrypt(text::String, route::Char, k)
2     sw, temp = split(text, ""), split(k, "")
3     nw, nk = length(sw), length(temp)
4     sk = [temp[1+(j%nk)] for j in 0:nw]
5     res = g_crypted(sw, sk, nw, route)
6     return res
7 end
8
```

Рис. 9: Основная функция Виженера

```

9 function g_crypted(w1::Vector, w2::Vector, n::Int, r::Char)
10     alfTable = [circshift([i for i in 'a':'я'], j) for j in 0:-1:-31]
11     m = fill("_", n)
12     for i in 1:n
13         if !(first(w1[i]) in 'a':'я')
14             m[i] = w1[i]
15         elseif r == 'e'
16             xn = Int(first(w1[i])) - Int('a') + 1
17             yn = Int(first(w2[i])) - Int('a') + 1
18             m[i] = string(alfTable[yn][xn])
19         else
20             yn = Int(first(w2[i])) - Int('a') + 1
21             xn = 0
22             for j in 1:32
23                 if first(w1[i]) == alfTable[yn][j]
24                     xn = j
25                 end
26             end
27             m[i] = string(alfTable[1][xn])
28         end
29     end
30     return join(m)
31 end
32

```

Рис. 10: функция шифрования через таблицу Виженера

```

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

```

Рис. 11: Вызов и вывод шифра

Результат 3

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 решетки и таблица Вижинера. А также реализовал на языке программирования Julia методы шифрования и дешифрования для каждого шифра.

Список литературы

1. Лабораторная работа №2
2. Strings - Julia documentation
3. Functions - Julia documentation