Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Рязанский государственный радиотехнический

университет имени В.Ф. Уткина»

Кафедра «ЭВМ»

Отчет о лабораторной работе №2

«Шифры перестановки на примере шифра кардано»

по дисциплине

«Защита информации»

Выполнили:

Студенты группы 045

Анохин В.А.

Вашкулатов Н.А.

Проверили:

доц. Крошилина С.В.

доц. Тишкина В.В.

**Цель работы**: разработать алгоритмы шифрования и дешифрования сообщений с применением решетки Кардано.

**Ход работы**

На рисунках 1 – 4 показаны схемы алгоритмов данной программы.

**Код программы:**

public class Cardano {

public static String encrypt(String source, boolean[][] grid) {

//Делаем таблицу с использованием сетки

var table = encryptToTable(source, grid);

//Центральную ячейку пробелом

if (grid.length % 2 != 0) {

int center = grid.length / 2;

table[center][center] = ' ';

}

//прошлись по таблице для вывода результата

return tableToString(table);

}

public static String decrypt(String encrypted, boolean[][] grid) {

var table = stringToTable(encrypted, grid.length);

var sb = new StringBuilder();

for (int i = 0; i < 4; i++) {

//Чтение из таблицы по сетке

sb.append(readFromTable(table, grid));

//поворот сетки

grid = moveRight(grid);

}

return sb.toString();

}

static boolean[][] moveRight(boolean[][] grid) {

boolean[][] result = new boolean[grid.length][grid.length];

for (int i = 0; i < grid.length; i++) {

for (int j = 0; j < grid.length; j++) {

if (grid[i][j]) {

//поворот на 90

result[j][grid.length - i - 1] = true;

}

}

}

return result;

}

private static char[][] encryptToTable(String source, boolean[][] grid) {

char[][] table = new char[grid.length][grid.length];

char[] chars = source.toCharArray();

int writtenCount = 0;

for (int i = 0; i < 4; i++) {

//записали, используя сетку

writtenCount = writeToTable(chars, writtenCount, table, grid);

//повернули сетку

grid = moveRight(grid);

}

return table;

}

private static String tableToString(char[][] table) {

var sb = new StringBuilder();

for (int i = 0; i < table.length; i++) {

for (int j = 0; j < table.length; j++) {

sb.append(table[i][j]);

}

}

return sb.toString();

}

private static int writeToTable(char[] chars, int writtenCount, char[][] table, boolean[][] grid) {

for (int i = 0; i < grid.length; i++) {

for (int j = 0; j < grid.length; j++) {

if (grid[i][j]) {

table[i][j] = chars[writtenCount++];

}

}

}

return writtenCount;

}

private static char[][] stringToTable(String encrypted, int tableSize) {

var chars = encrypted.toCharArray();

char[][] table = new char[tableSize][tableSize];

for (int i = 0; i < tableSize; i++) {

for (int j = 0; j < tableSize; j++) {

//Записываем зашифрованую строку в таблицу

table[i][j] = chars[i \* tableSize + j];

}

}

return table;

}

private static StringBuilder readFromTable(char[][] table, boolean[][] grid) {

var sb = new StringBuilder();

for (int i = 0; i < grid.length; i++) {

for (int j = 0; j < grid.length; j++) {

if (grid[i][j]) {

sb.append(table[i][j]);

}

}

}

return sb;

}

}

**Вывод**: в ходе выполнения работы были разработаны алгоритмы шифрования и дешифрования сообщений с применением решетки Кардано.