МИНИСТЕРСТВО ЦИФРОВОГО РАЗВИТИЯ, СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ

Ордена Трудового Красного Знамени федеральное государственное

бюджетное образовательное учреждение

высшего образования

«Московский технический университет связи и информатики»

Кафедра «Информационная безопасность»

По дисциплине: «Криптографические методы и средства обеспечения информационной безопасности инфокоммуникаций»

Лабораторная работа на тему:

«Простые симметричные шифры».

Выполнили ст. группы БПЗ1801

Купцова Д.О.

Зелик М.Б.

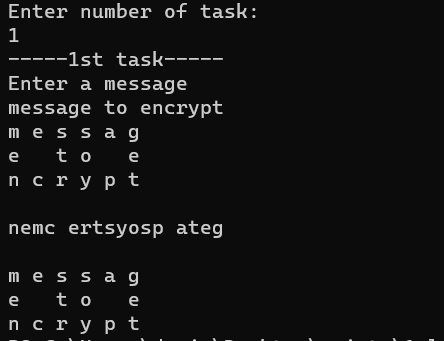
Москва

2021

**Цель работы:** Ознакомление с простыми симметричными криптографическими шифрами.

**Ход работы**

**Задание 1:** Придумайте свой перестановочный шифр, реализуйте его на любом языке программирования, зашифруйте с помощью вашего шифра произвольное сообщение, после чего дешифруйте его. Опишите достоинства и недостатки шифра

****

import java.util.Random;

import java.util.Scanner;

public class Lab1{

    public static void main(String[] args){

        Scanner vvod = new Scanner(System.in);

        System.out.print("Enter number of task: \n");

        int n = vvod.nextInt();

        if (n == 1)

            zadanie1();

        else if (n == 2)

            zadanie2();

        else if (n == 3)

            zadanie3();

        else if (n == 4)

            zadanie4();

        else if (n == 5)

            zadanie5();

        else if (n == 6)

            zadanie6();

        else if (n == 7)

            zadanie7();

        else

        {

            System.out.print("Invalid number\n");

        }

        vvod.close();

    }

    public static void zadanie1()

    {

        System.out.println("-----1st task-----");

        int x;

        int y;

        int col = 6;

        int i = 0;

        Scanner vvod = new Scanner(System.in);

        System.out.println("Enter a message");

        String str = vvod.nextLine();

        int len = str.length();

        int line;

        if (len % col > 0)

            line = len / col + 1;

        else

            line = len / col;

        char[] word = new char[len];

        char[][] array = new char[col][line];

        while (i < len)

        {

            word[i] = str.charAt(i);

            i++;

        }

        i = 0;

        y = 0;

        while (y < line)

        {

            x = 0;

            while (x < col)

            {

                if (i < len)

                {

                    array[x][y] = word[i];

                    i++;

                }

                else

                    array[x][y] = ' ';

                x++;

            }

            y++;

        }

        i = 0;

        y = 0;

        while (y < line)

        {

            x = 0;

            while (x < col)

            {

                System.out.printf("%c ", array[x][y]);

                x++;

            }

            System.out.printf("\n");

            y++;

        }

        System.out.printf("\n");

        char [] new\_arr = new char[col \* line];

        x = 0;

        int x1 = 0;

        while (x < col)

        {

            y = line - 1;

            while (y >= 0)

            {

                new\_arr[x1] = array[x][y];

                x1++;

                y--;

            }

            x++;

        }

        x1 = 0;

        while (x1 < line \* col)

        {

            System.out.printf("%c", new\_arr[x1]);

            x1++;

        }

        System.out.printf("\n");

        x1 = 0;

        x = 0;

        while (x < col)

        {

            y = line - 1;

            while (y >= 0)

            {

                array[x][y] = new\_arr[x1];

                y--;

                x1++;

            }

            x++;

        }

        System.out.printf("\n");

        y = 0;

        while (y < line)

        {

            x = 0;

            while (x < col)

            {

                System.out.printf("%c ", array[x][y]);

                x++;

            }

            System.out.printf("\n");

            y++;

        }

        vvod.close();

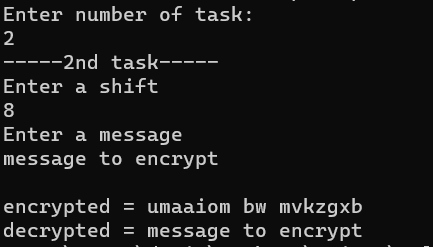
    }

Достоинства: так как при шифровании меняется не открытый текст, а порядок символов, то при помощи таблиц частоты использования букв в языках, данный шифр не взломать.

Недостатки: нужно чтобы все ячейки таблицы были заполнены. При необходимости строка текста дополняется нужным количеством пробелов.

Задание 2.

Придумайте свой простой подстановочный шифр (моноалфавитный шифр), реализуйте его на любом языке программирования зашифруйте с помощью вашего шифра произвольное сообщение, после чего дешифруйте его.

****

public static void zadanie2()

    {

        Scanner vvod\_str = new Scanner(System.in);

        Scanner vvod\_int = new Scanner(System.in);

        System.out.println("-----2nd task-----");

        System.out.println("Enter a shift");

        int n = vvod\_int.nextInt();

        System.out.println("Enter a message");

        String str = vvod\_str.nextLine();

        System.out.println();

        str = encrypt\_2(str, n);

        System.out.println("encrypted = " + str);

        str = encrypt\_2(str, -n);

        System.out.println("decrypted = " + str);

        vvod\_str.close();

        vvod\_int.close();

    }

public static String encrypt\_2(String str, int n)

    {

        int i = 0;

        int len = str.length();

        char[] word = new char[len];

        while (i < len)

        {

            word[i] = str.charAt(i);

            i++;

        }

        char[] encrypt\_w = new char[len];

        i = 0;

        while (i < len)

        {

            if (word[i] > 64 && word[i] < 91)

            {

                if (word[i] + n > 90)

                    encrypt\_w[i] = (char)(word[i] + n - 26);

                else

                {

                    if (word[i] + n < 65)

                        encrypt\_w[i] = (char)(word[i] + n + 26);

                    else

                        encrypt\_w[i] = (char)(word[i] + n);

                }

            }

            else if (word[i] > 96 && word[i] < 123)

            {

                if (word[i] + n > 122)

                    encrypt\_w[i] = (char)(word[i] + n - 26);

                else

                {

                    if(word[i] + n < 97)

                        encrypt\_w[i] = (char)(word[i] + n + 26);

                    else

                        encrypt\_w[i] = (char)(word[i] + n);

                }

            }

            else

                encrypt\_w[i] = word[i];

            i++;

        }

        str = String.valueOf(encrypt\_w);

        return (str);

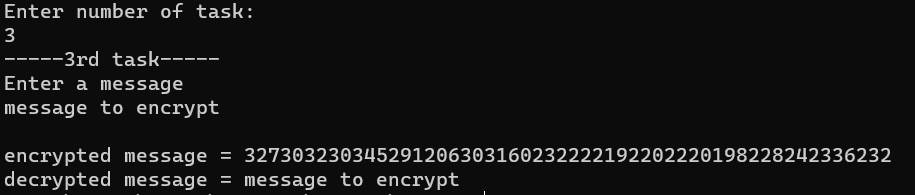
    }

Достоинства: метод использует случайный ключ для каждой отдельной буквы для шифрования, который делает моноалфавитный шифр защищенным от атаки грубой силы.

Недостатки: сложно запомнить порядок букв в ключе, и поэтому требуется много времени и усилий для шифрования или расшифровки текста вручную. Моноалфавитная замена уязвима для частотного анализа.

Задание 3.

Придумайте свой однозвучный подстановочный шифр, реализуйте его на любом языке программирования, зашифруйте с помощью вашего шифра произвольное сообщение, после чего дешифруйте его. Опишите достоинства и недостатки шифра.

****

public static void zadanie3()

    {

        Scanner vvod = new Scanner(System.in);

        System.out.println("-----3rd task-----");

        System.out.println("Enter a message");

        String str = vvod.nextLine();

        System.out.println();

        str = encrypt\_3(str);

        System.out.println("encrypted message = " + str);

        str = decrypt\_3(str);

        System.out.println("decrypted message = " + str);

    }

public static String encrypt\_3(String str)

    {

        String res = "";

        Random ran = new Random();

        int i;

        i = 0;

        while (i < str.length())

        {

            if (str.charAt(i) == 32)

                res += (int)str.charAt(i) \* (ran.nextInt(7 - 5) + 5);

            else

                res += (int)str.charAt(i) \* (ran.nextInt(4 - 2) + 2);

            i++;

        }

        return (res);

    }

    public static String decrypt\_3(String str)

    {

        String res = "";

        int cod = 0;

        int i = 0;

        while (i < str.length())

        {

            int k = Integer.parseInt((str.substring(i, i + 3)));

            if (k / 2 >= 97 && k / 2 <= 122)

                res += (char)(k / 2);

            else if (k / 3 >= 97 && k / 3 <= 122)

                res += (char)(k / 3);

            else if (k / 4 >= 97 && k / 4 <= 122 || k / 4 == 32)

                res += (char)(k / 4);

            else if (k / 5 == 32)

                res += (char)(k / 5);

            else if (k / 6 == 32)

                res += (char)(k / 6);

            i += 3;

        }

        return (res);

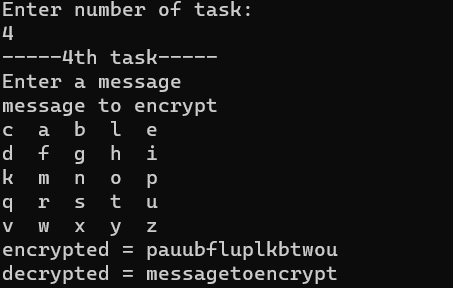
    }

Достоинства: так как при шифровании для каждого символа используется по три различных ключа, то при помощи таблиц частоты использования букв в языках, вероятность взлома данного шифра снижается.

Недостатки: вероятность взлома данного шифра возрастает при увеличении длины текста.

Задание 4.

Придумайте полиграммный подстановочный шифр, реализуйте его на любом языке программирования, зашифруйте с помощью вашего шифра произвольное сообщение, после чего дешифруйте его. Опишите достоинства и недостатки шифра.

****

public static void zadanie4(){

        Scanner vvod = new Scanner(System.in);

        System.out.println("-----4th task-----");

        System.out.println("Enter a message");

        String str = vvod.nextLine();

        String key = "cable";

        int len = str.length();

        char[][] array = new char[5][5];

        int i = 0;

        while (i < key.length())

        {

            array[0][i] = key.charAt(i);

            i++;

        }

        i = 1;

        char ch = 'a';

        while (i  < 5)

        {

            int j = 0;

            while (j < 5)

            {

                if (ch == 'j')

                {

                    ch++;

                    j--;

                }

                else if (check(key, ch) == 0)

                {

                    array[i][j] = ch;

                    ch++;

                }

                else if (check(key, ch) == 1)

                {

                    j--;

                ch++;

                }

                j++;

            }

            i++;

        }

        i = 0;

        while (i < 5)

        {

           int j = 0;

            while (j < 5)

            {

                System.out.printf("%c  ", array[i][j]);

                j++;

            }

            System.out.println();

            i++;

        }

        String encr = encrypt\_4(str, array);

        System.out.println("encrypted = " + encr);

        String dec = decrypt\_4(encr, array, str);

        System.out.println("decrypted = " + dec);

        vvod.close();

    }

public static int j\_found(char ch, char[][] array)

        {

            int i = 0;

            while (i < 5)

            {

                int j = 0;

                while (j < 5)

                {

                    if (array[i][j] == ch)

                        return (j);

                    j++;

                }

                i++;

            }

            return (0);

        }

    public static int i\_found(char ch, char[][] array)

    {

        int i = 0;

        while (i < 5)

        {

            int j = 0;

            while (j < 5)

            {

                if (array[i][j] == ch)

                    return (i);

                j++;

            }

            i++;

        }

        return (0);

    }

    public static String encrypt\_4(String str, char[][] array)

    {

        char[]word = new char[str.length()];

        int i = 0;

        int j = 0;

        while (i < str.length())

        {

            if (str.charAt(i) == ' ')

                i++;

            else{

            word[j] = str.charAt(i);

            j++;

            i++;}

        }

        int len = j;

        i = 0;

        str = String.valueOf(word);

        int k;

        int k1, j1;

        while (i < len)

        {

            k = i\_found(word[i], array);

            j = j\_found(word[i], array);

            k1 = i\_found(word[i+1], array);

            j1 = j\_found(word[i+1], array);

            if (j1 == j)

            {

                word[i] = array[k][j + 2];

                word[i+1] = array[k1][j1+2];

                i = i+2;

            }

            else

            {

                word[i] = array[k][j1];

                word[i + 1] = array[k1][j];

                i = i + 2;

            }

        }

            String res = String.valueOf(word);

            return(res);

    }

    public static String decrypt\_4(String str, char[][] array, String val)

    {

        char[]word = new char[str.length()];

        int i = 0;

        while (i < str.length())

        {

            word[i] = str.charAt(i);

            i++;

        }

        int len = i;

        i = 0;

        int k, j;

        int k1, j1;

        while (i < len - 2)

        {

            k = i\_found(word[i], array);

            j = j\_found(word[i], array);

            k1 = i\_found(word[i+1], array);

            j1 = j\_found(word[i+1], array);

            if ((k == 0 && j == 0) || (k1 == 0 && j1 == 0))

                i++;

            if (j1 == j)

            {

                word[i] = array[k][j - 2];

                word[i+1] = array[k1][j1 - 2];

                i = i + 2;

            }

            else

            {

                word[i] = array[k][j1];

                word[i + 1] = array[k1][j];

                i = i + 2;

            }

            }

            String res = String.valueOf(word);

            return (res);

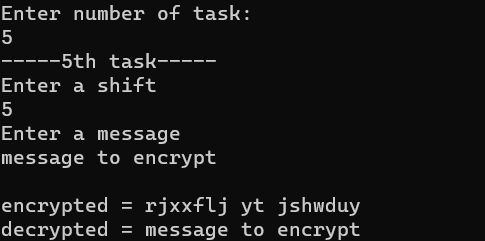
    }

**Достоинства:** Так как символы шифруются парами, из-за этого возникает множество различных ключей. Первое преимущество такого способа заключается в том, что распределение частот групп букв значительно более равномерное, чем отдельных символов. Во-вторых, для продуктивного частотного анализа требуется больший размер зашифрованного текста, так число различных групп букв значительно больше, чем просто алфавит.

**Недостатки:** Имеет только открытый ключ.

Задание 5.

Придумайте полиалфавитный подстановочный шифр, реализуйте его на любом языке программирования, зашифруйте с помощью вашего шифра произвольное сообщение, после чего дешифруйте его. Опишите достоинства и недостатки шифра.

****

    public static void zadanie5()

    {

        Scanner vvod\_str = new Scanner(System.in);

        Scanner vvod\_int = new Scanner(System.in);

        System.out.println("-----5th task-----");

        System.out.println("Enter a shift");

        int n = vvod\_int.nextInt();

        System.out.println("Enter a message");

        String str = vvod\_str.nextLine();

        System.out.printf("\n");

        str = encrypt\_2(str, n);

        System.out.println("encrypted = " + str);

        str = encrypt\_2(str, -n);

        System.out.println("decrypted = " + str);

        vvod\_str.close();

        vvod\_int.close();

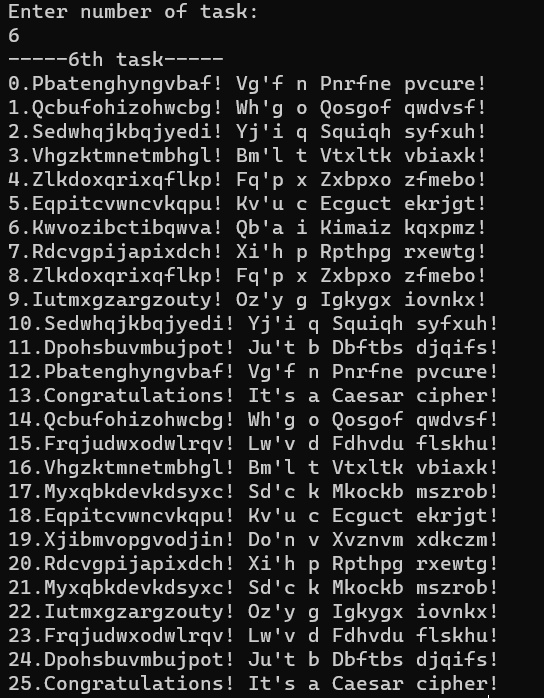
    }

Достоинства: так как используется два различных вида шифрования, то в связи с этим вероятность взлома полученного шифра стремится к нулю.

Недостатки: имеет только открытый ключ.

Задание 6.

Известно, что сообщение " Pbatenghyngvbaf! Vg'f n Pnrfne pvcure!" было зашифровано шифром Цезаря, но ключ k неизвестен. Напишите вспомогательную программу, которая поможет вам в расшифровке этого сообщения.

****

public static void zadanie6()

    {

        System.out.println("-----6th task-----");

        String str = "Pbatenghyngvbaf! Vg'f n Pnrfne pvcure!";

        for (int i = 0; i < 26; i++)

        {

            str = encrypt\_2(str, i);

            System.out.println(i + "." + str);

        }

    }

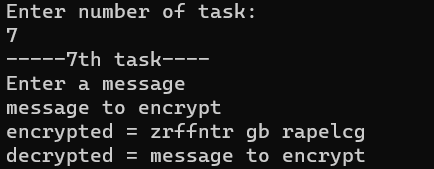
Недостатки: подстановки, выполняемые в соответствии с системой Цезаря, не маскируют частот появления различных букв исходного открытого текста;

число возможных ключей К мало; шифр Цезаря легко вскрывается на основе анализа частот появления букв в шифр тексте.

Достоинства: простота шифрования и расшифровывания.

Задание 7.

Реализуйте собственный аналог функции ROT13 его на любом языке программирования, зашифруйте с помощью вашего шифра произвольное сообщение, после чего дешифруйте его. Опишите достоинства и недостатки шифра.

****

public static void zadanie7(){

        Scanner vvod = new Scanner(System.in);

        System.out.println("-----7th task----");

        System.out.println("Enter a message");

        String str = vvod.nextLine();

        str = encrypt\_2(str, 13);

        System.out.println("encrypted = " + str);

        str = encrypt\_2(str, -13);

        System.out.println("decrypted = " + str);

        vvod.close();

    }

Достоинства: прост в реализации.

Недостатки: при помощи таблиц частоты использования букв в языках, вероятность взлома данного шифра значительно возрастает.