Лабораторная работа №5

Тема «Исследование симметричных криптоалгоритмов»

ВЫПОЛНИЛ

Олишкевич Игорь Русланович

Группа 25-ПО

Задание: изучить симметричный шифр; обосновать его название; привести примеры, код, дать описание.

Симметричные криптографические алгоритмы шифрования – способ шифрования, в котором для шифрования и расшифрования применяется один и тот же криптографический ключ:

*Ek(P) = C* и *Dk(C) = P,* где *Е* – функция зашифрования, *k* – ключ, *P* – открытый текст, *С* – шифртекст,    
*D* – функция расшифрования.

При этом справедливо следующее равенство:

*Dk(Ek(P)) = P.*

Критерием качественного шифрования служит следующий принцип: стойкость шифра должна определяться только секретностью ключа (правило Кирхгоффса -Dutchman A. Kerckhoffs).

Из вышесказанного можно сделать вывод, что название происходит от использования одного и того же ключа, используемого как в шифровании, так и в дешифровании.

Классическим примером таких алгоритмов являются симметричные криптографические алгоритмы, перечисленные ниже:

* простая перестановка;
* одиночная перестановка по ключу;
* двойная перестановка.

Простая перестановка без ключа является одним из самых простых методов шифрования. Сообщение записывается в таблицу по столбцам. После того, как открытый текст записан колонками, для образования шифровки он считывается по строкам. Для использования шифра обе стороны должны договориться о ключе и виде размера таблицы.

Пример:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Windows.Forms;

namespace ZILAB1

{

    public partial class Form1 : Form

    {

        int state;

        int[] key,unkey;

        public Form1()

        {

            InitializeComponent();

            state = 1;

            int[] k = {2,5,4,3,1};

            key = k;

        }

        private void button2\_Click(object sender, EventArgs e)

        {

            Close();

        }

        int[] getunkey(int[] k)

        {

            int[] res = new int[k.Length];

            for (int i = 0; i < k.Length; i++)

                res[k[i] – 1] = i + 1;

            return res;

        }

        string crypt(string msg,bool f)

        {

            string tmp1, tmp2, res;

            int[] k ;

            if(f) k=key;

            else k = getunkey(key);

            res = "";

            for (int i = 0; i < (int)(1+(msg.Length/k.Length)); i++)

            {

                if ((i + 1) \* k.Length > msg.Length) return res += msg.Substring(i \* k.Length, msg.Length – (i \* k.Length));

                tmp1 = msg.Substring(i \* k.Length, k.Length);

                  tmp2 = "";

                for (int n = 0; n < k.Length; n++)

                     tmp2 += tmp1.Substring(k[n] – 1, 1);

                res += tmp2;

            }

            return res;

        }

        private void button1\_Click(object sender, EventArgs e)

        {

            switch (state)

            {

                case 1://Зашифровать

                    {

                        groupBox1.Text = "Криптограмма";

                        button1.Text = "Расшифровать";

                        textBox1.Text = crypt(textBox1.Text,true);

                        state = 2;

                        break;

                    }

                case 2://Расшифровать

                    {

                        groupBox1.Text = "Исходный текст";

                        button1.Text = "Зашифровать";

                        textBox1.Text = crypt(textBox1.Text,false);

                        state = 1;

                        break;

                    }

            }

        }

    }

}

Одиночная перестановка по ключу являет собой простую перестановку, отличием которой является то, что колонки таблицы переставляются по ключевому слову, фразе или набору чисел длиной в строку таблицы.

Под двойной перестановкой понимается шифрование уже зашифрованного сообщения. Для этого размер второй таблицы подбирают так, чтобы длины её строк и столбцов были другие, чем в первой таблице. Наилучший вариант – взаимно простыми. Также в первой таблице можно переставлять столбцы, во второй – строки, заполнять зигзагом, змейкой или другими способами.

Схема симметричных криптоалгоритмов представлена на рисунке 1.

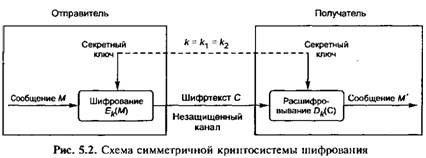


Рисунок 1 – Схема симметричного криптоалгоритма