МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное автономное бюджетное образовательное учреждение высшего образования

«**Севастопольский государственный университет**»

кафедра Информационных систем

Институт информационных технологий и управления в технических системах

Лисянский Александр Игоревич

курс 4 группа ИС/б-42-о

09.03.02 Информационные системы (уровень бакалавриата)

**ОТЧЁТ**

о лабораторном практикуме №3

СЕТИ ФЕЙШТЕЛЯ

по дисциплине «Основы защиты информации»

Отметка о зачёте \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_

(дата)

Руководитель практикума

\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(должность) (подпись) (инициалы, фамилия)

Севастополь

2016

**Цель работы**: Изучить принципы работы сети Фейштеля, научиться шифровать информацию посредством использования блочного криптоалгоритма.

**Вариант задания**:

|  |  |  |
| --- | --- | --- |
| Номер  вар. | Количество  раундов | Образующая  Функция |
| 13 | 12 | Сложение |

**Сети Фейштеля**

Сеть Фейштеля получила широкое распространение, поскольку обеспечивает выполнение требования о многократном использовании ключа и материала исходного блока информации. Классическая сеть Фейштеля имеет следующую структуру:

Независимые потоки информации, порожденные из исходного блока, называются ветвями сети. В классической схеме их две. Величины Vi именуются параметрами сети, обычно это функции от материала ключа. Функция F называется образующей. Действие, состоящее из однократного вычисления образующей функции и последующего наложения ее результата на другую ветвь с обменом их местами, называется циклом или раундом (англ. round) сети Фейштеля. Оптимальное число раундов K – от 8 до 32. Часто количество раундов не фиксируется разработчиками алгоритма, а лишь указываются разумные пределы (обязательно нижний, и не всегда – верхний) этого параметра.

**Ход работы**: В соответствии с вариантом задания был разработан алгоритм шифрования сообщений пользователя.

Проверяем работу программы. Введем самое простое выражение для двух видов кодировки

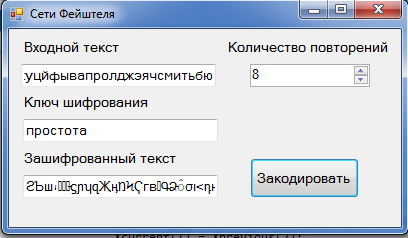


Рисунок 1 – кодирование сообщения

В результате проверки было получено заключение, что программа работает правильно, кодировка удается.

Код программы:

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace lab3

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

private int ToBite(string s)

{

byte[] Key = Encoding.UTF8.GetBytes(s);

int res = Key[0] + Key[1] \* 256 + Key[2] \* 256 \* 256 + Key[3] \* 256 \* 256 \* 256;

return res;

}

private string sum(string s1, string s2)

{

string res = "";

for (int i = 0; i < s1.Length;i++ )

res += (char)((int)s1[i] + (int)s2[i]);

return res;

}

private string ByteToString(int i, string s)

{

string res = "";

byte[] Key = Encoding.UTF8.GetBytes(s);

res += (char)((i % 256) ^ Key[0]);

res += (char)(((i >> 8) % 256) ^ Key[1]);

res += (char)(((i >> 16) % 256) ^ Key[2]);

res += (char)(((i >> 24) % 256) ^ Key[3]);

return res;

}

private UInt32 RotateLeft(UInt32 x, Byte n)

{

return (UInt32)((x << n) | (x >> (32 - n)));

}

private UInt32 RotateRight(UInt32 x, Byte n)

{

return (UInt32)((x >> n) | (x << (32 - n)));

}

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

{

if (inTextBox.TextLength % 16 != 0)

{

MessageBox.Show("Строка должна быть кратной 16 символам");

return;

}

if (keyTextBox.TextLength % 8 != 0)

{

MessageBox.Show("Ключ должен быть кратен 8 символам");

return;

}

string key = keyTextBox.Text;

string lkey = key.Substring(0, 4);

string rkey = key.Substring(4, 4);

string[] Xcurrent = new string[4];

string[] Xprevious = new string[4];

int left\_temp = ToBite(lkey);

int right\_temp = ToBite(rkey);

for (int j = 0; j < inTextBox.TextLength; j += 16)

{

string[] temp = new string[4];

temp[0] = inTextBox.Text.Substring(j, 4);

temp[1] = inTextBox.Text.Substring(j + 4, 4);

temp[2] = inTextBox.Text.Substring(j + 8, 4);

temp[3] = inTextBox.Text.Substring(j + 12, 4);

for (byte i = 0; i < loopNumeric.Value; i++)

{

if (i == 0)

{

int temp\_mod\_key = (int)RotateLeft((UInt32)left\_temp, i) ^ (int)RotateRight((UInt32)right\_temp, i);

Xcurrent[0] = sum(temp[1], ByteToString(temp\_mod\_key, temp[0]));

Xcurrent[1] = temp[2];

Xcurrent[2] = temp[3];

Xcurrent[3] = temp[0];

Xprevious[0] = Xcurrent[0];

Xprevious[1] = Xcurrent[1];

Xprevious[2] = Xcurrent[2];

Xprevious[3] = Xcurrent[3];

}

else

{

int temp\_mod\_key = (int)RotateLeft((UInt32)left\_temp, i) ^ (int)RotateRight((UInt32)right\_temp, i);

Xcurrent[0] = sum(Xprevious[1], ByteToString(temp\_mod\_key, Xprevious[0]));

Xcurrent[1] = Xprevious[2];

Xcurrent[2] = Xprevious[3];

Xcurrent[3] = Xprevious[0];

Xprevious[0] = Xcurrent[0];

Xprevious[1] = Xcurrent[1];

Xprevious[2] = Xcurrent[2];

Xprevious[3] = Xcurrent[3];

}

}

}

string res = Xcurrent[0] + Xcurrent[1] + Xcurrent[2] + Xcurrent[3];

outTextBox.Text = res;

}

}

}

Вывод: В результате выполнения лабораторной работы были получены навыки по разработке алгоритмов кодирования методами сетей Фейштеля, на основе полученного алгоритма разработаны функции кодирования, а так же написана и отлажена программа с полученными функциями. Результаты тестирования программы показали, что алгоритм разработан верно, программа работает правильно.