КІТ-м120Б

Богомаз Олексій

Лабораторна робота №6

Тема: Електронно-цифровий підпис.

Мета роботи: Створення програми для формування та перевірки повідомлень за допомогою електронно-цифрового підпису.

Завдання:

1. Необхідно розробити і налагодити дві програми:

Програма формування підпису повідомлення.

* У якості повідомлення використовувати копію файлу з розробленою програмою.
* Програма перевірки повідомлення.

Алгоритм «RSA»

Виконання роботи.

Для виконання роботи використаємо на наш розсуд найбільш простий варіант побудови віконної програми - C#.

Код програми Шифрування C# :

Program.cs

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace lab7

{

static class Program

{

/// <summary>

/// Главная точка входа для приложения.

/// </summary>

[STAThread]

static void Main()

{

//try

//{

Application.EnableVisualStyles();

Application.SetCompatibleTextRenderingDefault(false);

Application.Run(new Form1());

}

}

}

Form1.cs

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.Numerics;

using System.Windows.Forms;

namespace lab7

{

public partial class Form1 : Form

{

private Form2 form2;

public Form1()

{

InitializeComponent();

form2 = new Form2();

}

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

{

pgBar.Value = 0;

pgBar.Visible = true;

pgBar.Maximum = textBox4.Text.Length;

string[] split = textBox7.Text.Split('-');

string[] splitE = split[0].Split(':');

string[] splitN = split[1].Split(':');

string ms = textBox4.Text;

BigInteger number;

textBox5.Text = "";

do

{

for (int i = 0; i < 8; i++)

{

if (!ms.Equals(""))

{

int e1 = Convert.ToInt32(splitE[i], 16);

int n = Convert.ToInt32(splitN[i], 16);

char buf;

{

buf = ms[0];

ms = ms.Remove(0, 1);

}

number = BigInteger.Pow(buf, e1);

number = number % n;

int c = (int)number;

buf = (char)c;

textBox5.Text += buf.ToString();

pgBar.Value++;

}

}

} while (!ms.Equals(""));

pgBar.Visible = false;

}

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

{

progressBar1.Value = 0;

progressBar1.Visible = true;

progressBar1.Maximum = textBox3.Text.Length;

string[] split = textBox2.Text.Split('-');

string[] splitD = split[0].Split(':');

string[] splitN = split[1].Split(':');

string ms = textBox3.Text;

BigInteger number;

textBox8.Text = "";

do

{

for (int i = 0; i < 8; i++)

{

if (!ms.Equals(""))

{

int d = Convert.ToInt32(splitD[i], 16);

int n = Convert.ToInt32(splitN[i], 16);

char buf;

{

buf = ms[0];

ms = ms.Remove(0, 1);

}

number = BigInteger.Pow(buf, d);

number = number % n;

int c = (int)number;

buf = (char)c;

textBox8.Text += buf.ToString();

progressBar1.Value++;

}

}

} while (!ms.Equals(""));

progressBar1.Visible = false;

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

private void textBox2\_TextChanged(object sender, EventArgs e)

{

}

private void textBox3\_TextChanged(object sender, EventArgs e)

{

}

private void textBox5\_TextChanged(object sender, EventArgs e)

{

}

private void textBox4\_TextChanged(object sender, EventArgs e)

{

}

private void textBox6\_TextChanged(object sender, EventArgs e)

{

}

private void textBox7\_TextChanged(object sender, EventArgs e)

{

}

private void textBox8\_TextChanged(object sender, EventArgs e)

{

}

private void label8\_Click(object sender, EventArgs e)

{

}

private void label9\_Click(object sender, EventArgs e)

{

}

private void button3\_Click(object sender, EventArgs e)

{

form2.ShowDialog();

}

private void textBox5\_TextChanged\_1(object sender, EventArgs e)

{

}

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

{

}

private void progressBar1\_Click(object sender, EventArgs e)

{

}

}

}

Form2.cs

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 lab7

{

public partial class Form2 : Form

{

byte[] easyC = { 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79, 83, 89, 97, 101, 103, 107, 109, 113, 127, 131, 137, 139, 149, 151, 157, 163, 167, 173, 179, 181, 191, 193, 197, 199, 211, 223, 227, 229, 233, 239, 241, 251 }; // Каждый раз считать алгоритмом не имеет смысла, а так Решето Эратосфена поможет

public Form2()

{

InitializeComponent();

}

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

{

Random rd = new Random();

int[] p = new int[8];

int[] q = new int[8];

int[] n = new int[8];

int[] f = new int[8];

int[] d = new int[8];

int[] em = new int[8];

int[] k = new int[8];

bool prime = true;

int cou;

textBox2.Text = "";

textBox1.Text = "";

for (int i = 0; i < 8; i++)

{

p[i] = easyC[rd.Next(0, 54)];

q[i] = easyC[rd.Next(0, 54)];

n[i] = p[i] \* q[i];

f[i] = (p[i] - 1) \* (q[i] - 1);

do

{

prime = true;

k[i] = rd.Next(2, 6);

cou = k[i] \* f[i] + 1;

for (int j = 2; j <= Math.Sqrt(cou); j++)

{

if (cou % j == 0)

{

prime = false;

break;

}

}

em[i] = rd.Next(3, 230);

} while (prime);

do

{

em[i]--;

if (em[i] < 2)

{

em[i] = rd.Next(2, 255);

}

}

while (((k[i] \* f[i] + 1) % em[i]) != 0);

d[i] = (k[i] \* f[i] + 1) / em[i];

}

for (int i = 0; i < 8; i++)

{

textBox2.Text += em[i].ToString("X3");

textBox2.Text += ":";

textBox1.Text += d[i].ToString("X3");

textBox1.Text += ":";

}

textBox2.Text = textBox2.Text.Remove(textBox2.Text.Length - 1, 1);

textBox1.Text = textBox1.Text.Remove(textBox1.Text.Length - 1, 1);

textBox2.Text += "-";

textBox1.Text += "-";

for (int i = 0; i < 8; i++)

{

textBox2.Text += n[i].ToString("X4");

textBox2.Text += ":";

textBox1.Text += n[i].ToString("X4");

textBox1.Text += ":";

}

textBox2.Text = textBox2.Text.Remove(textBox2.Text.Length - 1, 1);

textBox1.Text = textBox1.Text.Remove(textBox1.Text.Length - 1, 1);

}

private void textBox2\_TextChanged(object sender, EventArgs e)

{

}

private void textBox1\_TextChanged(object sender, EventArgs e)

{

}

}

}

Результат:

Генерація ключів для підпису(рис. 1):

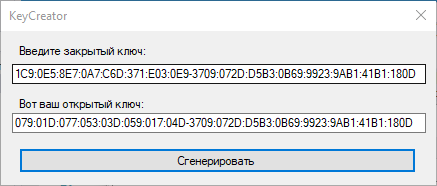


Рисунок 1

Шифруємо(рис. 2):

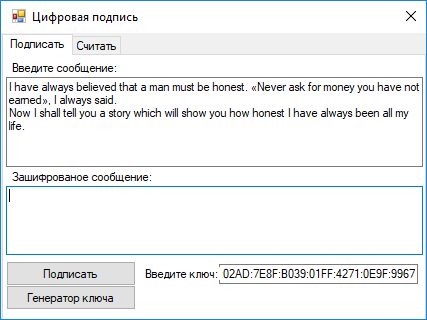


Рисунок 2

Після успішного шифрування отримуємо(рис. 3):

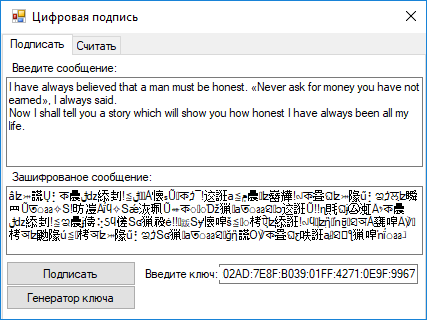


Рисунок 3

Вікно дешифрування(рис. 4):

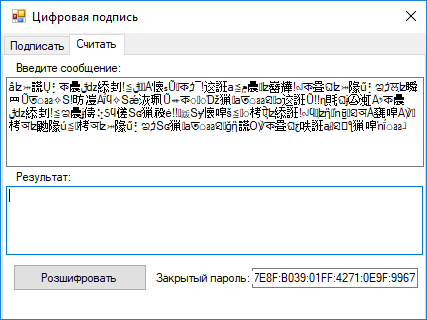


Рисунок 4

Після успішної дешифрації отримуємо(рис. 5):

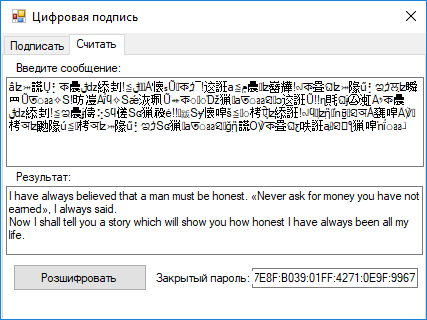


Рисунок 5

Дешифрована програма повністю функціональна й готова до повторного циклу.

**Висновок:** у ході лабораторної роботи розроблено програму для формування та перевірки повідомлень за допомогою електронно-цифрового підпису з використанням мови C#.