Федеральное агентство связи

Ордена Трудового Красного Знамени федеральное государственное бюджетное образовательное учреждение высшего образования

«Московский технический университет связи и информатики»

Кафедра «Информационная безопасность»

Отчёт по лабораторной работе №5

по дисциплине

«Методы и средства защиты компьютерной информации»

Применение средств криптографической защиты

Выполнил:

студент группы БВТ1301

Редько Е.Ю.

Проверил:

ассистент кафедры ИБ

Барков В.В.

Москва 2016

**Цель работы**

Используя разработанные в предыдущих работах библиотеки, создать приложение, организующее работу симметричного и ассиметричного алгоритмов шифрования, хэширования и электронной цифровой подписи. Также предусмотреть пользовательский интерфейс.

**Код программы**

namespace RedkoApp

{

public partial class Form1 : Form

{

RedkoLib.DES des;

public Form1()

{

InitializeComponent();

}

private void Form1\_Load(object sender, EventArgs e)

{

}

public static byte[] FromHex(string hex)

{

hex = hex.Replace(" ", "");

byte[] raw = new byte[hex.Length / 2];

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

{

raw[i] = Convert.ToByte(hex.Substring(i \* 2, 2), 16);

}

return raw;

}

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

{

if (textBox1.Text.Length > 0 && textBox2.Text.Length > 0)

{

string text = textBox1.Text;

string key = textBox2.Text;

byte[] textByteArray = Encoding.ASCII.GetBytes(text);

MemoryStream textStream = new MemoryStream(textByteArray);

byte[] keyByteArray = Encoding.ASCII.GetBytes(key);

MemoryStream keyStream = new MemoryStream(keyByteArray);

Stream output = new MemoryStream();

des = new RedkoLib.DES(keyStream);

des.Encrypt(textStream, output);

StreamReader reader = new StreamReader(output);

String encrypted = reader.ReadToEnd();

if (encrypted != null)

{

textBox1.Text = encrypted;

}

}

}

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

{

if (textBox1.Text.Length > 0 && des != null)

{

string text = textBox1.Text;

byte[] textByteArray = Encoding.ASCII.GetBytes(text);

MemoryStream textStream = new MemoryStream(textByteArray);

Stream output = new MemoryStream();

des.Decrypt(textStream, output);

StreamReader reader = new StreamReader(output);

String decrypted = reader.ReadToEnd();

if (decrypted != null)

{

textBox1.Text = decrypted;

}

}

}

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

{

RedkoLib.cypherRSAclass bufferRSA = new RedkoLib.cypherRSAclass();

RedkoLib.RSAKey publicKey = new RedkoLib.RSAKey(17, 249);

Stream encryptedDataStream = new MemoryStream();

string sourceDataString = textBox3.Text;

MemoryStream sourceDataStream = new MemoryStream(Encoding.Default.GetBytes(sourceDataString));

bufferRSA.dataToRsa(sourceDataStream, encryptedDataStream, publicKey);

encryptedDataStream.Position = 0;

StreamReader reader = new StreamReader(encryptedDataStream);

string encryptedText = reader.ReadToEnd();

byte[] byteArray = Encoding.Default.GetBytes(encryptedText);

var hexString = BitConverter.ToString(byteArray);

hexString = hexString.Replace("-", " ");

textBox4.Text = hexString;

}

private void button4\_Click(object sender, EventArgs e)

{

RedkoLib.cypherRSAclass bufferRSA = new RedkoLib.cypherRSAclass();

RedkoLib.RSAKey privateKey = new RedkoLib.RSAKey(29, 249);

Stream decryptedDataStream = new MemoryStream();

byte[] data = FromHex(textBox3.Text);

string sourceDataString = Encoding.Default.GetString(data);

MemoryStream sourceDataStream = new MemoryStream(Encoding.Default.GetBytes(sourceDataString));

bufferRSA.rsaToData(sourceDataStream, decryptedDataStream, privateKey);

decryptedDataStream.Position = 0;

StreamReader reader = new StreamReader(decryptedDataStream);

textBox4.Text = reader.ReadToEnd();

}

private void button5\_Click(object sender, EventArgs e)

{

string sourceDataString = textBox5.Text;

RedkoLib.MD5 md5 = new RedkoLib.MD5();

MemoryStream input = new MemoryStream(Encoding.Default.GetBytes(sourceDataString));

MemoryStream output = new MemoryStream();

md5.GetHash(input, output);

string MD5String = Encoding.Default.GetString(output.ToArray()).TrimEnd('\0');

byte[] byteArray = Encoding.Default.GetBytes(MD5String);

var hexString = BitConverter.ToString(byteArray);

hexString = hexString.Replace("-", " ");

textBox6.Text = hexString;

}

private void button6\_Click(object sender, EventArgs e)

{

string sourceDataString = textBox7.Text;

MemoryStream input = new MemoryStream(Encoding.Default.GetBytes(sourceDataString));

MemoryStream output = new MemoryStream();

RedkoLib.SignatureRSA rsa = new RedkoLib.SignatureRSA();

rsa.RSA\_Params();

rsa.SetHashFunction(new RedkoLib.MD5());

rsa.Sign(input, output);

if (rsa.Verify(output))

{

textBox9.Text = "Correct";

}

else

{

textBox9.Text = "Incorrect";

}

string SighString = Encoding.Default.GetString(output.ToArray()).TrimEnd('\0');

byte[] byteArray = Encoding.Default.GetBytes(SighString);

var hexString = BitConverter.ToString(byteArray);

hexString = hexString.Replace("-", " ");

textBox8.Text = hexString;

}

}

}

 