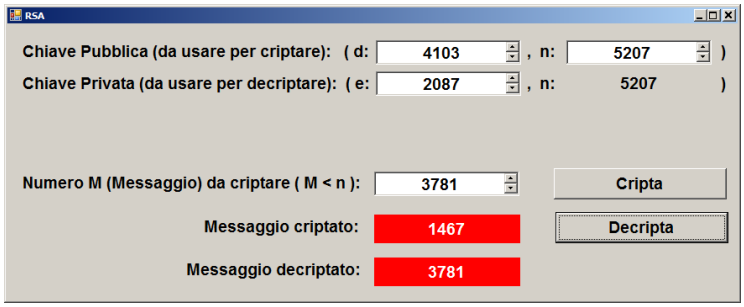
Esempio Pilota:

CodificaDecodificaRSAconBigInteger

Definizione della Form



nudD As NumericUpDown
nudN As NumericUpDown
nudE As NumericUpDown
nudM As NumericUpDown

plsCripta As Button
plsDecripta As Button

lblMcriptato As Label
lblMdecriptato As Label

Codice della Form

```
using System;
using System.Windows.Forms;

// Per poter utilizzare la classe BigInteger è necessario
// attivare il namespace "System.Numerics" con il comando
// "Progetto / Aggiungi Riferimento / Assembly / Framework ..."

using  System.Numerics;

namespace CodificaDecodificaRSAconBigInteger
{
   public partial class frmRSA : Form
   {
      public frmRSA()
      {
         InitializeComponent();
      }

      // Possibili coppie di chiavi RSA ...
      // (3,33) (7,33)
      // (215,551) (143,551)
      // (4103,5207) (2087,5207)

      // chiave pubblica (d,n)
      long _d;
      long _n;

      // chiave privata (d,n)
      long _e;

      // messaggi ...
      long M;          // ... originale
      BigInteger  Mc;     // ... criptato
      BigInteger  Md;     // ... decriptato

      private void plsCripta_Click(object sender, EventArgs e)
      {
         // acquisisci il messaggio (numero intero < n) da criptare ...
         M = Convert.ToInt64(nudM.Value);

         // acquisisci le chiavi (d,n) e (e,n) ...
         _d = Convert.ToInt64(nudD.Value);
         _n = Convert.ToInt64(nudN.Value);
         _e = Convert.ToInt64(nudE.Value);

         // completa la visualizzazione delle chiavi ...
         lblN.Text = _n.ToString();

         // verifica che il messaggio M sia inferiore a _n ...
         if (M < _n)
         {

            // cripta il messaggio:   Mc = (M ^ d) mod n ...


            // ... calcola la Potenza "M ^ d" con il metodo Pow ...
            //    (NOTA BENE: Pow accetta solo un esponente di tipo "int")

            BigInteger Potenza = BigInteger.Pow( M, Convert.ToInt32(_d) );


            // ... calcola il resto della divisione (Potenza / n)
            //    usando il metodo Remainder della classe BigInteger ...

            Mc = BigInteger.Remainder(Potenza, _n);


            // visualizza il messaggio criptato ...
            lblMcriptato.Text = Mc.ToString();

            plsDecripta.Enabled = true;
            lblMdecriptato.Text = "";
         }
         else
            MessageBox.Show("Il messaggio M deve essere minore di n.");
      }


      private void plsDecripta_Click(object sender, EventArgs e)
      {
         // decripta il messaggio:   Md = (Mc ^ e) mod n ...


         // ... calcola la Potenza "Mc ^ e" con il metodo Pow ...
         //    (nota come Pow accetta solo un esponente "int")

         BigInteger Potenza = BigInteger.Pow(Mc, Convert.ToInt32(_e));


         // ... calcola il resto della divisione (Potenza / n)
         //    usando il metodo Remainder della classe BigInteger ...

         Md = BigInteger.Remainder(Potenza, _n);


         // visualizza il messaggio criptato ...
         lblMdecriptato.Text = Md.ToString();
      }
   }
}
```