La Chat

Il problema

Progettare l'architettura di un' applicazione VB 6.0 che fornisca su una rete LAN un servizio per la gestione di una Chat e realizzarla.

Richiami teorici (tratti dalla guida del VB 6.0 e dal sito http://www.dia.unisa.it)

Protocollo TCP

- Il protocollo TCP è un protocollo basato sulla connessione ed è analogo a un telefono, in quanto l'utente deve stabilire una connessione prima di poter procedere; consente ,quindi, di effettuare un riconoscimento del mittente.
- Il protocollo TCP richiede una connessione esplicita prima di inviare o ricevere dati.
- Il protocollo TCP la mantiene e garantisce l'integrità dei dati, e permette , quindi, l'invio dati di grandi dimensioni.

Crittosistema RSA

- Il crittosistema RSA(ideato da Rivest , Shamir e Adleman nel 1978) è anche oggi molto utilizzato grazie alla sua **semplicità strutturale**.
- La **sicurezza** del sistema viene garantita dalla **difficoltà di fattorizzare** grandi numeri.
- Operazione di cifratura con l'RSA:
  c = m ^ pub % n (c = messaggio cifrato)
  Operazione di decifratura con l'RSA:
  m = c^ pri % n (m = messaggio in chiaro)
  dove :
  n = a * b (a e b numeri primi molto grandi)
  pri viene scelto in modo tale che non abbia fattori primi in comune con **z**, dove:
  z = (a-1) * (b – 1)
  pub viene scelto in modo tale che si soddisfi l'equazione: **(pub * pri) % z = 1.**
  NB: n deve essere maggiore di m.
- Un semplice algoritmo sull'elevazione a potenza modulare (ma non il più efficiente) è il metodo naive . Esso calcola x ^ y mod z semplicemente moltiplicando x per se stesso y volte e facendo ad ogni passo modulo con z, evitando numeri grandi. (vedi pagina 25).

Analisi del problema

L'applicazione sarà basata sul modello Client/Server. Per stabilire la connessione ed effettuare lo scambio di dati tra l'applicazione lato Server e le applicazioni concorrenti lato Client si userà il controllo Winsock impostato sul protocollo **TCP (Transmission Control Protocol),** per le sue caratteristiche esposte in precedenza. Inoltre per la connessione si utilizzerà la porta 1001 , non usata dai programmi presenti di default nei PC a cui è destinato l'uso.

Le funzionalità individuate

L'applicazione lato server si occuperà di :

- Accettare le richieste di connessione dei vari utenti provenienti dalle applicazioni lato client, verificando che non si è superato il limite di connessioni massimo stabilito dall'amministratore del servizio;

- Autentificare gli utenti , verificando che il nome e la password ,inviati dai client che desiderano connettersi, siano eguali a quelli immessi dall'amministratore nel file delle password;
- Inviare l'elenco degli utenti connessi agli utenti che sono stati autentificati.
- Consentire ai vari client di aggiornare la propria lista quando un utente si disconnette
- Consentire ai vari client di inviare messaggi agli altri client connessi.
- Consentire all'amministratore del servizio di :
    - vedere in Real-Time la lista degli utenti connessi
    - Contattare un client .
    - Leggere i messaggi inviati dai client .
    - Chiudere la connessione ad un client.

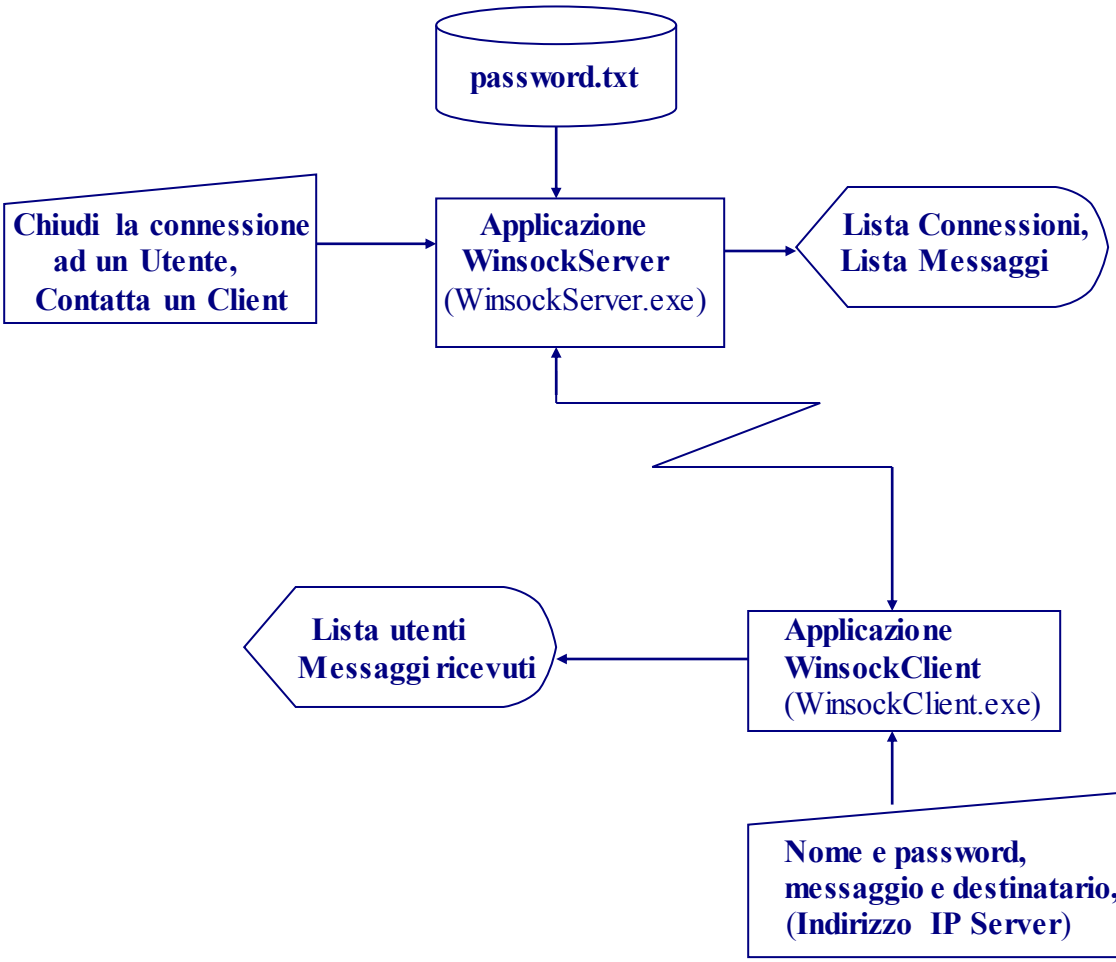
L'applicazione lato client si occuperà di :

- Acquisire il nome e la password dell'utente da inviare al server subito dopo che il server ha risposto alla richiesta di connessione con un "OK".
- Richiedere la lista degli utenti connessi.
- Inviare al server i messaggi che l'utente decide di spedire ad un destinatario presente nella lista degli utenti connessi;formattandoli secondo il seguente **protocollo**: **<destinatario scelto> + < Chr(13)> + <messaggio>.** Il server quando riceverà il messaggio , si ricaverà il destinatario e il messaggio , e lo invierà al destinatario secondo il **protocollo**:
  <mittente> + <" dice:"> + <messaggio>
  - crittografare i messaggi da inviare e decodificare i messaggi ricevuti utilizzando il metodo RSA.

I vincoli

- L'applicazione sarà provvista di un semplice sistema di protezione:
    - il file delle password (**password.txt**) si troverà in una zona protetta dell'server , nella stessa directory dell'applicazione lato server (WinsockServer.exe) .
    - il file delle password sarà crittografato con il metodo RSA.
    - Le applicazioni lato Client concorrenti (WinsockClient.exe) dovranno essere di **un numero inferiore a 10**(aumentando tale numero si potrebbero verificare degli errori di connessione non gestiti).
    - Per semplicità la **chiavi pubbliche e private** utilizzate nella codifica/decodifica saranno **uguali per tutte le connessioni.**

System resources chart



La progettazione

Le specifiche tecniche per la creazione dei moduli del progetto.

Le due applicazioni (WinsockClient.exe e WinsockServer.exe) dovranno essere realizzare in ambiente VB 6.0 su due progetti distinti (ProgettoClient.vbp e ProgettoServer.vbp), ciascuno aventi un singolo *progetto di avvio* .
Il ProgettoClient conterrà al suo interno due form (frmClient.frm e FormLogin.frm) e un modulo(Module1.bas). FormLogin.frm si occuperà dell'acquisizione del nome e della password di login, mentre frmClient.frm si occuperà di :

- consentire all'utente di connettersi al server(Tramite un command Botton chiamato cmdConnetti)
- visualizzare la lista degli utenti connessi (tramite una combo box chiamata Combo1)
- inviare al server i messaggi inseriti in una text-box chiamata txtSend quando l'utente clicca sul comand botton cmdInvia) secondo il protocollo specificato in analisi.
- Visualizzare i messaggi inviati dagli altri utenti o dal server verso l'utente in una text box chiamata txtOutput.

Il ProgettoServer conterrà al suo interno un form ,chiamato frmServer, che si occuperà di consentire all' amministratore di :

- visualizzare la lista delle connessioni tramite la combo box Combo1
- visualizzare la lista degli utenti connessi (tramite una combo box chiamata Combo1)

- visualizzare i messaggi inviati dai client (tramite una text box chiamata txtOutput)
- spedire il testo inserito dall'utente in una text box chiamata txtSendData al client selezionato nella Combo1 quando al click su un Command Botton chiamato Command1

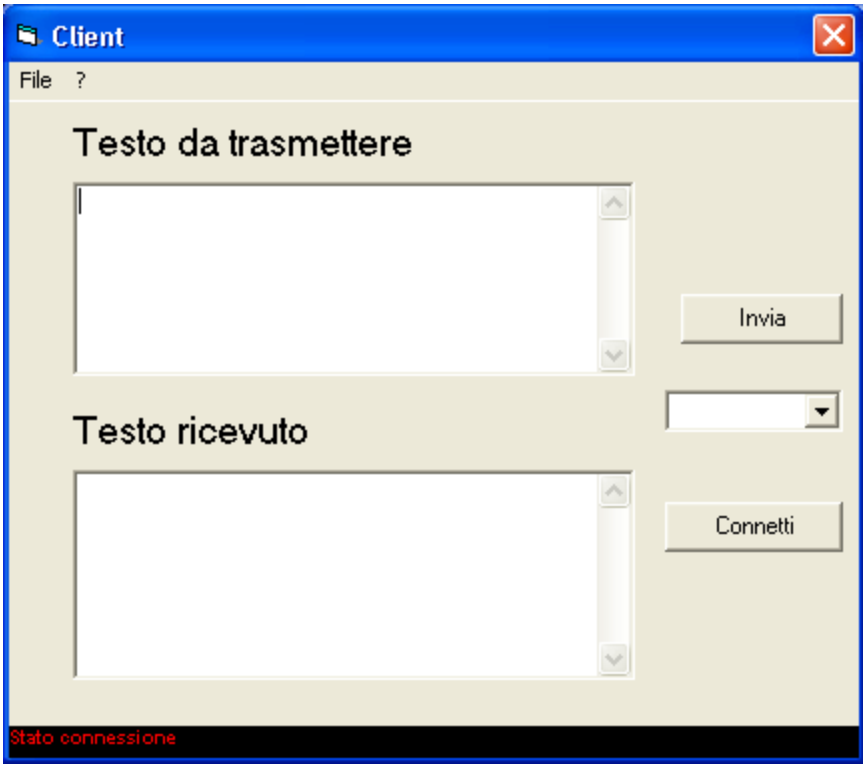
La realizzazione dell' applicazione

In basso viene presentato l'elenco di principali oggetti inseriti nel ProgettoClient e nel ProgettoServer ; individuati seguendo le specifiche tecniche precedentemente esposte.

Tris (ProgettoClient.vbp)
- **form** frmLogin (frmLogin.frm)
  - **label** Label1
  - **label** Label2
  - **Text** Text1
  - **Text** Text2
  - **CommandButton** Command1
- **form** frmClient (frmClient.frm)
  - **barra dei menù**
  - **Text** txtSend
  - **Text** txtOutput
  - **Winsock** tcpClient
  - **label** Label1
  - **label** Label2
  - **label** Label3
- **modulo** Module1 (Module1.bas)

Tris (ProgettoServer.vbp)
- **form** frmServer (frmServer.frm)
  - **Text** txtSendData
  - **Text** txtOutput
  - **Timer** Timer2
  - **Timer** Timer3
  - **Winsock** tcpServer

La disposizione dei controlli sui rispettivi **form** sarà la seguente:



NB: Il semaforo dovrebbe indicare la connessione (attraverso l'accensione della luce verde) e la disconnessione (attraverso l'accensione della luce Rossa) di un utente.

Scrittura del codice

File WinsockServer.vbp

Form=frmServer.frm
IconForm="frmServer"
Startup="frmServer"
HelpFile=""
Title="ProgettoWinsockServer"
ExeName32="WinsockServer.exe"

File frmServer.frm

```
VERSION 5.00
'Importazione del oggetto Winsock
Object = "{248DD890-BB45-11CF-9ABC-0080C7E7B78D}#1.0#0"; "MSWINSCK.OCX"
Begin VB.Form frmServer
   'impostazioni attributi della form
   BorderStyle     =  1  'Fixed Single
   Caption         =  "Server TCP"
   ClientHeight    =  5310
   ClientLeft      =  150
   ClientTop       =  840
   ClientWidth     =  6315
   MaxButton       =  0   'False
   MinButton       =  0   'False
   ScaleHeight     =  5310
   ScaleWidth      =  6315
```

```
  StartUpPosition = 3 'Windows Default
'dichiarazione degli oggetti contenuti nel form e impostazione dei valori degli attributi
  Begin VB.Timer Timer3
    Enabled        =  0  'False
    Interval       =  1
    Left           =  4920
    Top            =  2520
  End
  Begin VB.Timer Timer2
    Enabled        =  0  'False
    Interval       =  1
    Left           =  4920
    Top            =  3240
  End
  Begin VB.CommandButton Command1
    Caption        =  "Invia"
    Height         =  375
    Left           =  5040
    TabIndex       =  7
    Top            =  1560
    Width          =  855
  End
  Begin VB.ComboBox Combo1
    Height         =  315
    Left           =  5040
    Style          =  2  'Dropdown List
    TabIndex       =  5
    Top            =  4560
    Width          =  975
  End
  Begin MSWinsockLib.Winsock tcpServer
    Index          =  0
    Left           =  5760
    Top            =  3720
    _ExtentX       =  741
    _ExtentY       =  741
    _Version       =  393216
  End
  Begin VB.TextBox txtOutput
    Height         =  1575
    Left           =  480
    Locked         =  -1  'True
    MultiLine      =  -1  'True
    ScrollBars     =  2  'Vertical
    TabIndex       =  1
    Top            =  2760
    Width          =  4215
  End
  Begin VB.TextBox txtSendData
    Height         =  1455
    Left           =  480
    MultiLine      =  -1  'True
```

```
    ScrollBars      =  2  'Vertical
    TabIndex        =  0
    Top             =  600
    Width           =  4215
  End
  Begin VB.Label Label4
    Caption         =  "Seleziona l'Indice della connessione "
    BeginProperty Font
      Name          =  "MS Sans Serif"
      Size          =  12
      Charset       =  0
      Weight        =  700
      Underline     =  0  'False
      Italic        =  0  'False
      Strikethrough =  0  'False
    EndProperty
    ForeColor       =  &H000000FF&
    Height          =  375
    Left            =  480
    TabIndex        =  6
    Top             =  4560
    Width           =  4455
  End
  Begin VB.Label Label3
    BackColor       =  &H80000007&
    Caption         =  "Stato connessione"
    BeginProperty Font
      Name          =  "Arial"
      Size          =  6.75
      Charset       =  0
      Weight        =  700
      Underline     =  0  'False
      Italic        =  0  'False
      Strikethrough =  0  'False
    EndProperty
    ForeColor       =  &H000000FF&
    Height          =  255
    Left            =  0
    TabIndex        =  4
    Top             =  5040
    Width           =  6375
  End
  Begin VB.Label Label2
    Caption         =  "Testo ricevuto"
    BeginProperty Font
      Name          =  "MS Sans Serif"
      Size          =  13.5
      Charset       =  0
      Weight        =  400
      Underline     =  0  'False
      Italic        =  0  'False
      Strikethrough =  0  'False
```

```
    EndProperty
    Height          =   375
    Left            =   480
    TabIndex        =   3
    Top             =   2280
    Width           =   3975
  End
  Begin VB.Label Label1
    Caption         =   "Testo da trasmettere"
    BeginProperty Font
      Name          =   "MS Sans Serif"
      Size          =   13.5
      Charset       =   0
      Weight        =   400
      Underline     =   0   'False
      Italic        =   0   'False
      Strikethrough =   0   'False
    EndProperty
    Height          =   375
    Left            =   480
    TabIndex        =   2
    Top             =   120
    Width           =   3975
  End
  Begin VB.Shape Shape2
    BackColor       =   &H000000FF&
    BackStyle       =   1  'Opaque
    Height          =   255
    Left            =   5400
    Shape           =   3  'Circle
    Top             =   360
    Width           =   375
  End
  Begin VB.Shape Shape1
    BackColor       =   &H00000000&
    BackStyle       =   1  'Opaque
    Height          =   255
    Left            =   5400
    Shape           =   3  'Circle
    Top             =   720
    Width           =   375
  End
  Begin VB.Shape Shape3
    BackColor       =   &H00808080&
    BackStyle       =   1  'Opaque
    BorderColor     =   &H00000000&
    Height          =   855
    Left            =   5400
    Top             =   240
    Width           =   375
  End
```

```
  'Impostazione della barra di menù
  Begin VB.Menu mnuFile
    Caption        =   "&File"
    Begin VB.Menu mnuFileItem
      Caption        =   "&Propietà"
      Index          =   0
    End
    Begin VB.Menu mnuFileItem
      Caption        =   "-"
      Index          =   1
    End
    Begin VB.Menu mnuFileItem
      Caption        =   "&Esci"
      Index          =   6
    End
  End
  Begin VB.Menu mnuHelp
    Caption        =   "&?"
    Begin VB.Menu mnuHelpItem
      Caption        =   "&Help On Line"
      Enabled        =   0   'False
      Index          =   0
    End
    Begin VB.Menu mnuHelpItem
      Caption        =   "-"
      Index          =   1
    End
    Begin VB.Menu mnuHelpItem
      Caption        =   "&Informazioni su"
      Enabled        =   0   'False
      Index          =   2
    End
  End
End
Attribute VB_Name = "frmServer"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False

'dichiarazione delle variabili private della form
Option Explicit
Private intMax As Long
Dim vettNone(10) As String
Dim UltimoUtente As String
Dim UltimoUtenteDisconnesso As String
Dim IndiceAvviso As Integre
```

```
'definizione procedure
Private Sub Command1_Click()
   On Error GoTo errore
   tcpServer(Combo1.Text).SendData txtSendData.Text
   Exit Sub
errore:
   If (txtSendData.Text <> "") Then
      txtSendData.Text = ""
      MsgBox "Sei Disconnesso!", , "Errore"
   End If
End Sub

'Procedura di risposta all'evento caricamento del form
Private Sub Form_Load()
   intMax = 0
   tcpServer(0).LocalPort = 1001
   tcpServer(0).Listen
   Dim i
   i = 0
   Do
      vettNone(i) = ""
      i = i + 1
   Loop While i < 10
   AggiornaCombo
End Sub

'Procedura di risposta all'evento scaricamento del form
Private Sub Form_Unload(Cancel As Integer)
   On Error Resume Next
   Dim i As Integer
   For i = 0 To 10
      tcpServer(i).Close
   Next i
End Sub

'Procedura di risposta all'evento click sulla voce del menù File
Private Sub mnuFileItem_Click(Index As Integer)
   If Index = 6 Then
      Unload Me
   Else
      If Index = 0 Then
         Dim stri As String
         stri = "UDPProtocol"
         If tcpServer(0).Protocol = sckTCPProtocol Then stri = "TCPProtocol"
         MsgBox ("RemoteHost: " & tcpServer(0).RemoteHost _
         & Chr(13) & "RemotePort: " & tcpServer(0).LocalPort) & Chr(13) _
         & "Protocollo connessione: " & stri, , "Informazioni sulla connessione"
      End If
   End If
End Sub
```

```
'Procedura di risposta all'evento chiusura della connessione del winsock
Private Sub tcpServer_Close(Index As Integer)
   UltimoUtenteDisconnesso = vettNone(Index)
   Shape1.BackColor = 0
   Shape2.BackColor = &HFF&
   tcpServer(Index).Close
   Unload tcpServer(Index)
   vettNone(Index) = ""
   AggiornaCombo
   Timer3.Enabled = True
End Sub

'Procedura di risposta all'evento richiesta di connessione
Private Sub tcpServer_ConnectionRequest _
(Index As Integer, ByVal requestID As Long)
   Dim vuoti As Integer
   Shape1.BackColor = &HFF00&
   Shape2.BackColor = 0
   vuoti = 0
   Do
      intMax = intMax + 1
      If intMax > 10 Then intMax = 0
      vuoti = vuoti + 1
      If vuoti > 10 Then
         MsgBox "Il server ha rifiutato una richiesta di connessione"
         Exit Sub
      End If
   Loop While Not esistetsp()
   tcpServer(intMax).Accept requestID
   tcpServer(intMax).SendData "Send"
   tcpServer(intMax).Tag = "Init"
   AggiornaCombo
End Sub

Private Function esistetsp() As Boolean
   On Error GoTo errore
   esistetsp = True
   Load tcpServer(intMax)
   Exit Function
errore:
   esistetsp = False
End Function

Private Function esiste(i As Integer) As Boolean
   On Error GoTo errore
   esiste = tcpServer(i).State
   esiste = True
   Exit Function
errore:
   esiste = False
End Function
```

```
Private Sub AggiornaCombo()
On Error Resume Next
   Dim i As Integer
   Combo1.Clear
   i = 1
   Do
      If esiste(i) Then Combo1.AddItem i
      i = i + 1
   Loop While i < 11
   Combo1.Text = Combo1.List(0)
End Sub

'Procedura di risposta all'evento arrivo nuovo messaggio
Private Sub tcpServer_DataArrival _
(Index As Integer, ByVal bytesTotal As Long)
   Dim strData As String
   tcpServer(Index).GetData strData
   Select Case tcpServer(Index).Tag
      Case "Init"
         If verNome(strData) Then     ' autenticazione Nome
            tcpServer(Index).SendData "OK"
            tcpServer(Index).Tag = "Init1"
            vettNone(Index) = strData
         Else
            tcpServer(Index).SendData "KO"
         End If
    Case "Init1"
         If verPass(strData) Then    ' autenticazione Password
            tcpServer(Index).SendData "OK"
            tcpServer(Index).Tag = "Send"
         Else
            tcpServer(Index).SendData "KO"
         End If
    Case "Send"
         If strData = "OK" Then
            tcpServer(Index).Tag = "Lista"
            tcpServer(Index).SendData "Lista?"
         Else
            tcpServer(Index).SendData "KO"
         End If
     Case "Lista"
            If strData = "Lista" Then
               Dim strin As String, i As Integer
               i = 0
               Do
                  If vettNone(i) <> "" Then strin = strin & vettNone(i) & Chr(13)
                  i = i + 1
               Loop While i < 10
               UltimoUtente = vettNone(Index)
               tcpServer(Index).SendData strin
               Timer2.Enabled = True
               tcpServer(Index).Tag = "Connesso"
```

```
            Else
               tcpServer(Index).SendData "KO"
            End If
      Case "Connesso"
         If strData = "Nuovo Utente" Then
            tcpServer(Index).SendData UltimoUtente
         Else
            If strData = "Nuovo Messaggio" Then
               tcpServer(Index).SendData "OK"
               tcpServer(Index).Tag = "Send Message"
            Else
               If strData = "Utente Disconnesso" Then
                  tcpServer(Index).SendData UltimoUtenteDisconnesso
               End If
            End If
         End If
       Case "Send Message"
         Dim destinatario As String, inizio As Integer, fine As Integer
         Dim s As String, cont As Integer
         destinatario = ""
         inizio = 1
         cont = 0
         fine = Len(strData)
         Do While inizio <= fine
            s = Mid(strData, inizio, 1)
            If s = Chr(13) Then
              'trova l'indice
               Do
                  If vettNone(cont) = destinatario Then tcpServer(cont).SendData vettNone(Index) & "
dice: " & Mid(strData, inizio + 1)
                  cont = cont + 1
               Loop While cont < 10
            Else
               destinatario = destinatario & s
            End If
           inizio = inizio + 1
         Loop
         tcpServer(Index).Tag = "Connesso"
   End Select
   txtOutput.Text = txtOutput.Text & "Host N° " & Index & " dice ..." & Chr(13) & Chr(10)
   txtOutput.Text = txtOutput.Text & strData & Chr(13) & Chr(10)
   txtOutput.Text = txtOutput.Text & ">fine<" & Chr(13) & Chr(10)
End Sub

'Procedura di risposta all'evento Timer di Timer2
Private Sub Timer2_Timer()
   On Error Resume Next
   tcpServer(IndiceAvviso).SendData "Nuovo Utente"
   IndiceAvviso = IndiceAvviso + 1
   If IndiceAvviso > 10 Then
      Timer2.Enabled = False
      IndiceAvviso = 0
```

```
    End If
End Sub

'Procedura di risposta all'evento Timer di Timer3
Private Sub Timer3_Timer()
    On Error Resume Next
    tcpServer(IndiceAvviso).SendData "Utente Disconnesso"
    IndiceAvviso = IndiceAvviso + 1
    If IndiceAvviso > 10 Then
        Timer3.Enabled = False
        IndiceAvviso = 0
    End If
End Sub


Private Function verNome(strData As String) As Boolean
    Dim c As Double, m As Double
    Dim messaggio As String
    Dim n As Double, a As Double, b As Double, z As Double, pri As Double, pub As Double
    Dim nomeAtt As String, passAtt As String, i As Integer
    Dim codificato, ia, tmp


    a = 17
    b = 6
    pri = 21
    n = a * b

    verNome = False
    Open ".\password.txt" For Input As #1       ' Apre il file per l'output.
    For i = 0 To 9
        Input #1, nomeAtt, passAtt
        'decodifica
        messaggio = nomeAtt
        codificato = ""
        For ia = 1 To Len(messaggio)
            tmp = Mid(messaggio, ia, 1)
            m = Asc(tmp)
            m = PotMod(m, pri, n)
            codificato = codificato & Chr(m)
        Next ia
        If codificato = StrConv(strData, 1) Then verNome = True
    Next
    Close #1   ' Chiude il file
    For i = 0 To 10
        If vettNone(i) = StrConv(strData, 1) Then verNome = False
    Next
End Function

Private Function verPass(strData As String) As Boolean
    Dim c As Double, m As Double
    Dim messaggio As String
```

```
    Dim n As Double, a As Double, b As Double, z As Double, pri As Double, pub As Double
    Dim nomeAtt As String, passAtt As String, i As Integer
    Dim codificato, ia, tmp
    a = 17
    b = 6
    pri = 21
    n = a * b

    verPass = False
    Open ".\password.txt" For Input As #1       ' Apre il file per l'output.
    For i = 0 To 9
       Input #1, nomeAtt, passAtt
       'decodifica
       messaggio = passAtt
       codificato = ""
       For ia = 1 To Len(messaggio)
          tmp = Mid(messaggio, ia, 1)
          m = Asc(tmp)
          m = PotMod(m, pri, n)
          codificato = codificato & Chr(m)
       Next ia
       If codificato = StrConv(strData, 1) Then verPass = True
    Next
    Close #1   ' Chiude il file
End Function

'La funzione calcola x ^ y mod z utilizzando il metodo naive
Private Function PotMod(x As Double, y As Double, z As Double) As Double
    Dim a As Double, i As Double
    a = 1
    For i = 1 To y
       a = (a * x) Mod z
    Next i
    PotMod = a
End Function
```

File WinsockClient.vbp

```
Form=frmClient.frm
Form=FormLogin.frm
Module=Module1; Module1.bas
IconForm="frmClient"
Startup="FormLogin"
Title="WinsockClient"
ExeName32="WinsockClient.exe"
Name="ProgettoClient"
```

File FormLogin.frm

```
VERSION 5.00
Begin VB.Form FormLogin
   'impostazioni attributi della form
   BackColor       =   &H00000000&
   BorderStyle     =   3  'Fixed Dialog
   Caption         =   "Login"
   ClientHeight    =   2850
   ClientLeft      =   45
   ClientTop       =   435
   ClientWidth     =   4680
   LinkTopic       =   "Form1"
   MaxButton       =   0   'False
   MinButton       =   0   'False
   ScaleHeight     =   2850
   ScaleWidth      =   4680
   StartUpPosition =   3  'Windows Default
   Begin VB.CommandButton Command1
      Caption         =   "OK"
      Height          =   375
      Left            =   1800
      TabIndex        =   4
      Top             =   2160
      Width           =   1575
   End
   Begin VB.TextBox Text2
      DataSource      =   "*"
      BeginProperty Font
         Name            =   "MS Sans Serif"
         Size            =   18
         Charset         =   0
         Weight          =   400
         Underline       =   0   'False
         Italic          =   0   'False
         Strikethrough   =   0   'False
      EndProperty
      Height          =   495
      IMEMode         =   3  'DISABLE
      Left            =   2400
      PasswordChar    =   "*"
      TabIndex        =   3
      Top             =   1200
      Width           =   1815
   End
   Begin VB.TextBox Text1
      BeginProperty Font
         Name            =   "MS Sans Serif"
         Size            =   18
         Charset         =   0
         Weight          =   400
         Underline       =   0   'False
```

```
      Italic          =   0   'False
      Strikethrough   =   0   'False
    EndProperty
    Height          =   495
    Left            =   2400
    TabIndex        =   0
    Top             =   360
    Width           =   1815
  End
  Begin VB.Label Label2
    BackStyle       =   0   'Transparent
    Caption         =   "Password"
    BeginProperty Font
      Name          =   "MS Sans Serif"
      Size          =   13.5
      Charset       =   0
      Weight        =   400
      Underline     =   0   'False
      Italic        =   0   'False
      Strikethrough =   0   'False
    EndProperty
    ForeColor       =   &H00FFFFFF&
    Height          =   375
    Left            =   240
    TabIndex        =   2
    Top             =   1200
    Width           =   1815
  End
  Begin VB.Label Label1
    BackStyle       =   0   'Transparent
    Caption         =   "Nick Name"
    BeginProperty Font
      Name          =   "MS Sans Serif"
      Size          =   13.5
      Charset       =   0
      Weight        =   400
      Underline     =   0   'False
      Italic        =   0   'False
      Strikethrough =   0   'False
    EndProperty
    ForeColor       =   &H00FFFFFF&
    Height          =   375
    Left            =   240
    TabIndex        =   1
    Top             =   360
    Width           =   1815
  End
End
Attribute VB_Name = "FormLogin"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
```

```
Attribute VB_Exposed = False

'procedura di risposa all'evento click su Command1
Private Sub Command1_Click()
    None = Text1.Text
    Password = Text2.Text
    Unload Me
    frmClient.Show
End Sub
```

File frmClient.frm

```
VERSION 5.00
'Importazione del oggetto Winsock
Object = "{248DD890-BB45-11CF-9ABC-0080C7E7B78D}#1.0#0"; "MSWINSCK.OCX"
Begin VB.Form frmClient
   'impostazioni attributi della form
   BorderStyle     =  1 'Fixed Single
   Caption         =  "Client"
   ClientHeight    =  4920
   ClientLeft      =  150
   ClientTop       =  840
   ClientWidth     =  6375
   LinkTopic       =  "Form1"
   MaxButton       =  0 'False
   MinButton       =  0 'False
   ScaleHeight     =  4920
   ScaleWidth      =  6375
   StartUpPosition =  3 'Windows Default
   Begin VB.ComboBox Combo1
      Height          =  315
      ItemData        =  "frmClient.frx":0000
      Left            =  4920
      List            =  "frmClient.frx":0002
      Style           =  2 'Dropdown List
      TabIndex        =  7
      Top             =  2160
      Width           =  1335
   End
   Begin VB.CommandButton Command1
      Caption         =  "Invia"
      Height          =  375
      Left            =  5040
      TabIndex        =  6
      Top             =  1440
      Width           =  1215
   End
   Begin MSWinsockLib.Winsock tcpClient
      Left            =  5640
      Top             =  3840
```

```
    _ExtentX        =  741
    _ExtentY        =  741
    _Version        =  393216
    RemoteHost      =  "sistemi10"
    RemotePort      =  1001
  End
  Begin VB.CommandButton cmdConnect
    Caption         =  "Connetti"
    Height          =  375
    Left            =  4920
    TabIndex        =  2
    Top             =  3000
    Width           =  1335
  End
  Begin VB.TextBox txtOutput
    Height          =  1575
    Left            =  480
    Locked          =  -1  'True
    MultiLine       =  -1  'True
    ScrollBars      =  2  'Vertical
    TabIndex        =  1
    Top             =  2760
    Width           =  4215
  End
  Begin VB.TextBox txtSend
    Height          =  1455
    Left            =  480
    MultiLine       =  -1  'True
    ScrollBars      =  2  'Vertical
    TabIndex        =  0
    Top             =  600
    Width           =  4215
  End
  Begin VB.Label Label3
    BackColor       =  &H80000007&
    Caption         =  "Stato connessione"
    BeginProperty Font
      Name          =  "Arial"
      Size          =  6.75
      Charset       =  0
      Weight        =  700
      Underline     =  0   'False
      Italic        =  0   'False
      Strikethrough =  0   'False
    EndProperty
    ForeColor       =  &H000000FF&
    Height          =  255
    Left            =  0
    TabIndex        =  5
    Top             =  4680
    Width           =  6375
  End
```

```
  Begin VB.Label Label2
    Caption         =   "Testo ricevuto"
    BeginProperty Font
      Name          =   "MS Sans Serif"
      Size          =   13.5
      Charset       =   0
      Weight        =   400
      Underline     =   0   'False
      Italic        =   0   'False
      Strikethrough =   0   'False
    EndProperty
    Height          =   375
    Left            =   480
    TabIndex        =   4
    Top             =   2280
    Width           =   3975
  End
  Begin VB.Label Label1
    Caption         =   "Testo da trasmettere"
    BeginProperty Font
      Name          =   "MS Sans Serif"
      Size          =   13.5
      Charset       =   0
      Weight        =   400
      Underline     =   0   'False
      Italic        =   0   'False
      Strikethrough =   0   'False
    EndProperty
    Height          =   375
    Left            =   480
    TabIndex        =   3
    Top             =   120
    Width           =   3975
  End
  'Impostazione della barra di menù
  Begin VB.Menu mnuFile
    Caption         =   "&File"
    Begin VB.Menu mnuFileItem
      Caption       =   "&Login"
      Index         =   0
    End
    Begin VB.Menu mnuFileItem
      Caption       =   "&Nuova Connessione"
      Index         =   1
    End
    Begin VB.Menu mnuFileItem
      Caption       =   "-"
      Index         =   2
    End
    Begin VB.Menu mnuFileItem
      Caption       =   "Propietà"
      Index         =   3
```

```
    End
    Begin VB.Menu mnuFileItem
      Caption        =  "-"
      Index          =  4
    End
    Begin VB.Menu mnuFileItem
      Caption        =  "&Esci"
      Index          =  5
    End
  End
  Begin VB.Menu mnuHelp
    Caption        =  "&?"
    Begin VB.Menu mnuHelpItem
      Caption        =  "&Help Online"
      Enabled        =  0  'False
      Index          =  0
    End
    Begin VB.Menu mnuHelpItem
      Caption        =  "-"
      Index          =  1
    End
    Begin VB.Menu mnuHelpItem
      Caption        =  "&Informazioni su"
      Index          =  2
    End
  End
End
Attribute VB_Name = "frmClient"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
Option Explicit
‘dichiarazione delle variabili private della form
Dim IndirizzoIp As String
Dim conferma As Boolean
‘definizione procedure
Private Sub Command1_Click()
   On Error GoTo Errore
   tcpClient.SendData "Nuovo Messaggio"
   tcpClient.Tag = "Send Message"
   Exit Sub
Errore:
   If (txtSend.Text <> "") Then
      txtSend.Text = ""
      MsgBox "Sei Disconnesso!", , "Errore"
   End If
End Sub

‘Procedura di risposta all’evento caricamento del form
Private Sub Form_Load()
   ' Nota: per specificare un host remoto, è possibile
```

```
    ' utilizzare l'indirizzo IP (es: "121.111.1.1")
    ' oppure il nome del computer, come qui indicato.
    ' IndirizzoIp = "127.1.1.1"
    tcpClient.RemoteHost = IndirizzoIp
    tcpClient.RemotePort = 1001
    conferma = False
End Sub

Private Sub cmdConnect_Click()
  If cmdConnect.Caption = "Connetti" Then
      ' Richiama il metodo Connect per stabilire una
      ' connessione.
      tcpClient.Connect IndirizzoIp, 1001
      cmdConnect.Caption = "Disconnetti"
      Label3.Caption = "Connessione in corso"
    Else
      tcpClient.Close
      cmdConnect.Caption = "Connetti"
      Label3.Caption = ""
    End If
End Sub

'Procedura di risposta all'evento scaricamento del form
Private Sub Form_Unload(Cancel As Integer)
    tcpClient.Close
End Sub

'Procedura di risposta all'evento click sulla voce del menù File
Private Sub mnuFileItem_Click(Index As Integer)
    Dim stri As String
    Select Case Index
    Case 0
      Unload Me
      FormLogin.Show
    Case 1
      IndirizzoIp = InputBox("Inserisci il nome del PC Server o il suo indirizzo IP", "Crea Nuova
Connessione", IndirizzoIp)
      tcpClient.Close
      tcpClient.Connect IndirizzoIp, 1001
      cmdConnect.Caption = "Disconnetti"
      Label3.Caption = "Connessione in corso"
    Case 3
      stri = "UDPProtocol"
      If tcpClient.Protocol = sckTCPProtocol Then stri = "TCPProtocol"
      MsgBox ("RemoteHost: " & tcpClient.RemoteHost _
      & Chr(13) & "RemotePort: " & tcpClient.LocalPort) & Chr(13) _
      & "Protocollo connessione: " & stri, , "Informazioni sulla connessione"
    Case 5
      Unload Me
    End Select
End Sub
```

```
Private Sub mnuHelpItem_Click(Index As Integer)
   If Index = 2 Then MsgBox "Questo programma è stato creato da .. " & Chr(13) & "Davide
Caruso", , "Informazoni su Client"
End Sub

'Procedura di risposta all'evento apertura della connessione del winsock
Private Sub tcpClient_Connect()
   tcpClient.Tag = "Init"
End Sub

'Procedura di risposta all'evento errore nella connessione
Private Sub tcpClient_Error(ByVal Number As Integer, Description As String, ByVal Scode As
Long, ByVal Source As String, ByVal HelpFile As String, ByVal HelpContext As Long,
CancelDisplay As Boolean)
   Label3.Caption = "Errore " & Number & " - " & Description
End Sub

'Procedura di risposta all'evento arrivo di Dati
Private Sub tcpClient_DataArrival _
(ByVal bytesTotal As Long)
   Dim strData As String
   Dim n As Double, a As Double, b As Double, z As Double, pri As Double, pub As Double
   Dim c As Double, m As Double
   Dim messaggio As String
   Dim mittente, inizio1, cont, fine1, s
   Dim cod
   Dim i
   Dim inizio As Integer, fine As Integer, nomeAtt As String
   Dim tmp As Integer
   tcpClient.GetData strData
   Select Case tcpClient.Tag
   Case "Init"
      tcpClient.SendData None
      tcpClient.Tag = "Init1"
   Case "Init1"
      If strData = "OK" Then
         tcpClient.SendData Password
         tcpClient.Tag = "Send"
      Else
         Label3.Caption = "Nome non valido."
         tcpClient.Close
      End If
   Case "Send"
      If strData = "OK" Then
         tcpClient.SendData "OK"
         tcpClient.Tag = "Connesso"
         Label3.Caption = "Connesso"
      Else
         Label3.Caption = "Password non valida."
         tcpClient.Close
      End If
   Case "Connesso"
```

```
      Select Case strData
      Case "Lista?"
         tcpClient.SendData "Lista"
         tcpClient.Tag = "Lista"
         Combo1.Clear
      Case "Nuovo Utente"
         tcpClient.SendData "Nuovo Utente"
         tcpClient.Tag = "Nuovo Utente"
      Case "Utente Disconnesso"
         tcpClient.SendData "Utente Disconnesso"
         tcpClient.Tag = "Utente Disconnesso"
      Case Else ‘messaggio utente
      'decodifica
         mittente = ""
         inizio1 = 1
         cont = 0
         fine1 = Len(strData)

         Do While inizio1 <= fine1
            s = Mid(strData, inizio1, 2)
            If s = ": " Then
               strData = Mid(strData, inizio1 + 2)
               inizio1 = fine1
            Else
               mittente = mittente & Mid(strData, inizio1, 1)
            End If
           inizio1 = inizio1 + 1
         Loop
         '-------------------------
         a = 17
         b = 6
         pri = 21
         n = a * b
         z = (a - 1) * (b - 1)
         '-------------------------

         cod = ""
         messaggio = strData
         strData = ""


         For i = 1 To Len(messaggio)
            m = Asc(Mid(messaggio, i, 1))
            m = PotMod(m, pri, n)
            strData = strData & Chr(m)
         Next i
         txtOutput.Text = mittente & ": " & strData & Chr(13) & Chr(10) & txtOutput.Text '
decodifica
      End Select
   Case "Nuovo Utente"
      If None <> strData Then
         Combo1.AddItem strData
```

```
    Else
       conferma = True
    End If
    tcpClient.Tag = "Connesso"
  Case "Utente Disconnesso"
    tmp = 0
    Do
       If Combo1.List(tmp) = strData Then
          Combo1.RemoveItem (tmp)
          On Error Resume Next
          Combo1.Text = Combo1.List(0)
       End If
       tmp = tmp + 1
    Loop While tmp < Combo1.ListCount
    tcpClient.Tag = "Connesso"
  Case "Lista"
    tcpClient.Tag = "Connesso"
    nomeAtt = ""
    inizio = 1
    fine = Len(strData)
    Do While inizio <= fine
       s = Mid(strData, inizio, 1)
       If s = Chr(13) Then
          Combo1.AddItem nomeAtt
          nomeAtt = ""
       Else
          nomeAtt = nomeAtt & s
       End If
       inizio = inizio + 1
    Loop
    Combo1.Text = Combo1.List(0)
  Case "Send Message"
    If strData = "OK" Then
       'codifica
       '-------------------------
       a = 17
       b = 6
       pub = 13
       n = a * b
       z = (a - 1) * (b - 1)
       '-------------------------
       cod = ""
       messaggio = txtSend.Text
       For i = 1 To Len(messaggio)
          m = Asc(StrConv(Mid(messaggio, i, 1), 1))
          c = PotMod(m, pub, n)
          cod = cod & Chr(c)
       Next i
       '-----------------
       tcpClient.SendData Combo1.Text & Chr(13) & cod
       txtSend.Text = ""
       tcpClient.Tag = "Connesso"
```

```
        End If
    End Select
End Sub

'La funzione calcola x ^ y mod z utilizzando il metodo naive
Private Function PotMod(x As Double, y As Double, z As Double) As Double
    Dim a As Double, i As Double
    a = 1
    For i = 1 To y
        a = (a * x) Mod z
    Next i
    PotMod = a
End Function
```

File Module1.bas

```
Attribute VB_Name = "Module1"
Public None As String
Public Password As String
```

Commento

Al momento della codifica si è aggiunto un ulteriore vincolo:
I messaggi da inviare al momento della codifica RSA vengono convertiti in Maiuscolo.
Unico errore riscontrato:
Avvolte i client non ricevevano il messaggio di dis-connessine di un client , nonostante l'uso del Timer.