6.10.2 Verschlüsseln und Entschlüsseln einer Datei

Im vorliegenden Projekt können Sie eine Datei mit einem hochwertigen Verschlüsselungsalgorithmus verschlüsseln oder entschlüsseln. Es wird stets geprüft, ob das System den verwendeten Schlüssel unterstützt.

Wenn die ausgewählte Datei mit einem starken Passwort verschlüsselt wurde, dann wird die verschlüsselte Datei unter dem originalen Namen mit neuer Extension .enc im gleichen Verzeichnis abgespeichert. Die Original-Datei wird im vorgestellten Projekt nicht gelöscht.

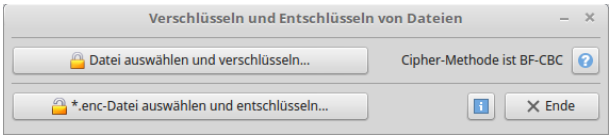

Abbildung 6.10.2.1: Hauptprogramm

Im Dialog können Sie die zu verschlüsselnde Datei auswählen:

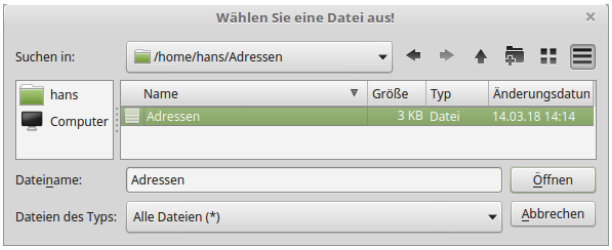

Abbildung 6.10.2.2: Datei-Auswahldialog

Im einem weiteren Dialog ist das Passwort einzugeben:

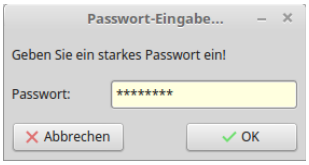

Abbildung 6.10.2.3: Passwort-Abfrage

Über den Erfolg oder Misserfolg beim Ver- oder Entschlüsseln der Datei werden Sie informiert:

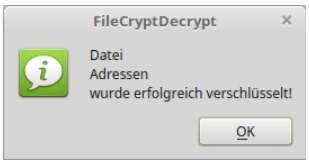

Abbildung 6.10.2.4: Erfolgsmeldung

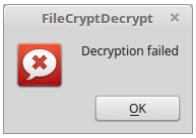

Abbildung 6.10.2.5: Fehlermeldung

Die Quelltexte werden vollständig angegeben:

```
' Gambas class file

Private sMethod As String = "BF-CBC" ' Definition of the encryption algorithm
Private sCurrentFile As String

Public Sub Form_Open()
  FMain.Caption = ("Encrypting and decrypting files")
  FMain.Resizable = False
```

```
  If Cipher.List.Find(sMethod) = -1 Then
     Message.Error(Subst("\n&1 &2 &3", ("The system does not support the Cipher-Method:"), sMethod, "!"))
     FMain.Close()
  Else
     lblMethod.Text = Subst("&1 &2", ("Cipher-Method is"), sMethod)
  Endif
End

Public Sub btnEncryptFile_Click()

  Dim sPassword, sSalt, sPath, sMessage As String

  Dialog.Title = ("Select a file!")
  Dialog.Path = Application.Dir
  If Dialog.OpenFile() Then Return

  sPassword = GetPassword() ' Password from a password dialog
  If sPassword Then
     sSalt = "123Abc#*" ' Free defined string (8 bytes)
     sPath = Dialog.Path & ".enc"
     btnEncryptFile.Text = ("Encrypt a selected file ... in progress")
       File.Save(sPath, Cipher[sMethod].EncryptSalted(File.Load(Dialog.Path), sPassword, sSalt))
       Wait 1
     btnEncryptFile.Text = ("Encrypt a selected file ...")
     sMessage = Subst("&1\n&2\n&3", ("File"), File.Name(Dialog.Path), ("has been successfully encrypted!"))
     Message.Info(sMessage)
  Else
     Message.Warning(("The password dialog was aborted by the user\nor the password is empty!"))
     Return
  Endif

  Catch
    Message.Error(Error.Text)
    btnEncryptFile.Text = ("Encrypt a selected file ...")
End

Public Sub btnDecryptFile_Click()

  Dim sPassword, sPath, sMessage As String

  Dialog.Title = ("Select a file!")
  Dialog.Path = Application.Dir
  If Dialog.OpenFile() Then Return

  sPassword = GetPassword() ' Password from a password dialog
  If sPassword Then
     If Dialog.Path Ends ".enc" Then
        sPath = Replace(Dialog.Path, ".enc", "")
     Else
        sPath = Dialog.Path
     Endif
     btnDecryptFile.Text = ("Decrypt a selected file... in progress")
       File.Save(sPath, Cipher[sMethod].DecryptSalted(File.Load(Dialog.Path), sPassword))
       Wait 1
     btnDecryptFile.Text = ("Decrypt a selected file ...")
     sMessage = Subst("&1\n&2\n&3", ("File"), File.Name(Dialog.Path), ("has been successfully decrypted!"))
     Message.Info(sMessage)
  Else
     Message.Warning(("The password dialog was aborted by the user\nor the password is empty!"))
     Return
  Endif

  Catch
    Message.Error(Error.Text)
    btnDecryptFile.Text = ("Decrypt a selected file ...")
End

Public Sub btnPWInformation_Click()
  ShowPWInformation()
End

Public Sub btnHelp_Click()
  ShowHelp()
End

Public Sub Form_Resize()
  Separator1.Width = FMain.ClientWidth
End

Private Sub ShowPWInformation()

  Dim sMessage As String
```

```
  sMessage = "<hr>"
  sMessage &= "<p>"
  sMessage &= Subst("&1 ", ("A"))
  sMessage &= "<font color='red'>" & ("strong") & "</font>"
  sMessage &= Subst(" &1", ("password contains at least:"))
  sMessage &= "<p>"
  sMessage &= ("- 8 characters,") & "<br>"
  sMessage &= ("- a small letter,") & "<br>"
  sMessage &= ("- one uppercase letter,") & "<br>"
  sMessage &= ("- one digit and") & "<br>"
  sMessage &= ("- a special character from the set {+ # _ @ ! ? § $ % *}")
  sMessage &= "<hr>"

  Message.Info(sMessage)

End

Private Sub ShowHelp()

  Dim sMessage As String

  sMessage = "<hr>"
  sMessage &= "<p>"
  sMessage &= ("The encrypted file retains its original name,\n")
  sMessage &= ("to which the extension 'enc' is appended.")
  sMessage &= "<hr><p>"
  sMessage &= ("The decrypted file gets its original name. \n")
  sMessage &= ("An existing file with the original name will be overwritten!")
  sMessage &= "<hr>"

  Message.Info(sMessage)

End

Private Function GetPassword() As String

  Dim hFormLogin As New FLogin

  Return hFormLogin(("Enter a password!")) ' Start dialog ...

End

Public Sub btnClose_Click()
  FMain.Close()
End
```

Den Quelltext des Dialogs zur Eingabe des Passwortes finden Sie in der Datei FLogin.class, deren wichtigste Prozedur _call(…) ist:

```
' Gambas class file

Public Sub _call(sMessage As String) As String
  lblMessage.Text = sMessage
  ' Returns when one of the buttons is clicked! The return value is specified in the Me.Close() call and
  ' indicates whether or not it was canceled.
  Select Me.ShowModal()
    Case 1
      Return txtPassword.Text
    Case 0
      Return ""
  End Select

End

Public Sub btnOK_Click()
  Me.Close(1)
End

Public Sub btnCancel_Click()
' 0 is also indicated by the cross in the window bar is returned - this corresponds to an abort.
  Me.Close(0)
End

Public Sub txtPassword_Activate()
  btnOK_Click() ' Me.Close(1)
End
```