6.10.1 Hash-Wert einer Datei

Im vorliegenden Projekt können Sie im Dialog eine Datei auswählen und den speziellen SHA256-Hash-Wert der Datei berechnen. Als Erweiterung können Sie den berechneten Hash-Wert mit dem bekannten Hash-Wert der ausgewählten Datei vergleichen – wenn Sie ihn kennen.

Als Beispiel wird Ihnen gezeigt, wie Sie die Integrität eines ISO-Images prüfen, dass bereits im Download-Verzeichnis liegt.

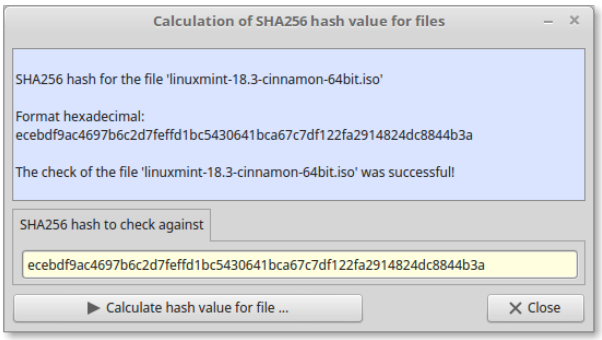

Abbildung 6.10.1.1: Prüfung Integrität ISO-Images

Für die Datei linuxmint-18.3-cinnamon-64bit.iso können Sie den Hash-Wert der Datei sha256sum.txt entnehmen, die Sie sich über den folgenden Befehl in das Download-Verzeichnis kopieren:

```
wget https://ftp.heanet.ie/mirrors/linuxmint.com/stable/18.3/sha256sum.txt -O $HOME/Downloads/sha256sum.txt
```

Diese Zeile aus dem Inhalt der Datei ist wichtig:

```
...
ecebdf9ac4697b6c2d7feffd1bc5430641bca67c7df122fa2914824dc8844b3a *linuxmint-18.3-cinnamon-64bit.iso
...
```

Tragen Sie zuerst den Hash-Wert der Version Mint 18.3 in das gelbe Textfeld ein. Wählen Sie dann im Dialog die passende ISO-Datei im Download-Verzeichnis aus. Der Vergleich von berechnetem SHA256-Hash-Wert und dem Wert von der Mint-Website in der Datei sha256sum.txt war erfolgreich!

Beachten Sie, dass die Berechnung des Hash-Wertes bei einer 2GB großen Datei beispielsweise durchaus einige Sekunden dauern kann!

6.10.1.1 Exkurs

Für die Prüfung der Authentizität liefert die gleiche Quelle mit diesem Befehl in einer Konsole:

```
wget https://ftp.heanet.ie/mirrors/linuxmint.com/stable/18.3/sha256sum.txt.gpg -O \
$HOME/Downloads/sha256sum.txt.gpg
```

die Datei sha256sum.txt.gpg. Mit folgendem Befehl wird geprüft, ob die Datei sha256sum.txt korrekt von Mint unterschrieben ist:

```
hans@mint-183 ~/Downloads $ gpg --verify sha256sum.txt.gpg sha256sum.txt
gpg: Unterschrift vom Mi 13 Dez 2017 17:16:15 CET mittels RSA-Schlüssel ID A25BAE09
gpg: Korrekte Unterschrift von »Linux Mint ISO Signing Key <root@linuxmint.com>«
gpg: WARNUNG: Dieser Schlüssel trägt keine vertrauenswürdige Signatur!
gpg:          Es gibt keinen Hinweis, daß die Signatur wirklich dem vorgeblichen Besitzer gehört.
Haupt-Fingerabdruck  = 27DE B156 44C6 B3CF 3BD7  D291 300F 846B A25B AE09
hans@mint-183 ~/Downloads $
```

Die Warnung "Es gibt keinen Hinweis, daß die Signatur wirklich dem vorgeblichen Besitzer gehört." ist berechtigt – hier die schnelle Überprüfung in einer Konsole für die o.a. angegebene ID:

```
hans@mint-183 ~/Downloads $ gpg --list-key --with-fingerprint A25BAE09
pub   4096R/A25BAE09 2016-06-07
      Schl.-Fingerabdruck = 27DE B156 44C6 B3CF 3BD7  D291 300F 846B A25B AE09
uid   Linux Mint ISO Signing Key <root@linuxmint.com>
```

6.10.1.2 Projekt – Quelltext

Der Quelltext wird vollständig angegeben:

```
' Gambas class file
' A hash process generates a number from a string in a file.

Public sOriginalText As String
Private sFilePath As String = User.Home

Public aDigestList As String[]
Public aCipherList As String[]

Public Sub Form_Open()
  FMain.Center()
  FMain.Caption = "Calculation of SHA256 hash value for files"
  FMain.Resizable = False
  txaText.Wrap = True
  DigestSupportedSHA256()
End

Public Sub DigestSupportedSHA256()
  txaText.Clear()
  If Digest.IsSupported("SHA256") = True Then
     txaText.Text = "\nThe system supports 'SHA256'!"
  Else
     txaText.Text = "\nThe system does not support 'SHA256'!"
  Endif
End

Public Sub btnDigestFromFile_Click()

  Dim sPath As String
  Dim vRawData As Variant

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

  txaText.Clear()
  Inc Application.Busy
    vRawData = Digest["SHA256"].Hash(File.Load(sPath)) ' Attention! Highly compressed source code
  Dec Application.Busy

  txaText.Clear()
  txaText.Text = "\nSHA256 hash for the file '" & File.Name(Dialog.Path) & "'\n\n"
  txaText.Text &= "Format hexadecimal:" & gb.NewLine
  txaText.Text &= StringToHex(vRawData) & gb.NewLine

  If txbCheck.Text Then
     If StringToHex(vRawData) = txbCheck.Text Then
        txaText.Text &= Subst("\n&1 '&2' &3", ("The check of the file"), File.Name(sPath), ("was success-
ful!"))
     Else
        txaText.Text &= Subst("\n&1 '&2' &3", ("The check of the file"), File.Name(sPath), ("was *not* suc-
cessful!"))
     Endif
  Endif

  Catch
    Message.Error(sPath & " not available.\n" & Error.Text)

End

Private Function StringToHex(sString As String) As String

  Dim iByte As Byte
  Dim sResult As String

  For Each iByte In Byte[].FromString(sString)
    sResult &= Lower$(Hex$(CLong(iByte), 2))
  Next

  Return sResult

End

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