

```

1: { *****
2: *
3: * Project   : CryptoBox AES Crypto Series
4: * App Name  : 258_AES_cryptobox2.txt, loc's = 388
5: * Purpose   : crypt & decrypt a file with AES symmetric keys
6:               : example for tutorial 13
7: * History   : system for mX3.2, internal random generator, newdir
8:               : 220_CipherBox_form2.txt, loc's = 312
9:               : LockBox3 implemented for more accurate Standard
10:              : SHA256 integration
11: * ToDo      : check if key or password is loaded!, rename encrypted file to orig
12:              : code a better password dialog to reset the password
13: *****}
14:
15: Program CipherBox3_FORM_Lab;
16:
17: const LEFTBASE = 20;
18:       TOPBASE  = 25;
19:       MEDIAPATH = 'exercices\';
20:       KEYPATH   = 'crypt\';
21:       CIPHERLOG = 'cipherbox_log2.txt';
22:
23: var
24:   mpan: TPanel;
25:   mPlayer: TMediaPlayer; //for future use
26:   inFrm: TForm;
27:   clstbox: TListbox;
28:   stat: TStatusbar;
29:   idx: integer;
30:   selectFile, cryptlog, AESpasswd: string;
31:
32:
33: function MSecToTime(mSec: Int64): string;
34: var dt: TDateTime;
35: begin
36:   dt := (mSec/1000/86400); //SecsPerDay;
37:   Result := FormatDateTime('" song length is:" nn:ss:zzz',dt);
38: end;
39:
40: procedure OpenCD(self: TObject);
41: begin
42:   OpenCDDrive;
43: end;
44:
45:
46: //*****Event Handler*****
47: procedure CloseCryptClick(Sender: TObject; var action: TCloseAction);
48: begin
49:   //Form1.Close; free 308 kByte
50:   if MessageDlg('Wanna Leave Crypt?',mtConfirmation,[mbYes, mbNo],0) = mrYes then begin
51:     //inFrm.Free;
52:     MPlayer.Close;
53:     MPlayer.Free;
54:     action := caFree;
55:   end else
56:     Action := caNone;
57: end;
58:
59: procedure CryptoButtonCloseClick(sender: TObject);
60: begin
61:   //mbitmap.Free;
62:   inFrm.Close;
63: end;
64:
65: //AES CryptoBox
66: procedure BtnGetPassword(Sender: TObject);
67: begin
68:   AESpasswd := InputBox('AES CryptoBox Password' , 'Enter Password:', 'password');
69:   clstBox.Items.Add('Password is set! at: '+DateTimeToStr(Now));
70:   {with TPasswordDlg.Create(NIL) do begin
71:     showmodal;
72:     if modalresult = mrOK then begin
73:       AESpasswd := password.Text;
74:       with clstBox.Items do begin
75:         Add('Password is set! at: '+DateTimeToStr(Now));
76:       end;
77:     end else begin
78:       //if cancelbtn.modalresult = mrCancel then begin
79:         password.Text := '';
80:         AESpasswd := '';
81:         clstBox.Items.Add('Password empty reset at: '+DateTimeToStr(Now));
82:         //showmessage(AESpasswd); debug
83:       end;
84:     Free;
85:   end;
86:

```

```

87:
88: procedure EncryptMediaAES(sender: TObject);
89: begin
90:   if AESpasswd <> '' then begin
91:     if PromptForFileName(selectFile, 'Files (*.*)|*.*', //others
92:       '', 'Select your file to crypt', MEDIAPATH, False)
93:     then begin
94:       //Display this full file/path value
95:       mpan.font.color:= clyellow;
96:       mpan.font.size:= 18;
97:       mpan.caption:= ExtractFileName(selectFile)+' encrypt...';
98:       Application.ProcessMessages;
99:       Screen.Cursor:= crHourglass;
100:      with clstBox.Items do begin
101:        Add('SHA256 Hash of: '+ExtractFileName(selectFile));
102:        Add('is: '+ComputeSHA256(selectFile, 'F'));
103:        idx:= Add('File to Crypt: '+ExtractFileName(selectFile));
104:        stat.SimpleText:= clstBox.items[idx];
105:      end;
106:      with TStopwatch.Create do begin
107:        Start;
108:        AESSymmetricExecute(selectFile, selectFile+'_encrypt', AESpasswd);
109:        mpan.font.color:= clblue;
110:        mpan.font.size:= 30;
111:        mpan.caption:= 'File Encrypted!';
112:        Screen.Cursor:= crDefault;
113:        Stop;
114:        clstBox.Items.Add('Time consuming: ' +GetValueStr + ' of: ' +
115:          inttoStr(getFileSize(selectFile))+' File Size');
116:        Free;
117:      end;
118:      clstbox.Items.Add('Crypted file: '+ExtractFileName(selectFile)+'_encrypt');
119:      WriteLog(cryptLog, clstbox.items.text)
120:    end;
121:    end else Showmessage('Set your password first!');
122:  end;
123:
124:
125: procedure DecryptMediaAES(sender: TObject);
126: var clearout: string;
127: begin
128:   if AESpasswd <> '' then begin
129:     if PromptForFileName(selectFile, 'Files (*.encrypt)|*.encrypt', //others
130:       '', 'Select your file to decrypt', MEDIAPATH, False)
131:     then begin
132:       // Display this full file/path value
133:       mpan.font.color:= clyellow;
134:       mpan.font.size:= 18;
135:       mpan.caption:= ExtractFileName(selectFile)+' decrypt...';
136:       idx:= clstbox.Items.Add('File to Decrypt: '+ExtractFileName(selectFile));
137:       stat.SimpleText:= clstbox.Items[idx];
138:       Application.ProcessMessages;
139:       Screen.Cursor:= crHourglass;
140:       with TStopwatch.Create do begin
141:        Start;
142:        AESDecryptFile(selectFile+'_decrypt', selectFile, AESpasswd);
143:        clearout:= selectFile+'_decrypt';
144:        Delete(clearout, length(clearout)-15, 8); //-7!
145:        RenameFile(selectFile+'_decrypt', clearout);
146:        Screen.Cursor:= crDefault;
147:        Stop;
148:        clstBox.Items.Add('Time consuming: ' +GetValueStr + ' of: ' +
149:          intToStr(getFileSize(clearout))+' File Size');
150:        Free;
151:      end;
152:      mpan.font.color:= clGreen; //clLime
153:      mpan.font.size:= 30;
154:      mpan.caption:= 'File Decrypted!';
155:      with clstBox.Items do begin
156:        Add('SHA256 Hash of '+ExtractFileName(clearout));
157:        Add('is: '+ComputeSHA256(clearout, 'F'));
158:        Add('File Decrypted as: '+ExtractFileName(clearout));
159:      end;
160:      WriteLog(cryptLog, clstbox.items.text)
161:    end;
162:    end else Showmessage('Set your password first!');
163:  end;
164:
165: //*****Form Creator*****
166: procedure SetCryptFormAES;
167: var
168:   mi, mi1, mi2: TMenuItem;
169:   mt: TMainMenu;
170:   mlbl, mlbl1: TLabel;
171: begin
172:   inFrm:= TForm.Create(self);

```

```

173: mLbl:= TLabel.create(inFrm);
174: mLbl1:= TLabel.create(inFrm);
175: mpan:= TPanel.Create(inFrm);
176: stat:= TStatusbar.Create(inFrm);
177: cLstbox:= TListbox.create(inFrm);
178:
179: with inFrm do begin
180:   caption:= '*****CryptoBox3 AES256 and SHA256*****';
181:   height:= 610;
182:   width:= 1180;
183:   //color:= clred;
184:   Position:= poScreenCenter;
185:   onClose:= @CloseCryptClick;
186:   Show;
187: end;
188: with mpan do begin
189:   caption:= '****CryptoBox_AES256_SHA256****';
190:   parent:= inFrm;
191:   SetBounds(LEFTBASE+10,TOPBASE+40,416,400);
192:   color:= clsilver;
193:   font.color:= clyellow;
194:   font.size:= 20;
195:   Show;
196: end;
197: with cLstbox do begin
198:   parent:= inFrm;
199:   SetBounds(LEFTBASE+450, TOPBASE+40, 670, 400);
200:   font.size:= 10;
201:   color:= clyellow;
202:   //onClick:= ChangeKey;
203: end;
204: with TBitBtn.Create(inFrm) do begin
205:   Parent:= inFrm;
206:   setBounds(LEFTBASE+490,TOPBASE+460,150,40);
207:   caption:= '&Close';
208:   font.size:= 12;
209:   glyph.LoadFromResourceName(HINSTANCE,'CL_MPSTOP');
210:   onclick:= @CryptoButtonCloseClick; //CloseCryptClick;
211: end;
212: with TBitBtn.Create(inFrm) do begin
213:   Parent:= inFrm;
214:   setbounds(LEFTBASE+ 650, TOPBASE+ 460,150, 40);
215:   caption:= 'File to &Encrypt';
216:   font.size:= 12;
217:   //Font.Style:= [fsBold];
218:   glyph.LoadFromResourceName(getHINSTANCE,'CL_MPNEXT');
219:   onclick:= @EncryptMediaAES;
220: end;
221: with TBitBtn.Create(inFrm) do begin
222:   Parent:= inFrm;
223:   setbounds(LEFTBASE+ 810, TOPBASE+460,150, 40);
224:   caption:= 'File to &Decrypt';
225:   font.size:= 12;
226:   glyph.LoadFromResourceName(getHINSTANCE,'CL_MPPLAY');
227:   onclick:= @DecryptMediaAES;
228: end;
229: with TBitBtn.Create(inFrm) do begin
230:   Parent:= inFrm;
231:   setbounds(LEFTBASE+ 970, TOPBASE+460,150, 40);
232:   caption:= 'Enter &Password!';
233:   font.size:= 12;
234:   glyph.LoadFromResourceName(getHINSTANCE,'CL_MPPAUSE');
235:   //event handler
236:   onclick:= @BtngetPassword;
237: end;
238: with mlbl do begin
239:   parent:= inFrm;
240:   setbounds(LEFTBASE+15,TOPBASE-15,180,20);
241:   font.size:= 32;
242:   font.color:= clred;
243:   font.style:= [fsunderline];
244:   caption:= 'AES CryptoBox';
245: end;
246: with mlbl1 do begin
247:   parent:= inFrm;
248:   setbounds(LEFTBASE+455,TOPBASE-1,180,20);
249:   font.size:= 20;
250:   font.color:= clred;
251:   caption:= 'Log Register: ';
252: end;
253: mt:= TMainMenu.Create(infrm);
254: with mt do begin
255:   //parent:= frmMon;
256: end;
257: mi:= TMenuItem.Create(mt);
258: mil:= TMenuItem.Create(mt);

```

```

259:   mi2:= TMenuItem.Create(mt);
260:   with mi do begin
261:     //parent:= frmMon;
262:     Caption:='Encrypt File';
263:     Name:='ITEM';
264:     mt.Items.Add(mi);
265:     OnClick:= @EncryptMediaAES;
266:   end;
267:   with mil do begin
268:     Caption:='Decrypt File';
269:     Name:='ITEM2';
270:     mt.Items.Add(mil);
271:     OnClick:= @DecryptMediaAES;
272:   end;
273:   with mi2 do begin
274:     Caption:='Set Password';
275:     Name:='ITEM3';
276:     mt.Items.Add(mi2);
277:     OnClick:= @BtngetPassword;
278:   end;
279:   with Stat do begin
280:     parent:= inFrm;
281:     stat.SimplePanel:= true;
282:   end;
283:   MPlayer:= TMediaPlayer.create(self);
284:   with MPlayer do begin
285:     parent:= inFrm;
286:     height:= 38;
287:     top:= TOPBASE + 460;
288:     left:= LEFTBASE+ 10;
289:     Display:= mpan; //for video show
290:   end;
291:   if DirectoryExists(ExePath+KEYPATH) = false then
292:     CreateDir(ExePath + KEYPATH);
293:   cLstbox.Items.Add('Welcome to CryptoBox3 protecting your Data '+datetimetToStr(Now));
294:   idx:= cLstbox.Items.Add('Keypath is: '+ExePath+KEYPATH);
295:   stat.SimpleText:= cLstbox.Items[idx];
296:   cryptLog:= ExePath+CIPHERLOG;
297:   cLstbox.Items.Add('CryptoBox3 ready, Save or load a Password protected Cipher');
298: end;
299:
300: //main app
301: begin
302:   SetCryptFormAES;
303:   if DirectoryExists(ExePath+KEYPATH) = false then
304:     CreateDir(ExePath+ KEYPATH);
305:   cryptLog:= ExePath+ CIPHERLOG;
306:   writeln('Machine: '+GetHostName)
307:   writeln('User: '+GetUserName)
308:   writeln('Instance: '+intToStr(getHinstance))
309:   writeln('ProcId: '+intToStr(getProcessid))
310:   writeln('Processors: '+intToStr(GetNumberOfProcessors))
311:   WriteLog(cryptLog, memo2.text)
312: End.
313:
314: //-----
315:
316: cry-crypt-crypto-cryptologic
317: -----
318: Programmers never die, they just GOSUB without RETURN

```