

安全的ACCE 加密方法 PDF转换可能丢失图片或格式，建议
阅读原文

https://www.100test.com/kao_ti2020/251/2021_2022__E5_AE_89_E5_85_A8_E7_9A_84A_c97_251711.htm Microsoft的ACCESS数据库，是我们常用的桌面数据之一，大多中小企业的数据库管理系统都可以采用它，但其安全性一直令人担忧，试想，一套财务管理系统，用户直接打开数据库去更改数据，后果会如何？有些系统对ACCESS数据库可能只是更改扩展名，或加个密码，众所周知，破解ACCESS密码的方法和工具网上多的是！所以这样的加密一样令人担忧，下面介绍一个简单的方法，实现ACCESS数据的加密，供大家参考。用UltraEdit打开MDB文件可以看到，文件前16个字节的内容：00 01 00 00 53 74 61 6E 64 61 72 64 20 4A 65 74 现在随便更改几个，再用ACCESS打开，发现出现不同识别的文件格式错误，因为ACCESS前面保存的信息都是一些MDB文件的定义和口令，如果更改这些内容，别人就很难看出这个数据库的格式，无法打开它了，而且这样不会对数据库的内容作更改，不会破坏原有的数据。下面就用Delphi作个简单的加密解程序：用到的加密解函数如下：
const titlestr:array[0..15] of byte= (\$00,\$01,\$00,\$00,\$53,\$74,\$61,\$6E,\$64,\$61,\$72,\$64,\$20,\$4A,\$65,\$74) .//对应MDB文件的前16个字节 titlestr2:array[0..15] of byte= (\$48,\$4A,\$00,\$58,\$55,\$43,\$48,\$41,\$4E,\$47,\$59,\$4F,\$55,\$00,\$20,\$20) .//更改后的MDB文件的前16个字节，自己随便写吧，如写上自己公司的简称或自己的名
produce
EncrypMDB(filename:string). //用titlestr2内容替换MDB前16个字节，以便实现加密的作用
var F:TFileStream. begin if not

fileExists(filename) then exit.

```
F:=TFileStream.create(filename,fmopenwrite). try
```

```
F.seek($00,soFromBeginning). F.Write(titlestr2,16). finally F.free.
```

```
end. end. produce uncrpMDB(filename:string). //还原MDB前16  
个字节 var F:TFileStream. begin if not fileExists(filename) then exit.
```

```
F:=TFileStream.create(filename,fmopenwrite). try
```

```
F.seek($00,soFromBeginning). F.Write(titlestr,16). finally F.free.
```

```
end. end. 我们知道打开ACCESS数据库后会出现一个锁定文  
件(.ldb文件),因为我们自己也要使用数据库,所以必须在使用  
时还原数据库。如果还原后没有进行加密的话,用户同样可  
以复制MDB文件,然后用ACCESS或其它工具打开它,所以  
应该在数据打开前后都处于加密状态才能保证数据的安全。
```

```
用Delphi采用ADO连接数据库用以下方法可以实现: //还原
```

```
数据,以便自己使用数据库 copyfile(pchar(APP_path
```

```
\data\account.db),pchar(app_path data\temp.db),false). //app_path
```

```
表示程序的当前目录, account.db是个更改了扩展名的MDB文  
件 uncrpMDB(App_path data\temp.db).
```

```
copyfile(pchar(App_path data\temp.db),pchar(APP_path  
\data\account.db),false).
```

```
adoconn.connectionstring:=provider=Microsoft.Jet.OLEDB.4.0.Dat  
a Source= App_path data\account.db.Persist Security Info=false.
```

```
//adocon是个TADOConnection组件 try
```

```
adoconn.connected:=true. except MessageBox(handle,打开数据库  
出现致命的错误!!!,错误,MB_OK MB_ICONERROR). end.
```

```
//打开后马上对其加密 copyfile(pchar(APP_path
```

```
\data\account.db),pchar(app_path data\temp.db),false). //app_path
```

表示程序的当前目录，account.db是个更改了扩展名的MDB文件 EncrypMDB(App_path data\temp.db).

```
copyfile(pchar(App_path data\temp.db),pchar(APP_path  
\data\account.db),false). 0deletefile(App_path data\temp.db).
```

上面使用了两次临时文件，是因为数据库打开后再对MDB进行直接的写入会出现问题，而且你无法去确定多少个用户打开了程序。整个程序共用一个TADOConnection，只在打开数据库连接的时候还原MDB文件，其它时间MDB文件一直都处于加密状态！用户复制了MDB文件一般很难知道它是什么！打开数据库后会有一个.ldb文件，类型会出现ACCESS等字样，如果你不想让人看出是什么的话就修改注册表吧,如:

```
reg:=TRegistry.Create. try reg.RootKey:=HKEY_CLASSES_ROOT.  
reg.OpenKey(.ldb). reg.WriteString(,tempfile). finally reg.closekey.  
reg.free. end.
```

这样用户看到的文件类型是tempfile 100Test 下载
频道开通，各类考试题目直接下载。详细请访问
www.100test.com