DBASEⅢ软件信息保护技术

合肥市统计局计算站 葛晓滨

摘要：本文详述了在 *DBASE*Ⅲ程序中实现信息保护的一种实用方法，该方法不同于在外部环境中对 *DBASE*Ⅲ应用软件所采取的保护方法，而是直接在 *DBASE*Ⅲ系统环境下实现，因而易于使用并具有一定的方便性和实用性。

1.开发背景

软件信息的保护是软件版权保护的一个重要方面，它包括软件的版权信息、版本号、软件作者、软件名称等诸多内容。目前，在普遍使用的解释型 DBASEⅢ软件中，由于受软件执行方式的限制，应用程序中的源代码对用户是透明的，使得他人很容易非法篡改软件原有的信息，从而侵犯了软件作者的合法权益。这就需要我们在软件中对这些重要的软件信息采取保护措施，提高软件自身的防护能力。

2.方案选择

从根本上讲，对解释型的 DBASEⅢ应用软件采取的加密保护措施终究是不可靠的，而比较稳妥的方法是改变 DBASEⅢ应用软件的执行环境，采用与解释型 DBASEⅢ相兼容的编译型 DBASEⅢ，如 CLIPPER 等。这样通过 DBASEⅢ编译器对应用软件的源程序进行编译，产生可执行的二进制代码文件，就可以使应用软件的源程序及其软件信息不对用户透明，从而在一定程度上实现对软件信息的保密。如果用户对 DEBUG 或 PCTOOLS 等工具软件比较熟悉，仍可借助这些工具找到软件中的软件信息并加以篡改，这是因为虽然经过 DBASEⅢ编译器的加工，应用软件的源程序已变成二进制可执行文件，但软件中的文字信息仍是以 ASCⅡ码形式存放在可执行文件中的，用工具软件的字符搜索功能对软件信息进行查找和修改还是比较容易做到的。为此需要我们进一步对软件信息采取保护措施。

笔者经过研究和试验，采用加密保护同校验保护方式相结合的方法对软件信息进行保护，取得了较好的使用效果。实践证明，这种方法具有使被保护的软件信息不易篡改，实现方式简单，并且无需借助外部环境，在 DBASEⅢ应用软件的内部就可以实现等特点。

3.基本原理

加密校验保护方法是基于以下两种方法的结合：

(1)信息加密保护方法：它是将软件信息通过某种变换方式(加密算法)加工为不易被他人所理解和密文信息，通过这种变换，明文形式的软件信息转换为密文信息，使得工具软件无法对其直接进行修改等操作。

(2)信息校验保护方法：这种方法是通过提取校验信息，将之与预先设置的信息模型进行对比检验，如果结果是相同的，则证明所校验的信息是完整的，否则，信息已被篡改，据此可以采取相应的处理措施。

这两种方法的结合使用，增强了软件信息的保密性和坚固性。

4.实施方法

具体而言，实现软件信息保护的过程是：

首先编写一个加密程序，并采用两种不同的加密算法将软件的明文信息变换为两种不同形式的密文。一个用在应用程序中，供程序使用，称为“使用密文”；另一个用作信息校验使用，称为“校验密文”。然后，在编写应用程序时，读入“使用密文”及“校验密文”，并随之编写密文的还原解密算法，将“使用密文”解密，并将解密结果存放到某个设定的应用量中，以后程序可通过调用该变量获得软件的明文信息。通过这种变换，程序中便不再有可直接识别的软件信息，用 DEBUG 或 PCTOOLS 等工具软件已无法直接找到这些软件信息了。其次，为防止他人使用工具软件对软件信息进行动态跟踪调试以篡改其信息内容，我们在随后的软件程序中设置校验点，对所使用的软件信息进行检测，此即通过对比解密后的“使用密文”和“校验密文”内容，检测“使用密文”的完整性。这

© 中国科学院软件研究所 http://www.c-s-a.org.cn

种校验点可以在应用软件中设置多处，以加强对软件信息的保密效果。

5.实现步骤

我们通过具体的实例说明这种加密校验保护法：

第一步：加密软件信息

编写加密程序 JM・PRG，其源代码如下：

```
* 软件信息加密程序
SET TALK OFF
SET SAFE OFF
CLEAR
S=SPACE(78)
@4,20 SAY"加密程序：本程序产生密文文件MW・TXT。"
@5,20 SAY"其中变量DATAONE存放的是使用密文。"
@6,20 SAY"变量DATATWO存放的是校验密文。"
@7,20 SAY"请输入您需要加密的软件明文信息："
@8,20 GETS
READ
D1=SPACE(0)
D2=SPACE(0)
KEY=2
JS=1
DO WHILE JS<=LEN(TRIM(S))
  D1=D1+STR(ASC(SUBSTR(S,JS,1))+KEY,3)
  D2=D2+STR(ASC(SUBSTR(S,JS,1))*KEY,3)
  JS=JS+1
ENDDO
SET ALTE TO MW・TXT
SET AITE ON
?"DATAONE='&D1'"
?"DATATWO='&D2'"
SET ALTE OFF
SET ALTE TO
SET SAFE ON
SET TALK ON
CLEAR
RETU
```

随时运行这个通用的加密程序 JM・PRG，我们即可获得加密的软件信息，该信息存放在文件名为"MW・TXT"的文件中。

第二步：设置使用密文

在编写 DBASEⅢ应用软件时，我们可以利用文字编辑软件如 DBASEⅢ的字处理或 Word Star 等的文件读入功能(Ctrl－KR 命令)将"MW・TXT"文件读入当前编辑的源程序中，随后在源程序中安装对使用密文的还原解密程序，如下所示：

```
PUBL DATAONE,DATATWO
DATAONE='……'
DATATWO='……'
  …
D1=SPACE(0)
KEY=2
JS=1
DO WHILE JS<=LEN(DATAONE)/3
  D1=D1+CHR(VAL(SUBS(DATAONE,JS*3-2,3))
-KEY)
  JS=JS+1
ENDDO
…
```

以后编程者便可以在应用程序中直接使用变量 D1 输出软件信息。

第三步：软件信息校验

为进一步保证软件信息不被篡改，我们可以在完成上述第二步工作后，在软件中设置对软件信息的校验点，以检验其完整性。首先，我们设置一个检验结果变量"ERR"，并用"ERR=・F・"语句将其初值赋为逻辑值"・F・"，再使用"DO CHECK WITH D1,ERR"语句在校验点调入如下的检测程序 CHECK・PRG：

```
* 密文校验程序：CHECK・PRG
SET EXAC ON
D2=SPACE(0)
KEY=2
JS=1
DO WHILE JS<=LEN(DATATWO)/3
  D2=D2+CHR(VAL(SUBS(DATATWO,JS*3-2,3))
      /KEY)
  JS=JS+1
ENDDO
IF D1#D2
  ERR=・T・
ENDIF
SET EXAC OFF
RETU
```

这样就可以通过运行 CHECK・PRG 程序后返回的"ERR"变量值判定软件信息的完整性：返回值是"F・"表明信息是完整的，否则为"・F・"表明已信息被篡改过，可以在随后的应用软件中采取措施给随意篡改软件信息的用户以警告。

6.结束语

上述方法和步骤，我们在 DBASEⅢ应用软件内部实现了对软件信息的保护。需要说明的是：本文所采取的加密算法较为简单，笔者只是作些原理性的介绍，有关的加密算法还可以编写得更复杂些，读者可以通过借鉴本文的方法，编写出符合自身情况和特点的加密算法。▲

© 中国科学院软件研究所 http://www.c-s-a.org.cn