硬盘数据保密的探讨

江西拖拉机发动机厂 黄焕如

摘要：本文分析了在硬盘上设置口令、封锁硬盘和子目录等有效的加密方法的可能性及实现的方法，并提出在IBM PC机硬盘上一些数据保密的实例，供用户在使用硬盘资源时进行数据保密的选择和参考使用。

1.问题的提出

随着计算机的广泛使用，硬盘的配置也朝着大容量发展，硬盘的数据保密已提到重要的日程上来。

由于硬盘较软盘不易保护和管理。因此人们想出种种办法来保护硬盘上的数据，使其不受破坏或窃取。一般来说硬盘数据的保密可能通过硬件和软件两种办法来解决。所谓硬件设置主要在硬盘控制器选件板上ROM中修改硬盘自举引导系统，使其为用户的保密需要服务。如最近推出的计算机病毒免疫硬卡，就同时含有为硬盘设置口令的功能。相对来说通过软件实现硬盘数据保密比较简单，投资小、见效快，一般条件下均能实现，当然就可靠性来说就不如硬件设置了。本文讨论的是通过软件设置来实现硬盘数据保密的方法。

2.设置口令

如果把保险柜比作硬盘，密码锁是打开保险柜的关键，是那么在进入硬盘前设置口令就相当于给硬盘加了一把密码锁。因此不少人想办法为硬盘设置口令，将口令安排在COMMAND·COM文件内，让DOS执行该文件之前校核口令是否正确，来决定程序退出或继续运行。

笔者认为，口令字设置在COMMAND·COM文件中，对于用户来说透明度太大了，这对于稍微熟悉DOS系统的人利用DEBUG·COM来查阅口令是不困难的。如果将口令字通过某种变换设置在硬盘的某一位置，设置或修改口令字的并不交给用户，即使熟练的用户能找到口令字所在的地址，也只不过是一些已经伪装的乱七八糟的字符而已。

选择合适的位置是很重要的。该地址既要没有可能被其他程序或文件复盖，又不能影响程序的正常运行。

大家知道，PC—DOS的FORMAT·COM命令在格式化硬盘的同时，也建立了DOS的引导记录，该引导记录的前32字节信息，记录了各个硬盘介质特性的物理参数，其中：

00H～02H （3字节）转移引导程度的JMP指令
03H～0AH （8字节）制造商名和版本号
0BH～17H （13字节）BIOS参数块BPB
18H～19H （2字节）每个道(柱)的扇区数
1AH～1BH （2字节）磁头(面)数
1CH～1DH （2字节）隐藏扇区个数
1EH～1FH （2字节）引导盘标志

将上述非DOS占用扇区中各项逐一加以分析，不难发现在引导记录中的第03H～0AH制造商名和版本号这8个字节，相对来说是无关紧要的。利用其全部或部分地址为口令字提供设置地点是一个好办法。

从理论上来说，口令字的位数越多破译越困难，但位数多了输入嫌不方便。为了便于说明清楚，以下程序以3位数口令为例，用来伪装口令字而采用字符变换的公式选择也较简单。

（详见程序1和程序2）

程序1是用来设置或修改口令字的，第一次设置口令字必须首先使用它，该程序一般并不交给用户。而程序2即对DOS2·00版COMMAND·COM文件的修改，至于其他版本，只需在最后一条转移语句稍作改动，并修改相应的地址和CX寄存器。示例中口令字的输入采用不透明的全封闭的显示输入，尽管口令字仅3位数的宽度，允许输入5次，故总共可输入字符15个，如果用户不知道位数宽度试着破译口令，就会更困难。

考虑到一般用户手中能轻而易举地搞到相应的DOS版本软盘，为防止利用软盘启动或重新拷入COMMAND·COM文件复盖已修改的文件，来执行

硬盘中的程序。可以采用一些其他办法增加其执行指定程度的难度。例如修改自动批处理文件，结合隐藏某些子目录和文件的办法。

文件名为 AUTOEXEC·BAT 文件是一个特殊的批文件，每次启动 DOS 时，命令处理程序就在 DOS 盘根目录上寻找该文件并执行之。显然该文件是第一个执行的批命令文件，如果将该文件改名，（例如改为 HHR·DBF）让其混杂在某一子目录（例如＼DOS）内众多的·DBF 文件中，甚至将该文件的内容应当是进入重要程度之前必须执行的文件，这就要修改 COMMAND·COM 文件。

步骤一继续修改 COMMAND·COM 文件。

```
C>DEBUG COMMAND·COM
-RCX
CX 4630
:
-S100 L4630 'AUTOEXEC·BAT'
-*****:1077
-E1077'DOS HHR·DBF'
-W
```

可根据自己计算机硬盘上的文件分布的实际情况，自行选择子目录和文件名，以增加该文件的隐蔽性。同时也要注意，修改的字符总长度不应超过原字符总长度，以免破坏 COMMAND·COM 文件的其他内容。

步骤二：设置"假"的自动批文件

所谓"假"的自动批文件，就是一旦用户从软盘拷入相应 DOS 系统的 COMMAND·COM 文件复盖已修改的文件，重新启动计算机后执行的自动批处理文件。该文件可以在表面上于"真"的批文件一样，在某一关键地方设置障碍，以便引导其执行错误的程序，也可使其陷入死机状态。例如：

```
C>TYPE AUTOEXEC·BAT
 ECHO OFF
 DEL COMMAND·COM
 TYPE<文件名>
```

TYPE 后面的文件可任选一后缀为 COM 或 EXE 的文件，以在屏幕上出现一些莫名其妙的字符，并且死机或出现一些不正常的现象。

当然，口令字设置的地址可灵活选择，还可设在 BOOT 中的数据区，例如将 Non-System disk or……中的 S 改为小写的 s，d 改为大写的 D 等。值得注意的是，对于一些执行外部文件需在内部重新调用 COMMAND·COM 文件的软件，例如 DBASEⅢ、DBASEⅢPLUS，当使用 RUN 或！命令时，需要重新输入口令字。这在一定的程度上又增加了程序的保密性。

3.设置"硬锁"

在 COMMAND·COM 文件中设置口令仅仅为硬盘加了把"软锁"，因为熟悉 DOS 系统和硬盘结构的用户，仍然可利用 DEBUG 等工具软件查找口令或修改程序使其不校核口令字。

当计算机启动引导程序时，将检查硬盘的 0 磁道 1 扇区最后两字节是否为 55、AA，如果不是则认为硬盘无效，转入 ROM BASIC）有的机型则作死机处理），即便利用软盘启动，也无法使用硬盘，系统将认为是无效的驱动器符。这样就好象给硬盘加了把"硬锁"。

程序 3 可用字处理软件编辑，然后用宏汇编软件编译和连接，最后转成 COM 文件即可。（详见程序 3）

```
C>MASM HARDJM;
C>LINK HARDJM;
C>EXE2 BIN HARDJM·EXE HARDJM·COM
```

使用时请带选择参数，例如：

```
C>HARDJM 1（给硬盘加锁）
C>HARDJM 2（给硬盘解锁）
```

请注意：给硬盘加锁或解锁后，必须重新启动机器方能生效。

4.子目录加密

为硬盘加"软锁"或"硬锁"，实际上对全部硬盘资源进行保密。由于条件的限制（主要指计算机数量有限）多人使用同一台计算机的硬盘是很普遍的，因此人们常常将需要保密的文件和数据存入一个子目录或下一级子目录内，然后隐藏这些子目录。改成隐藏属性的子目录排除了 DIR 或 TREE 等命令的搜寻，但又可正常执行进入子目录内文件。这种方法加密的效果十分有限，利用 PCTOOLS 或 DEBUG 等软件很容易破译。

无论是硬盘或软盘，描述一个子目录或文件，在目录区内占 32 字节，其内容为：

0～10 字节文件名（含扩展名）

11 属性

12～21 DOS 保留区

22～23 时间

24～25 日期

26～27 起始簇号

28～31 文件长度

其中起始簇号指向 FAT 表的指示器，用来确定子目录或文件存放于磁盘的地址。如果将其数值加以改变，相应的子目录进不去，文件也无法执行。为了防止意外事故丢失正确的起始簇号，应该把它存放在较安全的地方。根据加密的要求可分别采取以下方法：

(1)将正确的起始簇号存入引导扇区中(如出版商和版本号，即 OCM 标志处)，然后在该位置填入 0 或其他不正确的数据。

(2)将正确的起始簇号存入本目录的 DOS 保留区，然后在该位置填入 0 或其他不正确的数据。

(3)将起始簇号的高、低字节对换。

(4)将起始簇号和另一子目录或文件的起始簇号对换，或者将正确的起始簇号存入它处，填入另一子目录的起始簇号。

利用 1 ～3 方法加密后，不能进入子目录或执行该文件；利用 4 方法加密后，当执行进入该子目录或文件时，却进入另一子目录或执行另一文件，同样起到了保护作用。

为了加强保密效果，加密时可将起始簇号进行某些运算后再存放，解密时将已经被变换的数据进行相应的逆运算后再填入。这样，即使存放的起始簇号被发现，也不是正确的数据。当然选择运算公式要慎重，如果数据变化使其产生溢出，可能会产生起始簇号无法还原的后果。

为了编制程序简单易行，不考虑采用修改 DOS 功能调用或读目录进内存再比较目录匹配以获得起始簇号的方法，而利用 DOS 中断 25、26 来直接读写磁盘，这对于仅仅保护少数几个子目录还是相当方便的。INT25 和 INT26 中断所需要的入口参数中，必须取得读写扇区的逻辑号。如果待加密的子目录或文件在根目录区，10M 或 20M 硬盘其值范围在 11～30 扇区。(均为十六进制)如果待加密的是下一级子目录或其内的文件，则必须根据根目录中子目录的起始簇号计算：

逻辑扇区号＝(起始簇号－2)×每簇扇区数+文件区起始区号

对于 10M 硬盘来说，每簇扇区数＝8，文件区起始区号＝31h，故逻辑扇区号＝起始簇号×8＋21h。

下面以方法 2 为例，举一具体例子说明如下：(在 IBM PC 及兼容机上 10M、DOS 2．00 版本下通过)

硬盘根上当内有一子目录 DOS，现要保护该子目录内部文件，即对 DOS 子目录加密。

首先利用 DEBUG 或 PCTOOLS 查找 DOS 的逻辑扇区号：

```
C>DEBUG
-100 2 11 5
-D160 17F
476 B:0160 44 4 F 53 20 20 20 20 20-20 20 20 10 00 00 00 00
      DOS·····
476 B:0170 00 00 00 00 00 00 1 B 00-21 00 0 E 00 00 00 00 00···
      ·····
```

DOS 子目录在第 11h 扇区第 60～7 Fh 字节上，采用起始簇号(7 A～7 B)和 DOS 保留区的一部分(70～71)数据对换的方法。

然后编制程序如下：(程序 4 附后)

使用该程序时请注意，执行第一次为该子目录加密，执行第二次为该子目录解密。

5.后记

在示例程序 1 和程序 2 中，为了增加破译的难度，可使用来伪装口令字的字符变换公式更复杂一些。为了加强数据和文件的保密性，可将已经放入“真”的自动批文件和其他文件、数据的子目录隐蔽的更深一些，并结合为子目录加密的方法一起实行，加密的效果可能更好。

在编制程序 4 之前，必须先找到该子目录的逻辑扇区号。如果将该程序修改为可带参数的通用加密的程序(即把查找某一子上当的逻辑扇区号交给程序本身完成)，可以加密硬盘的任意下一级子目录，由于各种硬盘的有关参数不一样，任意下一级子目录的逻辑扇区号的计算公式不同，程序编制较为复杂，限于编幅不再详叙，有兴趣的同行可参照该程序加以修改。以上程序均在 IMB PC 或 IBM 286 机上，DOS 2．0 下通过。

参考文献：

(1)张福炎 蒋新儿 李滨宇 微型计算机 *IBM PC*

的原理与应用南京大学出版社

（2）张载鸿局部网操作系统 DOS 高级技术析，国防工业出版社

程序 1 JTXZ · COM

```
C>DEBUG
-A100
0100    MOV     CX,0000
0103    NOV     DX,2479
0106    MOV     BH,07
0108    MOV     AC,0600
010 B   INT     10;设置屏幕
010 D   MOV     BH,00
010 F   MOV     DX,061 A
0112    MOV     AH,02
0114    INT         10
0116    MOV         AL,02
0118    MOV     CX,0001
011 B   MOV     DX,0000
011 E   MOV     BX,1000
-121    INT     25;读引导扇区
0123    ADD     BYTE PTR〔1003〕,55;逆变换字符
0128    ADD     BYTE PTR〔1004〕,44
012 D   ADD     BYTE PTR〔1005〕,33
0132    MOV     DX,0189
0135    MOV     AH,09
0137    INT     21
0139    MOV     CX,0003
013 C   MOV     DI,0000
013 F   MOV     DL,〔DI+1003〕
0143    MOV     AH,02
0145    INT     21
0147    INT     DI
0148    LOOP    013 F;显示口令字
014 A   MOV     DX,071 A
014 D   MOV     BH,00
014 F   MOV     AH,02
0151    INT     10
0153    MOV     DX,0199
0156    MOV     AH,09
0158    INT     21
015 A   MOV     DI,0000
015 D   MOV     CX,0003
0160    MOV     AH,07
0162    INT     21
0164    MOV     〔DI+1003〕,AL
0168    INC     DI
0169    LOOP    0160;建立或修改口令字
016 B   SUB     BYTE PTR〔1003〕,55
0170    SUB     BYTE PTR〔1004〕,44
0175    SUB     BYTE PTR〔1005〕,33;变换字符
017 A   MOV     AL,02
017 C   MOV     CX,0001
017 F   MOV     DX,0000
0182    MOV     BX,1000
0185    INT     26;写盘
0187    INT     20

-E189 'Password：$'

-E199 'Please Edit：$'

-RCX BO

-NJTXZ · COM

-W
```

程序 2 COMMAND · COM

```
C>DEBUG COMMAND · COM

09ED:0100 E97DOB JMP 0C80

-A 100

09ED:0100 JMP 4680
-A 4680
4680    MOV     CX,0000
4683    MOV     DX,2479
4686    MOV     BH,07
4688    MOV     AX,0600
468 B   INT     10 置屏幕
468 D   MOV     BH,00
468 F   MOV     DX,061 A
4692    MOV     AH,02
4694    INT     10
4696    MOV     AL,02
4698    MOV     CX,0001
469 B   MOV     DX,0000
469 E   MOV     BX,F000
46A1    INT     25;读盘
46A3    ADD     Byte Ptr〔F003〕,55
46A8    ADD     Byte Ptr〔F004〕,44
46AD    ADD     Byte Ptr〔F005〕,33;字符变换
46B2    MOV     DX,4717
46B5    MOV     AH,09
46B7    INT     21
46B9    MOV     CX,0005;输入口令字次数
46BC    PUSH    CX
46BD    MOV     DI,0000
46CO    MOV     CX,0003
46C3    MOV     AH,07
46C5    INT     21
46C7    MOV     〔DI+F000〕,AL
46CB    INC     DI
46CC    LOOP    46C3;输入口令字
46CE    MOV     DI,0000
46D1    MOV     CX,0003
46D4    MOV     BP,0000
46D7    MOV     AL,〔DI+F000〕
46DB    CMP     AL,〔DI+F003〕;校核
46DF    JNZ     46E2
46E1    INC     BP
46E2    CMP     BP,+03
46E5    JZ      4700
46E7    INC     DI
46E8    LOOP    46D7
46EA    P0 P    CX
46EB    LOOP    46BC
46ED    MOV     BH,00
46EF    MOV     DX,071 A
46F2    MOV     AH,02
46F4    INT     10
46F6    MOV     DX,4703
46F9    MOV     AH,09
46FB    INT     21
46FD    JMP     46ED
46FF    NOP

-E4703'Error Password !!!  $'

-E4717'Please Input Password：$'

-RCX 4630

W
```

程序 3 HARDJM·COM

利用字处理软件建立;使用方式:加密 C>HARDJM 1 解密 C>HARDJM 2

```
DISP MACRO TEST
        LEA DX,TEST
        MOV AH,9
        INT 21H;显示信息
        ENDM
CODE SEGMENT
        ORG 100 H
        ASSUME CS:CODE,DS:CODE
START:
        JMP BEGIN
SS1 DB'硬盘已加锁:(hard disk emcrypt)$'
SS2 DB '硬盘已解锁:(hard disk decrypt)$'
SS3 DB'参数错误:(error)$'
BUFF DB 200 H DUP(?)
GEGIN:
        MOV AX,201 H;1 个扇区
        MOV CX,1;0 道 1 扇区
        MOV BX,OFFSET BUFF;传送地址
        MOV DX,80H;C 盘 0 面
        INT 13H;读指定扇区
        MOV BX,〔82H〕
        CMP BYTE PTR〔BX〕,31H;是加密吗?
        JZ ABC1
        CMP BYTE PTR〔BX〕,32H;是解密吗?
        JNZ ABC2
        DISP SS2
        MOV BX,OFFSET BUFF;传递地址
        MOV WORD PTR〔BX+1FFH〕,00
      JMP     ABC3
ABC1:DISP     SS1
     MOV    BX,OFFSET    BUFF        ;传送地址
     MOV    WORD      PTR  [BX+1FEH],OO
ABC3:
        MOV AX,301H;1 个扇区
        MOV CX,1;0 道 1 扇区
        MOV BX,OFFSET BUFF;传送地址
        MOV DX,80H;C 盘 0 面
        INT 13H;写指定扇区
        JMP ABC
ABC2:DISP     SS3
ABC:MOV AH,4CH;退出
        INT 21H
CODE ENDS
        END START
```

程序 4 JM.COM

对 C:盘根目录区 DOS 子目录加、解密;使用格式:C>JM(执行奇数次为加密,执行偶数次解密)

```
DATA 在 SEGMENT PARA'DATA'
BUFF DB 512 DUP(0)
        SS1 DB ODH,OQH,'DOS 子目录解密!(decrypt)
,
        SS2 DB ODH,OAH,'DOS 子目录加密!(encrypt)
,
        DATA ENDS
ABC1 MACRO HHR
MOV AX,2;驱动器 C
        LEA BX,BUFF
        MOV CX,1;一个扇区
        MOV DX,11H;读写扇区逻辑号
        INT HHR
        POP AX
        ENDM
        ABC2 MACRO HHR
MOV AH,9
        MOV DX,OFFSET HHR
        INT 21H
        ENDM
        CODE SEGMENT PARA'CODE'
        BEGIN PROC FAR
        ASSUME DS:DATA,ES:DATA,CS:CODE
        PUSH DS
        SUB AX,AX
        PUSH AX
        MOV AX,DATA
        MOV DS,AX
        MOV ES,AX
        ABC1 25H;读盘
        MOV BX,WORD PTR BUFF+70H
        MOV CX,SORD PTR BUFF+7AH
        MOV WORD PTR BUFF+70H,CX
        MOV WORD PTR BUFF+7AH,BX;交换数据
        CM BX,0;是加密吗?
        JZ LOC1
        ABC2 SS1
        JMP LOC2
LOC1:ABC2 SS2
LOC2:ABC1 26H;写盘
        RET
BEGIN ENDP
CODE ENDS
        END BEGIN
```