实现软件试用功能的时间限制方案
（ET199）

北京坚石诚信科技股份有限公司

2007-6-24

试用和租赁软件的销售模式已经成为软件行业中一种常规的销售模式，越来越多的开发商希望对软件有时间限制的要求。用户可以先购买软件进行试用，试用到期后，需要向开发商购买正式版软件，或者交费继续租赁。

目前市场上实现时间限制功能的加密锁常见的是采用硬件时钟芯片，具有下面的特点，见下表:

	硬件时钟芯片	ET199
加密原理	软件通过获取硬件时钟芯片中的时间，来判断时间是否有效。	软件通过加密锁硬件中已经设置的时间，来判断时间是否有效。
加密效果	高。时间到期后，无法通过修改计算机时间，使软件继续使用。	高。时间到期后，无法通过修改计算机时间，使软件继续使用。
成本	高。由于需要增加电池和时钟芯片，造成加密锁成本本身增加数倍。	低。没有电池和时钟芯片，不需要额外增加成本。
稳定性	低。当电池没电，或者由于晶振偏差，导致时钟漂移时，会造成硬件时钟芯片失效，软件运行异常	高。不受电池，晶振等因素影响。
使用寿命	短。一般的电池使用寿命为 1~3 年。	长。不受电池的影响，可以一直使用。

售后维护	繁琐。当电池算坏或者寿命到期后，需要将用户手中的加密锁收回更换电池，或者更换新的加密锁，增加成本。	简单。不受电池的影响，无需收回或者更换加密锁。

综合来看，使用 ET199 实现时间限制功能，在加密强度相同的情况下，其在成本，维护，稳定性等方面都要优于有硬件时钟芯片的加密锁。下面就看看 ET199 如何完成限制时间的加密功能。

ET199 是以国外进口高性能智能卡为核心的加密锁产品，我们可以利用下面列举的特点完成对时间的加密方案:

特点一: 硬件无法被复制

众所周知，智能卡硬件在制造工艺上采取了通过产生额外的噪声和干扰信号，再加上若干保护层，采用特殊的材料（对电子束敏感的材料）覆盖等手段，有效的防止了电子探测或者物理攻击，从而硬件无法被复制，另外智能卡内部具有安全存储区，结合运行在智能卡中的 COS（Chip Operation System）系统，使存储在安全存储区中的数据无法被导出。

特点二: 锁内文件无法被导出

ET199 中，存在三种文件类型，可执行文件（使用 C51 语言编写，在 KEIL 编译环境中编译出的 BIN 格式）、数据文件（存储 DATA 的文件）、密钥文件（存储 RSA 公私钥的文件）。这些文件都是不能被导出的。可执行文件和密钥文件根本不能被读取，数据文件可以由可执行文件中通过提供的 C51 语言的读写接口，将文件中的数据返回给上层软件。

特点三：数据安全存储

同时，ET199 没有提供查看锁内文件结构的接口，只有开发商知道锁内哪些是数据文件，哪些是可执行文件，相应的文件 ID 是什么，其他人是没有办法得到的。即使破解者通过反编译工具，查看到 ET199 上层 API 中的输入参数，里面也只有可执行文件 ID，而数据文件的 ID 是在锁内的 C51 可执行文件中，由于破解者是没有办法跟踪到锁内的，因此无法获得锁内数据文件的 ID，锁内的数据文件的内容只能通过锁内的 C51 程序进行读写。对比传统的提供读写数据接口的加密锁而讲，这种模式从根本上决定了锁内数据文件的安全性。

特点四：高强度口令保护

另外，ET199 具有 24 个字符长度（192 位）开发商口令保护，只有验证该口令通过后，才可以向锁内下载 C51 可执行文件，该口令可以设置重试次数限制，防止暴力破解。有效的防止了破解者想通过向锁内下载非法 C51 可执行程序，然后枚举每个文件 ID，从而读取文件数据内容的破解手段。

特点五：逻辑结构安全隐藏

锁内的 C51 可执行程序具有无法被导出，无法被读取，只能在锁内运行的特点。任何由开发商设计的逻辑思想都可以放到 C51 中来实现，破解者根本没有任何手段获知和修改这样的逻辑结构。那么破解者最多也就是通过反编译得到可执行文件的 ID 号，然后自己写程序调用这个 ID 号的可执行文件，而文件是怎么执行的只有开发商知道，破解者的程序最多也就是造成锁内数据被破坏，但这时软件就无法正常运行。

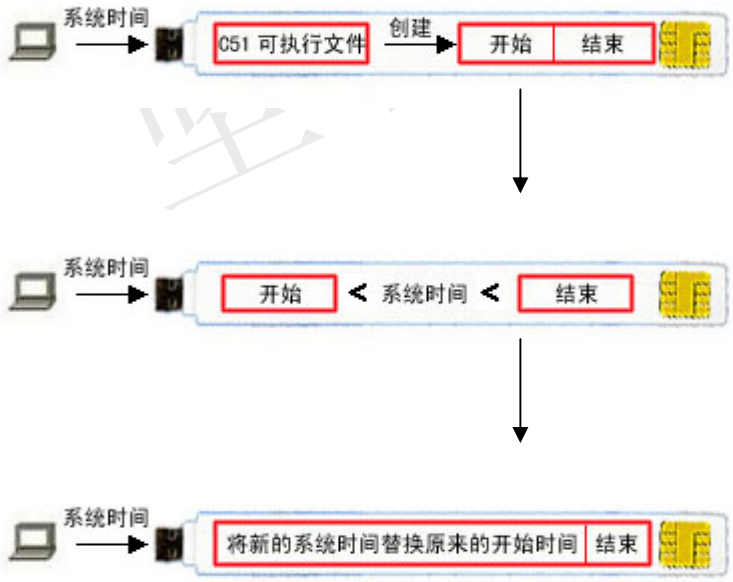
综合上述 ET199 特点分析，我们来具体看下如何完成时间限制功能。其原理

很简单，所有的时间创建，和时间判断都在锁内 C51 可执行文件中完成。

（1）软件第一次运行时，由这个 C51 文件查看锁内是否存在记录时间的文件，第一次运行应该是没有这个文件的，这时创建一个用于记录时间的数据文件，将上层软件传入的时间写入到这个文件中，同时计算出截止使用的时间也写到这个文件中。

（2）在以后的软件运行时，C51 文件这时判断时间文件已经存在，那么将文件中记录的开始时间和结束时间读取出来，然后与上层软件中传来的时间进行对比，如果传来的时间不在这个范围内，说明软件到期。如果传来的时间在范围内，那么将时间文件中的开始时间替换成传来的时间，也就是开始时间不断后移，直至等于或者大于结束时间，软件到期。软件到期后，可以在锁内设置一个标志，那么锁内其他的功能可执行程序判断这个标志，从而运行不正常或者不工作。

原理如下图：



开发需要的做的工作：

工作一：获取当前时间

这个时间应该是系统当前时间，基本上所有常见编程语言都会提供获取当前系统时间的 API 函数接口，我们这里要获取的是当前时间相对于 1970 年 1 月 1 日 0：00 时的一个以毫秒为单位的 DWORD 值。如 VC 的提供的接口：

```
time_t ltime;
    time(&ltime );
```

工作二：编写 C51 可执行程序

该 C51 可执行程序完成的功能

判断是否有时间文件

创建时间文件

根据起始时间来计算结束时间，这个结束时间换算成毫秒，如：1 个月是：30×24×60×60×1000=2592000000ms，那么将起始时间加上 2592000000，就是试用 1 个月的结束时间。然后将起始时间和结束时间写入到锁内

判断传入时间是否在起始时间和结束时间范围内，特别是要看是否大于起始时间

改写时间文件中的起始时间

设置标志，便于其他 C51 可执行程序来判断

这样的加密是否是安全的呢，回答是肯定的。我们从以上的特点可以分析出：

强度一：锁内时间文件的存储安全

由于没有读取接口，锁内的时间文件数据不能被读出

时间文件只能被可执行文件改写，破解者如果使用自己的程序执行这个可执行文件，那么最多也就是造成锁内数据被破坏，软件无法正常使用

破解者无法下载非法可执行程序，去枚举锁内数据文件的 ID，从而读取这个存储时间数据文件的内容

强度二：有效的防止修改计算机时间的破解手段

软件第一次运行时，锁内是没有时间文件的，那么这时得到的时间就是从系统中读取的时间，同时在锁内计算出结束时间。当破解者企图修改计算机时间时，这时传入的时间已经是不正确的，由于锁内对传入时间和起始时间进行了判断，并且只有在传入时间大于起始时间时，才视为正常。锁内的判断又是以毫秒为单位进行，且每次都必须大于存储的开始时间，彻底杜绝了修改时间的破解可能。

强度三：逻辑结构内部封装

所有的时间判断都是在加密锁中完成，破解者只能监控计算机的内存，截获 USB 端口的数据（ET199USB 端口传输也是硬件级加密，每次插拔都会变化），对加密锁内部的任何处理都是束手无策。

使用 ET199 完成对软件的时间保护不仅节省成本，维护方便，稳定性高，当使用到期后，可以通过远程升级，将锁内的时间文件删除或者重设标志，使软件继续使用。ET199 比目前市场上使用时钟芯片的加密锁具有更多的优势，同时以出色的质量，极高的性能价格比，为软件开发商提供了强大的保护后盾。