 目前，国内软件销售过程中采用了一种新的方式：开发者根据计算机中不同的硬件配置标志直接在应用程序中设置密钥，限制程序的使用次数或者限制某些先进功能的使用，然后将受限制的应用程序无偿提供给用户。用户在试用一段时间之后如果觉得很满意，就可以将安装程序提取的硬件配置解密密钥或已经采集机器配置情况的应用程序提供给开发者，并花少量费用购买自己机器中的电子注册密钥，从而能够充分利用应用程序的所有功能。 在应用程序中利用电子注册来限制应用程序的部分功能，这样既可以让用户先试用然后再决定是否购买应用程序，又保护了开发者的合法劳动成果，减少了用户与开发者之间的不必要的中间环节。开发者直接得到用户购买软件的费用，真正地体现了开发者所创造的价值；用户在试用软件之后再决定是否购买，从而使得用户能够得到称心如意、物有所值的软件。因此，不通过中间环节这种销售方式降低了软件的成本，使开发者和用户双方都受益。同时，这种方式还可以使得开发者能够直接获得用户的反馈信息，促使开发者开发出功能更加完善的应用程序。 然而，要想在应用程序中实现电子注册功能决不是件容易的事情，尤其是在Windows 平台推出以后，要想实现一个跨平台的应用程序电子注册功能，则要求开发者应具有丰富的编程技巧和实际开发经验以及广阔的开发视野。笔者通过实践探索，终于成功地实现了跨越DOS、Windows 3.X和Windows 95平台的应用

程序电子注册功能。下面将阐述其实现原理及技巧。 一、注册密钥点的选择与生成 实现应用程序的电子注册功能，最关键的问题是采集硬件配置中的密钥点。在DOS 系统下，可以通过硬盘端口1F6H和1F7H直接读取硬盘的序列号等作为密钥算法的数据，因为每块硬盘的型号、版本号和序列号均不同，只要用户提供上述内容，利用这种方法生成的注册密钥在每台计算机中均不同，从而实现电子注册的功能。著名的字表处理软件CCED 5.18中采用的就是类似的方法。虽然这种方法在绝大数场合下很有效，甚至可以在Windows 3.X系统和Windows 95系统的兼容模式下通过，但在最高性能配置的Windows 95保护模式下却行不通，原因是Windows 95保护模式下不允许通过端口方式读取硬盘类型参数，所以利用这种方法无法实现跨平台的通用电子注册功能。 本人仔细分析计算机中ROM区的F000H-FFFFH内容后，发现该区域中记录着很多与硬件配置有关的信息（如CMOS配置信息、主板名称、型号和序列号、主机标志字节和生产日期等）。可以采集其中一处或几处作为注册密钥算法的原始数据（如机器ROM区中的F000H:FFF5H-F000H:FFFFH中依次存放主机出厂日期和主机标志字节的内容），这些硬件特有的信息对于不同型号的计算机来说是不可能相同的。因此，完全可以将其作为注册密钥算法的原始数据，而且这些内容在DOS、Windows 3.X和Windows 95下均相同。需要注意的是，如果在实际应用中真的将该采集点作为算法的原始数据，则不应该包括F000:FFF0H开始的前五个字节的内容，原因是该地址已被用作机器热启动时的入口地址，在DOS、Windows 3.X和Windows 95系统中对热启动复合键Ctrl Alt Del的处理程序均

不同，因此该处的内容在三者之中也都不相同，读者应记住这一点。 利用上述方法取得注册密钥算法的原始数据后，开发者就可以确定自己的加密算法，这可以通过编程语言中丰富的位操作功能来实现。然后将注册加密算法增加到应用程序中需要限制的部分，并可根据应用程序的实际需要和限制的功能任意设置多处，使盗版者很难解密，从而有效地保护开发者的成果。利用这一方法，即使机器中有多个应用程序使用相同的硬件配置信息采集点，也不可能发生任意加密冲突问题；即便是使用了相同的算法原始数据，由于算法不同，注册密钥也不会完全相同；即使解密者知道加密算法的原始数据，由于无法知道加密算法，再加上加密算法贯穿于整个应用程序，所以很难解密。因此，上述方法可以有效地实现跨越DOS、Windows 3.X和Windows 95平台的电子注册功能。此外，由于ROM 区关键点的内容不可能发生变化，所以即使将来推出新型的操作系统平台，这种方法仍然会很有效。二、利用解密密钥建立联系 应用程序的加密部分完成之后，就需要建立相应的解密密钥。所谓解密密钥，就是将加密算法的原始数据经过加密之后，直接显示给用户并写入应用程序的相应位置。这样，用户既可以通过电话或计算机网络给开发者提供注册功能的算法原始数据，也可以将安装后的应用程序寄给开发者。加密密钥既可以是ROM 区域内的原始数据（最好不要原样提供），也可以是由原始数据经过一定换算后形成的新的数据。因此，开发者提供的应用程序中的加密算法部分应包括两部分：将机器ROM 区域内的数据经过解密密钥算法后形成解密密钥，再将解密密钥数据经注册算法后形成注册密钥。 应用程序中注册密钥的算法、注册密钥的

长度、显示或提供给开发者的方式可自己确定，但解密密钥的长度和算法应与注册密钥完全相同。解密密钥也没有必要做得那么复杂，只需进行简单处理后就可以实现，例如本文程序中实现的方法是将ROM中采集的数据简单地减去0x2020。 三、电子注册密钥生成程序 开发者得到用户提供的解密密钥原始数据后，需要利用专用的密钥生成程序将其转换成注册密钥，并将注册密钥交给用户。注册密钥的算法与应用程序中判断注册密钥的加密算法程序应该完全相同。该程序一般应具有以下三种取得注册密钥算法原始数据的方式，以方便进行密钥的处理。该程序的名称为READKEY.EXE，其功能如下：（1）当READKEY不带参数时，则直接从当前机器中取得注册密钥；（2）当READKEY带参数时，则从键盘输入解密密钥来获取注册密钥；（3）当READKEY带EXE文件名参数时，则从相应应用程序的特定位置取得解密密钥，并生成注册密钥。 用户得到注册密钥后，重新安装一次应用程序或在需要输入注册密钥处直接输入密钥，则应用程序会自动将这个注册密钥存放到文件的特定位置处，当应用程序被他人拷贝到其它机器中之后，由于注册密钥随机器的不同而不同，所以应用程序的功能或使用次数仍然受限，要想在其它机器中使用该应用程序，则必须重新注册。 应用程序中解密密钥和注册密钥的位置，可先用特殊字符来标识，然后用DEBUG 等程序直接查找其位置，再修改其它程序中读取或写入数据的地址值。至于解密密钥显示和注册密钥的输入方式，可由开发者确定是用安装程序的方法还是在应用程序中直接处理的方法。 