doi :10.3969/j.issn.1671-1122.2010.05.011

一种新型软件加密流程的设计

周利华[1,2]，周虎生[1]，文伟平[1]

（1. 北京大学 软件与微电子学院 信息安全系，北京 102600；
2. 东北大学 软件学院，辽宁沈阳 110819）

摘 要：为保护软件开发者对软件版权的自主控制，防止软件产品盗版，本文首先调研了当前主要软件保护技术，并对当前软件保护技术的不足进行了分析，最后提出了一种依赖于智能卡硬件设备的新型软件加密思路。本文结合信息安全的加密算法及智能卡技术实现了软件部分关键代码的隐蔽执行流程，大大增强了软件被破解的难度，提升了软件的保护能力，在实践中具有较强的应用价值。

关键词：软件版权；智能卡；软件加密

中图分类号：TP393.08 **文献标识码：**A

0 引言

随着计算机技术的不断发展，面向各应用领域或行业需求的软件不断地孕育而生。但无论哪种优秀的软件，其内部核心的技术往往是该软件的命脉，一旦被他人窃取或被非法复制，由此带来的安全风险和经济损失是无法估计的。目前，软件作为智力和知识的结晶也显得越来越重要，打击计算机软件的非法复制和销售，保护和推进软件产业有序地建立和发展，已成为世界各国不可忽视的问题。

从软件安全过程确保的各个阶段看，软件加密实际上可分为两种。其一是对软件的源代码进行加密，目的是将源代码作为具有知识产权的著作加以保护，以防止被剽窃；其二是软件的使用设置权限，不允许在未授权的情形下使用软件，即对软件的使用进行加密。这两种加密的最终目的都是防止软件作者的软件产品被无偿使用。

本文首先调研了当前主要软件保护技术，并对当前软件保护技术的不足进行了分析，然后提出了一种依赖于智能卡硬件设备的新型软件加密思路，该思路结合信息安全的加密算法及智能卡技术实现了软件部分关键代码的隐蔽执行流程，最后给出了该思路的具体应用场景，方便更多的同行做进一步的研究和参考。

1 软件加密技术分类

依据加密原理和方式，加密技术可分为三大类：软件加密、网络认证加密和硬件加密。

1.1 软件加密

软件加密是指不依赖特殊硬件，用纯软件的方式来实现对软件的加密保护。目前主要有序列号法、警告（NAG）窗口、时间限制、功能限制、注册文件法和加壳法等。

序列号法是以复杂的数学算法为核心的软件加密技术，同时也是应用最为广泛的一种。为了达到识别用户身份的目的，在销售软件产品的过程中，将一个序列号印刷在产品的说明书中，用户必须输入有效的序列号才能完成安装，在现有的软件中，序列号的使用率是极高的，包括微软的操作系统也曾使用过这种方法。

警告（NAG）窗口法，是开发者让程序在使用过程中或是软件启动的时间按照一定的时间段来不定时弹出一些窗口。提示用户来注册软件。

时间限制法，有些程序的试用版每次运行都有时间限制，比如限制使用 7 天、15 天、30 天等等。一旦使用时间到即退出运行，要求用户注册后使用。

功能限制法，通过限制部分软件功能的方式来促使用户购买软件授权。比如有的行业软件，如果没有注册，软件只能生成部分数据，或者不提供打印功能等等，一旦用户购买了正版授权，则放开功能限制。

注册文件法，是一种利用文件来注册软件的保护方式。KeyFile 或 LicenseFile 一般是一个小文件，可以是纯文本文件，也可以是包含不可显示字符的二进制文件，其内容是一些加密过或未加密的数据，其中可能有用户名、注册码等信息。文件格式则由软件作者自己定义。现在利用比较多的有某些加密软件，如 VmProtect[2] 等。

加壳法，就是在被加密程序上，加上一个外“壳”，也就是对 EXE、DLL 文件里的资源进行压缩，改变其原来的特征码，隐藏一些字符串等等，使一些资源编辑软件不能正常打开或者修改。用这个壳防止软件被复制，还通过这个壳，对付跟踪和反编译等。在出版发行的通用软件中，这种方法较为普遍，因为用加壳产品实施加密比较简便，所以软件作者在开发过程中，不需要过多考虑加密的问题。但由于“壳”在发布以后成为软件必须执行的一部分，有可能

会对程序的稳定性造成一定影响，甚至会引入一些意想不到的错误。而且如果某种壳被破解，那么所有使用同类壳的软件都可以使用一致的方法来破解。

在最初的加密过程中，大部分软件开发商采用的是软件加密方式，随着软件破解技术进一步发展，传统的软件加密方式无法满足软件版权保护的安全需求，网络认证加密和硬件加密开始逐渐出现并流行，成为最广泛采用的加密手段。

1.2 网络认证加密

互联网的广泛应用催生了共享软件这种软件形式，共享软件的销售特点促使了软件加密由传统的捆绑加密形式向软件注册机制发展，随着互联网的发展，网络的大范围推广解决了传统注册机制的缺点和所带来的一系列问题。在商业软件的带动下，软件加密技术开始向网络化发展。软件加密的网络化部署需要一定的软硬件基础环境，如软件中网络注册功能和远程服务器端验证功能的实现，以及远程服务器建立和设置、相应的互联网络支持等。

互联网的普及使得大多数软件都具备了相应的网络功能，为了支持这些网络功能，都设立了专用的服务器提供网络服务，如产品在线更新、技术支持等，这使得网络化的加密形式，诸如“ 激活 ”和“ 验证 ”等有了坚实的硬件和网络应用部署基础。借由这些提供用户网络服务的服务器，软件作者就可以方便地实施部署自己的网络化加密。同时，加密的网络化也促使软件加密和软件程序中的网络功能模块结合在一起，相辅相成，如 ：只有经过验证的正版注册的用户才享有软件的更新权限，在软件检查更新的同时校检用户的注册信息等。另外，软件在进行网络校检和认证的过程中，传输的只是较小的一部分数据信息，因此，对于服务器性能和网络带宽不会有太高的要求，同时也不会造成明显的负担。即便是对于租用的服务器来说，也不会带来过多额外的费用。

1.3 硬件加密

依赖特定硬件的加密软件不容易被复制，因此硬件加密比软件加密可靠。采用加密狗加密软件，由于指纹（即计算机的唯一标识，可以由某些不常更换硬件的序列号和类型码经过一定算法生成）和软件是分离的，可以对软件进行备份，以防止原盘损坏。而磁盘加密方法把指纹放在磁盘上，不能备份，如果原盘损坏，软件就不能继续使用 ；采用加密狗加密时，软件以不加密的形式存放，可以大批地快速复制，而采用磁盘加密方法，必须在每张盘上逐个制作指纹，而且指纹盘的质量难以保证，所以，加密狗加密法比磁盘加密法可靠。

硬件加密的原理是将加密信息固化在某个硬件中，然后将它作为一个软件的附加设备销售给用户。当运行该软件的时候，需要将该硬件设备接到计算机的输出端口，软件根据是否检测到对应的密钥来决定是否运行某些功能。这类加密中有软盘加密、光盘加密、加密卡和加密狗等。

钥匙盘的方式是最常见的软盘加密方式。它是通过 BIOS 的 INT13 中断对软盘格式化一些特殊的磁道，并在这些磁道里写入一定的信息，软件在运行时要校验这些信息。这种软盘就好像一把“ 钥匙 ”，被称为钥匙盘。比如当年的 KV3000 等杀毒盘就采用这种加密方式。由于软盘中特殊磁道的标记信息不能被平常的拷贝命令或软件读取，所以，钥匙盘类的软件不能被轻易复制，被加密的软件就比较安全了。

光盘方式是一种面向光盘软件的保护加密技术。采用“ 光盘 ”加密过的软件，它在运行时首先检测光盘特征，如果用户输入的特征码和该光盘的物理特征相符，便顺利运行，否则终止程序。这类保护多用于游戏软件，程序运行时要求将原版的光盘放在光驱中，然后输入光盘附带的 CDKey，或者是程序直接检查光盘上的特殊数据 (指纹等)，由此来判断使用的是否是正版光盘。这种方法的主要原理是利用特殊的光盘母盘上的某些特征信息是不可再现的，而且这些特征信息大多是光盘上非数据性的内容，分布在光盘复制时复制不到的地方。大规模的生产这种加密方案的光盘可以将成本降得很低。而且软件和数据在同一载体上，对用户使用很方便。它最适合于那些相对便宜的光盘版软件。但是它的问题也是与它的介质稳定性有关，那就是光盘也是一种耗材，一旦出现人为保管不善，磨损抑或丢失，则会给用户带来麻烦。

加密卡是指插在计算机总线上的加密产品，其加密强度较高，防跟踪措施完备，但使用起来不太方便，需要经常插换，会占用一定的系统资源，而且成本也高，所以一般为系统集成商采用。软件狗是目前流行的一种加密工具，它是插在计算机并行口或 USB 上的加密产品。软件狗一般都有几十或几百字节的非易失性存储空间可供读写，并且提供了各种语言的接口及外壳的加密方式供开发商使用。它具有加密可靠、使用方便等优点，成本相对加密卡来说要低很多。

加密狗是外形酷似 U 盘的一种硬件设备，正名加密锁，后来发展成如今的一个软件保护的通俗行业名词，加密狗是一种插在计算机并行口上的软硬件结合的加密产品（新型加密狗也有 USB 口的）。一般都有几十或几百字节的非易失性存储空间可供读写，现在较新的狗内部还包含了单片机。软件开发者可以通过接口函数和软件狗进行数据交换（即对软件狗进行读写），来检查软件狗是否插在接口上 ；或者直接用软件狗附带的工具加密自己 EXE 文件（又称“ 加壳 ”）。这样，软件开发者可以在软件中设置多处软件锁，利用软件狗做为钥匙来打开这些锁 ；如果没插软件狗或软件狗不对应，软件将不能正常执行。

2 一种新型智能锁硬件加密技术

2.1 智能锁相关加密技术特点

防反编译 ：将原本在 PC 中运行的部分关键代码移到锁中运行，加密锁相当于一个黑盒子，即使攻击者使用反编译和反汇编工具，也无法追踪到全部的代码流程，在加密锁中的运行部分将无法被知晓。

防伪造 ：加密锁一般不会这样做 ：即获得加密锁运行的结果，然后在 PC 端对其进行比较判断，从而来决定程序

是否可以继续运行，因为这样很容易被破解者将这部分比较直接跳过。而是将加密锁运行的结果用于其下继续运行的代码中，即这些变量或结果在下面继续运行的代码中将会继续用到，因为它本身是程序中的一部分，这样就可以达到，存在指定的加密锁，程序将会正常运行，不存在指定加密锁或加密部分被跳过，程序将不能正常运行，从而达到防伪造数据的目的。

防克隆 ：先进的智能锁都采用智能卡芯片，复制难度大，成本高，而且具有硬件序列号等唯一标识方式，使得克隆出来的二进制数据即使注入到新的芯片中也无法正常运行。

高强度加密 ：加密锁中使用 3DES 和 RSA 等高强度加密算法，使得文件或者数据以密文形式与 pc 机进行交互，在不知道密钥的情况下，黑客无法在可接受时间内完成对内容的破解。

2.2 一般智能锁的工作流程

常规智能锁一般包括加密和运行(即解密）两个过程（此两个过程分别见图 1 和图 2)。其中加密过程是将原本在 PC 中运行的关键代码移动到锁中，并使用加密算法将其加密成二进制的形式。运行过程是将传入锁中的二进制文件或数组通过锁中预设的密钥进行解密，并在锁中运行解密后的代码，将运行结果返回。该加密解密过程采用的是对称密钥方式，也就是说加密密钥必须与解密密钥相同，否则的话，加密后的代码将无法被加密锁在锁中正确解密。所以在加密时，必须先在锁中设置好正确的加密锁密钥，然后将加密锁派发给用户时，也需要在锁中设置相同的加密密钥。

图1 一般智能锁加密过程

2.3 一种基于公私钥结合的智能锁加密技术

本文提出的这种智能锁加密技术，将对称密钥及公开密钥技术相结合，可以实现更强大的保护功能。

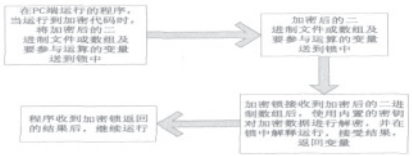

图2 一般智能锁运行过程

2.3.1 名词解释

锁 ID 是锁的身份标识，用于标识唯一的一把锁，长度为 256 个字节，它实际是一个模 N 及公钥的混合体，这个 ID 是全球唯一，且是不可以更改的。

主锁密钥，主锁密钥与开发密钥构成一个完整的加密密钥，设置主锁密钥的目的，是为了防止开发密钥被黑客盗取，设置了主锁密钥后，即便是开发密钥被别人知道，别人也没有办法生成与其对应的授权序列。

开发密钥，开发密钥与主锁密钥构成一个完整的加密密钥，开发密钥的存在使加密模式可以更加丰富。

授权序列，授权序列用于授权用户使用，只有与用户锁有相应的授权序列，加密代码才可能被相应的用户锁在锁内还原并运行。授权序列的产生与如下有关 ：主锁密钥，开发密钥及锁 ID 有关（删除)。这三个中的任何一个的改变，都会生成唯一不同的授权序列。也就是说 ：1) 对于同一主锁密钥及不同的用户锁，生成的授权序列也不相同，将某一用户锁的授权序列应用于另一用户锁，程序是不能正常运行的（原因是，授权序列与用户锁内置私钥不对应，加密代码无法被还原)。具体说明如下 ：A 锁授权序列—> A 锁内置私钥,加密文件可以被 A 锁在锁内还原。A 锁授权序列—> B 锁内置私钥，加密文件不可以被 B 锁正常还原。2) 对于同一主锁密钥，同一用户锁，不同的开发密钥，生成的授权序列也不相同。3) 对于同一开发密钥，同一用户锁，不同的主锁密钥，生成的授权序列也不相同。

2.3.2 加密过程

加密过程与一般的对称加密一样，将原本在 PC 中运行的关键代码移动到锁中，并使用加密算法如 3DES 将其加密成二进制的形式，达到的效果是，原本在 PC 机中运行的完整代码现在分别运行在两个硬件设备中，而加密锁中的代码又不是以明文形式存在，从而达到反编译的效果。这里使用的开发密钥可以看作是对称密钥，具体过程如图 3 所示 ：

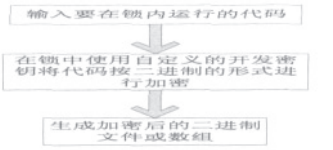

图3 公私钥结合加密技术的加密过程

2.3.3 授权过程

锁 ID 由两部分组成，即 RSA 算法中的公钥及模 KU(PK,n)，每一把锁的 ID 是唯一的，除此之外，每一把都内置有与公钥相对应的唯一私钥 EU(SK,n)，私钥内置在加密锁中，不可读，不可以更改。授权过程实际上就是根据锁 ID 对开发密钥 M 进行公钥加密，获得的授权序列 C 即为公钥加密后的密文。由于锁 ID 是唯一的，所以可以得出，不同的锁对应的授权序列也是唯一的，不同的开发密钥，对应于同一把锁，其授权序列也不相同。

2.3.4 注册过程

在加密锁中对加密文件或二进制数据进行注册时 ：1) 加密锁使用内置的私钥 EU(SK,n) 对授权序列进行解密，将开发密钥 M 还原 ；2) 加密锁使用还原后的开发密钥，对输入的二进制数据进行解密，还原成正常的二进制数据，这里是对称加密的逆过程 ；3) 加密锁中存在着另一密钥，该密钥是唯一的，每一把锁都不相同，注意，该密钥与私钥并不相同，是相互独立并不关联，它是一对称密钥，加密锁使用这一对称密钥重新对还原后的正常的二进制数据进行加密，生成另一加密后的二进数据，该加密数据只能被该加密锁正常解密。

2.3.5 运行过程

用户锁转换加密文件过程 ：加密文件通过用户锁的 IO 口传送到用户锁中，然后用户锁使用授权号将其解密还原成正常的文件储存在 RAM1 中，接着用户锁又使用其内置的密钥对还原后的文件进行加密，生成另一与用户锁相对应的加密文件，最后通过用户锁的 IO 口传送到用户计算机中。由于不同的用户锁，其内置的密钥是不相同的，因而对

于不同的用户锁，其生成的加密文件是不相同的。用户锁运行转换后的加密文件过程 ：当要加密软件使用转换后的加密文件时，通过 IO 口将加密文件及输入参数传送到用户锁中，然后用户锁使用内置 的加密密匙将其解密还原成正常的文件，执行该文件，然后将执行后的结果输出到用户计算机中。如果转换后的加密文件不是由该锁生成的，那么当加密文件被送到 用户锁后，将不能被还原成正常的文件，因而加密软件将不能被正常运行。（责编 杨晨）

参考文献：

[1] R.L. Rivest, A. Shamir and L. Adleman, A method for obtaining digital signatures and public-key cryptosystem[C], Commun. ACM, pp.120-126, Feb. 1978.
[2] 段钢 . 加密与解密 [M].3 版 . 北京 : 电子工业出版社 ,2006 1-5.
[3] 飞天诚信 . 软件加密原理与应用 [M]. 北京 : 电子工业出版 ,2004 2-5.
[4] 域天公司 . 域天加密技术过程 .2009.
[5] 徐海风，曹小军 . 软件加密方法及技术 [J]. 山西冶金，2007 1.
[6] 冯朝辉 . 软件加密市场需求与产品互动分析 [J]. 网络安全技术与应用 ,2006 9.
[7] 张书敏，韩文虹 . 软件加密技术及实现 [J]. 学术研究 ,2009 6.

本课题得到全球信息安全公司 SafeNet 2009 年支持项目“软件安全漏洞挖掘”的支持

作者简介 :周利华（1986-），男，硕士研究生，主要研究方向 : 系统与网络安全 ; 周虎生（1986-），男，硕士研究生，主要研究方向 : 系统与网络安全 ; 文伟平（1976-），男，副教授，博士，主要研究方向 : 网络攻击与防范、恶意代码研究、信息系统逆向工程和可信计算技术等。

上接第 22 页

设置中断等。相应的底层操作代码如下 ：

```
// 配置 QSM 模块
MOVE.W#$000a,QSMCR
MOVE.B #$1b,QILR ；定义中断优先级
MOVE.B #$40,QIVR ；定义中断向量基地址
// 配置 SCI 模块
MOVE.W #$0000,SCCR1 ；禁止 TXD, RXD
MOVE.WSCSR, D0
MOVE.WSCDR, D0
接口函数 ioctl 进行读写以外的一些操作，如设置波特率等 ：
//baudspeed 设置波特率
MOVE.W #$0032, SCCR0 ；设置波特率为 10400bps
```

接口函数 write,read 进行在串口通信中读写数据，相应的底层操作代码如下 ：

```
//read 读数据代码
MOVE.B RDR，（A0）；接收数据
MOVE.W#$000C, SCCR1
接收完毕，关闭接收中断
//write 写数据代码
MOVE.B -(A0), TDR ；发送数据
MOVE.W#$0088, SCCR1
打开发送中断允许
MOVE.W#$002C, SCCR1 ；发送完成，关闭发送中断
```

编写完接口函数以后，将 SerialPortInterface.c 复制到 / uclinux/linux/drivers/char 目录下，并且在 /uclinux/linux/drivers/char 目录下 mem.c 中，int chrdevinit ( ) 函数中增加如下代码 ：

```
#ifdef CONFIG_SERIALPORT_DRIVE
init_com();
#endif
```

编译完成后，把目标文件一起存入 Flash 中就可以使用了。在这其中，com_fops 提供了设备驱动程序的入口点，在设备驱动程序初始化的时候，通过调用 register_chrdev 向系统注册登记。注册成功后就可以看到串口设备文件，直接操作设备文件来访问串口。比如说 ：发送数据、接收数据、操作 modem 等都可以通过对设备文件的 open、read、write 等操作来完成。

本设计中，取样 PC 机作为主控端，控制指令必须及时接受，串口设备文件不是主动接收的，所以程序设计成中断接收控制信号指令的方式，并进行响应回复。控制命令的执行时间一般小于 25ms，对于中断信号的响应完成能胜任，当中断中完成数据转存和置标志位以后，主要的处理过程还是在取样的 PC 机中完成，实现控制和处理分离的结构设计。其流程图如下 ：

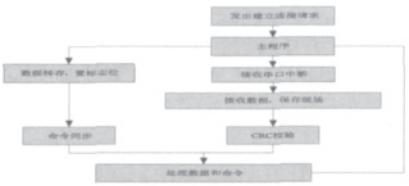

图2 串口访问程序流程图

对于取证设备的访问，根据消息帧结构，一旦进入串口终端以后，立刻以查询的方式接收数据，如果传送的串口字数间隔时间超过中断响应的最大时间，则认为一帧消息完成，整个响应流程在 115200 波特率的设置下，可以正常进行通信，不会发生丢帧现象。

整个程序响应的流程如图 ：

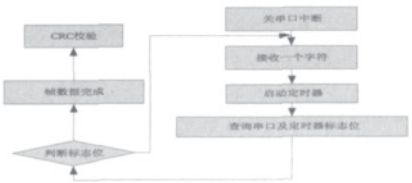

图3 串口设备数据接收流程图

串口通信发送过程中，还需要进行比较重要的 CRC 校验。我们使用的是 CRC-16 的查表法实现。

程序如下 ：

```
CRC: MOV R6,#0FFH
MOV R5, #0FFH;  先置 CRC 初值为 FFFFH
MOVX A,@DPTR
INC DPTR
XRL A, R5
MOV DH, A
INC 0A2H
MOV DPTR, #tab1; 指向 table1 表的下半区
MOVC A,@A+DPTR ；读高字节位
XRL A, R6;  计算高字节
MOV R5, A;  存入 R5
MOV DPTR, #tab2; 指向 table2 表的下半区
MOV A, DH
MOVC A, @A+DPTR 读低字节位
MOV R6, A;  存入 R6
INC 0A2H
DJNZ R7,CRC_LOOP  CRC 校验循环
RET  返回
```

3 结 论

串口通信是一种灵活和实用的通信方式。尤其在电子取证过程中，由于数据恢复 / 取证设备通常都不会带键盘和显示装置，这个时候，应用程序的调试、运行参数的设置以及测试数据的上传都需要通过串口通信方式进行。所以研究数据恢复 / 取证设备的串口通信机制，对于深入了解相关设备的性能，提高设备使用的准确率和可控性有非常大的帮助。（责编 杨晨）

作者简介 : 张鹏（1979-），男，司法鉴定人，硕士研究生，主要研究方向 : 网络安全、电子取证等 ; 周刚（1978-），男，讲师，博士，主要研究方向 : 网络存储 ; 麦永浩（1959-），男，教授，博士后，主要研究方向 : 网络安全、电子取证等。