坚石诚信

ET-ARM 产品介绍

V1.0

北京坚石诚信科技有限公司

网址：http://www.jansh.com.cn

修订记录：

修订日期	版本	修订内容
2013 年 7 月	V1.0	第一版发布

软件开发协议

北京坚石诚信有限公司（以下简称坚石）的所有产品，包括但不限于：开发工具包，磁盘，光盘，硬件设备和文档，以及未来的所有定单都受本协议的制约。如果您不愿接受这些条款，请在收到后的 7 天内将开发工具包寄回坚石，预付邮资和保险。我们会把货款退还给您，但要扣除运费和适当的手续费。

1. 许可使用

您可以将本软件合并、连接到您的计算机程序中，但其目的只是如开发指南中描述的那样保护该程序。您可以以存档为目的复制合理数量的拷贝。

2. 禁止使用

除在条款 1 中特别允许的之外，不得复制、反向工程、反汇编、反编译、修改、增加、改进软件、硬件和产品的其它部分。禁止对软件和产品的任何部分进行反向工程，或企图推导软件的源代码。禁止使用产品中的磁性或光学介质来传递、存储非本产品的原始程序或由坚石提供的产品升级的任何数据。禁止将软件放在服务器上传播。

3. 有限担保

坚石保证在自产品交给您之日起的 12 个月内，在正常的使用情况下，硬件和软件存储介质没有重大的工艺和材料上的缺陷。

4. 修理限度

当根据本协议提出索赔时，坚石唯一的责任就是根据坚石的选择，免费进行替换或维修。坚石对更换后的任何产品部件都享有所有权。

保修索赔单必须在担保期内写好，在发生故障 14 天内连同令人信服的证据交给坚石。当将产品返还给坚石或坚石的授权代理商时，须预付运费和保险。

除了在本协议中保证的担保之外，坚石不再提供特别的或隐含的担保，也不再对本协议中所描述的产品负责，包括它们的质量，性能和对某一特定目的的适应性。

5. 责任限度

不管因为什么原因，不管是因合同中的规定还是由于刑事的原因，包括疏忽的原因，而使您及任何一方受到了损失，由我方产品所造成的损失或该产品是起诉的原因或与起诉有间接关系，坚石对您及任何一方所承担的全部责任不超出您购买该产品所支付的货款。在任何情况下，坚石对于由于您不履行责任所导致的损失，或对于数据、利润、储蓄或其它的后续的和偶然的损失，即使坚石被建议有这种损失的可能性，或您根据第 3 方的索赔而提出的任何索赔均不负责任。

6. 协议终止

当您不能遵守本协议所规定的条款时，将终止您的许可和本协议。但条款 2，3，4，5 将继续有效。

Software Developer's Agreement

All Products of Jansh Technologies Co., Ltd. (Jansh) including, but not limited to, evaluation copies, diskettes, CD-ROMs, hardware and documentation, and all future orders, are subject to the terms of this Agreement. If you do not agree with the terms herein, please return the evaluation package to us, postage and insurance prepaid, within seven days of their receipt, and we will reimburse you the cost of the Product, less freight and reasonable handling charges.

1. Allowable Use – You may merge and link the Software with other programs for the sole purpose of protecting those programs in accordance with the usage described in the Developer's Guide. You may make archival copies of the Software.

2. Prohibited Use – The Software or hardware or any other part of the Product may not be copied, reengineered, disassembled, decompiled, revised, enhanced or otherwise modified, except as specifically allowed in item 1. You may not reverse engineer the Software or any part of the product or attempt to discover the Software's source code. You may not use the magnetic or optical media included with the Product for the purposes of transferring or storing data that was not either an original part of the Product, or a Jansh provided enhancement or upgrade to the Product.

3. Warranty –Jansh warrants that the hardware and Software storage media are substantially free from significant defects of workmanship or materials for a time period of twelve (12) months from the date of delivery of the Product to you.

4. Breach of Warranty – In the event of breach of this warranty, Jansh's sole obligation is to replace or repair, at the discretion of Jansh, any Product free of charge. Any replaced Product becomes the property of Jansh.

Warranty claims must be made in writing to Jansh during the warranty period and within fourteen (14) days after the observation of the defect. All warranty claims must be accompanied by evidence of the defect that is deemed satisfactory by Jansh. Any Products that you return to Jansh, or a Jansh authorized distributor, must be sent with freight and insurance prepaid.

EXCEPT AS STATED ABOVE, THERE IS NO OTHER WARRANTY OR REPRESENTATION OF THE PRODUCT, EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

5. Limitation of Jansh's Liability –Jansh's entire liability to you or any other party for any cause whatsoever, whether in contract or in tort, including negligence, shall not exceed the price you paid for the unit of the Product that caused the damages or are the subject of, or indirectly related to the cause of action. In no event shall Jansh be liable for any damages caused by your failure to meet your obligations, nor for any loss

of data, profit or savings, or any other consequential and incidental damages, even if Jansh has been advised of the possibility of damages, or for any claim by you based on any third-party claim.

6. Termination – This Agreement shall terminate if you fail to comply with the terms herein. Items 2, 3, 4 and 5 shall survive any termination of this Agreement.

章节目录

第1章 ET-ARM 入门

随着软件加密技术和硬件性能的提升，加密锁也在发生着日新月异的变化。从一开始只提供加解密和微量数据读写，到现在内嵌虚拟机和运行可执行文件；从一开始单一的生产和管理模式，到现在安全便捷的子母锁模式，这些都将加密锁的安全生产和软件保护强度提升到了一个前所未有的高度。而如今随着硬件性能的进一步提升和坚石对于加密锁领域十几年的专注与积累，一款新的产品呼之欲出，而 ET-ARM 正是这样一个加密锁产品的集大成者。

这是一款高性能加密锁，无论是从功能使用的角度，还是从运算速度的角度，ET-ARM 都可以视作加密锁产品中的翘楚，通过本章的介绍，您大约需要五分钟的时间，可以了解到以下内容：

1. ET-ARM 系列产品的分类。
2. ET-ARM 有哪些特点？为什么选择 ET-ARM？

1.1 ET-ARM 产品型号介绍

为了满足不同客户的需求，ET-ARM 系列产品分为标准锁（ET-ARM）、时钟锁（ETTime-ARM）、U 盘锁（StoreET-ARM），其中标准锁外观提供了可选择的普通外壳和 Mini 外壳，Mini 外壳因其小巧时尚的特性，更会被便携式移动设备（例如，笔记本电脑）的用户所青睐；时钟锁内置时钟芯片，在标准锁功能的基础上增加了时钟控制功能，可安全方便的对加密锁的使用期限进行有效控制；U 盘锁是在标准锁功能的基础上增加了大容量存储功能（U 盘或者光盘形式），将数据存储和软件保护完美的结合在了一起。

1.2 ET-ARM 产品特点

- 高端的配置，高效的速度

ET-ARM 系列产品全部采用了最先进的 32 位 ARM 芯片的高强度智能卡芯片，下载到锁内的代码可以在锁内进行本地执行。其实，你完全可以将 ET-ARM 加密锁视作是一台微型计算机。在采用高端配置的同时，ET-ARM 也兼顾了高效的运算速度和通讯传输速度，同时也采用更加安全的设计方式，使用 ET-ARM 保护后的软件，在安全性和速度方面都得到了质的飞跃。

- 简单易用，高度的安全保证

ET-ARM 以其合理的设计，极大的方便了用户的使用。简单易用的管理工具，规范完善的 API 接口、丰富的多语言示例程序、大量的软件保护解决方案，可使开发商在短时间内就学会使用，更是能清晰的明白加密锁的精髓之所在。

在简单易用和高强度的安全性能之间，ET-ARM 找到了二者完美的契合点。ET-ARM 采用双向随机数通

讯噪音干扰机制，通信加密的密钥的产生采用双向认证方式，不管是从 API 层面，还是从加密锁硬件层面，这样的方式将大大的提高了通讯数据的安全性，也更加有效的防止了破解或通讯数据被监听的可能性。

ET-ARM 采用了三级权限管理机制，开发商密码是由非公开的种子码算法产生的，极大的加强了加密锁的安全强度。通讯数据通过 3DES 进行加解密，通讯的 3DES 密钥随机生成，这样的设计，使用 ET-ARM 保护后的软件是几乎不可能被破解的。

- 双通讯协议，用户自定义切换

ET-ARM 既支持 HID 通讯协议，也支持 CCID 通讯协议。两种通讯协议开发商可根据自己软件的需要进行自由切换，简单方便。有关双通讯协议切换功能的具体介绍（详见 2.1.5 双通讯协议切换）

- 丰富的型号选择，满足用户挑剔的功能和审美需求

从 1.1 ET-ARM 产品型号介绍中就可以看出，ET-ARM 系列为用户提供了多种型号的产品，每种型号的产品都具有各自独特的特点。用户可以根据自己的需求，选择适合自己的产品型号。

- 完美的支持多语言开发，全面的操作系统兼容性

ET-ARM 支持 C/C++、Delphi、BCB、VB、PB、Java、C#等主流开发语言和开发平台，并且为每种语言提供了丰富而详细的示例程序，使得用户能在最短的时间内将 ET-ARM 的高强度保护性能运用到自己的软件中，从而缩短用户宝贵的研发周期。

ET-ARM 支持的系统平台有 Windows2000/XP/2003/2008/Windows7/Windows8。

1.3 ET-ARM 技术参数表

核心芯片	32 位 ARM 高性能智能卡芯片
硬件序列号	全球唯一硬件序列号
数据存储空间	128K（64K 文件存储区+64K 可执行文件存储区）
硬件内置算法	RSA、ECC、SM2、SM3、SM4、3DES、SHA1、专用种子码算法
数据保存年限	≥10 年
接口标准	标准 USB2.0 全速设备
时钟芯片使用年限	≥3 年
读次数	无限制

写次数	≥10 万次
USB 通讯	可选的 HID 或 CCID 通讯协议
保护方式	API 方式保护，外壳方式保护

第2章 ET-ARM 深入详解

ET-ARM 系列加密锁(以下简称 ET-ARM)是一款锁内代码本地执行的 32 位高强度的智能卡芯片加密锁，它集更高速，更方便，更快捷，更安全等特点于一身。它所提供的强大功能，为用户保护自己的软件提供了很强的灵活性，让软件被破解的可能性几乎为零。通过以上的介绍，能够对 ET-ARM 有个初步的认识，本章会将 ET-ARM 功能进行比较详细具体的介绍。接下来，您大约需要十分钟的时间来了解一下内容：

1. ET-ARM 有哪些功能？我们用 ET-ARM 能干什么？
2. ET-ARM 的通讯协议。
3. 不同型号的 ET-ARM 产品各自有哪些特点？

2.1 ET-ARM 基本功能

2.1.1 名词解释

为能更好的理解 ET-ARM 系列加密锁的功能，首先对以下名词进行解释。

产品 ID：使用相应种子码所产生的 8 位 16 进制数，方便开发商产生对自己产品的唯一标识。只要种子码不同，所产生的产品 ID 也不会相同。

用户 ID：由开发商设定的一个 8 位 16 进制数，方便开发商对其最终用户进行标识。

硬件 ID：加密硬件的唯一标识，坚石出厂时固化，不可更改，保证具有全球唯一性。

空锁：坚石出厂的锁，所有信息都是缺省值，特征是产品 ID 为 FFFFFFFF。

子锁：开发商通过种子码唯一化后的锁，特征是产品 ID 不再是 FFFFFFFF。

母锁：在子锁的基础上写入了母锁数据，方便开发商的安全生产，特征是硬件信息中的母锁标志位为 1。

一键恢复：将锁内数据全部清除，恢复到出厂状态，加密硬件将恢复成空锁。

匿名权限：无需任何验证 PIN 码操作，为访问加密锁的最小权限，可进行有限功能操作。

用户权限：需要成功验证用户 PIN 码，其权限高于匿名权限低于开发商权限，拥有部分功能操作权限。

开发商权限：需要成功验证开发商 PIN 码，为最高权限，可操作加密锁的所有功能。

PIN 码：即密码，分为用户 PIN 码和开发商 PIN 码。

2.1.2 数据存储

ET-ARM 提供了 8K 的数据存储区，其中前 4k(0~4095)为低级数据存储区，任意权限都可以对这 4K 的存储区进行读写操作。剩余的后 4K（4096~8191）为高级数据存储区，任意权限可以进行读操作，只有开发商权限才可对这 4K 的区域进行写操作。

2.1.3 共享内存区

ET-ARM 内有一个 32 字节的共享内存区，该区域没有权限限制，锁内可执行程序和锁外 API 都可以访问该内存区，掉电之后数据将被擦除。

2.1.4 文件管理

ET-ARM 的文件系统为用户提供了 128K 可用空间，可以存储数据文件、密钥文件和可执行文件，如此大的存储空间完全能够满足用户的需求，以下针对每种文件类型进行详细的介绍。

- 数据文件

ET-ARM 提供了创建、读写和删除数据文件的功能。数据文件可以和上层软件进行关联，即用 API 进行访问，数据文件也可以被锁内可执行文件读写，据此，开发商可以通过加密锁对自己的数据进行安全的存储。

- 密钥文件

ET-ARM 提供了三种类型的密钥文件，分别为：RSA 私钥文件、ECC 和 SM2 私钥文件、3DES 和 SM4 密钥文件。需要说明的是 ECC 和 SM2 的结构相同，锁内视作同一种文件类型，3DES 和 SM4 的结构相同，锁内也视作同一种文件类型。ET-ARM 可以创建，写入和删除锁内的这些密钥文件，同时还支持这些密码算法的加解密和签名验签运算。与这些密钥文件相关的算法外，ET-ARM 还支持 SHA1 和 SM3 哈希算法，从这些支持的算法中可以看出，ET-ARM 支持的算法涵盖了几乎所有目前主流的密钥算法，完全能够满足用户的需求。

用户可以使用 ET-ARM 所提供的这些密钥文件和相关算法，运用到自己的软件当中，实现远程升级（详见 2.1.6 远程升级）、数字签名、数字信封或者身份认证（详见 3.2 身份认证）等功能。

- 可执行文件

所谓的可执行文件，就是由锁内的处理器芯片进行运算的程序文件。ET-ARM 采用了最高端的 32 位 ARM 高性能智能卡芯片，与 C51 的内核相比，其运算速度有了质的飞跃。用户可以将程序中的核心代码编写成 ARM 可执行程序下载到锁内，在程序运行过程中，调用提供的 API 接口，将可执行文件传入输入数据，通过可执行文件在加密锁内部运行后，将结果返回给外部的应用程序，使被保护的软件与锁进行交互，从而达到软件保护的目的。需要说明的是：出于对安全性的考虑，ET-ARM 的锁内可执行文件下载，采用每次擦除后重新下载的方式。可执行程序访问锁内资源的权限为开发商权限。

与 C51 可执行程序相比，以 ARM 芯片为核心的本地可执行程序的运算速度和性能都要远远超越 C51 可执行程序。

■ 文件访问权限表

1）数据文件

	开发商	用户	匿名
读	√	-	-
写	√	-	-

2）私钥文件

	开发商	用户	匿名
读	×	×	×
写	√	×	×
调用	√	-	-
设定可调用次数	√	×	×

3）密钥文件

	开发商	用户	匿名
读	×	×	×
写	√	×	×
调用	√	-	-

4）可执行文件

	开发商	用户	匿名
读	×	×	×
写	√	×	×
调用	√	-	-

注：“-”表示由开发商权限设定

2.1.5 双通讯协议切换

ET-ARM 首次采用了 HID 和 CCID 两种 USB 通讯协议，并且可以通过 API 在管理员权限下实现协议的自主切换。

HID 通讯协议与 CCID 通讯协议相比，HID 通讯协议的系统兼容性相对好一些，Windows XP 及其以后的 Windows 操作系统无需安装驱动。而 CCID 通讯协议的速度相对会快一些，windows 7 以下的操作系统需要单独安装微软的 CCID 驱动程序。因此，用户可根据自身的需要来灵活的选择通讯协议。

2.1.6 远程升级

当开发商将加密锁出售给最终客户后，若想对锁内的数据进行更新维护，可以采用远程升级的方式。用户无需将已出售的加密锁收回，只需要制作一个远程升级包，升级包通过网络发送给最终客户，方便快捷。由于制作好的升级包是进行过 RSA 加密的，因此远程升级过程具有很高的安全强度的。

除了对已售出的加密锁进行远程升级操作外，在生产阶段也可以采用远程升级的方式进行批量生产，由于此种方式不需要开发商密码，而且升级包采用加密处理，完全可以确保生产过程中的安全性，建议开发商采用。

用户可以通过远程升级的方式对 ET-ARM 进行以下操作：

1. 创建文件（非可执行文件）
2. 写文件（非可执行文件）
3. 删除文件
4. 设置文件权限
5. 设置种子码调用次数
6. 批量下载可执行文件
7. 解锁用户 PIN
8. 修改加密锁使用期限。

2.2 ETTime-ARM（时钟锁）简介

ETTime-ARM 具备 ET-ARM 标准锁的所有功能，其性能也与 ET-ARM 标准锁相同。ETTime-ARM 增加了时钟芯片，时钟芯片在非联机状态下，可以使用三年之久，联机状态下，还可以使用更长的时间，开发商无需顾虑时钟锁使用年限的所带来的问题。开发商可以设置加密锁的使用截止日期和可使用的小时数，可使用的小时数是从第一次验证用户密码后开始计时的。当时钟到期后，用户权限的操作被禁止，所有需要用户权限的操作都无法进行，可以通过管理员权限和远程升级的方式进行解锁。

用户可以根据 ETTime-ARM 时钟锁的计时功能，进行“统一保护，分类销售”的销售模式，根据最终客户的不同，将软件的使用截止时间或者可使用的小时数写入到 ETTime-ARM 中，生产出不同级别的产品。ETTime-ARM 从硬件级别确保了锁内时间的准确性以达到安全的目的。

2.3 StoreET-ARM（U 盘锁）简介

StoreET-ARM 是标准锁的基础上支持文件存储，加密锁的功能与性能方面同标准锁也没有任何差别，在此之外又提供了大容量存储功能，方便用户在使用加密锁的同时也可将该产品视作一款移动存储设备。

第3章 ET-ARM 保护

对于软件的保护方案，每个用户应当需要结合自己软件特点设计出来具有独特性的方案。本章会结合 ET-ARM 的功能特点，介绍几个保护方式，一来为用户提供一些保护软件的方法，二来也可以让用户更加深刻的体会到 ET-ARM 不可超越的，强大的安全强度。你大约需要五分钟的时间来了解一下内容：

1. ET-ARM 如何保护我们的软件？
2. ET-ARM 如何进行身份验证？
3. ET-ARM 如何用子母锁模式？

3.1 基本保护方式

ET-ARM 系列加密锁采用的是 32 位 ARM 高性能智能卡芯片，这就为其强大的运算能力提供了保障，因此 ET-ARM 系列加密锁已经不仅仅是一个存储设备，可以将它完全看做是一个小型的计算机。

计算机最核心的功能就是计算，因此，可以将程序中需要计算的代码放入 ET-ARM 中，应用程序运行过程中，遇到关键算法时，将需要运行的数据传给 ET-ARM，ET-ARM 在加密锁内调用可执行文件，可执行文件运算完成，将结果传给应用程序，程序继续执行。

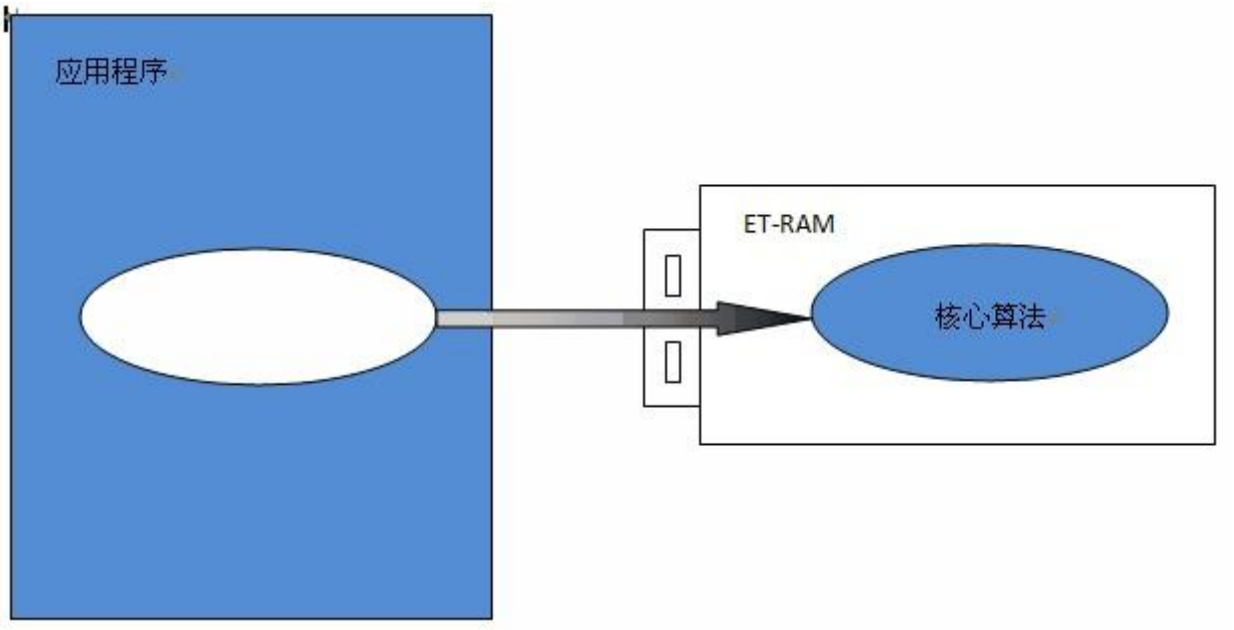

图 1 ET-ARM 基本保护方式

3.2 反调试保护

说到反调试功能，了解破解技术的人第一个联想到的就是赫赫有名的 OllyDbg 的 StrongOD 插件，因为它能使大部分的反调试功能失效。现在介绍的反调试是利用 ET-ARM 的固有功能，设计出硬件级的反调试功能。做法是在基本保护方式的基础上，利用锁内系统 API 接口 ***get_tickcount*** 进行设计。

具体流程为，假如程序 A 中有两个（或两个以上）算法可执行程序 S1 和 S2 可以移植，而 A 程序运行时调用 S1 和 S2 的时间间隔假设是一个固定范围，范围要尽可能的小（范围值可以从软件测试中获得），而且调用可执行程序的最好有固定的顺序（例如调一次 S1 后调一次 S2），这样就满足设计反调试的条件。可在 A 调用 S1 程序时获取一下锁上电以来的时间，并把这个时间值保存在数据区或者数据文件中，当 A 调用 S2 时再次获取上电以来的时间，并与第一次记录的时间进行对比，如果差值超出了正常运行所需时间的范围，则可认定是软件正在进行调试。这时候开发商可以作出你的决定，比如采取销毁锁内的关键数据等任何手段，使加密锁不再能够被 A 程序正常使用，这样一来就有效的防止了程序被软件调试破解，不给破解者任何可乘之机。这样的反调试自毁，能够非常有效的防止软件被跟踪破解。

有心的读者可能会发现，设计这样的功能需要满足很多条件，但是仔细想想，本质上就是分别调用两次可执行程序，至于这两次调用锁内可执行程序是否是软件中移植的算法并不重要（当然最好是软件中需要的算法，因为这样可以增强软件对加密锁的依赖性），而且对一个可执行程序调用两次也没有关系，我们要的只是两次调用之间的一个时间间隔而已，这就给了开发商非常大的想象空间，可据此设计出各种不同种类的反调试自毁功能。另外，最好在锁内存放有一些关键数据，比如软件中用到的加解密的密钥文件，软件正常运行所需要的数据文件等，因为当发现调试的时候，可将这些关键数据销毁掉，不给破解者第二次调试的机会，这样最大程度的保护了软件的安全。

3.3 身份认证

ET-ARM 的 API 接口提供单项散列算法(MD5)，预先在用户的 ET-ARM 和服务器中存储一个证明用户身份的密钥，当需要在网络上验证用户身份时，先由客户端向服务器发送一个验证请求。服务器收到此请求后生成一个随机数传回给客户端。ET-ARM 使用这个随机数与 ET-ARM 中的密钥进行 HMAC_MD5 运算得到一个运算结果，作为认证证据传给服务器。服务器使用这个随机数与服务器中该用户对应的密钥进行相同的运算，如果服务器的运算结果与客户发来认证证据结果一致，那么认为客户端是一个合法用户。

3.4 子母锁

ET-ARM 系列加密锁提供子母锁的机制，通过对母锁进行设置，可以使用同一把母锁生产出相同配置的子锁，而且子母锁的机制还可以运用在远程升级中。通过子母锁的方式进行远程升级，可以在生产流程上更加确保数据的安全，也方便了开发商对售出的加密锁进行管理。以下详细说明一下在生产和销售模式方面如何来灵活的使用子母锁。

（1） 子母锁的生产模式

子母锁的生产模式指的是，用母锁产生的升级包进行批量生产子锁的方式，这种方式可以将子锁生产单独从软件开发过程中分离。在软件发布之前，研发人员只需制作一把母锁，并安全保存好开发商 PIN 码和远程升级密钥，然后使用母锁制作升级包，升级包中包含了所有与软件关联的数据文件和可执行文件，将母锁与升级包交给生产人员，生产人员只需要使用提供的 RyARMInitSon.exe（基于母锁的子锁初始化工具）和 RyARMUpdater.exe（客户端远程升级工具）就可以批量的生产子锁。这样做的好处在于，研发人员或者管理人员可从大量的生产任务中解放出来，将工作重心放在软件开发以及如何保护软件上。在此过程中，生产人员也不会得到子锁内的任何数据信息，也无法解密升级包的内容，有效的防止了在生产流程中造成的数据泄露的情况。

（2） 子母锁的远程升级

当软件产品出售给了最终用户后，如果需要对加密锁内的数据进行升级维护，使用子母锁的方式会更加的简单。开发商可以在生产阶段为每一款产品配置一把母锁，当某个用户需要升级时，只要使用与其匹配的母锁制作升级包，对客户使用的子锁中的数据进行相应的操作即可，这样有利于简化开发商的管理和维护过程，节约成本。如果不使用子母锁的远程升级方式也可以，那么对开发商来说需要将对远程升级的公钥文件进行安全的管理，因为在子锁制作升级包的时候，需要导入相应的公钥进行加密升级包。

（3） 子母锁分类销售

子母锁分类销售，是指如果软件产品有针对不同用户的需求，开发商可针对同一款产品制作不同类别的子母锁，以区分最终用户，保证产品安全。这样的子锁区别在于远程升级的私钥不同，以防止不同类别的用户之间交叉使用同一个升级包的情况。