产品名称 Product name	密级 Confidentiality level
mToken GM3000	
产品版本 Product version	
V1.1	

mToken GM3000 国密 Key 用户手册

Prepared by 拟制		Date 日期	
Reviewed by 评审人		Date 日期	
Approved by 批准		Date 日期	

北京世纪龙脉科技有限公司
Beijing Century Longmai Technology Co., Ltd.

Revision Record 修改记录

Date 日期	Revision Version 修订 版本	Sec No. 修改 章节	Change Description 修改描述	Author 作者
2013/11/15	V1.0		初始版本	

目 录

第一章 前言

随着信息产业化的发展，信息安全问题也日益突出。PKI(Public Key Infrastructure,公共密钥基础设施)网络安全体系被广泛应用。USBKEY 作为结合了现代密码学技术、智能卡技术和USB技术的产品，在PKI应用的各个领域取代原先的认证方式得到广泛应用。

近两年，国家密码管理局推出的 SM2 算法逐渐应用到实际生产环境中，随着时间的推移，基于 SM2 算法产业链的产品会越来越多，使用环境也会越来越成熟，因此基于 SM2 算法的应用也会越来越简单。龙脉科技将顺应技术的发展，基于其密码技术的精湛造诣，致力于信息安全行业，在未来的终端安全市场中做出更大的成绩，不断助推我国公钥密码算法升级。

第二章 产品概述

GM3000 是由龙脉科技设计研发的基于智能卡安全芯片的 USB Key, 可用于网络安全认证和通讯加密等信息安全领域，它是一种 USB 接口设备，体积小巧便于使用者随身携带。中间件符合国密局发布的密码行业标准 GM/T 0016-2012，支持多种语言的 B\S、C\S 调用。

无需安装驱动

GM3000 采用 SCSI/HID/CCID 技术，无需安装额外驱动即可在操作系统中使用，从而彻底消除驱动兼容性风险，保证系统的稳定性。
GM3000 自带光驱，方便最终用户安装中间件。

高性能高安全性

GM3000 使用高性能 32 位 CPU，并通过高速 USB 接口与 PC 相连接，保护数据运算和传输的高性能，同时安全算法如 SM1、SM2、SM4 等均在芯片硬件内部进行，为应用提供安全性保证。

跨平台

GM3000 支持多种操作系统，如 Windows, Linux, Mac OS, 并提供标准 PKCS11 接口，为用户跨平台应用提供的保障。

无缝集成

GM3000 提供了符合业界规范的 Microsoft CryptoAPI 和 PKCS11 接口，并支持多个证书和密钥对，任何兼容这种接口的应用程序都可以立即集成 GM3000 使用。

硬件实现加密算法

详见第三章 GM3000 支持的算法描述。

硬件随机数发生器

GM3000 硬件芯片内置真随机数发生器。

大容量安全数据存储区

GM3000 标准支持 64K 安全数据区用于存放证书、密钥及敏感数据。

自带光驱

GM3000 自带 2M 光驱，可用于存放中间件方便最终用户使用，并可根据用户需求提供最大 8M 空间光驱。

第三章 接口应用规范

国密规范

在使用龙脉 GM3000 国密 KEY 时，建议您参考国家密码局发布的密码行业规范<GM/T 0016-2012 智能密码钥匙密码应用接口规范>.

GM3000 出厂设置

设置项	长度	默认出厂值	说明
Manufacturer	64	Longmai Tech	
Issuer	64	Longmai	
Label	32	GM3000	
TotalSpace	4	65535	64K 设备总空间
MaxRetryCount	4	8	设备认证最大重试次数
RemainRetryCount	4	8	设备认证剩余次数

GM3000 默认设备认证密钥

GM3000默认设备认证密钥：1234567812345678

GM3000 支持的算法描述

算法	默认长度(bit)	最小长度(bit)	最大长度(bit)	用途类别
SGD_SM1_ECB	128	128	256	加解密
SGD_SM1_CBC	128	128	256	
SGD_SM1_CFB	128	128	256	
SGD_SM1_OFB	128	128	256	
SGD_SM1_MAC	128	128	256	
SGD_SM4_ECB	128	128	256	
SGD_SM4_CBC	128	128	256	
SGD_SM4_CFB	128	128	256	
SGD_SM4_OFB	128	128	256	
SGD_SM4_MAC	128	128	256	
SGD_SSF33_ECB	128	128	128	
SGD_SSF33_CBC	128	128	128	

SGD_SSF33_CFB	128	128	128	
SGD_SSF33_OFB	128	128	128	
SGD_SSF33_MAC	128	128	128	
SGD_SM2_2	256	256	256	
SGD_SM2_3	256	256	256	
SGD_RSA	1024	1024	2048	
SGD_SM3	256	256	256	散列
SGD_SHA1	160	160	160	
SGD_SHA256	256	256	256	
SGD_SM2_1	256	256	256	签名验证
SGD_RSA	1024	1024	2048	

GM3000 支持的函数描述

名称	描述	支持情况
设备管理系列函数		
SKF_WaitForDevEvent	等待设备的插拔事件	已实现
SKF_CancelWaitForDevEvent	关闭等待事件	已实现
SKF_EnumDev	获得当前系统中的设备列表	已实现
SKF_ConnectDev	通过设备名称连接设备，返回设备的句柄	已实现
SKF_DisconnectDev	断开一个已经连接的设备，并释放句柄	已实现
SKF_GetDevState	获取设备是否存在的状态	已实现
SKF_SetLabel	设置设备标签	已实现
SKF_GetDevInfo	获取设备的一些特征信息	已实现
SKF_LockDev	获得设备的独占使用权	已实现
SKF_UnlockDev	释放对设备的独占使用权	已实现
SKF_Transmit	将命令直接发送给设备，并返回结果	已实现
访问控制系列函数		
SKF_ChangeDevAuthKey	更改设备认证密钥	已实现
SKF_DevAuth	设备认证是设备对应用程序的认证	已实现
SKF_ChangePIN	修改PIN，可以修改Admin和User的PIN，如果原PIN错误，返回剩余重试次数，当剩余次数为0时，表示PIN已经被锁死	已实现
SKF_GetPINInfo	获取PIN码信息	已实现
SKF_VerifyPIN	校验PIN码。	已实现
SKF_UnblockPIN	该函数来解锁用户PIN码	已实现
SKF_RemoteUnblockPIN	远程解锁用户PIN	已实现
SKF_ClearSecureState	清除应用当前的安全状态	已实现
应用管理系列函数		
SKF_CreateApplication	创建一个应用	已实现
SKF_EnumApplication	枚举设备中所存在的所有应用	已实现
SKF_DeleteApplication	删除指定的应用	已实现
SKF_OpenApplication	打开指定的应用	已实现
SKF_CloseApplication	关闭应用并释放应用句柄	已实现
文件管理		
SKF_CreateFile	创建一个文件。创建文件时要指定文件的名称，大小，以及文件的读写权限	已实现
SKF_DeleteFile	删除指定文件	已实现
SKF_EnumFiles	枚举一个应用下存在的所有文件	已实现

SKF_GetFileInfo	获取应用文件的属性信息	已实现
SKF_ReadFile	读取文件内容	已实现
SKF_WriteFile	写数据到文件中	已实现
容器管理系列函数		
SKF_CreateContainer	在应用下建立指定名称的容器	已实现
SKF_DeleteContainer	在应用下删除指定名称的容器	已实现
SKF_EnumContainer	枚举应用下的所有容器并返回容器名称列表	已实现
SKF_OpenContainer	获取容器句柄	已实现
SKF_CloseContainer	关闭容器句柄，并释放容器句柄相关资源	已实现
SKF_GetContainerType	获取容器的类型	已实现
SKF_ImportCertificate	向容器内导入数字证书	已实现
SKF_ExportCertificate	导出容器内的数字证书	已实现
密码服务系列函数		
SKF_GetRandom	产生指定长度的随机数	已实现
SKF_GenRSAKeyPair	由设备生成RSA密钥对并明文输出	已实现
SKF_ImportRSAKeyPair	导入RSA加密公私钥对	已实现
SKF_RSASignData	使用hContainer指定容器的签名私钥，对指定数据pbData进行数字签名	已实现
SKF_RSAVerify	验证RSA签名。用pRSAPubKeyBlob内的公钥值对待验签数据进行验签	已实现
SKF_RSAExportSessionKey	生成会话密钥并用外部公钥加密输出	已实现
SKF_GenECCKeyPair	生成ECC签名密钥对并输出签名公钥	已实现
SKF_ImportECCKeyPair	导入ECC公私钥对	已实现
SKF_ECCSignData	ECC数字签名。采用ECC算法和指定私钥hKey，对指定数据pbData进行数字签名	已实现
SKF_ECCVerify	用ECC公钥对数据进行验签	已实现
SKF_GenerateAgreement DataWithECC	为计算会话密钥而产生协商参数，密钥交换发起方调用	已实现
SKF_GenerateAgreement DataAndKeyWithECC	产生协商参数并计算会话密钥，密钥交换响应方调用	已实现
SKF_GenerateKeyWithECC	使用自身协商句柄和响应方的协商参数计算会话密钥，密钥交换发起方调用	已实现
SKF_ECCExportSessionKey	生成会话密钥并用外部公钥加密输出。	已实现
SKF_ExportPublicKey	导出容器中的签名公钥或者加密公钥	已实现
SKF_ImportSessionKey	导入会话密钥	已实现
SKF_EncryptInit	数据加密初始化。设置数据加密的算法相关参数	已实现

SKF_Encrypt	单一分组数据的加密操作	已实现
SKF_EncryptUpdate	多个分组数据的加密操作	已实现
SKF_EncryptFinal	结束多个分组数据的加密，	已实现
SKF_DecryptInit	数据解密初始化，设置解密密钥相关参数	已实现
SKF_Decrypt	单个分组数据的解密操作	已实现
SKF_DecryptUpdate	多个分组数据的解密操作	已实现
SKF_DecryptFinal	结束多个分组数据的解密	已实现
SKF_DigestInit	初始化消息杂凑计算操作，指定计算消息杂凑的算法	已实现
SKF_Digest	对单一分组的消息进行杂凑计算	已实现
SKF_DigestUpdate	对多个分组的消息进行杂凑计算	已实现
SKF_DegistFinal	结束多个分组消息的杂凑计算操作，将杂凑保存到指定的缓冲区	已实现
SKF_MACInit	初始化消息认证码计算操作，设置计算消息认证码的密钥参数	已实现
SKF_MAC	计算单一分组数据的消息认证码	已实现
SKF_MACUpdate	计算多个分组数据的消息认证码	已实现
SKF_MACFinal	结束多个分组数据的消息认证码计算操作	已实现
SKF_CloseHandle	关闭会话密钥、杂凑、消息认证码句柄	已实现

GM3000 重点函数解析

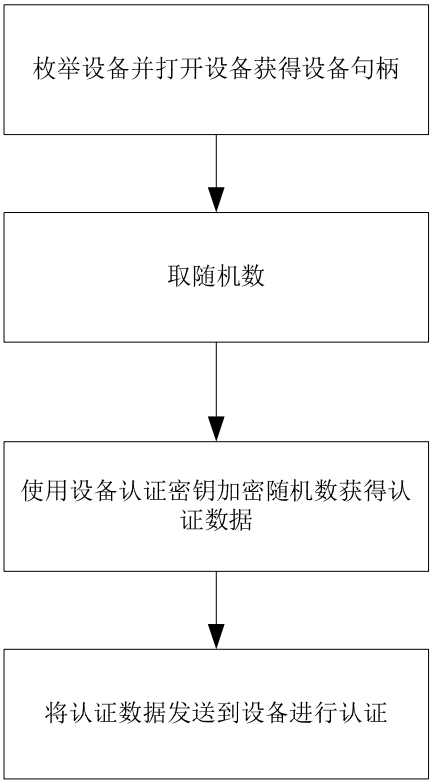
函数	使用说明
SKF_LockDev	当进行加密、解密、摘要、MAC 运算时，运算过程不能被其它运算打断， 直到调用 Final 函数才能进行另外一个运行，所以要求在做加解密等运算时先调用 SKF_LockDev，锁定设备，其它进程再调用加解密等运算调用 SKF_LockDev 时函数将被锁定，直到上一次锁定解锁。
SKF_DevAuth	设备认证函数，认证流程如下： 调用 SKF_GenRandom 函数，获取随机数； 将随机数发送到应用的服务端、或设备外部进行运算； 认证密钥做 key，使用 SM4 算法加密随机数； 将加密后的随机数传入 SKF_DevAuth，进行认证。 注意：认证流程不能被打断，SKF_GenRandom 产生随机数后，随机数将保留在设备的随机数发生器内。
SKF_ChangeDevAuthKey	修改设备认证函数，流程如下： 调用 SKF_GenRandom 函数，获取随机数；

	将随机数发送到应用的服务端、或设备外部进行运算； 认证密钥做 key，使用 SM4 算法加密新的认证密钥； 认证密钥做 key，随机数做 IV，对新的认证密钥做 SM4 的 MAC； 将新认证密钥的密文及 MAC 一起传给函数 SKF_ChangeDevAuthKey，进行设备认证；
SKF_RemoteUnblockPIN	此函数为龙脉科技在国密的基础上扩展的函数，目的主要是提高设备的安全性。使用流程： 在服务端对管理员 PIN 做 SHA1； SHA1 结果取前 16 字节作为 key，用 SM4 算法加密新的用户 PIN； 使用随机数做 IV，对 key 做 MAC 将加密后的密文及 MAC 传到客户端，并调用 SKF_RemoteUnblockPIN 函数； 注意：认证失败时管理员 PIN 的错误次数加 1
SKF_ReadFile	读取文件前可以先调用 SKF_GetFileInfo，获取文件长度，读取时需要传入偏移量，如果偏移量超过文件长度，函数将报错。
SKF_EncryptInit SKF_Encrypt SKF_EncryptUpdate SKF_EncryptFinal	加密前需要先锁定设备； 加密前需要先调用SKF_EncryptInit； 对于加密内容不长，一包数据能够处理时，可以调用 SKF_Encrypt函数，此函数只能调用一次，调用完成后会关闭加密引擎，重新调用需要先调用SKF_EncryptInit； 加密数据量大时可以调用 SKF_EncryptUpdate ，分组加密； 分组加密完成时要调用 SKF_EncryptFinal。 解密、摘要、MAC 算法同上。

第四章 产品使用

设备认证

设备出厂时，预置设备认证密钥，在此阶段只能修改设备认证密钥或创建应用操作，禁止其他任何操作。



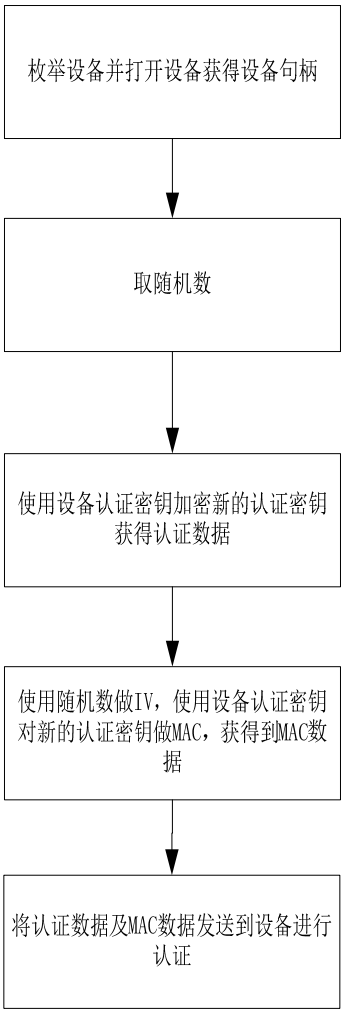
认证流程

认证过程中，产生随机数后，认证流程不能被其它产生随机数的流程打断。

认证时用外部设备或算法，使用设备认证密钥加密随机数。

修改认证密钥

设备认证完成后才能修改设备认证。



修改设备认证

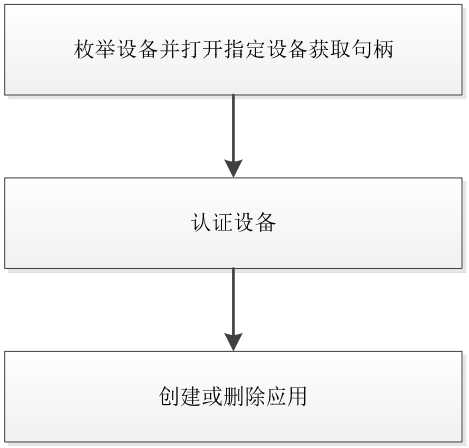
修改设备认证过程中，产生随机数后，认证流程不能被其它产生随机数的流程打断。

修改设备认证时用外部设备或算法，使用设备认证密钥加密新的认证密钥，并使用随机数做 IV，用设备认证密钥对新的认证密钥做 MAC。

应用管理

设备认证通过后可以创建或删除应用。

GM3000 支持多达 8 个应用。



创建或删除应用

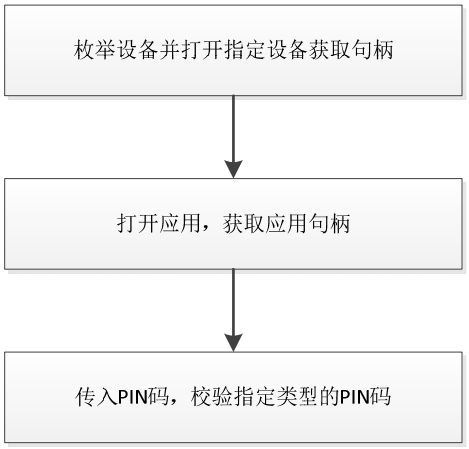
如果只是打开应用，不需要创建或删除应用，则无需设备认证。

创建应用时需指定管理员 PIN、用户 PIN 和应用下文件的读写权限。

PIN 码管理

PIN 码管理需先打开应用。

支持验证 PIN 码，修改 PIN 码和解锁 PIN 码。



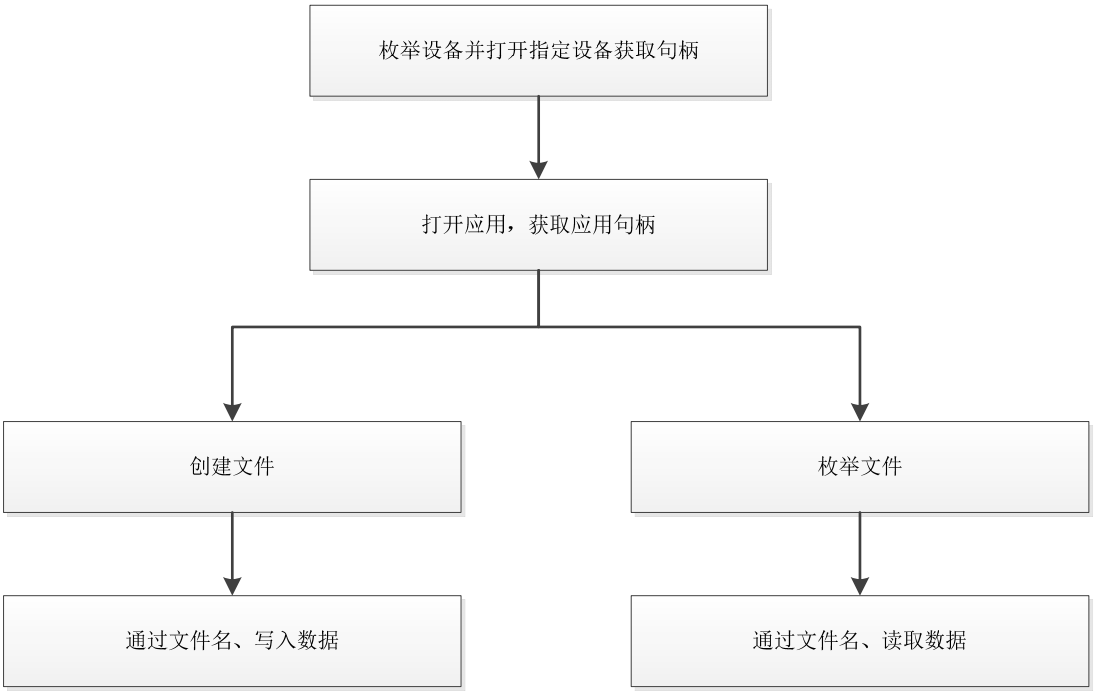
PIN 码管理

修改 PIN 码，需校验原 PIN。

解锁 PIN 码，需要管理员 PIN。

文件管理

读写或创建文件需要先验证用户身份，用户权限与创建应用时输入的权限相同。



文件管理

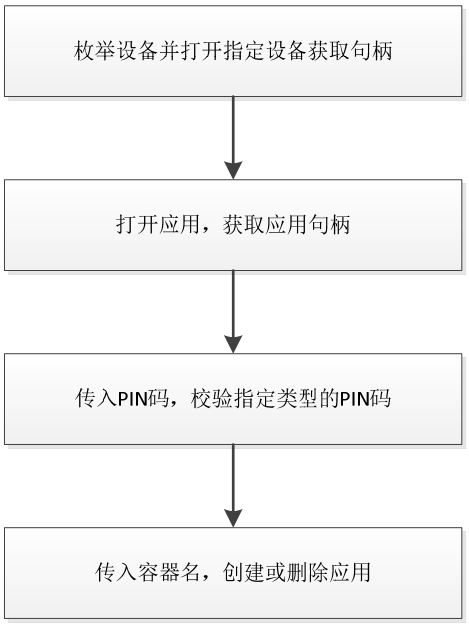
读取或写入文件时，需要传入读取或写入的文件位置偏移量。
偏移量为 0，表示从文件头开始读取或写入。

容器管理

创建或删除容器，需先打开应用并登陆。

GM3000 最大支持 255 个容器（与存储空间相关）。

枚举或打开容器无需登陆。

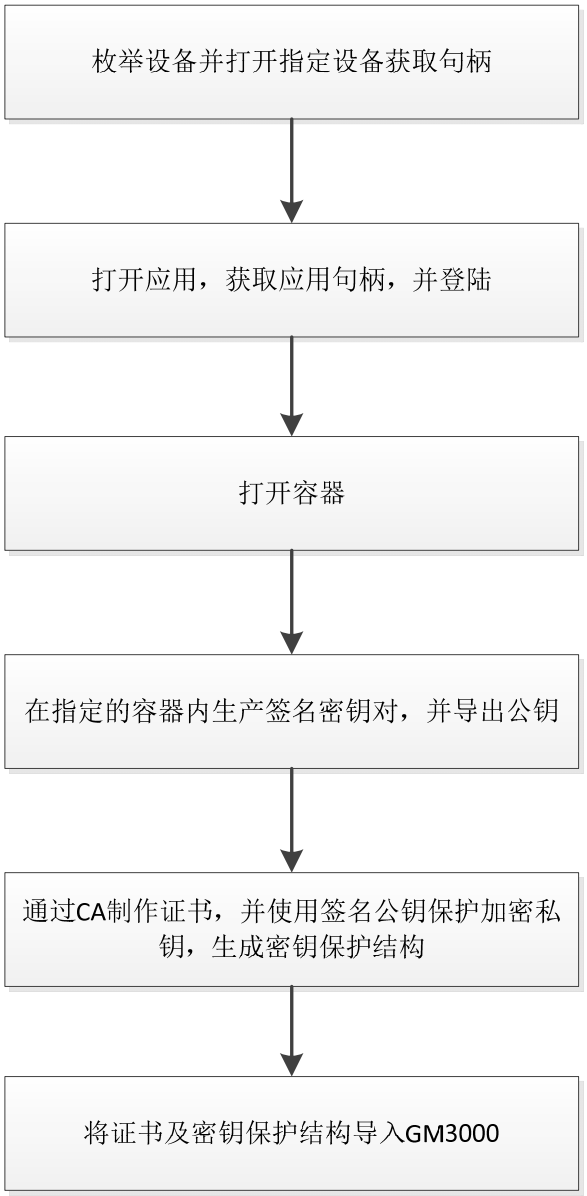


容器管理

容器用于存放非对称密钥和会话密钥。

证书制作

制作证书时需要先通过身份认证，校验 PIN 码。



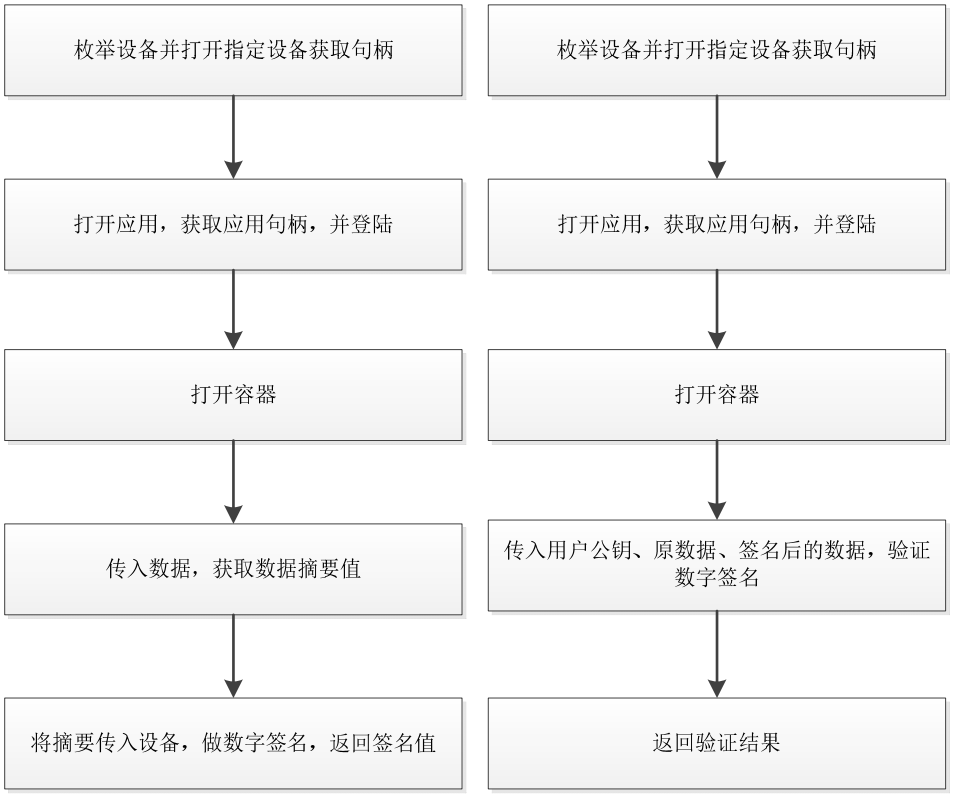
证书制作

签名证书和加密证书导入时需要对应签名密钥和加密密钥。

非对称密钥支持 SM2 和 RSA1024、RSA2048。

数字签名

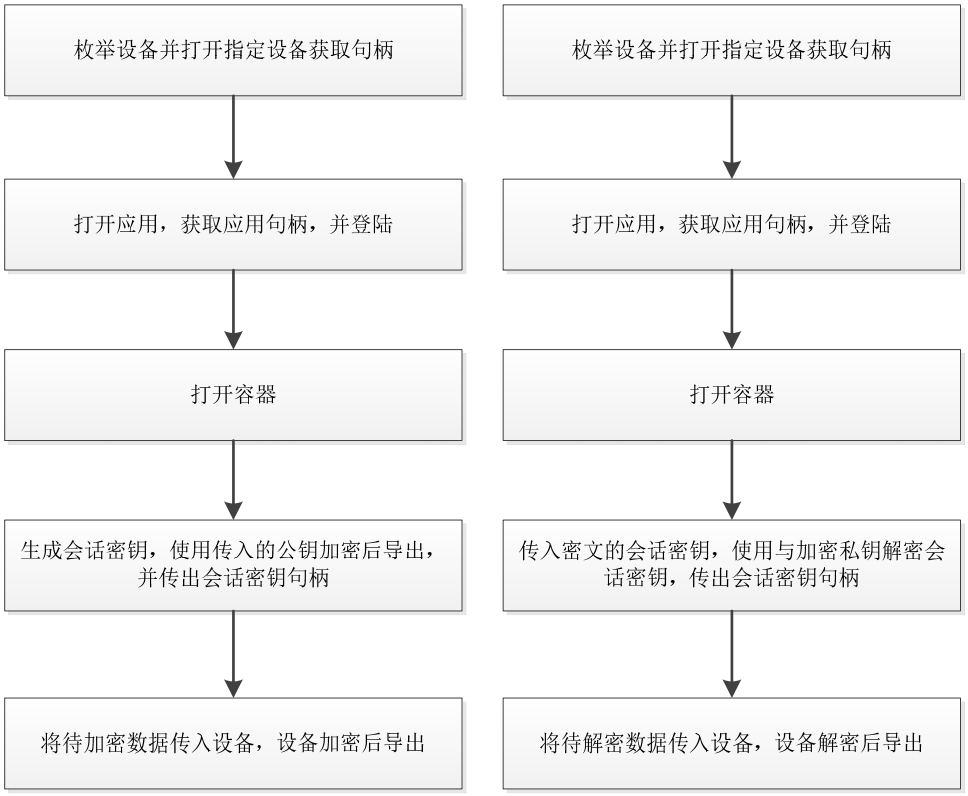
签名处理需要先打开容器，签名算法与打开的容器内密钥对的算法相同。签名前需要先做摘要。



数字签名　　数据验签

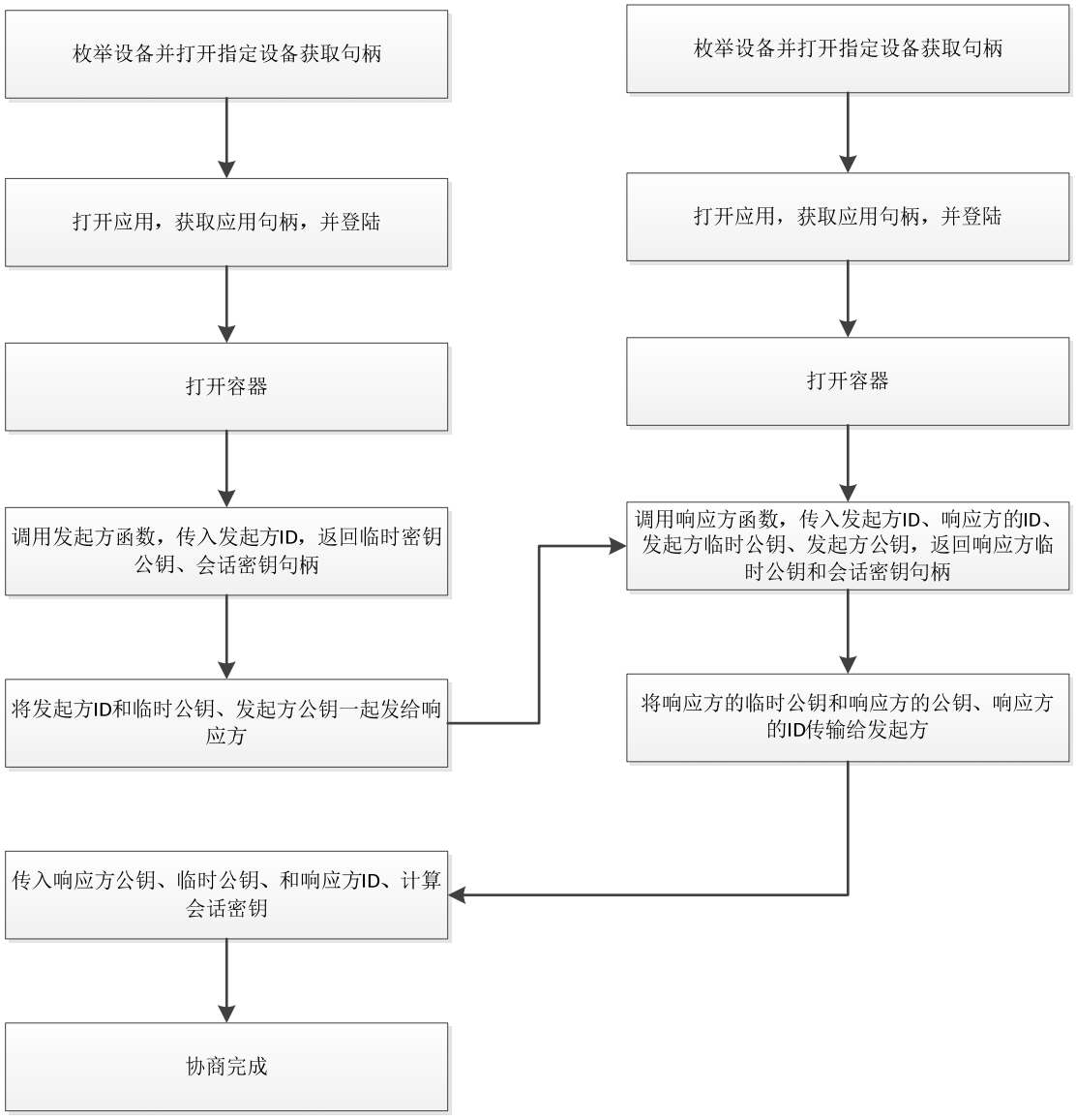
加解密

数据加密需要先导出会话密钥。数据解密需要先导入会话密钥。
导出的会话密钥使用传入的公钥加密后导出。



加解密时，如果数据量过大，需要调用分组加解密函数。

密钥协商



密钥协商

联系我们

公司总部

地址：北京市海淀区王庄路甲1号工控办公楼三层

电话：010-62323636

传真：010-62313636

热线：400-666-0811 (7x24免费热线)

邮箱：longmai@longmai.com.cn

网站：http://www.longmai.com.cn

商城：http://smart2000.taobao.com/

邮编：100083

杭州办事处

地址：浙江省杭州市西湖区文三西路499号西溪风尚5幢420

电话：0571-86908895

邮箱：hangzhou@longmai.com.cn

邮编：310012

广州办事处

地址：广东省广州市天河区黄埔大道中路262号恒安大厦恒福轩14D

电话：020-85272610

邮箱：guangzhou@longmai.com.cn

邮编：510630

申请试用

如果您对产品感兴趣，可先申请试用，测试通过后再进行购买。申请试用，请到龙脉官方网站填写试用单或直接打电话与我们联系：

试用：http://www.longmai.com.cn/apply/index.htm

邮箱：sales@longmai.com.cn

购买产品

如果您希望咨询产品价格或正式购买产品，可以通过如下方式与销售人员联系：

电话：010-62323636

邮箱：sales@longmai.com.cn

技术支持

我们提供了多种方式的技术支持服务，您可通过如下方式向技术人员咨询：

电话：010-62323636-661

邮箱：support@longmai.com.cn