ICS 35.040
L 80
备案号：

中华人民共和国密码行业标准

GM/T XXXX—XXXX

密码设备管理技术规范

Public Key Infrastructure Application Technology

Basic Specifications of Cryptography Device Management

（征求意见稿）

在提交反馈意见时，请将您知道的相关专利连同支持性文件一并附上。

××××-××-××发布　　××××-××-××实施

国家密码管理局 发布

目　次

前　言

本标准依据GB/T1.1—2009给出的规则起草。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别这些专利的责任。

本规范由密码行业标准化技术委员会提出并归口。

本规范的附录A、附录B是规范性附录。

本规范起草单位：兴唐通信科技有限公司、无锡江南信息安全工程技术中心、成都卫士通信息产业股份有限公司、济南得安计算机技术有限公司、上海格尔软件股份有限公司、北京海泰方圆科技有限公司、长春吉大正元信息技术股份有限公司、北京数字证书认证中心有限公司、上海市数字证书认证中心有限公司、万达信息股份有限公司。

本规范主要起草人：刘平、王妮娜、李玉峰、林岳嵩、王海霞、徐强、李元正、高志权、谭武征、柳增寿、李伟平、李述胜、韩玮、周栋。

本规范凡涉及密码算法相关内容，按国家有关法规实施。

引　言

密码设备管理向上层管理应用提供设备管理应用接口，为实现远程密钥管理、设备维护、设备监控、设备合规性检查等上层管理应用提供设备管理功能，将上层管理应用的管理请求转换为标准的消息调用，通过安全协议建立应用层安全通道，实现管理应用与密码设备间的消息传递。

本规范规定了密码设备管理的应用接口、管理流程、管理协议、管理信息结构，明确了密码设备实现管理代理的具体要求，实现设备管理应用与具体密码设备的无关性，达到依据本规范设计开发的密码设备可以由依据本规范开发的管理系统进行统一管理、统一配置的目的。本规范为密码设备和上层管理应用的研制和开发提供指导和依据。

本规范制定一套密码设备管理应用接口，确定密码设备实现管理代理的具体要求，实现设备管理应用与具体密码设备的无关性，达到依据本规范设计、开发的密码设备，可以进行统一管理、统一配置的目的。

本规范 5、6、7、8、9 章针对密码设备管理系统厂商开发商使用。

本规范 5、6、7、8 章针对密码设备厂商使用。

本规范 5、9 章针对管理应用厂商使用。

本规范的编制过程中得到了国家商用密码应用体系总体工作组的指导。

密码设备管理技术规范

1　范围

本规范规定了密码设备管理的体系结构、管理流程、安全通道协议、管理信息结构、应用接口和标准管理消息格式。

为应用技术体系框架内的密码设备和上层管理应用的研制和开发提供指导和依据。

本规范适用于密码设备管理系统、密码设备管理应用、密码机等密码设备的研制和开发，也可用于指导密码设备管理系统、密码设备的检测。

2　规范性引用文件

下列文件对于本文件是必不可少的。凡是注日期的引用文件，仅注日期的版本适用于本文件，凡是不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

GM/T 0006 密码应用标识规范

GM/T 0015 基于 SM2 密码算法的数字证书格式

GM/T 0018 密码设备应用接口规范

FIPS198　Key-Hash Message Authentication Code

3　术语和定义

下列术语及定义仅适用于本规范。

3.1

密码设备　cryptography device

为密钥等秘密信息提供安全存储，并基于这些秘密信息提供密码安全服务的设备。本规范中，专指可以接受设备管理操作的密码设备，主要包括网络密码机、应用密码机/卡；但不包括智能密码终端、密码芯片等部件级设备。

3.2

设备证书　device certificate

可以标识密码设备身份的数字信息，包含密码设备的基本信息、设备公钥信息及其他补充信息等。设备证书可以由专门的 CA 系统签发，也可以由设备管理平台签发。

3.3

安全通道　security tunnels

通过设备管理中心与密码设备管理代理之间的数据交互安全协议建立起来的应用层安全连接，目的是为设备管理应用与密码设备之间的应用层信息交互提供机密性和完整性保护。

3.4

设备密钥 device key pair

存储在设备内部的用于设备管理的非对称密钥对，包含签名密钥对和加密密钥对。

3.5

被管对象　be-managed object

指接受管理的密码设备，通过设备管理代理成为被管对象。

3.6

设备管理代理 device-managed agent

设备管理代理是实现安全通道建立、设备管理消息解析的逻辑实体，它处理设备管理中心下发的消息命令，将处理结果返回给设备管理中心。每个设备管理代理对应一个密码设备，设备管理代理可以在密码设备内部实现，也可以由密码设备外部主机实现。如果在外部实现，必须保证外部设备代理和所代理密码设备之间的安全连接。

3.7

安全通道消息 security tunnels message

指密码设备管理平台在被管设备与管理中心之间建立和维护安全会话连接的初始化协议消息。

3.8

管理应用 manage application

指密码设备管理、密钥管理、设备维护、设备监控等对密码设备进行状态或数据管理的应用。

3.9

密码设备管理平台 cryptography device management platform

为管理应用提供与被管对象建立远程安全通道的管理系统。

3.10

密码设备管理消息 cryptography device management message

在密码设备管理平台上发送的对密码设备进行远程管理和控制的协议消息。

3.11

密码设备管理信息 cryptography device management data

指对密码设备管理系统对密码设备进行远程管理时查询或配置的标准数据。

4 符号和缩略语

下列缩略语适用于本部分。

API：应用程序接口(Application Program Interface)

CA：证书认证中心（Certification Authority）

PDU：分包数据单元（Package Data Unit）

AID：被管对象的属性标识符(Attribute ID)

5 密码设备管理体系

5.1 密码设备管理在密码基础设施应用技术体系框架中的位置

密码设备管理服务位于公钥密码基础设施应用技术体系框架中的通用密码服务层，采用设备管理总中心、设备管理分中心两级树形结构体系。

密码设备管理平台向上层管理应用提供设备管理应用接口，为远程密钥管理、设备维护、设备监控等上层管理应用提供设备管理功能，通过安全通道进行传递应用相关的管理消息。

密码设备管理平台在提供设备管理服务时调用管理中心专用密码设备的密码设备应用接口以及外部证书认证系统提供的服务，密码服务要求的密码运算在密码设备内部实现。

密码设备管理服务在密码基础设施体系结构中的位置如图 1 所示：

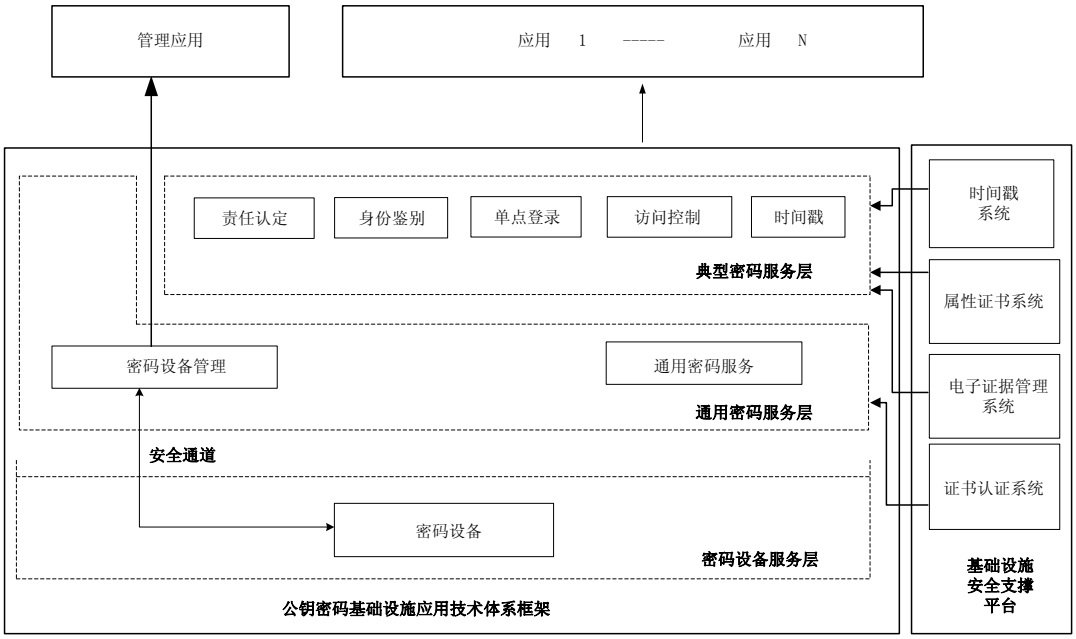

图 1 密码设备管理接口在密码基础设施体系结构中的位置

5.2 密码设备管理平台结构

密码设备管理体系采用两级树形结构体系，基于设备证书的管理机制。建立密码设备管理体系首先是在密码设备管理层的各级设备管理中心和密码设备层的被管对象中安装相应的设备证书。设备证书可由密码设备管理体系中的设备管理总中心签发，也可以由外部 CA 签发。在该树形体系中，各级设备管理中心之间、设备管理中心和被管对象间都是通过设备证书进行密钥协商，通过安全协议建立安全通道，实现管理信息的安全传递。

5.3 密码设备管理应用体系结构

密码设备管理应用体系为如图 2 所示：

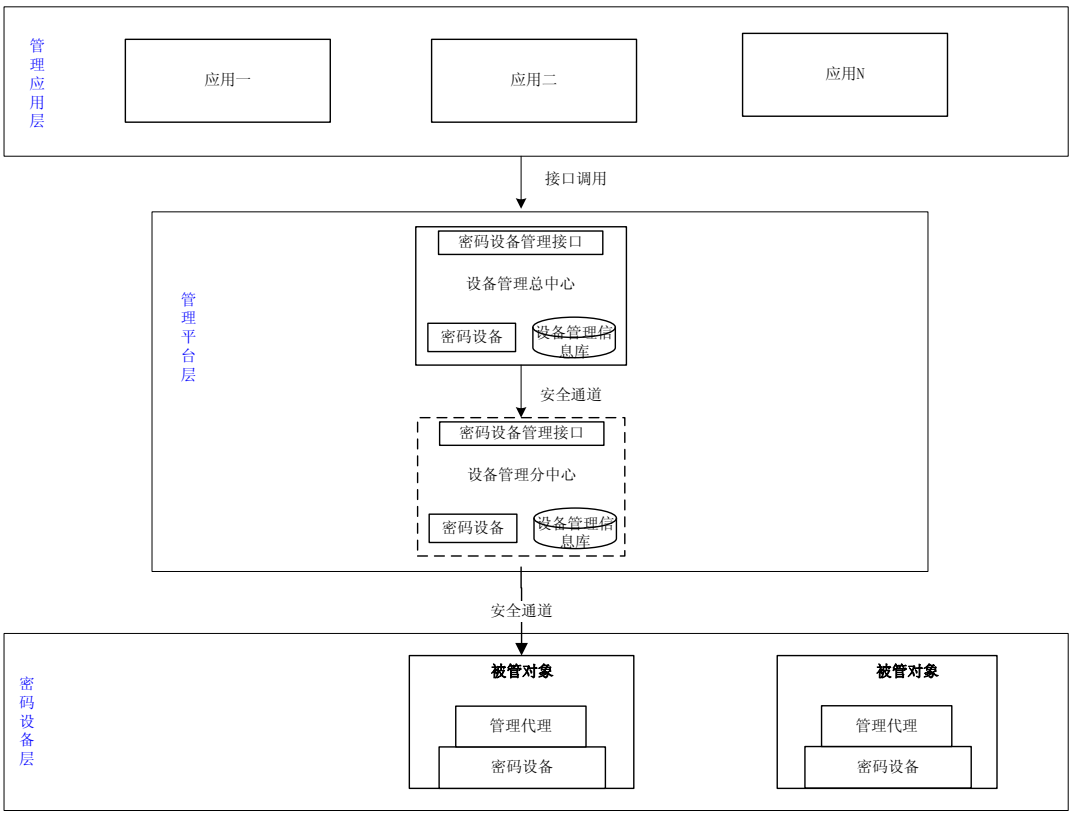

图 2 密码设备管理应用体系结构图

密码设备管理以数字证书体系为基础，按照功能进行层次划分可分为三层，分别为：管理应用层、管理平台层和密码设备层。管理应用层主要是对密码设备进行的各种具体的管理应用，包括：远程密钥管理、远程设备监控、远程设备维护、远程设备合规性检测等。管理平台层包括设备管理总中心、设备管理分中心，其主要作用是在设备管理中心与密码设备之间构建一条安全通道，使得上层管理中心可以准确定位底层密码设备，同时与其进行安全的数据交互。密码设备层主要包括各种密码设备如：服务器密码设备、密码卡等，但不包含终端型密码设备如：UKEY、IC 卡等。

密码设备管理平台与密码设备之间，通过安全协议实现双向身份认证，建立应用层安全通道并通过所协商的会话密钥实现传输指令和数据的机密性、完整性保护。各种管理应用通过密码设备管理平台与密码设备之间建立的安全通道对密码设备进行各种具体的管理。

5.4 管理应用层

管理应用为密码设备管理、密钥管理、设备维护、设备监控等对密码设备进行状态或数据管理的应用。管理应用通过设备管理总中心提供的密码设备管理接口，获取被管设备在数据库中的基本信息，并向被管设备发送各应用的具体管理指令。

5.5 设备管理平台层

5.5.1 设备管理平台层结构及功能

设备管理平台层即密码设备管理服务层，只采用两级结构，由设备管理总中心、设备管理分中心组成。

密码设备管理总中心可以下设多个密码设备管理分中心。管理总中心为上层的管理应用提供设备管理功能，总中心通过分中心统一管理所有被管对象，分中心不对管理应用提供服务。总中心与分中心、分中心与密码设备管理代理之间分别建立安全通道。

密码设备管理平台的主要功能是通过标准的密码设备管理接口为管理应用层提供安全

通道服务；

5.5.2 设备管理总中心

设备管理总中心主要功能有：

- 为设备管理分中心和被管对象签发设备证书（可选）；
- 处理设备管理分中心和被管对象的注册和注销请求；
- 实现密码设备管理接口，提供与管理应用层交互的接入点；
- 提供设备监测管理功能：
  - 与设备管理信息库进行交互，对被管对象进行基础维护和管理。包括设备轮询时间的设置、轮询属性的添加和修改、设备告警信息的处理等；
  - 对下级管理中心和被管对象的管理：包括设备状态查询、下级中心上报设备信息的接收处理等；

5.5.3 设备管理信息库

在管理平台层的各级中心，都设有设备管理信息库，信息库中所包含的设备基本信息主要有以下几项：

表 1 设备基本信息表

信息内容	说明
设备唯一标识	由设备管理总中心根据设备用户单位编号、设备型号、设备编号三项内容串接组成的，是密码设备在设备管理系统中的唯一标识。
设备名称	一个容易识别，容易记忆的名称。
设备厂商	设备生产厂商名称
设备型号	由主管部门统一批准的设备型号
设备序列号	厂商对设备的自行编号，包含：生产日期（8 字符）、生产批次号（3 字符）、流水号（5 字符）
设备软件版本号	密码设备内部软件版本号
设备支持接口规范版本号	密码设备支持的接口规范版本号
设备证书	密码设备的设备证书，证书格式符合 GM/T0015 规范
设备非对称算法能力描述	说明密码设备所支持的各种非对称算法，具体描述方式参见 GM/T0018 中设备信息定义的相关内容。
设备对称算法能力描述	说明密码设备所支持的各种对称密码算法，具体描述方式参见 GM/T0018 中设备信息定义的相关内容。
设备杂凑算法能力描述	说明密码设备所支持的各种杂凑密码算法，具体描述方式参见 GM/T0018 中设备信息定义的相关内容。
设备类型	总中心，分中心或被管对象
设备当前状态	设备当前是否工作正常
设备所属中心标识串	设备树形结构中所属父节点的设备唯一标识串，即该设备的拓扑结构。由本级管理中心到该设备的父节点的唯一标识按照管理层级、自顶至下排列、中间用英文句号隔开，格式为：总中心 ID.分中心 ID。
设备 IP 地址	设备的 IP 地址

设备管理总中心的设备管理信息库中存储了包括设备管理分中心在内的所有被管对象的基本信息。分中心的设备信息管理库里只存储所属被管对象的信息。

5.5.4 设备管理分中心

设备管理分中心的主要功能有：

- 维护其所辖范围内的被管对象；

- 为其所辖范围内的被管对象提供设备证书申请和设备注册代理服务；
- 处理或转发设备管理总中心与被管对象之间的消息；
- 定期轮询其所辖范围内的被管对象的工作状态，收集整理后提交设备管理总中心。

设备管理分中心设有分中心的设备管理信息库，存储其所辖范围内的被管对象的基本信息，内容同表 1。

5.6 密码设备层

密码设备层是指各种具体的被管对象，包括密码机、密码卡等提供密码服务功能的设备。被管对象通过设备管理代理与设备管理平台进行消息交互。

被对象通过安全通道与管理中心相连。在设备端需实现管理代理，提供接受系统管理的能力。对被管对象的基本要求如下：

- 算法资源要求
  - 支持管理平台的设备管理算法，否则中心会拒绝该被管对象的注册，管理算法应使用国家认可的密码算法，使用对称算法 SM4，使用非对称算法 SM2，使用摘要算法 SM3；
  - 可以产生公私钥对以产生证书申请，或者从外部证书认证系统获得证书；
  - 对证书的安全存储要求：可以安全存储被管对象自身证书和私钥，以及直属分中心和总中心证书；
- 管理代理要求：应实现管理代理，完成安全通道建立，响应标准管理消息包。
  - 内部管理代理：管理代理在被管对象内实现，实现安全通道协议。安全通道用到的密码服务功能，由被管对象内部提供。
  - 外部管理代理：对于某些无法支持管理代理功能的被管对象，可以采用外部设备管理代理进行维护。外部设备管理代理应存储被代理设备的设备证书，负责处理所有的管理类操作。
  - 外部设备管理代理操作流程：
    - 设备管理平台通过安全通道访问外部设备管理代理，发出操作请求。
    - 外部设备管理代理对接收的数据包进行包解析，确认目标是所代理的某台被管对象。
    - 外部设备管理代理将指令中数据 PDU 解密获得管理应用的操作指令，通过内部途径将操作指令转交访问被管设备执行。
    - 操作结果用外部管理代理建立的安全通道返回给设备管理平台。

5.7 设备证书管理

设备证书可以由设备管理总中心签发，也可以由第三方 CA 签发。

设备证书的申请、更新等管理过程遵循有关 CA 系统的相关管理技术规范。

5.8 注册流程

5.8.1 注册要求

系统中的所有被管对象在使用前需在设备管理总中心进行注册，用以获得设备唯一标识。

当设备证书由设备管理总中心签发时，设备注册时需提交设备信息表、证书申请和支持的算法，总中心负责分配设备唯一标识和签发证书。

当设备证书由外部第三方 CA 签发时，设备注册时需提交设备信息表和支持的算法，总中心负责分配设备唯一标识。

注册分为设备管理分中心注册和被管设备注册流程。

5.8.2 设备管理分中心注册

设备管理分中心注册流程如下：

1. 设备管理分中心产生证书申请。
2. 设备管理分中心将设备信息表、证书申请和支持的算法一起提交给设备管理总中心，算法标识定义参见 GM/T0006。
3. 设备管理总中心对申请进行审核。审核通过则在设备管理信息库中记录该分中心信息。如果不支持分中心使用的算法则拒绝该申请。
4. 设备管理总中心签发分中心双证书，证书格式符合《基于 SM2 算法的数字证书格式》规范，导出将分中心设备证书和总中心设备证书，下发给设备分中心。
5. 设备管理分中心导入分中心设备证书和总中心设备证书。

注：分中心注册的信息传递采用离线方式。

5.8.3 被管对象注册

被管对象注册流程如下：

1. 被管设备产生证书申请。
2. 被管设备将设备信息表、证书申请和支持的算法一起提交给设备管理分中心，算法标识定义参见 GM/T0006。
3. 设备管理分中心通过与总中心之间建立的安全通道，将该注册申请转发给总中心。
4. 设备管理总中心对申请进行审核。审核通过则在设备管理信息库中记录该设备信息。如果不支持设备采用的算法则拒绝该申请。
5. 设备管理总中心签发设备证书，导出设备证书和总中心设备证书导出，通过安全通道下发给设备分中心。
6. 设备管理分中心将设备证书，分中心证书，总中心证书下发给被管设备。
7. 被管设备导入设备证书、分中心设备证书和总中心设备证书。

注：被管设备向管理分中心的注册为离线方式，分中心已注册，与总中心间通过安全通道传递注册信息。

6 安全通道消息

6.1 安全通道协议

安全通道协议是设备管理中心与密码设备管理代理之间的管理信息交互应用安全协议，实现设备管理应用与密码设备之间应用层安全连接的建立，为应用层信息交互提供机密性和完整性保护。安全通道协议框架见附录 B。

6.2 安全通道消息

6.2.1 安全通道消息格式定义

安全通道消息是密码设备管理平台在被管设备与管理中心之间建立和维护安全会话连接的初始化协议消息，用于确保会话两端的可信身份，以及所承载管理消息的机密性和完整性。

本部分定义了安全通道消息的格式，并说明了安全通道建立和使用的时机。

下文消息格式中的签名值为对图 3 所示结构的除签名值以外的所有内容的签名，由消息产生者签名。采用 SM2 算法时签名的格式遵循 GM/T0009《SM2 密码算法使用规范》，采用 RSA 算法时，遵循 PKCS#1 规范。

消息格式如图 3 所示：

图 3 安全通道消息格式定义

传送数据按照网络字节序传输。

- 版本号：目前为 1；
- 消息安全模式：目前只设置低 3 位（D2D1D0），分别代表是否需要回复信息(D2)，是否加密(D1)和是否签名(D0)。D2 置为 0 表示不需回复，1 表示需要回复；D1 置为 0 表示未加密，1 表示已加密；D0 置为 0 表示未签名/未计算 HMAC(遵循 FIPS198 Key-Hash Message Authentication Code 标准)，1 表示已签名/已计算 HMAC。
- 消息 ID：用来防止重放，每个被管设备自己维护，依次递增，当大于某个指定值时，应重新建立安全通道，重新建立以后该消息 ID 清 0。
- PDU 长度：消息 PDU 字节长度。
- 目的方 ID：目的方注册时从设备管理总中心获得的设备唯一性标识（deviceID）。
- 发送方 ID：设备管理总中心或分中心的唯一性标识（deviceID）。当目的方 ID 与接受方标识不一致，则通过接受方与目的方的安全通道转发这条消息，如找不到信息的目的地，返回一个错误
- 操作类型：密码设备管理平台管理消息的类型。
- 消息 PDU：密码设备管理平台承载的管理应用所发送的管理消息，包括安全通道消息、设备管理的消息和管理应用的消息。管理应用消息首字节为管理应用的标识（密钥管理：0XC0，远程监控：0XC1，参数配置：0XC2，远程维护：0XC3，有效性检测：0XC4，其它应用标识待扩展），用于设备管理代理将管理应用指令转发至对应管理应用代理。安全通道消息和设备管理的消息的数据包格式参见第 9 章，管理应用的消息由各应用自主制定，并调用管理应用接口发送。
- 签名值/HMAC：对消息头和消息 PDU 的签名值（建立安全通道时）或 HMAC 值（安全通道发送数据时）。

6.2.2 安全通道建立请求消息格式

安全通道建立请求消息格式如图 4 所示：

名称	长度（字节）	说明
版本号	1 byte	消息头
安全模式	1 byte	
保留	2 byte	
消息ID	8 byte	
PDU长度	4 byte	
目的方ID （管理中心）	32 byte	
发送方ID （代理）	32 byte	
操作类型 （0xA1）	1 byte	
管理中心公钥 对随机数A的加密结果	N byte	消息PDU
管理代理私钥 对随机数A的签名值	N byte	
管理代理私钥 对消息头和消息PDU的签名值	N byte	消息尾

图 4 安全通道建立请求消息包格式

6.2.3 安全通道建立响应消息格式

安全通道建立响应消息格式如图 5 所示：

名称	长度（字节）	说明
版本号	1 byte	版本号
安全模式	1 byte	消息头
保留	2 byte	
消息ID	8 byte	
PDU长度	4 byte	
目的方ID （管理中心）	32 byte	
发送方ID （代理）	32 byte	
操作类型 （0xA2）	1 byte	
管理代理公钥对随机数A和随机数B的加密结果	N byte	消息PDU
管理中心私钥对随机数B的签名值	N byte	
管理中心私钥对消息头和消息PDU的签名值	N byte	消息尾

图 5 安全通道建立响应消息包格式

6.2.4 安全通道数据发送消息格式

安全通道数据发送的管理消息由所承载的应用定义，首字节为管理应用的标识。数据发送消息格式如图 6 所示：

名称	长度（字节）	说明
版本号	1 byte	消息头
安全模式	1 byte	
保留	2 byte	
消息ID	8 byte	
PDU长度	4 byte	
目的方ID （管理中心）	32 byte	
发送方ID （代理）	32 byte	
操作类型 （0xA3）	1 byte	
被安全通道会话密钥保护的消息	N byte	消息PDU
管理中心私钥对消息头和消息PDU的签名值	N byte	消息尾

图 6 安全通道数据发送消息包格式

6.2.5 通知重启安全通道消息格式

管理中心发起的要求下级管理节点重新建立安全通道的消息格式。下级节点收到该消息后使用 9.1 节消息重新建立安全通道。具体消息格式如图 7 所示：

名称	长度（字节）	说明
版本号	1 byte	消息头
安全模式	1 byte	
保留	2 byte	
消息ID	8 byte	
PDU长度	4 byte	
目的方ID （管理中心）	32 byte	
发送方ID （代理）	32 byte	
操作类型 （0xA4）	1 byte	
管理中心私钥对消息头的签名值	N byte	消息尾

图 7 通知重启安全通道消息包格式

6.3 安全通道建立时机

安全通道采用自下向上分层建立的方式（安全通道建立流程参见附录 B），安全通道可以采用长连接方式也可以采用短连接方式，其建立时机如下：

- 管理分中心启动时，与管理总中心建立安全通道。
- 被管设备启动时，与管理分中心建立安全通道。
- 如果安全通道连接超时或者出错，管理中心通过其他手段通知分中心管理员或者设备管理员，重新建立安全通道。
- 当防止重放的消息 ID 大于某个特定值时,上级管理中心应和底层设备重新建立安全通道。上级管理中心必须向该设备发送清零通知，被管设备将消息 ID 清零，然后由被管设备发起建立安全通道。

当安全通道采用长连接方式时，底层设备或分中心与上层中心建立并长期保持该连接。当安全通道采用短连接方式时，底层设备或分中心与上层中心建立连接并完成数据发送后即断开与上层中心的连接直到下次需要发送数据时再次启动连接。

6.4 安全通道的使用

管理中心与被管设备之间的所有消息，都通过安全通道发送。发送方和目的方间如需经过分中心，发送方需把该信息打包传递给分中心，分中心负责转发给目的方。

7 设备管理信息结构

7.1 设备管理信息定义

设备管理信息，指对密码设备管理系统对密码设备进行远程管理时查询或配置的标准数据格式。

管理中心根据第 8 章定义的设备管理消息对密码设备进行管理信息的查询和配置，被管密码设备需支持本章定义的管理属性，用于管理中心查询和配置。

7.2 数据类型定义

表 2 数据类型定义表

数据类型定义	类型名称	描述
这几个数据类型为基本类型，开发者可根据硬件平台定义，方便移植。本规范中的其它数据类型都是基于这几种基本类型。	INT8	有符号 8 位整数
	INT32	有符号 32 位整数
	UINT8	无符号 8 位整数
	UINT32	无符号 32 位整数
	INT64	有符号 64 位整数
	UINT64	无符号 64 位整数
typedef UINT8	SDM_BYTE	无符号字符
typedef INT32	SDM_INT32	32 位整数
typedef INT64	SDM_INT64	64 位整数
typedef UINT32	SDM_UINT32	32 位非负整数
typedef UINT64	SDM_UINT64	64 位非负整数
typedef struct { SDM_UINT32 len; SDM_BYTE * ptr;}	SDM_OCTET_STRING	零或更多位的八位数组
typedef 0	SDM_NULL	空
typedef SDM_UINT32	SDM_COUNTER32	32 位计数器，从 0 开始，

		到最大值后复位；
typedef SDM_UINT64	SDM_COUNTER64	64 位计数器，从 0 开始，到最大值后复位；
typedef SDM_UINT64	SDM_AID	被管设备属性标识
typedef SDM_UINT32	SDM_TimeTicks	计时值，计量某一时刻后的时间，以百分之一秒计时
typedef SDM_OCTET_STRING	SDM_DisplayString	可显示的 OCTET STRING，0-255 位
typedef enum(0,1)	SDM_TruthValue	0 或 1；
typedef SDM_OCTET_STRING	SDM_MacAddress	6 字节的 OctetString,代表 MAC 地址；
typedef SDM_UINT32	SDM_TestAndIncr	代表互斥操作的锁，自动递增。
typedef enum(active(1),notInService(2), notReady(3),createAndWait(4),createAndGo(5),destroy(6))	SDM_RowStatus	表中某一行的状态，提供了对表内的行进行添加删除的功能。
typedef SDM_OCTET_STRING	SDM_DateAndTime	8 字节的 OCTET STRING
typedef SDM_UINT32	SDM_StorageType	1：other； 2：volatile 存储在内存中，重启会丢失； 3：nonVolatile 重新启动值会恢复初始值； 4：permanent，可以更改但不能删除； 5：readOnly，不能被修改和删除；

typedef
{
SDM_INT32 ulPubKeyInfoLen;
SDM_BYTE *pPubKeyInfo;
}
	SDM_PubKey_Info	公钥信息结构

typedef
{
SDM_INT32 ulSignInfoLen;
SDM_BYTE *pSignInfo;
}
	SDM_Sign_Info	签名信息结构

typedef { SDM_INT32 uiCertApplicationLen; SDM_BYTE *pCertApplication;	SDM_Cert_Application	证书申请

}		

Typedef
{
SDM_INT32 uiCertLen;
SDM_BYTE *pCert;
}
	SDM_Cert	证书

其中 SDM_DateAndTime 定义如表 3 所示：

表 3 时间定义表

字节	说明	取值范围
1-2	年	0-65535
3	月	1-12
4	日	1-31
5	小时	0-23
6	分钟	0-59
7	秒	0-60（闰秒）

7.3 管理信息层次结构

密码设备管理面向被管对象，管理被管对象可以通过读取和设置密码设备属性操作来完成。管理信息按照层次式树型结构组织和排列。如图 8 所示：

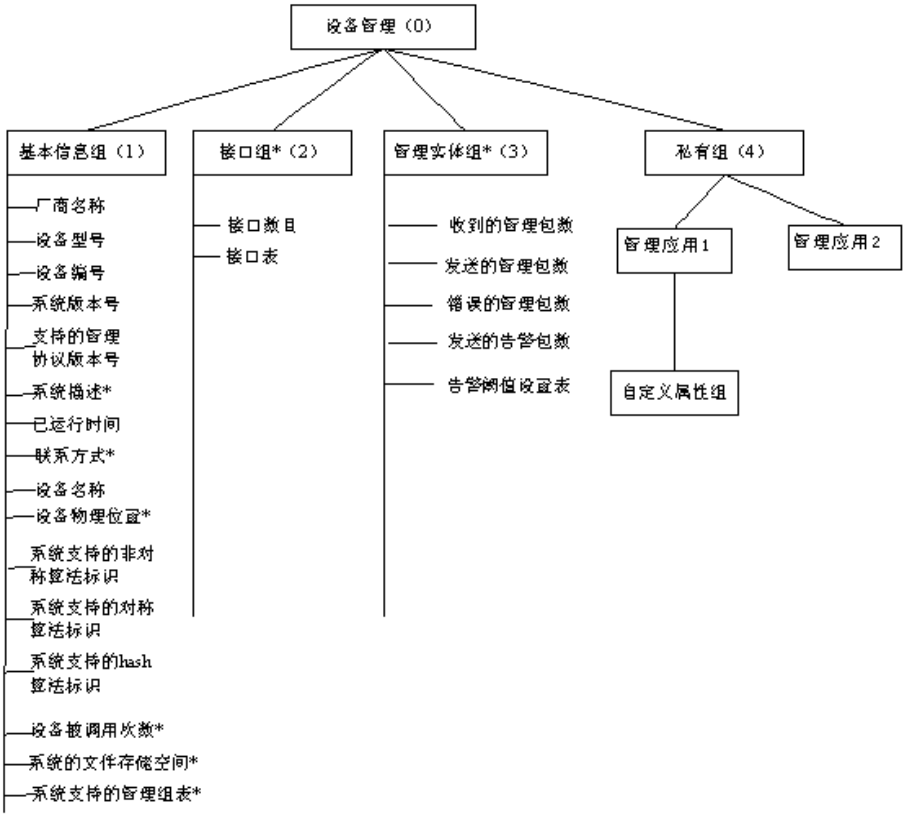

图 8 被管对象属性树形结构

被管对象的属性分为基本信息组、接口组、管理实体组和私有组。

基本信息组：被管对象基本属性，带星号的是可选项。

接口组：被管对象所支持的物理接口，本组可选。

管理实体组：管理消息的属性，本组可选。

私有组：管理应用下发的、被管对象解析的属性信息。各管理应用自主定义私有组内容。

属性的标识定义为 AID，采用整数划分范围的方式。AID 为 64 位整数，0-7 位代表组别，8-23 位代表管理应用的标识号（通用项为 0），24-31 位代表不同的属性编号，如果是表对象，32-55 位代表行号，最后 8 位代表表对象的子属性编号。

本章 7.3 节对设备部分属性的 AID 进行定义，设备厂商和设备管理中心平台厂商应按照此规则开发。未定义的部分可按照 AID 定义规则自行扩展。

7.4 属性定义

指密码设备的管理属性，是密码设备可被管理的具体数据对象。

7.4.1 基本信息组

表 4 基本信息定义表

名称	AID	类型	访问权限	描述
厂商名称	0x01000001 0x00000000	SDM_DisplayString	只读	设备生产厂商名称
设备型号	0x01000002 0x00000000	SDM_DisplayString	只读	设备型号
设备序列号	0x01000003 0x00000000	SDM_DisplayString	只读	设备序列号，包含：日期（8 字符）、批次号（3 字符）、流水号（5 字符）
系统版本号	0x01000004 0x00000000	SDM_DisplayString	只读	被管对象内部软件的版本号
管理协议版本号	0x01000005 0x00000000	SDM_DisplayString	只读	被管对象支持的接口规范版本号
设备描述	0x01000006 0x00000000	SDM_DisplayString	只读	设备描述，可选
已运行时间	0x01000007 0x00000000	SDM_TimeTicks	只读	设备启动时开始计时
联系方式	0x01000008 0x00000000	SDM_DisplayString	读写	设备管理员联系方式
设备名称	0x01000009 0x00000000	SDM_DisplayString	读写	设备名称
设备位置	0x0100000a 0x00000000	SDM_DisplayString	读写	设备的物理位置
系统支持的非对称算法	0x0100000b 0x00000000	SDM_INT64	只读	前 4 字节表示支持的算法,表示方法为非对称算法标识按位或的结果；后 4 字节表示算法的最大模长，表示方法为支持的模长按位或的结果
系统支持的对称算法	0x0100000c 0x00000000	SDM_INT32	只读	对称算法标志按位或
系统支持的杂凑算法	0x0100000d 0x00000000	SDM_INT32	只读	杂凑算法标志按位或

系统支持的管理组（表对象）	0x0100000e 0x00000000	管理组对象 ID：SDM_INT64 0x0100000e 0x00000001	只读	支持的属性组，一行代表一个组别，至少有一行，代表基本信息组。
		管理组对象描述：SDM_DisplayString 0x0100000e 0x00000002	只读	
		上次该行赋值时间：SDM_TimeTicks 0x0100000e 0x00000003	读写	
		ControlStatus ：SDM_RowStatus 0x0100000e 0x00000004	读写	控制添加和删除的状态变量 置为 0 表示删除该行。
设备唯一标识	0x0100000f 0x00000000	SDM_DisplayString SDM_INT32	只读	注册时总中心向下分配设备唯一标识时使用，设备型号、设备编号、所属系统
证书申请	0x01000011 0x00000000	SDM_Cert_Application	只读	注册时分中心通过安全通道向总中心转发证书申请时使用
总中心证书	0x01000012 0x00000000	SDM_Cert	只读	如果总中心是 CA,利用安全通道下发证书时使用
父节点证书	0x01000013 0x00000000	SDM_Cert	只读	利用安全通道发证书时使用
被管设备证书	0x01000014 0x00000000	SDM_Cert	只读	利用安全通道发证书时使用

7.4.2 接口组

表 5 接口组定义表

名称	AID	类型	访问权限	描述
接口数目	0x02000001 0x00000000	SDM_INT32	读写	设备上存在的接口数目，包括物理接口和虚拟接口
接口信息表	0x02000002 0x00000000	ifName ：SDM_DisplayString 0x02000002 0x00000001	只读	接口名称
		IfDescr ：SDM_DisplayString 0x02000002	只读	接口描述

		0x00000002		
		ifType： SDM_INT32 0x02000002 0x00000003	只读	接口类型
		IfAddress： SDM_OCTET_STRING 0x02000002 0x00000004	只读	物理地址
		ifMTU：SDM_INT32 0x02000002 0x00000005	只读	最大包长
		ifSpeed： SDM_INT64 0x02000002 0x00000006	只读	接口速率
		IfInPacket： SDM_INT64 0x02000002 0x00000007	只读	收到总包数
		IfOutPacket： SDM_INT64 0x02000002 0x00000008	只读	发送总包数
		IfErrPakcet： SDM_INT64 0x02000002 0x00000009	只读	出错包数
		IfLinkUpDownTrapEnable： SDM_INT32 0x02000002 0x0000000a	只读	是否允许接口发送 trap 包
		IfLastChange： SDM_TimeTicks 0x02000002 0x0000000b	读写	接口被修改时间
		IfControlStatus： SDM_RowStatus 0x02000002 0x0000000c	读写	控制接口添加和删除的状态变量

7.4.3 管理实体组

表 6 管理实体组定义表

名称	AID	类型	访问权限	描述
收到的管理包数	0x03000001 0x00000000	SDM_INT32	只读	
发送的管理包数	0x03000002 0x00000000	SDM_INT32	只读	
错误的管理包数	0x03000003 0x00000000	SDM_INT32	只读	
发送的告警包数	0x03000004 0x00000000	SDM_INT32	只读	
告警阈值设置表	0x03000005 0x00000000	TrapOid SDM_AID 0x03000005 0x00000001	只读	设置阈值的对象
		TrapLValue SDM_INT32 0x03000005 0x00000002	读写	低于多少告警
		TrapRValue SDM_INT32 0x03000005 0x00000003	读写	高于多少告警
		TrapSetTime SDM_TimeTicks 0x03000005 0x00000004	读写	上次设置时间
		TrapMapControlStatus: SDM_RowStatus 0x03000005 0x00000005	读写	告警行添加删除控制变量

8 设备管理消息

8.1 设备管理消息格式定义

设备管理总中心和设备管理分中心，设备管理分中心和被管设备之间通过安全通道进行消息交互，消息的格式见本规范的 6.2 章节。

在安全通道协议报文中，设备管理消息封装在安全通道消息 PDU 中，被安全通道会话密钥加密保护。消息头中的“操作类型”为 0XA3。消息尾为 16 字节的 HMAC，用安全通道会话密钥对消息头和消息 PDU 密文作 HMAC，密码设备管理的应用标识为 0X00。如图 9 所示：

名称	长度（字节）	说明
版本号	1 byte	消息头
安全模式	1 byte	
保留	2 byte	
消息ID	8 byte	
PDU长度	4 byte	
目的方ID （管理中心）	32 byte	
发送方ID （代理）	32 byte	
操作类型 （0xA3）	1 byte	
密码设备管理应用标识（0x00）	1 byte	被安全通道会话密钥保护的设备管理消息PDU
密码设备管理消息	N byte	
安全通道会话密钥对消息头和消息PDU计算的16字节的MAC值	16 byte	消息尾

图 9 标准管理应用消息包格式

HMAC 是用 SM3 算法，用安全通道会话密钥计算 HMAC，取前 16 字节。加密填充方法：第一个字节为 0x80，其后为若干 0x00，填充到分组长度整数倍。

被管实体中的管理代理应该实现以下 7 个标准操作，接受管理中心对密码设备的管理属性（第 7 章定义）的查询和配置：

- get 操作，获得某个属性值；
- get-next 操作，对于包里的每个属性，获得下一个字典序的属性值；
- response 操作，响应 get、get-next、get-bulk、set、inform 操作；
- set 操作，修改某个属性；
- get-bulk 操作，批量获得属性值；
- trap 操作，达到告警值时发送一个告警信息；
- inform 操作，发送需要应答的告警信息

这 7 个标准操作的消息格式定义如下，封装在安全通道消息 PDU 中。其它针对特定设备的操作由设备厂商自行扩展。

8.2 get 操作消息

获得某个属性值，其中管理应用消息流水号用于防重放，包格式如下表 7 所示：

表 7 get 操作消息数据包格式定义表

应用标识 0x00	类型 0xB0	属性标识 1	……	属性标识 n 参见 7.3

8.3 get-next 操作消息

对于包里的每个属性，获得下一个字典序的值(字典序的概念参考 SNMP 协议)，包格式如下表 8 所示：

表 8 get-nex 操作消息数据包格式定义表

应用标识 0x00	类型 0xB1	属性标识 1	……	属性标识 n

8.4 response 操作消息

响应 get、get-next、get-bulk、set、inform 操作，包格式如下表 9 所示：

表 9 response 操作消息数据包格式定义表

应用标识 0x00	类型 0xB2	错误响应标识(1 字节)	变量绑定

其中，错误响应标识用来表明消息是正常响应还是错误响应，正常响应用 0x00 标识，包格式具体如下表 10 所示：

表 10 正常响应消息数据包格式定义表

应用标识 0x00	类型 0xB2	0x00 (1 字节)	属性标识 1，值长度，值	属性标识 2，值长度，值	… …	属性标识 n，值长度，值

其中，值长度是对应属性值的字节数，为网络字节序的 4 字节整型，以下同。

错误响应用 0x01 标识包格式具体如下表 11 所示：

表 11 错误响应消息数据包格式定义表

应用标识 0x00	类型 0xB2	0x01 (1 字节)	属性标识 1，错误码 1	属性标识 2，错误码 2	… …	属性标识 n，错误码 n

其中，错误码紧跟属性标识，描述该属性的错误情况。

8.5 set 操作消息

修改某个属性，包格式如下表 12 所示：

表 12 set 操作消息数据包格式定义表

应用标识 0x00	类型 0xB3	属性标识 1，	属性标识 2，	…	属性标识 n，

		值长度，值	值长度，值	…	值长度，值

8.6 get-bulk 操作消息

N 和 M 分别代表对变量绑定的前 N 个变量做 get-next 操作，对后面的变量做 M 次的 get-next 循环操作，包格式如下表 13 所示：

表 13 get-bulk 操作消息数据包格式定义表

应用标识 0x00	类型 0xB4	N，M 2 字节	属性标识 1	属性标识 2	… …	属性标识 n

8.7 inform 操作消息

发送通知信息，需要应答以确认发送成功。当是注册消息时，通知类型为 0x00；当是告警消息时，通知类型为 0x01；当是注销消息时，通知类型为 0x02。包格式如下表 14 所示：

表 14 inform 操作消息数据包格式定义表

应用标识 0x00	类型 0xB5	通知类型（1 字节）	属性标识 1，值长度，值	属性标识 2，值长度，值	… …	属性标识 n，值长度，值

8.8 trap 操作消息

发送通知信息，且不需要应答。通知类型定义同 inform 操作。包格式如下表 15 所示：

表 9 response 操作消息数据包格式定义表

应用标识 0x00	类型 0xB6	通知类型（1 字节）	属性标识 1，值长度，值	属性标识 2，值长度，值	… …	属性标识 n，值长度，值

9 设备管理平台对管理应用提供的接口

设备管理平台对管理应用提供以下接口：

A. 初始化设备管理环境：SMF_ Initialize
B. 退出设备管理环境：SMF_ Finalize
C. 获取被管设备总数：SMF_GetDeviceCount
D. 根据设备号得到设备标识和信息 SMF_GetDeviceInfo
E. 批量获取设备属性值：SMF_GetMultiDeviceValue
F. 设置设备属性值：SMF_SetDeviceValue
G. 导出设备证书：SMF_GetDeviceCert
H. 发送数据：SMF_SecTunnelSendData
I. 获得告警信息数量：SMF_GetTrapCount
J. 获得一条告警信息：SMF_GetTrapInfo
K. 设置告警信息为已处理：SMF_SetTrapInfo

9.1 初始化设备管理环境

原型： int SMF_ Initialize (void **pMangeHandle);

描述： 初始化设备管理，获得设备管理安全通道句柄

参数： pMangeHandle [out] 返回的设备管理句柄

返回值： 0 成功

非 0 失败，返回错误代码

备注： 接口内部连接拓扑数据库，获得连接标识，用于获取被管设备的管理位置。

9.2 退出设备管理环境

原型： int SMF_Finalize (void *MangeHandle);

描述： 释放设备管理句柄

参数： pMangeHandle [in] 设备管理句柄

返回值： 0 成功

非 0 失败，返回错误代码

备注：

9.3 获取设备总数

原型：
```
int SMF_GetDeviceCount(
                        void * MangeHandle ,
                        unsigned int *pDeviceCount,
                        unsigned char pDeviceID[]
                        );
```

描述： 获取被管设备总数

参数： MangeHandle[in] 设备管理句柄

pDeviceCount [out] 返回被管设备数目

pDeviceID[out] 返回设备唯一标识列表

返回值： 0 成功

非 0 失败，返回错误代码

备注：

9.4 根据编号获得设备信息

原型：
```
int SMF_GetDeviceInfo(
     void * MangeHandle ,
     unsigned char deviceID[32],
     DEVICEINFO * pDeviceInfo
     );
```

描述： 根据设备在系统中的编号获得设备的信息

参数： MangeHandle[in] 设备管理句柄

pDeviceID[in] 设备唯一标识

pDeviceInfo[out] 设备信息

返回值： 0 成功

非 0 失败，返回错误代码

备注： 1、设备唯一标识在注册时由设备管理总中心统一分发，以后接口参数也按照此规定

2、下面是设备信息结构体的定义，其中 FatherTopo 是设备管理信息库中的设备所属中心标识串。

```
typedef struct DeviceInfo_st{
    unsigned char DeviceID[32];
    unsigned char DeviceName[40];
```

```
    unsigned char IssuerName[40];
    unsigned char DeviceType[16];
    unsigned char DeviceSerial[16];
    unsigned int  DeviceVersion;
    unsigned int  StandardVersion;
    unsigned int  AsymAlgAbility[2];
    unsigned int  SymAlgAbility;
    unsigned int  HashAlgAbility;
    unsigned int  BufferSize;              //支持的最大文件存储空间（单位字节）
    unsigned char  DeviceRole;             //0 是总中心，1 分中心，2 是被管设备
    unsigned char DeviceStatus;            //设备状态，0 正常，1 异常
    unsigned char Resv[2];
    unsigned char FatherTopo[256];         //设备所属父节点
}DEVICEINFO;
```

9.5 批量获取设备属性值

原型：
```
int SMF_GetMultiDeviceInfo(
    void * MangeHandle ,
    unsigned char deviceID[32],
    unsigned int AIDCount,
    SDM_AID typeAID[],
    unsigned char *pData,
    unsigned int *pDataLen []
    );
```

描述：获取设备多项属性

参数：

	MangeHandle[in]	设备管理句柄
	deviceID[in]	设备唯一标识
	AIDCount[in]	要获得设备属性的数量
	typeAID[in]	设备属性的编码数组
	pData [out]	设备属性值，按照属性依次排列，根据属性长度划分
	pDatalen [in,out]	设备属性值长度数组
返回值：	0	成功
	非 0	失败，返回错误代码

备注：

9.6 设置设备属性值

原型：
```
int SMF_SetDeviceInfo(
    void * MangeHandle ,
    unsigned char deviceID[32],
    SDM_AID  typeAID,
    unsigned char *pData,
    unsigned int dataLen
    );
```

描述：设置指定设备的某项属性

参数：

	MangeHandle[in]	设备管理句柄

	deviceID[in]	设备唯一标识
	typeAID[in]	设备属性的编码
	dataLen [in]	设备属性值长度
	pData [in]	设备属性值
返回值：	0	成功
	非 0	失败，返回错误代码

备注：

9.7 导出设备证书

原型：
```
int SMF_GetDeviceCert (
    void * MangeHandle ,
    unsigned char deviceID[32],
    unsigned char *certData,
    unsigned int *pCertLen
    );
```

描述： 导出指定设备的证书

参数：	MangeHandle[in]	设备管理句柄
	deviceID[in]	要导出的设备唯一标识
	certData [out]	证书数据
	pCertLen[in,out]	证书长度
返回值：	0	成功
	非 0	失败，返回错误代码

备注：

9.8 使用安全通道发送数据

原型：
```
int SMF_SecTunnelSendData (
        void * MangeHandle ,
        unsigned char deviceID[32],
        unsigned char *sendData,
        unsigned int sendDatalen,
        unsigned char *replyData ,
        unsigned int * replyDatalen );
```

描述： 使用和指定设备之间的安全通道发送数据

参数：	MangeHandle[in]	设备管理句柄
	deviceID	设备唯一标识
	sendDatalen [in]	要发送的消息数据长度
	sendData[in]	要发送的管理应用消息
	replyDatalen[out]	返回的数据长度
	replyData[in,out]	返回的数据
返回值：	0	成功
	非 0	失败，返回错误代码

备注：

9.9 获得告警信息数量及告警编号

原型：
```
int SMF_GetTrapCount (
        void * MangeHandle ,
```

```
                unsigned char deviceID[32],
                int trapType,
                unsigned int * pTrapCount,
                unsigned int *pTrapNum);
```

描述： 获得设备的告警信息数量

参数：

MangeHandle[in]	设备管理句柄
deviceID [in]	设备唯一标识
trapType[in]	告警类型，0 为未处理，1 为已处理,2 为所有
pTrapCount[out]	返回的告警数量
pTrapNum[out]	返回的告警号

返回值：

0	成功
非 0	失败，返回错误代码

备注： 如果被管设备没有远程告警功能，本接口可不实现。

9.10 获得一条告警信息

原型：

```
int SMF_GetTrapInfo (
        void * MangeHandle ,
        unsigned char deviceID[32],
        int trapType,
        unsigned int trapNum,
        OBJECT IDENTIFIER  *pWarningAID,
        unsigned char *pAIDValue,
        unsigned int * pAIDValueLen,
        unsigned char *info,
        unsigned int *pInfoLen
        );
```

描述： 获得设备的一条告警信息

参数：

MangeHandle[in]	设备管理句柄
deviceID [in]	设备唯一标识，0 为所有设备，其他值为指定设备唯一标识。设备的所有告警按设备号顺序排列
trapType[in]	告警类型，0 为未处理，1 为已处理,2 为所有
trapNum[in]	告警编号
pWarningAID[out]	发生告警的属性标识
pAIDValue[out]	发生告警的属性值
pAIDValueLen[out]	发生告警的属性值长度
Info[out]	告警的处理信息
pInfoLen[in,out]	告警的处理信息长度

返回值：

0	成功
非 0	失败，返回错误代码

备注： 如果被管设备没有远程告警功能，本接口可不实现。

9.11 设置告警信息为已处理

原型：

```
int SMF_SetTrapInfo (
```

```
        void * MangeHandle ,
        unsigned char deviceID[32],
        int trapType,
        unsigned int trapNum,
        unsigned char * info,
        unsigned int infoLen);
```

描述：	设置指定的告警为已处理	
参数：	MangeHandle[in]	设备管理句柄
	deviceID [in]	设备唯一标识，0 为所有设备，其他值为指定设备唯一标识
	trapType[in]	告警类型，0 为未处理，1 为已处理,2 为所有
	trapNum[in]	告警编号
	info[in]	处理措施描述
	InfoLen[in]	处理措施信息长度
返回值：	0	成功
	非 0	失败，返回错误代码
备注：	如果被管设备没有远程告警功能，本接口可不实现。	

附录 A
（规范性附录）
错误代码定义

表 7 错误代码定义表

错误码标识		
宏描述	预定义值	说明
#define SMR_OK	0x00000000	操作成功
#define SMR_UNKNOWERR	0x03000001	未知错误
#define SMR_NOTSUPPORT	0x03000002	不支持的功能
#define SMR_COMMFAIL	0x03000003	通信超时
#define SMR_VERIFY	0x03000004	设备证书验证失败
#define SMR_DEVNUM	0x03000005	设备号非法
#define SMR_NOSUCHDEV	0x03000006	无此属性设备
#define SMR_ERRVAL	0x03000007	不支持此属性
#define SMR_TRAPTYPE	0x03000008	告警类型错
#define SMR_TRAPNUM	0x03000009	告警号非法
#define SMR_TOOBIG	0x0300000A	包长度大于支持包长
#define SMR_READONLY	0x0300000B	试图修改只读属性
#define SMR_NOROUTE	0x0300000C	目标不可达
… …	0x0300000D 至 0x03FFFFFF	预留

附录 B
（规范性附录）
安全通道协议框架

安全通道是指通信双方通过证书机制验证彼此身份，协商会话密钥对后续信息进行加密传输的应用安全协议处理流程。具体协议流程如图 10 所示：

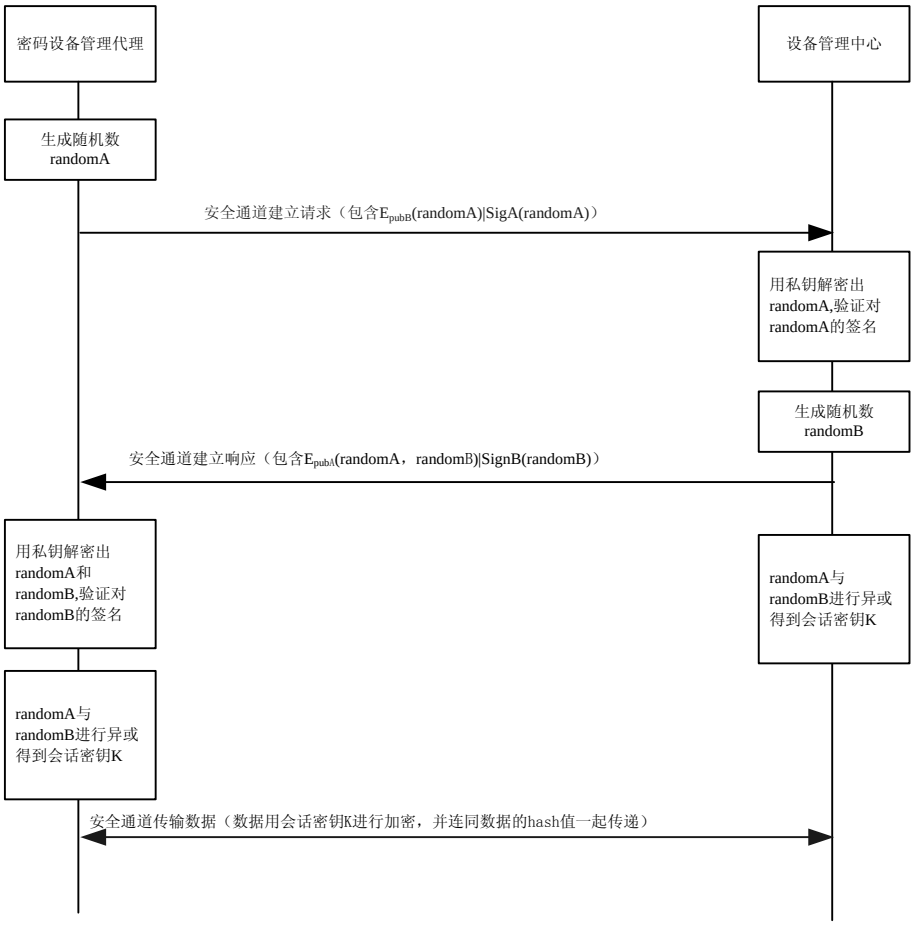

图 10 设备安全通道协议流程图

安全通道协议包括：安全通道建立和安全通道使用。

- 安全通道建立：
  - 管理代理产生一个随机数 randomA，对 randomA 进行签名后，用管理中心的证书公钥对其进行加密，通过安全通道建立请求消息发送给管理中心。
  - 管理中心解密获取 randomA 明文，并验证签名。验证通过后，管理中心保存 randomA 明文。同时，产生一个随机数 randomB，对 randomB 进行签名后，用管理代理的证书公钥对 randomA 和 randomB 进行加密，通过安全通道建立响应消息返回给管理代理。
  - 管理代理解密获取 randomA 和 randomB 的明文，比较 randomA 是否与自己发送的随机数一致，验证 randomB 的签名。
  - 双方将 randomA 和 randomB 进行异或，得到的结果作为安全通道的会话密钥。
- 安全通道使用：
  - 是指通信双方利用建好的安全通道进行信息的安全传递。通信双方传递的数据通过协商好的安全通道会话密钥进行加密保护，使用约定好的 HMAC 算法

（SM3）计算整包数据的 MAC 值，以保证数据在传递过程中的机密性和完整性。

- 在安全通道的有效期内，会话密钥应存储在安全通道两端的密码设备内存中。设备管理中心的密码设备内存中应维护其与下级所有直属密码设备的会话密钥，并在设备管理中心端维护会话密钥与安全通道的对应关系。

参考文献

[1] GB/T 17903.3-1999 信息技术 安全技术 密钥管理 第1部分：框架
[2] GB/T 17903.3-1999 信息技术 安全技术 密钥管理 第2部分：使用对称技术的机制
[3] GB/T 17903.3-1999 信息技术 安全技术 密钥管理 第3部分：使用非对称技术的机制
[4] GB/T 18336-2001 信息技术 安全技术 信息技术安全性评估准则