ICS 35.030
CCS L 80

GM

中华人民共和国密码行业标准

GM/T 0119—2022

PLC 控制系统及 PLC 控制器密码应用技术规范

Cryptography applications technical specification for PLC system and PLC controller

2022-11-20 发布　　2023-06-01 实施

国家密码管理局　　发布

目　　次

前　言

本文件按照GB/T 1.1—2020《标准化工作导则　第1部分：标准化文件的结构和起草规则》的规定起草。

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专利的责任。

本文件由密码行业标准化技术委员会提出并归口。

本文件起草单位：华大半导体有限公司、中电智能科技有限公司、成都天瑞芯安科技有限公司、国家密码管理局商用密码检测中心、中国电子标准化研究院、成都密码技术研究院、北京和利时系统工程有限公司、国电南京自动化股份有限公司、广州数控设备有限公司、南方电网科学研究院有限责任公司、工业信息安全(四川)创新中心有限公司、北京华大云创科技有限公司。

本文件主要起草人：兰天、傅一帆、张兴波、尚望、张众、齐晶晶、张五一、江楠、何英武、陈剑飞、姚相振、龚洁中、杜志波、向春玲、杨祎巍、张文科、陈跃。

PLC 控制系统及 PLC 控制器密码应用技术规范

1 范围

本文件描述了 PLC 控制系统和 PLC 控制器的基本组成；定义了 PLC 控制系统的密码应用功能，PLC 控制系统密码应用总体要求和密码应用基本流程；定义了 PLC 控制系统中各组成设备的密码应用功能和密码应用要求；给出了 PLC 控制系统密码应用接口参考。

本文件适用于集成密码功能的 PLC 控制系统和 PLC 控制器的设计、开发，也可用于指导相关产品的密码应用检测。

2 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中，注日期的引用文件，仅该日期对应的版本适用于本文件；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。

GB/T 32905 信息安全技术 SM3 密码杂凑算法

GB/T 32907 信息安全技术 SM4 分组密码算法

GB/T 32918（所有部分） 信息安全技术 SM2 椭圆曲线公钥密码算法

GB/T 37092—2018 信息安全技术 密码模块安全要求

GB/T 38635（所有部分） 信息安全技术 SM9 标识密码算法

GM/Z 4001 密码术语

IETF RFC4627 The application/json Media Type for JavaScript Object Notation (JSON)

3 术语和定义

GM/Z 4001 界定的以及下列术语和定义适用于本文件。

3.1

SM2 密码算法 SM2 cryptographic algorithm

由 GB/T 32918（所有部分）定义的一种椭圆曲线公钥密码算法。

3.2

SM3 密码算法 SM3 cryptographic algorithm

由 GB/T 32905 定义的一种密码杂凑算法。

3.3

SM4 密码算法 SM4 cryptographic algorithm

由 GB/T 32907 定义的一种分组密码算法。

3.4

SM9 密码算法 SM9 cryptographic algorithm

由 GB/T 38635（所有部分）定义的一种基于身份标识的非对称密码算法。

3.5

工业控制系统 industrial control system

对工业生产过程安全、信息安全和可靠运行产生作用和影响的人员、硬件和软件的集合。

3.6

工业局域网　industrial local area network

实现工业设备互联的局域网，具备更高的稳定性和确定性。

3.7

可编程逻辑控制器　programmable logic controller

一种用于工业环境的数字式操作的电子系统。这种系统用可编程的存储器作为面向用户指令的内部寄存器，完成规定的功能，如逻辑、顺序、定时、计数、运算等，通过数字或模拟的输入/输出，控制各种类型的机械或过程。

3.8

可编程逻辑控制系统　programmable logic controller system

用户根据所要完成的自动化系统要求而建立的由可编程逻辑控制器及其相关外围设备组成的系统。

3.9

现场总线　fieldbus

应用于生产现场，在现场设备之间、现场设备和控制装置之间实行双向、串形、多结点的数字通信技术。

3.10

用户逻辑组态　user logic configuration

用户使用 PLC 控制器配套的应用软件中提供的工具、方法，完成系统中具体逻辑控制任务的脚本集合。

3.11

用户监控组态　user monitor configuration

用户使用 PLC 控制器配套的应用软件中提供的工具、方法，完成系统中具体监控任务的脚本集合。

4　缩略语

下列缩略语适用于本文件。

APDU：应用协议数据单元(Application Protocol Data Unit)

PLC：可编程逻辑控制器(Program Logic Controller)

5　PLC 控制系统概述

5.1　基本组成

PLC 控制系统基本组成如图 1 所示。PLC 控制系统各组成部分详细描述见附录 A。

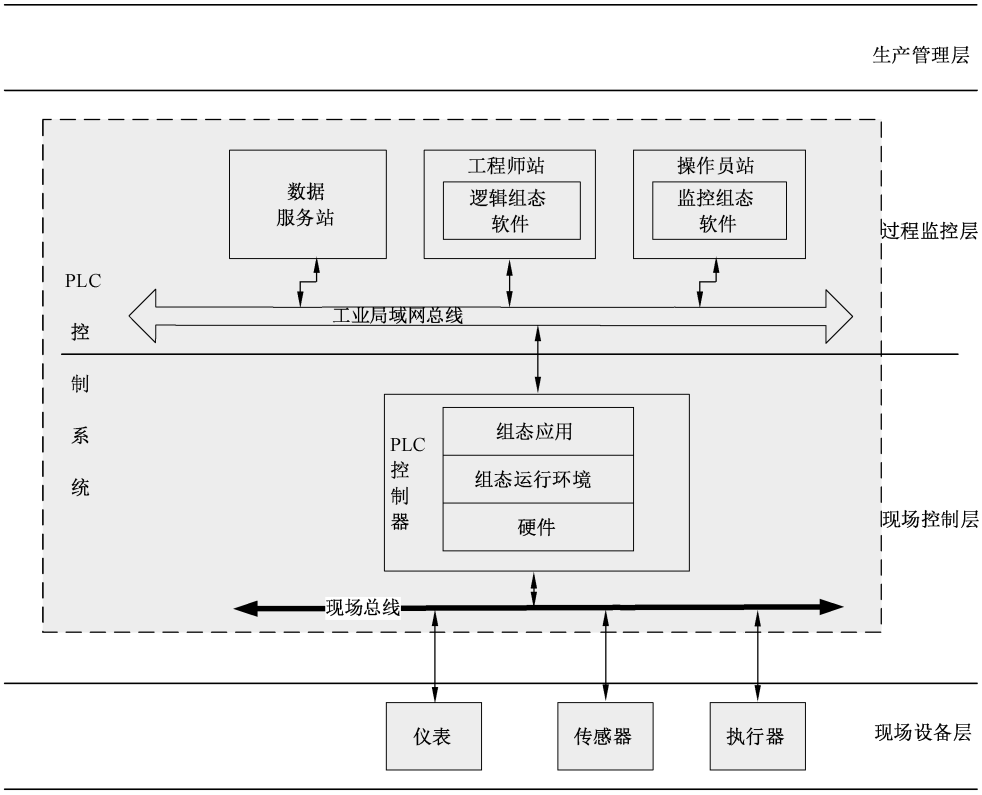

图 1　PLC 控制系统组成

PLC 控制系统以 PLC 控制器为中心构成，基本组成包括：

a）　PLC 控制器：运行用户逻辑组态，实现逻辑控制、现场总线管理、执行指令传递、感测数据回收等功能；

b）　数据服务站：实现数据和事件的记录存档；

c）　工程师站：运行逻辑组态软件，实现用户逻辑组态的编辑、打包和下装；

d）　操作员站：运行监控组态软件，实现用户监控组态的编辑和执行，PLC 控制系统的运行状态和事件的监测；

e）　工业局域网总线：实现 PLC 控制系统内各设备间的局域网通信；

f）　现场总线：实现 PLC 控制器和各现场设备间的通信。

其中，工程师站、操作员站、数据服务站、PLC 控制器之间通过工业局域网总线互联；PLC 控制器和现场设备之间通过现场总线互联。

5.2　基本功能

PLC 控制系统的基本功能包括：

a）　用户逻辑组态的编辑开发；

b）　用户逻辑组态的下装；

c）　用户逻辑组态运行；

d）　用户监控组态的编辑开发；

e）　用户监控组态运行；

f）　系统运行状态数据和事件上送；

g）　系统运行状态数据和事件的存档；

h）　工业局域网总线控制；

i） 现场总线控制。

6 PLC 控制系统密码应用概述

6.1 安全生命周期

PLC 控制系统安全生命周期包括初始化态、编程配置态、运行态和注销态。不同的安全生命周期状态下，PLC 控制系统需要使用不同的密码安全功能。

a） 初始化态：PLC 控制系统首次接入系统网络时，完成密码安全初始化的状态。

b） 编程配置态：PLC 控制系统处于用户逻辑组态和用户监控组态开发状态。该状态下，用户可以编辑和测试用户逻辑组态和用户监控组态，进行用户逻辑组态的重新更换和启用。

c） 运行态：PLC 控制系统处于正常运行状态，该状态下按照已启用用户逻辑组态的控制流程自主执行。处于运行态的系统，可以进行启动、锁定/解锁、挂起/解挂、关闭运行周期管理。

d） 注销态：PLC 控制系统处于不可响应的状态，该状态下密钥和敏感数据被销毁。

状态迁移如图 2 所示。

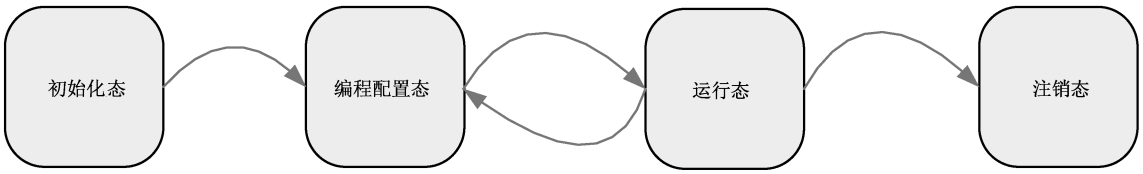

图 2 PLC 控制系统安全生命周期

6.2 应用域划分

PLC 控制系统至少包括两个应用域：设备系统域和用户应用域（见图 3）。不同应用域的管理方不同，信息资产归属不同，需要使用密码技术进行隔离，并配置不同的密钥及敏感数据。

a） 设备系统域：由系统开发方或系统运维方管理，主要包括 PLC 控制系统的软硬件系统环境、系统的配置，设备系统运行的状态。

b） 用户应用域：由用户方管理，主要包括用户开发的用户逻辑组态和用户监控组态，用户逻辑组态和用户监控组态的配置，用户逻辑组态和用户监控组态运行产生的数据。

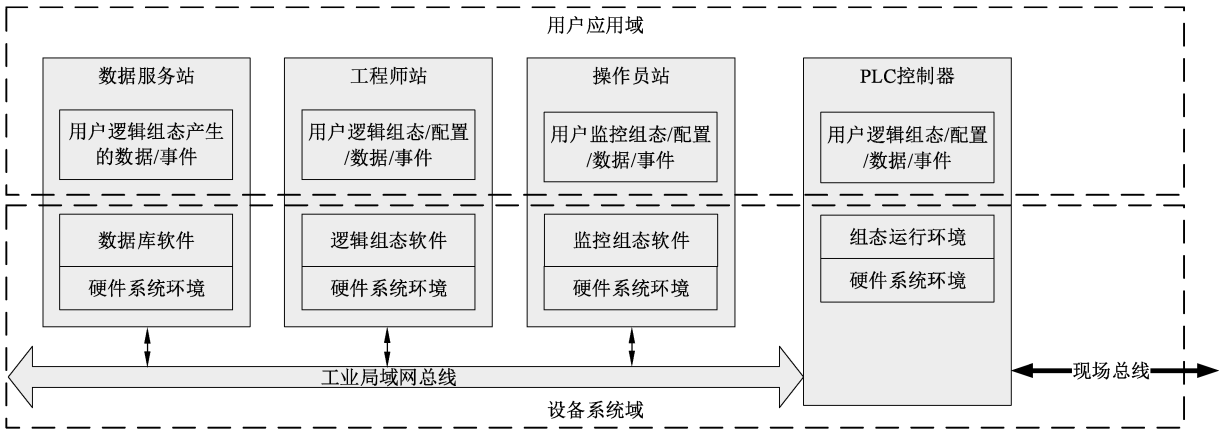

图 3 PLC 控制系统应用域划分

6.3 密码应用功能组成

PLC 控制系统中，密码应用系统如图 4 所示。

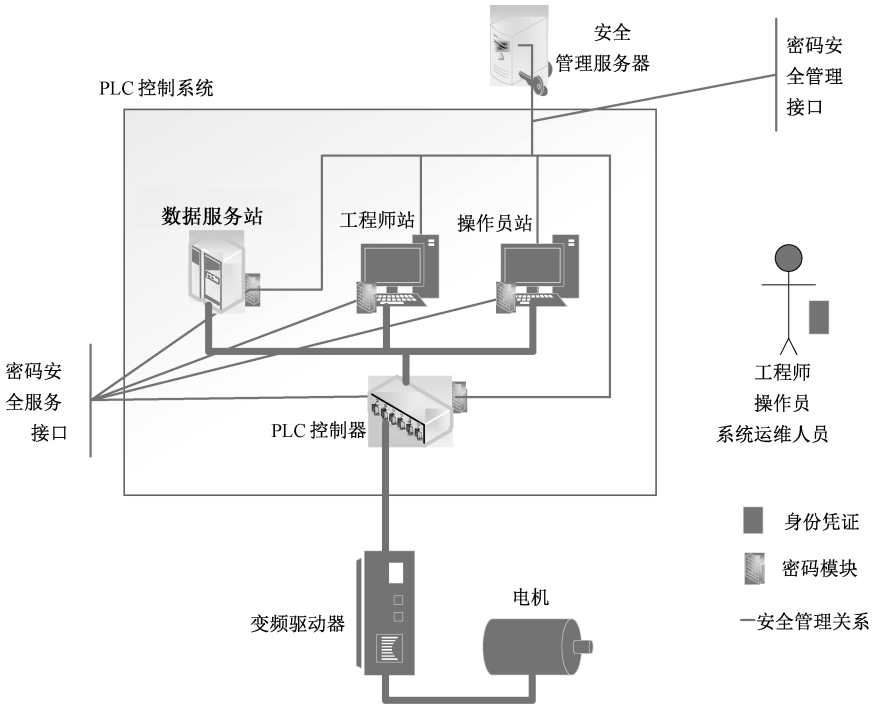

图 4 密码应用系统组成

PLC 控制系统中，密码应用系统主要包括 PLC 控制器、工程师站、操作员站、数据服务站中的密码模块及其密码应用功能，操作人员身份凭证，以及安全管理服务器。

密码模块嵌入到 PLC 控制系统的各组成设备中，主要实现密码算法、密码安全功能和安全管理功能，对外提供密码计算服务接口和安全管理接口。密码模块型态上可以是物理模块，也可以是软件模块或混合模块，应符合 GB/T 37092 中规定的安全要求。

操作人员身份凭证用于用户在 PLC 控制器、工程师站、操作员站和数据服务站上的登录认证。

安全管理服务器是一个后台服务器，和密码模块配合，实现对 PLC 控制系统的密码安全配置和管理。

PLC 控制系统中，通过嵌入到设备中的密码模块提供“密码安全管理接口”和“密码安全服务接口”两个接口，其中“密码安全管理接口”存在于设备和安全管理服务器之间，用于实现密码安全管理；“密码安全服务接口”存在于设备内部，用于设备内业务应用调用密码安全功能。

6.4 密码应用总体要求

6.4.1 功能要求

PLC 控制系统密码应用功能要求如下。

a) PLC 控制系统应使用 SM2、SM3、SM4 和 SM9 等在 GB/T 37092—2018 附录 C 中给出的密码算法。

b) PLC 控制系统应采用分域分状态的安全管理机制，不同应用域间通过密码技术进行隔离并配置不同的密钥及敏感数据。同一应用域下，针对不同安全生命周期状态使用不同的密码安全功能。

c) PLC 控制系统应具有安全管理服务接口，实现系统中主要设备、应用、人员的注册、身份管理、密钥配置和密码安全功能配置；实现设备和应用的安全生命周期管理。

d) PLC 控制系统中的 PLC 控制器、操作员站、工程师站等关键设备应采用密码技术进行注册，接入认证。

e) PLC 控制器的设备系统的更新和组态的更新，应采用密码技术进行完整性校验和操作授权控制。

f) PLC 控制系统的工程师站、操作员站和 PLC 控制器设备间，传递的关键控制命令等应使用密码技术进行真实性、完整性和抗抵赖性保护。

g) PLC 控制系统的工程师站、操作员站、PLC 控制器设备间，互联访问应使用密码技术进行接入认证和通信保护。

h) PLC 控制系统中关键设备的操作人员应发放身份凭证，采用密码技术进行登录认证和权限控制。

6.4.2 性能要求

PLC 控制系统密码应用性能要求如下：

a) 增加密码功能后，PLC 控制系统运行可靠性无显著下降；

b) PLC 控制器中，密码模块应至少支持 4 个彼此独立的安全通道，每个安全通道的对称算法加解密速率建议不低于 0.5 Mb/s；

c) 工程师站、操作员站、数据服务站中，密码模块应至少支持 4 个彼此独立的安全通道，每个安全通道的对称算法加解密速率建议不低于 0.5 Mb/s。

6.5 密码应用基本流程

PLC 控制系统密码应用基本流程中，增强要求为可选流程，用户需要增加系统安全性时可选用。

a) 系统处于初始化态：设备注册。

PLC 控制器、工程师站、操作员站、数据服务站等各设备首次接入到 PLC 控制系统时，完成设备密码安全功能的初始化。

b) 系统处于编程配置态。

1) 设备安全启动

PLC 控制器应使用密码技术对应用代码进行完整性校验，实现安全启动。

2) 设备签到与同步

本步骤是增强要求。

PLC 控制系统的设备启动后，使用密码技术接入到安全管理服务器，并进行状态和消息同步。

3) 设备系统更新

本步骤是增强要求。

PLC 控制系统的设备签到后，如果设备系统环境有更新，设备发起系统环境更新过程，使用密码技术进行设备系统更新的真实性、完整性校验和操作授权。

4) 用户应用组态开发

用户应用组态包括用户逻辑组态和用户监控组态。

编程人员登录逻辑组态软件，完成用户逻辑组态开发并发布。使用密码技术实现人员安全登录和用户逻辑组态文件操作权限控制，保证用户逻辑组态文件完整性和真实性。

编程人员登录监控组态软件，完成应用监控组态开发并发布。使用密码技术实现人员登录和监控组态文件操作权限控制，保证用户监控组态文件完整性和真实性。

5) 用户逻辑组态下装

PLC 控制系统运行过程中，可以向 PLC 控制器下载和安装用户逻辑组态。使用密码技术保护用户逻辑组态下装过程的真实性、完整性、不可抵赖性和操作授权。

6） 用户逻辑组态个人化

本步骤是增强要求。

PLC 控制器的用户逻辑组态安装完成后，首次运行时要求进行个人化设置。使用密码技术为用户逻辑组态建立独立安全区，配置密钥和敏感数据、安全策略等。

c） 系统处于运行态。

1） PLC 控制器运行周期管理

本步骤是增强要求。

PLC 控制器运行周期包括了启动、挂起/解挂、锁定/解锁、关闭。使用密码技术保护上述运行周期管理的切换，保障运行周期状态的完整性。

2） 用户逻辑组态运行周期管理

本步骤是增强要求。

PLC 控制器上的用户逻辑组态运行周期包括了启动、挂起/解挂、锁定/解锁、关闭，使用密码技术保护上述运行周期管理的切换，保障运行周期状态的完整性。

3） 设备间接入和访问

PLC 控制系统内 PLC 控制器、工程师站、操作员站等设备之间的接入和访问，使用密码技术实现接入认证和访问控制，完成安全通道的建立和数据传输保密。

d） 系统处于注销态：无。

7 PLC 控制器密码应用要求

7.1 密码应用功能组成

PLC 控制器包括芯片、操作系统及运行环境、PLC 控制功能模块和密码模块。PLC 控制功能模块用于运行用户逻辑组态，实现业务控制流程；密码模块用于存储密钥等敏感数据、提供密码安全服务以及和后台安全管理服务器的交互。PLC 控制器密码应用功能组成如图 5 所示。

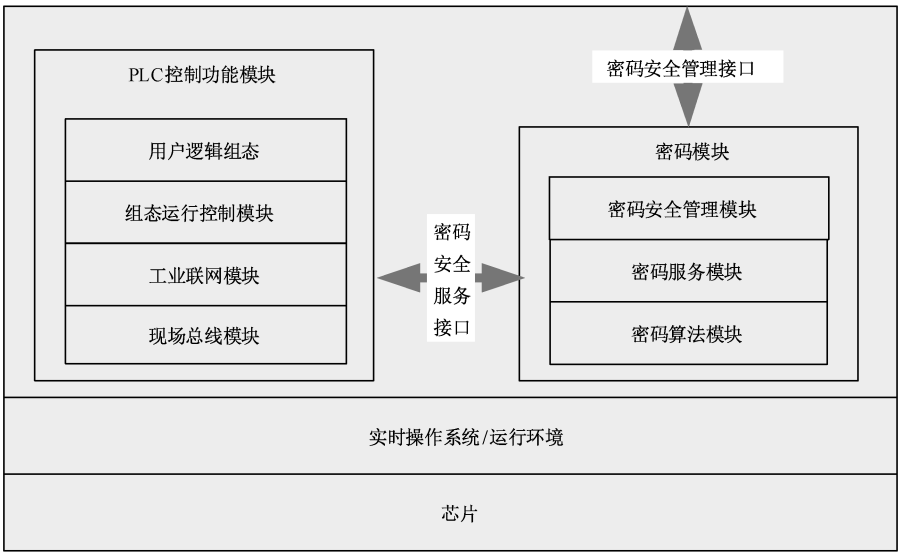

图 5 PLC 控制器密码应用功能组成

7.2 密码应用要求

7.2.1 初始化态

设备注册是指 PLC 控制器在首次接入到 PLC 控制系统时，完成设备安全初始化的过程。

a) PLC 控制器设备注册前应具备如下预置信息：
   1) PLC 控制器应配置设备系统域，由该域存储和使用设备注册相关密钥和敏感数据；
   2) PLC 控制器应具有唯一标识，且标识不可被更改；
   3) PLC 控制器应具有和该标识绑定的初始化密钥。
b) PLC 控制器设备注册中密码应用如下：
   1) PLC 控制器使用初始化密钥，和安全管理服务器完成双向认证，建立安全通道；
   2) PLC 控制器产生新的密钥对，由安全管理服务器发放设备证书或其他身份证明；
   3) PLC 控制器在注册完成前，只能执行注册功能，其他功能禁止使用；
   4) 注册完成后，PLC 控制器应修改其安全状态为编程配置态。

7.2.2 编程配置态

7.2.2.1 设备系统更新

本条是增强要求。

设备系统更新是 PLC 控制器中的底层运行环境（如驱动、运行库、组态运行环境）更新的过程。设备系统更新可以采用在线方式或离线方式。

a) PLC 控制器设备系统更新前应具备如下预置信息：
   1) PLC 控制器应配置设备系统域，由该域存储和使用设备系统更新相关密钥和敏感数据；
   2) PLC 控制器已经完成设备注册。
b) PLC 控制器设备系统更新中密码应用如下：
   1) PLC 控制器发起设备系统更新；安全管理服务器收到申请后，生成下装授权码；准备好下装包并对下装包签名；
   2) 如果使用在线方式，PLC 控制器和安全管理服务器完成基于证书或其他身份证明的双向认证，建立安全通道，然后传输下装授权码和下装包到 PLC 控制器；如果使用离线方式，安全管理服务器加密下装授权码和下装包，存储到载体中，离线传递到 PLC 控制器；
   3) PLC 控制器验证下装授权码，通过后接收下装包。接收过程中对下装包进行完整性度量，并验证下装包签名；验证通过后，设备系统完成更新。

7.2.2.2 用户逻辑组态下装

用户逻辑组态下装是 PLC 控制器获取并安装用户逻辑组态的过程。用户逻辑组态下装可以是在线方式或离线方式。

a) PLC 控制器用户逻辑组态下装前应具备如下预置信息：
   1) PLC 控制器应配置用户应用域，由该域存储和使用用户逻辑组态下装相关密钥和敏感数据；
   2) PLC 控制器已经完成设备注册。
b) PLC 控制器用户逻辑组态下装中密码应用如下：
   1) 安全管理服务器生成下装授权码，准备好用户逻辑组态下装包并签名；
   2) 如果使用在线方式，PLC 控制器和安全管理服务器完成基于证书或其他身份证明的双向认证，建立安全通道，然后传输下装授权码和下装包到 PLC 控制器；如果使用离线方式，安全管理服务器加密下装授权码和下装包，存储到载体中，离线传递到 PLC 控制器；

3） PLC 控制器验证下装授权码，通过后开始接收下装包。接收过程中对下装包进行完整性度量，并验证下装包签名；验证通过后，安装用户逻辑组态。

7.2.2.3 用户逻辑组态个人化

本条是增强要求。

用户逻辑组态个人化是对用户逻辑组态进行基本数据和基本配置初始化的过程。用户逻辑组态个人化可以是在线方式或离线方式。

a） PLC 控制器用户逻辑组态个人化前应具备如下预置信息：

1） PLC 控制器应配置用户应用域，由该域存储和使用用户逻辑组态个人化相关密钥和敏感数据；

2） 用户逻辑组态具有唯一标识。

b） PLC 控制器用户逻辑组态个人化中密码应用如下：

1） 安全管理服务器生成下装授权码，准备好用户逻辑组态个人化数据包并签名；

2） 如果使用在线方式，PLC 控制器和安全管理服务器完成基于证书或其他身份证明的双向认证，建立安全通道，然后传输下装授权码和数据包到 PLC 控制器；如果使用离线方式，安全管理服务器加密下装授权码和数据包，存储到载体中，离线传递到 PLC 控制器；

3） PLC 控制器接收安全管理服务器下发的配置数据，写入到用户逻辑组态中；

4） 用户逻辑组态在个人化完成前，只能执行个人化功能，其他功能禁止使用；

5） 个人化完成后，PLC 控制器应修改用户逻辑组态的安全状态。

7.2.2.4 切换到运行态并启用用户逻辑组态

PLC 控制器完成系统更新或用户逻辑组态下装后，在用户现场干预条件下切换到运行态，PLC 控制器将使用更新后的用户逻辑组态。

PLC 控制器切换到运行态并启用用户逻辑组态中密码应用如下：PLC 控制器在用户现场干预的条件下，切换到运行态，PLC 控制器应修改设备安全生命周期状态。

7.2.3 运行态

7.2.3.1 安全启动

安全启动是 PLC 控制器保证启动代码和启动序列完整性，并按照预定步骤完成启动的过程。

a） PLC 控制器安全启动前应具备如下预置信息：

1） PLC 控制器应配置设备系统域，由该域存储和使用安全启动相关密钥和敏感数据；

2） PLC 控制器已完成设备注册。

b） PLC 控制器安全启动中密码应用如下：PLC 控制器验证启动代码的完整性。

7.2.3.2 设备签到

本条是增强要求。

设备签到是 PLC 控制器完成安全启动后，接入到安全管理服务器并同步状态的过程。

a） PLC 控制器在设备签到前应具备如下预置信息：

1） PLC 控制器应配置设备系统域，由该域存储和使用设备签到相关密钥和敏感数据；

2） PLC 控制器已安全启动。

b） PLC 控制器设备签到密码应用如下：

1） PLC 控制器和安全管理服务器完成基于证书或其他身份证明的双向认证，建立安全通道；

2） PLC 控制器向安全管理服务器同步安全事件记录。

7.2.3.3 用户逻辑组态运行

用户逻辑组态运行是 PLC 控制器中逻辑组态按照预定控制流程自主执行的过程。

a） PLC 控制器在用户逻辑组态运行前应具备如下预置信息：PLC 控制器应配置用户应用域，由该域存储和使用用户逻辑组态相关密钥和敏感数据。

b） PLC 控制器用户逻辑组态运行密码应用如下：

1） PLC 控制器扫描并校验用户逻辑组态，验证其完整性；

2） PLC 控制器启动用户逻辑组态。

7.2.3.4 上位机设备间交互

上位机设备间交互是 PLC 控制器的用户逻辑组态和工程师站、操作员站、数据服务站中的上位机应用之间的接入和交互过程。

a） PLC 控制器和上位机设备间交互前应具备如下预置信息：

1） PLC 控制器应配置用户应用域，由该域存储和使用上位机设备间交互相关密钥和敏感数据；

2） PLC 控制器、工程师站、操作员站、数据服务站已经接入到 PLC 控制系统中。

b） PLC 控制器上位机设备间交互密码应用如下：

1） PLC 控制器和上位机之间完成基于证书或其他身份证明的双向认证，建立安全通道；PLC 控制器和不同上位机之间，和同一上位机的不同的应用之间，应使用不同的密钥建立独立隔离的安全通道；

2） PLC 控制器和上位机之间传输的信息应由安全通道保证完整性和机密性；

3） 上位机下发的关键命令应签名，PLC 控制器应验证关键命令的签名，验证通过后执行。

7.2.3.5 PLC 控制器设备间交互

PLC 控制器设备间交互是 PLC 控制器和其他 PLC 控制器用户逻辑组态之间的接入和交互过程。

a） PLC 控制器设备间交互前应具备如下预置信息：

1） 每个 PLC 控制器应配置用户应用域，由该域存储和使用上位机设备间交互相关密钥和敏感数据；

2） 每个 PLC 控制器已经接入到 PLC 控制系统中。

b） PLC 控制器设备间交互密码应用如下：

1） PLC 控制器和其他 PLC 控制器之间完成基于证书或其他身份证明的双向认证，建立安全通道；PLC 控制器和不同的 PLC 控制器之间，以及和同一 PLC 控制器的不同用户逻辑组态之间，应使用不同的密钥建立独立隔离的安全通道；

2） PLC 控制器和 PLC 控制器之间传输的信息应由安全通道保证完整性和机密性。

7.2.3.6 现场设备间交互

现场设备间交互是 PLC 控制器中的用户逻辑组态和现场设备间的交互过程。

受限于现场设备的资源情况，以及现场控制的实时性要求，该交互的密码机制不做要求。

7.2.3.7 运行周期管理

本条是增强要求。

运行周期管理是 PLC 控制器启动、关闭、挂起/解挂、锁定/解锁的过程。

a） PLC 控制器在运行周期管理前应具备如下预置信息：

1） PLC 控制器应配置设备系统域，由该域存储和使用运行周期管理相关密钥和敏感数据；

2） PLC 控制器已经接入到 PLC 控制系统中。

b） PLC 控制器运行周期管理密码应用如下：

1） PLC 控制器和安全管理服务器完成基于证书或其他身份证明的双向认证，建立安全通道；

2） PLC 控制器收到运行周期状态切换的指令，验证指令合法性；

3） PLC 控制器执行指令，修改运行周期状态。

7.2.4 注销态

注销态是指 PLC 控制器销毁密钥和敏感数据，设备处于不可响应的状态。PLC 控制器注销可以是在线方式或离线方式。

a） PLC 控制器设备注销前应具备如下预置信息：

1） PLC 控制器应配置设备系统域，由该域存储和使用注册相关密钥和敏感数据；

2） PLC 控制器已完成设备注册。

b） PLC 控制器设备注销中密码应用如下：

1） 安全管理服务器生成设备注销指令并签名；

2） 如果使用在线方式，PLC 控制器和安全管理服务器完成基于证书或其他身份证明的双向认证，建立安全通道，然后传输注销命令到 PLC 控制器；如果使用离线方式，安全管理服务器加密注销命令，存储到载体中，离线传递到 PLC 控制器；

3） PLC 控制器收到设备注销的指令，验证指令合法性；

4） PLC 控制器执行指令，销毁密码和敏感数据，修改设备安全生命周期状态。

8 工程师站密码应用要求

8.1 密码应用功能组成

工程师站包括芯片硬件、操作系统及运行环境、用户应用组态功能模块和密码模块。用户应用组态功能模块用于设计编辑用户逻辑组态、存储和管理用户逻辑组态，以及向 PLC 控制器下装用户逻辑组态；密码模块用于存储敏感数据、提供密码安全服务，以及和后台安全管理服务器的交互。工程师站密码应用功能组成如图 6 所示。

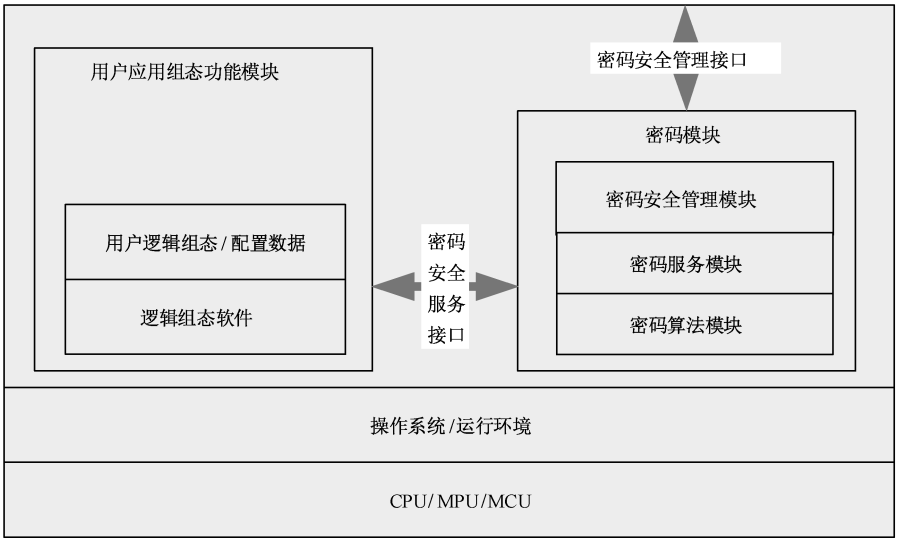

图 6 工程师站密码应用功能组成

8.2 密码应用要求

8.2.1 初始化态

设备注册是指工程师站在首次接入到 PLC 控制系统时，完成设备安全初始化的过程。

a) 工程师站设备注册前应具备如下预置信息：
   1) 工程师站应配置设备系统域，由该域存储和使用设备注册相关密钥和敏感数据；
   2) 工程师站应具有唯一标识，且标识不可被更改；
   3) 工程师站应具有和该标识绑定的初始化密钥。
b) 工程师站设备注册中密码应用如下：
   1) 工程师站使用初始化密钥，和安全管理服务器完成双向认证，建立安全通道；
   2) 工程师站产生新的密钥对，由安全管理服务器发放设备证书或其他身份证明；
   3) 工程师站在注册完成前，只能执行注册功能，其他功能禁止使用；
   4) 注册完成后，工程师站应修改其安全状态。

8.2.2 运行态

8.2.2.1 设备签到

本条是增强要求。

设备签到是工程师站注册后，接入到安全管理服务器并同步状态的过程。

a) 工程师站在设备签到前应具备如下预置信息：
   1) 工程师站应配置设备系统域，由该域存储和使用设备签到相关密钥和敏感数据；
   2) 工程师站已完成设备注册。
b) 工程师站设备签到密码应用如下：
   1) 工程师站和安全管理服务器完成基于证书或其他身份证明的双向认证，建立安全通道；
   2) 工程师站向安全管理服务器同步安全事件记录。

8.2.2.2 逻辑组态软件运行

本条规定了逻辑组态软件运行过程的密码应用要求。

a) 工程师站在逻辑组态软件运行前应具备如下预置信息：工程师站应配置用户应用域，由该域存储和使用逻辑组态软件相关密钥和敏感数据。
b) 工程师站逻辑组态软件运行密码应用如下：
   1) 工程师站扫描并校验逻辑组态软件，验证其完整性；
   2) 工程师站启动逻辑组态软件。

8.2.2.3 逻辑组态软件登录

本条规定了逻辑组态软件登录过程的密码应用要求。

a) 工程师站在逻辑组态软件登录前应具备如下预置信息：
   1) 工程师站应配置用户应用域，由该域存储和使用逻辑组态软件相关密钥和敏感数据；
   2) 用户已发放身份安全凭证。
b) 工程师站逻辑组态软件登录密码应用如下：
   1) 逻辑组态软件验证用户登录用安全凭证；
   2) 逻辑组态软件根据用户身份进行授权。

8.2.2.4 用户逻辑组态编辑

用户逻辑组态编辑是用户设计并编程用户逻辑组态的过程。

a) 用户逻辑组态编辑前应具备如下预置信息：
   1) 工程师站应配置用户应用域，由该域存储和使用用户逻辑组态相关密钥和敏感数据；
   2) 用户已发放身份安全凭证。
b) 用户逻辑组态编辑中密码应用如下：逻辑组态软件应验证用户具备逻辑组态编辑权限。

8.2.2.5 用户逻辑组态文件管理

用户逻辑组态文件管理是对用户逻辑组态文件的选择、打开、关闭、保存的过程。

a) 用户逻辑组态文件管理应具备如下预置信息：
   1) 工程师站应配置用户应用域，由该域存储和使用用户逻辑组态文件管理相关密钥和敏感数据；
   2) 用户已发放身份安全凭证。
b) 用户逻辑组态文件管理中密码应用如下：逻辑组态软件应验证用户具备文件管理相关权限。

8.2.2.6 用户逻辑组态文件发布

用户逻辑组态文件发布是对已完成的用户逻辑组态文件发布的过程。

a) 用户逻辑组态文件发布应具备如下预置信息：
   1) 工程师站应配置用户应用域，由该域存储和使用用户逻辑组态文件发布相关密钥和敏感数据；
   2) 用户已发放身份安全凭证。
b) 用户逻辑组态文件发布中密码应用如下：
   1) 工程师站应验证用户具备用户逻辑组态文件提交权限；
   2) 工程师站应对发布的用户逻辑组态文件进行签名。

8.2.2.7 设备间交互

设备间交互是工程师站的逻辑组态软件和 PLC 控制器、数据服务站应用之间的接入和交互过程。

a) 工程师站在设备间交互前应具备如下预置信息：
   1) 工程师站应配置用户应用域，由该域存储和使用上位机设备间交互相关密钥和敏感数据；
   2) PLC 控制器、工程师站、数据服务站已经接入到 PLC 控制系统中。
b) 工程师站和其他设备间交互密码应用如下：
   1) 工程师站和 PLC 控制器、数据服务站之间完成基于证书或其他身份证明的双向认证，建立安全通道；工程师站和不同的 PLC 控制器、数据服务站之间，应使用不同的密钥建立独立隔离的安全通道；工程师站和同一 PLC 控制器、数据服务站的不同应用之间，应使用不同的密钥建立独立隔离的安全通道；
   2) 工程师站和 PLC 控制器、数据服务站之间传输的信息应由安全通道保证完整性和机密性。

8.2.3 注销态

注销态是指工程师站销毁密钥和敏感数据，设备处于不可响应的状态。工程师站注销可以是在线方式或离线方式。

a) 工程师站注销前应具备如下预置信息：

1） 工程师站应配置设备系统域，由该域存储和使用注册相关密钥和敏感数据；
2） 工程师站已完成设备注册。

b） 工程师站注销中密码应用如下：
1） 安全管理服务器生成设备注销指令并签名；
2） 如果使用在线方式，工程师站和安全管理服务器完成基于证书或其他身份证明的双向认证，建立安全通道，然后传输注销命令到工程师站；如果使用离线方式，安全管理服务器加密注销命令，存储到载体中，离线传递到工程师站；
3） 工程师站收到设备注销的指令，验证指令合法性；
4） 工程师站执行指令，销毁密码和敏感数据，修改设备安全生命周期状态。

9 操作员站密码应用要求

9.1 密码应用功能组成

操作员站包括芯片硬件、操作系统及运行环境、用户应用组态功能模块和密码模块。用户应用组态功能模块用于编辑用户监控组态、存储和管理用户监控组态、运行用户监控组态，以及和 PLC 控制器互联并监测用户逻辑组态运行；密码模块用于存储敏感数据、提供密码安全服务，以及和后台安全管理服务器的交互。操作员站密码应用功能组成如图 7 所示。

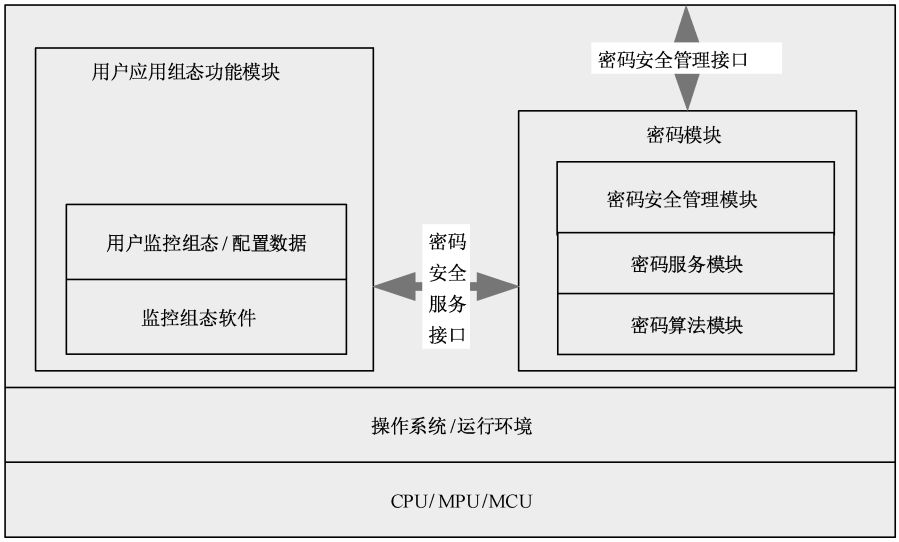

图 7 操作员站密码应用功能组成

9.2 密码应用要求

9.2.1 初始化态

设备注册是指操作员站在首次接入到 PLC 控制系统时，完成设备安全初始化的过程。

a） 操作员站设备注册前应具备如下预置信息：
1） 操作员站应配置设备系统域，由该域存储和使用注册相关密钥和敏感数据；
2） 操作员站应具有唯一标识，且标识不可被更改；
3） 操作员站应具有和该标识绑定的初始化密钥。

b） 操作员站设备注册中密码应用如下：
1） 操作员站使用初始化密钥，和安全管理服务器完成双向认证，建立安全通道；

2） 操作员站产生新的密钥对，由安全管理服务器发放设备证书或其他身份证明；
3） 操作员站在注册完成前，只能执行注册功能，其他功能禁止使用；
4） 注册完成后，操作员站应修改其安全状态。

9.2.2 运行态

9.2.2.1 设备签到

本条是增强要求。

设备签到是操作员站完成注册后，接入到安全管理服务器并同步状态的过程。

a） 操作员站在设备签到前应具备如下预置信息：
1） 操作员站应配置系统设备系统域，由该域存储和使用设备签到相关密钥和敏感数据；
2） 操作员站已完成设备注册。
b） 操作员站设备签到密码应用如下：
1） 操作员站和安全管理服务器完成基于证书或其他身份证明的双向认证，建立安全通道；
2） 操作员站向安全管理服务器同步安全事件记录。

9.2.2.2 监控组态软件运行

本条规定了监控组态软件运行过程的密码应用要求。

a） 操作员站在监控组态软件运行前应具备如下预置信息：操作员站应配置用户应用域，由该域存储和使用监控组态软件相关密钥和敏感数据。
b） 操作员站监控组态软件运行密码应用如下：
1） 操作员站扫描并校验监控组态软件，验证其完整性；
2） 操作员站启动监控组态软件。

9.2.2.3 监控组态软件登录

本条规定了监控组态软件登录过程的密码应用要求。

a） 操作员站在监控组态软件登录前应具备如下预置信息：
1） 操作员站应配置用户应用域，由该域存储和使用监控组态软件相关密钥和敏感数据；
2） 用户已发放身份安全凭证。
b） 操作员站监控组态软件登录密码应用如下：
1） 监控组态软件验证用户登录用安全凭证；
2） 监控组态软件根据用户身份进行授权。

9.2.2.4 用户监控组态编辑

用户监控组态编辑是用户在操作员站设计并编程用户监控组态的过程。

a） 用户监控组态编辑前应具备如下预置信息：
1） 操作员站应配置用户应用域，由该域存储和使用用户监控组态相关密钥和敏感数据；
2） 用户已发放身份安全凭证。
b） 操作员站用户监控组态编辑中密码应用如下：用户监控组态软件应验证用户具备监控组态编辑权限。

9.2.2.5 用户监控组态文件管理

用户监控组态文件管理是对用户监控组态文件的选择、打开、关闭、保存的过程。

a) 操作员站用户监控组态文件管理应具备如下预置信息：
   1) 操作员站应配置用户应用域，由该域存储和使用用户监控组态文件管理相关密钥和敏感数据；
   2) 用户已发放身份安全凭证。
b) 操作员站用户监控组态文件管理中密码应用如下：监控组态软件应验证用户具备文件管理相关权限。

9.2.2.6 用户监控组态运行

用户监控组态运行是对编辑完成的用户监控组态文件执行的过程。

a) 操作员站用户监控组态运行应具备如下预置信息：
   1) 操作员站应配置用户应用域，由该域存储和使用用户监控组态运行相关密钥和敏感数据；
   2) 操作员站应为用户配置登录和权限管理用的安全凭证。
b) 操作员站用户应用组态文件管理中密码应用如下：监控组态软件应验证用户具备监控组态运行权限。

9.2.2.7 设备间交互

设备间交互是操作员站监控组态软件和数据服务站、PLC 控制器的应用之间的接入和交互过程。

a) 操作员站在设备间交互前应具备如下预置信息：
   1) 操作员站应配置用户应用域，由该域存储和使用设备间交互相关密钥和敏感数据；
   2) 操作员站、PLC 控制器、数据服务站已经接入到 PLC 控制系统中。
b) 操作员站和其他设备间交互密码应用如下：
   1) 操作员站和操作员站、PLC 控制器、数据服务站之间完成基于证书或其他身份证明的双向认证，建立安全通道；操作员站和不同的操作员站、PLC 控制器、数据服务站之间，应使用不同的密钥建立独立隔离的安全通道；操作员站和同一操作员站、PLC 控制器、数据服务站的不同应用之间，应使用不同的密钥建立独立隔离的安全通道；
   2) 操作员站和操作员站、PLC 控制器、数据服务站之间传输的数据应由安全通道保证完整性和机密性；
   3) 操作员站下发给 PLC 控制器的关键命令应签名。

9.2.3 注销态

注销态是指操作员站销毁密钥和敏感数据，设备处于不可响应的状态。操作员站注销可以是在线方式或离线方式。

a) 操作员站注销前应具备如下预置信息：
   1) 操作员站应配置设备系统域，由该域存储和使用注册相关密钥和敏感数据；
   2) 操作员站已完成设备注册。
b) 操作员站注销中密码应用如下：
   1) 安全管理服务器生成设备注销指令并签名；
   2) 如果使用在线方式，操作员站和安全管理服务器完成基于证书或其他身份证明的双向认证，建立安全通道，然后传输注销命令到操作员站；如果使用离线方式，安全管理服务器加密注销命令，存储到载体中，离线传递到操作员站；
   3) 操作员站收到设备注销的指令，验证指令合法性；
   4) 操作员站执行指令，销毁密码和敏感数据，修改设备安全生命周期状态。

10 数据服务站密码应用要求

10.1 密码应用功能组成

数据服务站包括芯片硬件、操作系统及运行环境、数据功能模块、密码模块。数据功能模块用于存储和管理 PLC 控制器上送的运行数据、运行状态和事件；密码模块用于存储敏感数据、提供密码安全服务，以及和后台安全管理服务器的交互。数据服务站密码应用功能组成如图 8 所示。

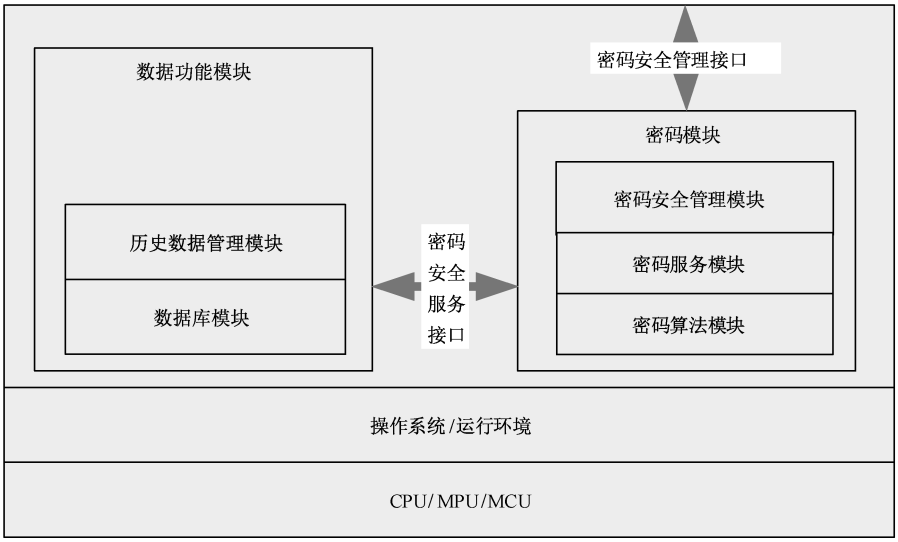

图 8 数据服务站密码应用功能组成

10.2 密码应用要求

10.2.1 初始化态

设备注册是指数据服务站在首次接入到 PLC 控制系统时，完成数据服务站安全初始化的过程。

a) 数据服务站设备注册前应具备如下预置信息：
   1) 数据服务站应配置设备系统域，由该域存储和使用注册相关密钥和敏感数据；
   2) 数据服务站应具有唯一标识，且标识不可被更改；
   3) 数据服务站应具有和该标识绑定的初始化密钥。
b) 数据服务站设备注册中密码应用如下：
   1) 数据服务站使用初始化密钥，和安全管理服务器完成双向认证，建立安全通道；
   2) 数据服务站产生新的密钥对，由安全管理服务器发放设备证书或其他身份证明；
   3) 数据服务站在注册完成前，只能执行注册功能，其他功能禁止使用；
   4) 注册完成后，数据服务站应修改其安全状态。

10.2.2 运行态

10.2.2.1 设备签到

本条是增强要求。

设备签到是数据服务站完成启动后，接入到安全管理服务器并同步状态的过程。

a) 数据服务站在设备签到前应具备如下预置信息：

1） 数据服务站应配置系统设备系统域，由该域存储和使用设备签到相关密钥和敏感数据；
2） 数据服务站已完成设备注册。

b） 数据服务站设备签到密码应用如下：
1） 数据服务站和安全管理服务器完成基于证书或其他身份证明的双向认证，建立安全通道；
2） 数据服务站向安全管理服务器同步安全事件记录。

10.2.2.2 数据服务软件运行

数据服务软件运营要求如下。

a） 数据服务站在数据服务软件运行前应具备如下预置信息：数据服务站应配置用户应用域，由该域存储和使用数据服务相关密钥和敏感数据。

b） 数据服务站数据服务软件运行密码应用如下：
1） 数据服务站启动数据服务软件；
2） 数据服务站对关键数据加密存储。

10.2.2.3 设备间交互

设备间交互是数据服务站和操作员站、PLC 控制器之间的接入和交互过程。

a） 数据服务站在设备间交互前应具备如下预置信息：
1） 数据服务站应配置用户应用域，由该域存储和使用设备间交互相关密钥和敏感数据；
2） 操作员站、PLC 控制器、数据服务站已经接入到 PLC 控制系统中。

b） 数据服务站和设备间交互密码应用如下：
1） 数据服务站和 PLC 控制器、操作员站之间完成基于证书或其他身份证明的双向认证，建立安全通道；数据服务站和不同的 PLC 控制器、操作员站之间，和同一 PLC 控制器、操作员站的不同应用之间，应使用不同的密钥建立独立隔离的安全通道；
2） 数据服务站和 PLC 控制器、操作员站之间传输的信息应由安全通道保证完整性和机密性。

10.2.3 注销态

注销态是指数据服务站销毁密钥和敏感数据，设备处于不可响应的状态。工程师站注销可以是在线方式或离线方式。

a） 数据服务站注销前应具备如下预置信息：
1） 数据服务站应配置设备系统域，由该域存储和使用注册相关密钥和敏感数据；
2） 数据服务站已完成设备注册。

b） 数据服务站注销中密码应用如下：
1） 安全管理服务器生成设备注销指令并签名；
2） 如果使用在线方式，数据服务站和安全管理服务器完成基于证书或其他身份证明的双向认证，建立安全通道，然后传输注销命令到数据服务站；如果使用离线方式，安全管理服务器加密注销命令，存储到载体中，离线传递到数据服务站；
3） 数据服务站收到设备注销的指令，验证指令合法性；
4） 数据服务站执行指令，销毁密码和敏感数据，修改设备安全生命周期状态。

11 安全管理服务器密码应用要求

安全管理服务器是一个后台服务器，和 PLC 控制系统各设备中的密码模块配合，实现安全配置和

安全管理功能。安全管理服务器提供密码安全管理接口，可见附录 B。

安全管理服务器密码应用功能如下：

a） 为 PLC 控制系统提供密钥管理功能；

b） 配合设备完成设备注册，为设备分发密钥、证书或其他身份证明；

c） 配合设备完成系统更新，为系统数据包提供签名，为操作提供授权；

d） 配合设备完成应用更新与个人化，为应用数据包提供签名，为操作提供授权；

e） 管理设备安全生命周期，控制设备安全状态切换；

f） 配合设备完成签到，实现设备接入认证；

g） 配合设备完成安全状态、安全事件记录同步；

h） 配合设备间的交互，提供接入授权。

12 PLC 控制系统密码应用接口

PLC 控制系统中，密码应用接口主要通过密码模块提供，包括“密码安全管理接口”和“密码安全服务接口”两个部分，如图 9 所示。

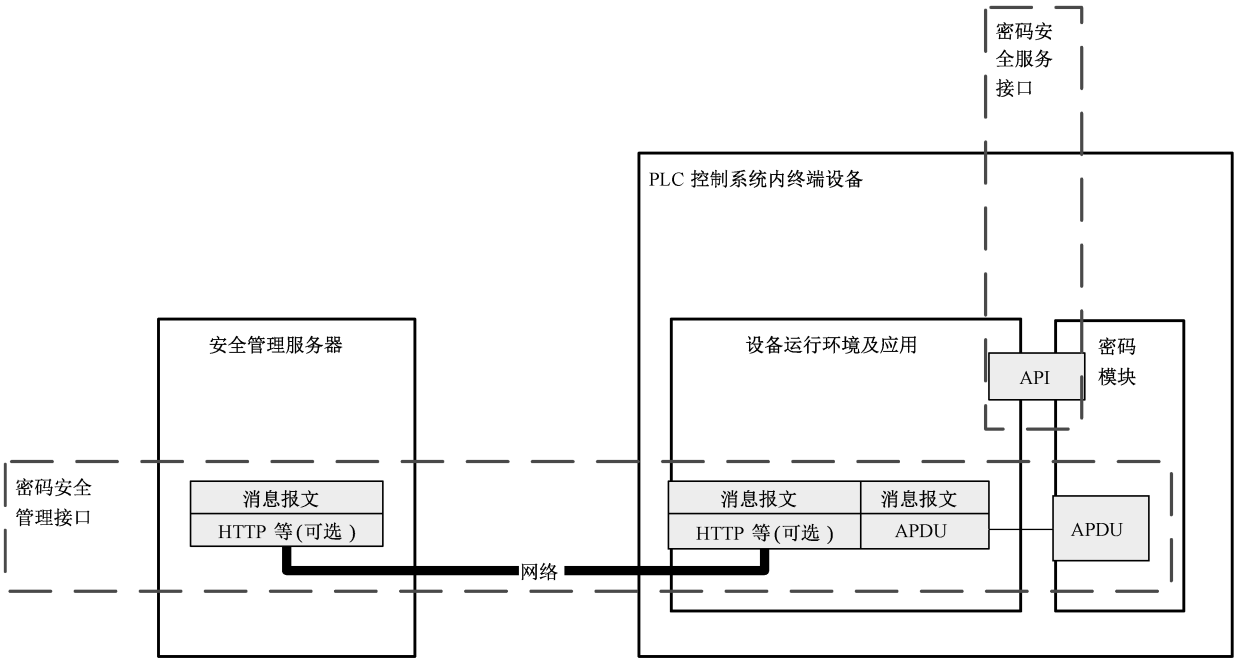

图 9 PLC 控制系统密码应用接口概要图

密码安全管理接口是安全管理服务器和 PLC 控制系统中的终端设备之间的接口，用于实现 PLC 控制系统中设备的注册、接入、状态同步、安全生命周期等密码安全管理功能。该接口需要穿越网络，为保证通用性，宜采用消息报文形式提供。为了和密码模块的通用 APDU 命令格式兼容，在消息报文中可包含 APDU 命令，密码模块执行 APDU 命令完成密码安全功能。附录 B 给出了一个采用 JSON 报文的接口参考，JSON 报文格式遵循 IETF RFC4627。该接口可承载在 HTTP 传输协议或其他传输协议上。

密码安全服务接口是 PLC 控制系统终端设备内部业务应用和密码模块之间的接口，用于为系统设备内部业务应用提供本地密码计算、安全保护等密码安全功能。该接口存在于系统设备内的运行环境中，宜使用 API 形式提供。附录 C 给出了一个 API 格式的接口参考。

附 录 A
（资料性）
PLC 控制系统各组成部分

A.1 概述

参考 GB/T 33008.1 规范定义，PLC 控制系统是以 PLC 控制器为中心、控制技术和信息技术结合的基本工业控制系统，是工作在现场控制层的基础系统。PLC 控制系统在局域环境内，实现对被控对象的实时逻辑控制，具有逻辑控制功能丰富、可靠性高、实时性强、资源受限的特点。PLC 控制系统一般由 PLC 控制器、工程师站、操作员站和数据服务站组成。

A.2 PLC 控制器

PLC 控制器实现用户逻辑组态的配置和运行。PLC 控制器向上通过工业局域网总线和过程监控层的上位机互联，实现用户逻辑组态运行、组态监测、数据存储和上送；向下通过现场总线与现场设备层的传感器、执行器等互联构成控制回路，通过用户逻辑组态中的逻辑运算、顺序控制、状态推演、定时、计数与基本算术运算等操作指令，实现对各类机械电子装置的控制。PLC 控制器基本组成如图 A.1 所示。

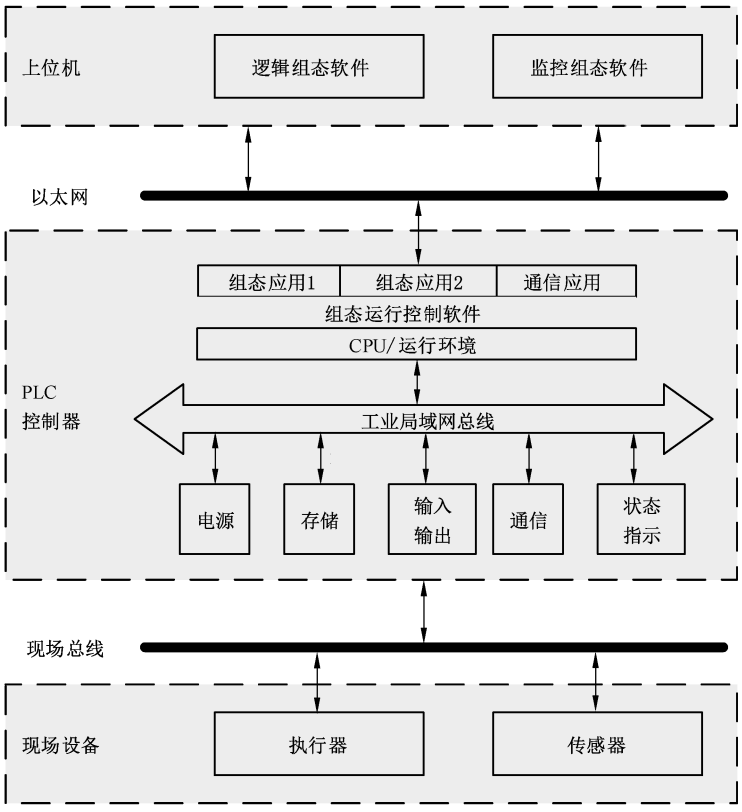

图 A.1 PLC 控制器基本组成

A.3 工程师站

工程师站一般采用通用计算机设备，在设备上运行逻辑组态软件，实现用户逻辑组态的编辑、生成

和下装。

A.4 操作员站

操作员站一般采用通用计算机设备或专业计算设备，在设备上运行监控组态软件，实现用户监控组态的编辑和运行，对 PLC 控制系统的运行状态、事件进行监测。

A.5 数据服务站

数据服务站一般采用通用计算机设备，在设备上运行数据库系统软件，实现 PLC 控制系统的运行状态、数据和事件的存档。

附 录 B
（资料性）
密码安全管理接口参考

B.1 密码安全管理接口概述

本附录给出了使用JSON报文形式的安全管理接口参考，其中每个应用接口均采用请求(request)—响应(response)的命令交互流程，即每次交互都是一个完整事务，包括一个请求报文和一个响应报文，请求报文只能由服务端发起，密码模块只能被动响应并返回响应报文。为了和密码模块的通用APDU命令格式兼容，在应用接口的JSON报文中包含APDU命令，密码模块执行APDU命令完成密码安全功能。

B.2 密码安全管理接口列表

PLC控制系统中，密码安全管理接口列表见表B.1。

表 B.1 密码安全管理接口列表

	应用接口定义	接口说明
安全管理接口	TEEPowerOnNotify	设备上电通知。该消息可携带设备状态等数据；如果安全管理服务器有请求，那么可以响应该通知，启动一个请求—响应事务
	TEEQueryNotify	设备轮询通知，可携带设备状态等数据。如果服务器有请求，那么可以响应该通知，启动一个请求—响应事务
	TEEResultRequest TEEResultResponse	设备非标准功能的扩展报文。对于非标定义的功能，服务器通过该报文中携带APDU，设备处理完成后返回响应。该报文提供了一种双方交互的扩展机制
	TEEGetUUIDRequest TEEGetUUIDResponse	获取设备唯一标识报文。服务器发出获取设备唯一ID的请求，设备返回唯一标识
	TEEGetPropertyRequest TEEGetPropertyResponse	设备安全配置信息报文。服务器发出请求，获取设备安全配置信息；设备返回安全配置信息响应
	TEEStorePropertyRequest TEEStorePropertyResponse	存储设备安全配置信息报告。服务器发出请求，下发设备安全配置信息；设备写入安全配置信息返回响应
	TEESyncStatusRequest TEESyncStatusResponse	设备状态同步报文。服务器发出请求同步设备生命周期状态、应用状态、审计信息；设备获取信息并返回响应
	TEELockRequest TEELockResponse TEEUnlockRequest TEEUnlockResponse	锁定—解锁设备报文。服务器发出请求，锁定设备；设备更改状态后返回响应。服务器发出请求，解锁设备；设备更改状态后返回响应
	TEESuspendRequest TEESuspendResponse TEEUnSuspendRequest TEEUnSuspendResponse	挂起—解挂设备报文。服务器发出请求，挂起设备；设备更改状态后返回响应。服务器发出请求，解挂设备；设备更改状态后返回响应

表 B.1　密码安全管理接口列表（续）

	应用接口定义	接口说明
安全管理接口	TEEExitRequest TEEExitResponse	设备关闭退出报文。服务器发出请求，关闭设备；设备更改状态后返回响应
	TEEFactoryNotify TEEFactoryResetRequest TEEFactoryResetResponse	设备重置报文。服务器发出请求，重置设备安全状态为出厂时状态。设备清空敏感数据，复位为初始状态，然后返回响应
	TEEGetStateRequest TEEGetStateResponse	获取设备生命周期状态报文。服务器发出请求，获取设备生命周期状态；设备提取状态信息返回响应
	TEEStoreStatusRequest TEEStoreStatusResponse	存储设备生命周期状态报文。服务器发出设备生命周期变更请求，设备设置生命周期状态后返回
	TEEVerifyPINRequest TEEVerifyPINResponse TEEChangePINRequest TEEChangePINResponse TEEReloadPINRequest TEERelaodPINResponse	PIN 验证报文；PIN 变更报文；PIN 重置报文。服务器发出请求，设备完成操作后返回响应
	TEEAPPDownloadNotify TEEAPPInstallRequest TEEAPPInstallResponse	应用下装报文。设备发出下装通知，服务器下发下装请求，设备完成 APP 安装，并为其创建和绑定安全域，然后返回响应
	TEEAPPInitNotify TEEAPPStorePropertyRequest TEEAPPStorePropertyResponse	应用初始化报文。设备发出 APP 初始化通知，服务器下发安全配置信息，设备将 APP 安全配置信息写入，返回响应
	TEEAPPUpdateRequest TEEAPPUpdateResponse	应用更新报文。服务器下发更新请求，设备完成更新后返回响应
	TEEAPPDeleteNotify TEEAPPDeleteRequest TEEAPPDeleteResponse	应用删除报文。设备发出 APP 删除通知，服务器下发删除请求，设备删除 APP 及其绑定安全域，完成后返回响应
	TEEAPPLockRequest TEEAPPLockResponse TEEAPPUnlockRequest TEEAPPUnlockResponse	应用锁定—解锁报文。服务器发出请求，锁定 APP；设备更改 APP 状态后返回响应。服务器发出请求，解锁 APP；设备更改 APP 状态后返回响应
	TEEAPPSuspendRequest TEEAPPSuspendResponse TEEAPPUnSuspendRequest TEEAPPUnSuspendResponse	应用挂起—解挂报文。服务器发出请求，挂起 APP。可信运行环境更改 APP 关联安全域状态后返回响应。服务器发出请求，解挂 APP。可信运行环境更改 APP 关联安全域状态后返回响应
	TEEAPPExitRequest TEEAPPExitResponse	应用关闭报文。服务器发出请求，关闭 APP；设备更改 APP 状态后返回响应

表 B.1　密码安全管理接口列表（续）

	应用接口定义	接口说明
安全管理接口	TEEAPPFactoryNotify TEEAPPFactoryResetRequest TEEAPPFactoryResetResponse	应用重置报文。服务器发出请求，重置 APP 为出厂时状态；设备清空 APP 安全数据后返回响应
	TEEAPPGetStatusRequest TEEAPPGetStatusResponse	获取应用生命周期状态报文。服务器发出应用生命周期状态请求命令，设备获取指定 APP 生命周期状态后返回
	TEEAPPStoreStatusRequest TEEAPPStoreStatusResponse	存储应用生命周期状态报文。服务器发出应用生命周期变更请求，设备设置指定 APP 生命周期状态后返回
	TEEAPPGenKeyPairRequest TEEAPPGenKeyPairResponse	应用生成密钥对报文。服务器发出密钥对生成请求，设备产生密钥对后返回公钥
	TEEAPPPutKeyRequest TEEAPPPutKeyResponse	应用密钥配置或替换报文。服务器发出密钥配置或替换命令请求，设备完成操作后返回
	TEEAPPGetPropertyRequest TEEAPPGetPropertyResponse	获取应用安全配置报文。服务器发出 APP 配置请求，设备返回指定 APP 配置信息响应
	TEEAPPStorePropertyRequest TEEAPPStorePropertyResponse	存储应用安全配置报文。服务器下发 APP 配置信息；设备为指定 APP 写入配置信息后返回响应
	TEEAPPCteateFileRequest TEEAPPCreateFileResponse	应用创建文件报文。服务器发出请求，设备为指定 APP 操作后返回响应
	TEEAPPDeleteFileRequest TEEAPPDeleteFileResponse	应用删除文件报文。服务器发出请求，设备为指定 APP 操作后返回响应
	TEEAPPReadBinaryRequest TEEAPPReadBinaryResponse	应用读二进制文件报文。服务器发出请求，设备为指定 APP 操作后返回响应
	TEEAPPUpateBinaryRequest TEEAPPUpdateBinaryResponse	应用更新二进制文件报文。服务器发出请求，设备为指定 APP 操作后返回响应
	TEEAPPAuthInitRequest TEEAPPAuthInitResponse TEEAPPAuthFinalRequest TEEAPPAuthFinalResponse	设备双向认证报文。服务器发出认证初始化请求，设备中指定 APP 认证操作后返回响应。服务器发出认证完成请求，设备中指定 APP 认证操作返回响应
	TEEP2PAccessVerifyNotify TEEP2PAccessVerifyRequest TEEP2PAccessVerifyResponse	设备申请访问授权报文。设备发出授权申请通知，服务器在请求中携带授权，设备收到授权后验证返回响应

B.3 密码安全管理接口详细定义

B.3.1 通用 Request 结构

```
{
  "TEEXXXRequest":{
    "seId":"String",
    "appId":"String",
    "cmds":[cmdData]
  }
}
```

TEEXXXRequest 表示该请求的名称，在实际使用中 XXX 应替换为对应的请求名称。

seId：密码模块的唯一标识。

appId：密码模块内安全域的唯一标识。

cmds：需要 SE 执行的 APDU 命令序列。cmdData 结构如下：

```
"cmdData":{
    "cmdIndex":int,
    "apduList":[apduData],
  }
```

cmdIndex：命令的序号，由命令发出者生成一个唯一的序号，用于区分每个命令。

apduList：该命令需要执行的 APDU 序列。apduData 结构如下：

```
"apduData":{
    "index": int,
    "apdu":"String",
    "result":"String",
    "resultSW":int,
    "expectSW":[int]
  }
```

index：APDU 序列编号。

apdu：APDU 命令序列，BCD 码表示，两个字符表示一个 16 进制数。

result：命令执行结果，BCD 码表示，两个字符表示一个 16 进制数。在 Request 命令中不使用，默认填“”。

resultSW：命令执行的结果。在 Request 命令中不使用，默认填 0。

expectSW：命令执行的期望结果集合，可以有多个期望的结果。

B.3.2 通用 Response 结构

```
{
  " TEEXXXResponse":{
    "seId":"String",
    "appId":"String",
    "code":int,
    "message":"String",
    "cmds":[cmdData],
```

```
    }
}
```

seId:密码模块的唯一标识。

appId: 密码模块内安全域的唯一标识。

code: 命令执行结果。

message: 命令执行结果的说明性文字。

cmds :SE 执行 APDU 后的结果返回列表。cmdData 结构如下:

```
"cmdData":{
    "cmdIndex":int,
    "needWaitResult":boolean,
    "apduList":[apduData],
  }
```

cmdIndex:命令的序号。

needWaitResult: 是否需要等待返回。在 Response 命令中无意义,将 Request 中的数据直接返回。

apduList: APDU 执行结果序列。apduData 结构如下:

```
"apduData":{
    "index": int,
    "apdu":"String",
    "result":"String",
    "resultSW":int,
    "expectSW":int[]
  }
```

index: APDU 序列编号。

apdu: APDU 命令序列,在 Response 命令中无意义,默认填“”。

result: 命令执行结果,BCD 码表示,两个字符表示一个 16 进制数。

resultSW: 命令执行的结果。

expectSW: 命令执行的期望结果集合,在 Response 命令中无意义,将 Request 中的数据直接返回。

B.3.3 TEEPowerNotify

```
{
  "TEEPowerNotify":{
    "seId":"String",
    "seName":"String",
    "seStatus ": int,
    "appStatusList":[appStatus]
  }
}
```

seId: 密码模块的唯一标识。

seName: 密码模块的名称。

seStatus: 密码模块的状态。

appStatusList: appStatus 的链表,记录该 SE 中所有安全域的状态。appStatus 的结构如下:

```
"appStatus":{
    "appId":"String",
```

```
    "appStatus": int,
    }
```

appId：安全域的唯一标识。

appStatus：安全域状态。

SE 上电后，会发出该通知。安全服务器收到通知后，会更新对应 SE 及 SE 中安全域的状态。同时检查是否有需要该 SE 执行的命令，如果有，采用通用 Request 结构返回。

B.3.4 TEEQueryNotify

```
{
  "TEEQueryNotify":{
    "seId":"String",
    "seStatus ": int,
    "seName":"String",
    "appStatusList":[appStatus]
  }
}
```

SE 上电后，会定时发出该通知查询是否有需要执行的命令。安全服务器收到通知后，会检查是否有需要该 SE 执行的命令，如果有，采用通用 Request 结构返回。

B.3.5 TEEGetUUID

TEEGetUUIDRequest 结构见“通用 Request 结构”，其中 cmdData 定义如下：

```
"cmdData":{
  "cmdIndex":具体值由服务器生成序列码填写,
  "needWaitResult":false,
  "apduList":[
    {
      "index": 0,
      "apdu":  GET SE UUID,
      "result": "",
      "resultSW":0,
      "expectSW":[9000]
    }]
}
```

GET SE UUID 详细定义见 GET SE UUID 命令说明

TEEGetUUIDResponse 结构见“通用 Response 结构”，其中 cmdData 定义如下：

```
"cmdData":{
  "cmdIndex":与 request 命令中保持一致,
  "apduList":[
    {
      "index": 0,
      "apdu": "" ,
      "result":具体值由 SE 根据处理情况填写,
```

```
        "resultSW":具体值由 SE 根据处理情况填写,
        "expectSW":[9000]
      }]
}
```

B.3.6 TEEGetProperty

TEEGetPropertyRequest 结构见“通用 Request 结构”,其中 cmdData 定义如下:

```
"cmdData":{
  "cmdIndex":具体值由服务器生成序列码填写,
  "needWaitResult":false,
  "apduList":[
    {
      "index": 0,
      "apdu":  GET APP PROPERTY,
      "result": "",
      "resultSW":0,
      "expectSW":[9000]
    }]
}
```

GET APP PROPERTY 详细定义见 GET APP PROPERTY 命令定义。

TEEGetPropertyResponse 结构见“通用 Response 结构”,其中 cmdData 定义如下:

```
"cmdData":{
  "cmdIndex":与 request 命令中保持一致,
  "apduList":[
    {
      "index": 0,
      "apdu": "" ,
      "result":具体值由 SE 根据处理情况填写,
      "resultSW":具体值由 SE 根据处理情况填写,
      "expectSW":[9000]
    }]
}
```

B.3.7 TEEStoreProperty

TEEStorePropertyRequest 结构见“通用 Request 结构”,其中 cmdData 定义如下:

```
"cmdData":{
  "cmdIndex":具体值由服务器生成序列码填写,
  "needWaitResult":false,
  "apduList":[
    {
      "index": 0,
      "apdu":  SET APP PROPERTY,
      "result": "",
```

```
      "resultSW":0,
      "expectSW":[9000]
    }]
}
```

SET APP PROPERTY 详细定义见 SET APP PROPERTY。

TEEStorePropertyResponse 结构见“通用 Response 结构”，其中 cmdData 定义如下：

```
"cmdData":{
  "cmdIndex":与 request 命令中保持一致,
  "apduList":[
    {
      "index": 0,
      "apdu": "" ,
      "result":具体值由 SE 根据处理情况填写,
      "resultSW":具体值由 SE 根据处理情况填写,
      "expectSW":[9000]
    }]
}
```

B.3.8 TEESyncStatus

TEESyncStatusRequest 结构见“通用 Request 结构”，其中 cmdData 定义如下：

```
"cmdData":{
  "cmdIndex":具体值由服务器生成序列码填写,
  "needWaitResult":false,
  "apduList":[
    {
      "index": 0,
      "apdu":  GET APP STATUS,
      "result": "",
      "resultSW":0,
      "expectSW":[9000]
    }]
}
```

GET APP STATUS 详细定义见 GET APP STATUS 命令定义。

TEESyncStatusResponse 结构见“通用 Response 结构”，其中 cmdData 定义如下：

```
"cmdData":{
  "cmdIndex":与 request 命令中保持一致,
  "apduList":[
    {
      "index": 0,
      "apdu": "" ,
      "result": 具体值由 SE 根据处理情况填写,
      "resultSW":具体值由 SE 根据处理情况填写,
      "expectSW":[9000]
```

```
    }]
}
```

B.3.9 TEELock

TEELockRequest 结构见“通用 Request 结构”,其中 cmdData 定义如下:

```
"cmdData":{
  "cmdIndex":具体值由服务器生成序列码填写,
  "needWaitResult":false,
  "apduList":[
    {
      "index": 0,
      "apdu":  SET APP STATUS,
      "result": "",
      "resultSW":0,
      "expectSW":[9000]
    }]
}
```

SET APP STATUS 详细定义见 SET APP STATUS 命令定义。

TEESyncStatusResponse 结构见“通用 Response 结构”,其中 cmdData 定义如下:

```
"cmdData":{
  "cmdIndex":与 request 命令中保持一致,
  "apduList":[
    {
      "index": 0,
      "apdu": "" ,
      "result": 具体值由 SE 根据处理情况填写,
      "resultSW":具体值由 SE 根据处理情况填写,
      "expectSW":[9000]
    }]
}
```

B.3.10 TEEUnlock

TEEUnlockRequest 结构见“通用 Request 结构”,其中 cmdData 定义如下:

```
"cmdData":{
  "cmdIndex":具体值由服务器生成序列码填写,
  "needWaitResult":false,
  "apduList":[
    {
      "index": 0,
      "apdu":  SET APP STATUS,
      "result": "",
      "resultSW":0,
```

"expectSW":[9000]
}]
}

SET APP STATUS 详细定义见 SET APP STATUS 命令定义。

TEEUnlockResponse 结构见“通用 Response 结构”，其中 cmdData 定义如下：

"cmdData":{
"cmdIndex":与 request 命令中保持一致，
"apduList":[
{
"index": 0，
"apdu": "" ，
"result":具体值由 SE 根据处理情况填写，
"resultSW":具体值由 SE 根据处理情况填写，
"expectSW":[9000]
}]
}

B.3.11 TEESuspend

TEESuspendRequest 结构见“通用 Request 结构”，其中 cmdData 定义如下：

"cmdData":{
"cmdIndex":具体值由服务器生成序列码填写，
"needWaitResult":false，
"apduList":[
{
"index": 0，
"apdu": SET APP STATUS，
"result": ""，
"resultSW":0，
"expectSW":[9000]
}]
}

SET APP STATUS 详细定义见 SET APP STATUS 命令定义。

TEESuspendResponse 结构见“通用 Response 结构”，其中 cmdData 定义如下：

"cmdData":{
"cmdIndex":与 request 命令中保持一致，
"apduList":[
{
"index": 0，
"apdu": "" ，
"result":具体值由 SE 根据处理情况填写，
"resultSW":具体值由 SE 根据处理情况填写，
"expectSW":[9000]
}]

```
}
```

B.3.12 TEEUnSuspend

TEEUnSuspendRequest 结构见“通用 Request 结构”,其中 cmdData 定义如下:

```
"cmdData":{
  "cmdIndex":具体值由服务器生成序列码填写,
  "needWaitResult":false,
  "apduList":[
    {
      "index": 0,
      "apdu":  SET APP STATUS,
      "result": "",
      "resultSW":0,
      "expectSW":[9000]
    }]
}
```

SET APP STATUS 详细定义见 SET APP STATUS 命令定义。

TEEUnSuspendResponse 结构见“通用 Response 结构”, 其中 cmdData 定义如下:

```
"cmdData":{
  "cmdIndex":与 request 命令中保持一致,
  "apduList":[
    {
      "index": 0,
      "apdu": "" ,
      "result":具体值由 SE 根据处理情况填写,
      "resultSW":具体值由 SE 根据处理情况填写,
      "expectSW":[9000]
    }]
}
```

B.3.13 TEEExit

TEEExitRequest 结构见“通用 Request 结构”,其中 cmdData 定义如下:

```
"cmdData":{
  "cmdIndex":具体值由服务器生成序列码填写,
  "needWaitResult":false,
  "apduList":[
    {
      "index": 0,
      "apdu":  SET APP STATUS,
      "result": "",
      "resultSW":0,
      "expectSW":[9000]
    }]
```

```
}
```

SET APP STATUS 详细定义见 SET APP STATUS 命令定义。

TEEExitResponse 结构见“通用 Response 结构”，其中 cmdData 定义如下：

```
"cmdData":{
  "cmdIndex":与 request 命令中保持一致，
  "apduList":[
    {
      "index": 0,
      "apdu": "" ,
      "result":具体值由 SE 根据处理情况填写，
      "resultSW":具体值由 SE 根据处理情况填写，
      "expectSW":[9000]
    }]
}
```

B.3.14 TEEFactoryNotify

TEEFactoryNotify 结构如下：

```
{
  "TEEFactoryNotify":{
    "seId":"String",
  }
}
```

seId：密码模块的唯一标识。

B.3.15 TEEFactoryReset

TEEFactoryResetRequest 结构见“通用 Request 结构”，其中 cmdData 定义如下：

```
"cmdData":{
  "cmdIndex":具体值由服务器生成序列码填写，
  "needWaitResult":false,
  "apduList":[
    {
      "index": 0,
      "apdu":  FACTORY RESET,
      "result": "",
      "resultSW":0,
      "expectSW":[9000]
    }]
}
```

FACTORY RESET 详细定义见 FACTORY RESET 命令定义。

TEEFactoryResetResponse 结构见“通用 Response 结构”，其中 cmdData 定义如下：

```
"cmdData":{
  "cmdIndex":与 request 命令中保持一致，
  "apduList":[
```

```
        {
          "index": 0,
          "apdu": "" ,
          "result": 具体值由 SE 根据处理情况填写,
          "resultSW":具体值由 SE 根据处理情况填写,
          "expectSW":[9000]
        }]
    }
```

B.3.16 TEEGetStatus

TEEGetStatusRequest 结构见“通用 Request 结构”,其中 cmdData 定义如下:

```
"cmdData":{
  "cmdIndex":具体值由服务器生成序列码填写,
  "needWaitResult":false,
  "apduList":[
    {
      "index": 0,
      "apdu":  GET APP STATUS,
      "result": "",
      "resultSW":0,
      "expectSW":[9000]
    }]
}
```

GET APP STATUS 详细定义见 GET APP STATUS 命令定义。

TEEGetStatusResponse 结构见“通用 Response 结构”,其中 cmdData 定义如下:

```
"cmdData":{
  "cmdIndex":与 request 命令中保持一致,
  "apduList":[
    {
      "index": 0,
      "apdu": "" ,
      "result":具体值由 SE 根据处理情况填写,
      "resultSW":具体值由 SE 根据处理情况填写,
      "expectSW":[9000]
    }]
}
```

B.3.17 TEEStoreStatus

TEEStoreStatusRequest 结构见“通用 Request 结构”,其中 cmdData 定义如下:

```
"cmdData":{
  "cmdIndex":具体值由服务器生成序列码填写,
  "needWaitResult":false,
  "apduList":[
```

```
    {
      "index": 0,
      "apdu":  SET APP STATUS,
      "result": "",
      "resultSW":0,
      "expectSW":[9000]
    }]
}
```

SET APP STATUS 详细定义见 SET APP STATUS 命令定义。

TEEStoreStatusResponse 结构见“通用 Response 结构”，其中 cmdData 定义如下：

```
"cmdData":{
  "cmdIndex":与 request 命令中保持一致,
  "apduList":[
    {
      "index": 0,
      "apdu": "" ,
      "result":具体值由 SE 根据处理情况填写,
      "resultSW":具体值由 SE 根据处理情况填写,
      "expectSW":[9000]
    }]
}
```

B.3.18 TEEVerifyPIN

TEEVerifyPINRequest 结构见“通用 Request 结构”,其中 cmdData 定义如下：

```
"cmdData":{
  "cmdIndex":具体值由服务器生成序列码填写,
  "needWaitResult":false,
  "apduList":[
    {
      "index": 0,
      "apdu":  VERIFY PIN,
      "result": "",
      "resultSW":0,
      "expectSW":[9000]
    }]
}
```

VERIFY PIN 详细定义见 VERIFY PIN 命令定义。

TEEVerifyPINResponse 结构见“通用 Response 结构”，其中 cmdData 定义如下：

```
"cmdData":{
  "cmdIndex":与 request 命令中保持一致,
  "apduList":[
    {
      "index": 0,
```

```
        "apdu": "" ,
        "result":具体值由 SE 根据处理情况填写,
        "resultSW":具体值由 SE 根据处理情况填写,
        "expectSW":[9000]
      }]
}
```

B.3.19 TEEChangePIN

TEEChangePINRequest 结构见“通用 Request 结构”,其中 cmdData 定义如下:

```
"cmdData":{
  "cmdIndex":具体值由服务器生成序列码填写,
  "needWaitResult":false,
  "apduList":[
    {
      "index": 0,
      "apdu": CHANGE PIN,
      "result": "",
      "resultSW":0,
      "expectSW":[9000]
    }]
}
```

CHANGE PIN 详细定义见 CHANGE PIN 命令定义。

TEEChangePINResponse 结构见“通用 Response 结构”,其中 cmdData 定义如下:

```
"cmdData":{
  "cmdIndex":与 request 命令中保持一致,
  "apduList":[
    {
      "index": 0,
      "apdu": "" ,
      "result":具体值由 SE 根据处理情况填写,
      "resultSW":具体值由 SE 根据处理情况填写,
      "expectSW":[9000]
    }]
}
```

B.3.20 TEEReloadPIN

TEEReloadPINRequest 结构见“通用 Request 结构”,其中 cmdData 定义如下:

```
"cmdData":{
  "cmdIndex":具体值由服务器生成序列码填写,
  "needWaitResult":false,
  "apduList":[
    {
```

```
        "index": 0,
        "apdu": RELOAD PIN,
        "result": "",
        "resultSW":0,
        "expectSW":[9000]
      }]
}
```

RELOAD　PIN 详细定义见 RELOAD PIN 命令定义。

TEEReloadPINResponse 结构见“通用 Response 结构”，其中 cmdData 定义如下：

```
"cmdData":{
  "cmdIndex":与 request 命令中保持一致,
  "apduList":[
    {
      "index": 0,
      "apdu": "" ,
      "result": "具体值由 SE 根据处理情况填写",
      "resultSW":具体值由 SE 根据处理情况填写,
      "expectSW":[9000]
    }]
}
```

B.3.21　TEEAPPDownloadNotify

TEEAPPDownloadNotify 结构如下：

```
{
  " TEEAPPDownloadNotify ":{
    "seId":"String",
    "appId":"String",
  }
}
```

seId：密码模块的唯一标识。

appId：密码模块内应用(APP)的唯一标识。

B.3.22　TEEAPPInstall

TEEAPPInstallRequest 结构见“通用 Request 结构”，其中 cmdData 定义如下：

```
"cmdData":{
  "cmdIndex":具体值由服务器生成序列码填写,
  "needWaitResult":false,
  "apduList":[
    {
      "index": 0,
      "apdu":APP DOWNLOAD,
      "result": "",
      "resultSW":0,
```

```
        "expectSW":[9000]
      }]
}
```

APP DOWNLOAD 详细定义见 APP DOWNLOAD 命令定义。

TEEAPPInstallResponse 结构见“通用 Response 结构”，其中 cmdData 定义如下：

```
"cmdData":{
  "cmdIndex":与 request 命令中保持一致,
  "apduList":[
    {
      "index": 0,
      "apdu": "" ,
      "result":具体值由 SE 根据处理情况填写,
      "resultSW":具体值由 SE 根据处理情况填写,
      "expectSW":[9000]
    }]
}
```

B.3.23 TEEAPPInitNotify

TEEAPPInitNotify 结构如下：

```
{
  " TEEAPPInitNotify ":{
    "seId":"String",
    "appId":"String",
  }
}
```

seId：密码模块的唯一标识。

appId：密码模块内安全域的唯一标识。

B.3.24 TEEAPPUpdate

TEEAPPUpdateRequest 结构见“通用 Request 结构”，其中 cmdData 定义如下：

```
"cmdData":{
  "cmdIndex":具体值由服务器生成序列码填写,
  "needWaitResult":false,
  "apduList":[
    {
      "index": 0,
      "apdu": APP DOWNLOAD,
      "result": "",
      "resultSW":0,
      "expectSW":[9000]
    }]
}
```

APP DOWNLOAD 详细定义见 APP DOWNLOAD 命令说明。

TEEAPPUpdateResponse 结构见“通用 Response 结构”，其中 cmdData 定义如下：

```
"cmdData":{
  "cmdIndex":与 request 命令中保持一致，
  "apduList":[
    {
      "index": 0,
      "apdu": "" ,
      "result": 具体值由 SE 根据处理情况填写，
      "resultSW":具体值由 SE 根据处理情况填写，
      "expectSW":[9000]
    }]
}
```

B.3.25 TEEAPPDeleteNotify

TEEAPPDeleteNotify 结构如下：

```
{
  " TEEAPPDeleteNotify ":{
    "seId":"String",
    "appId":"String",
  }
}
```

seId：密码模块的唯一标识。

appId：密码模块内安全域的唯一标识。

B.3.26 TEEAPPDelete

TEEAPPDeleteRequest 结构见“通用 Request 结构”，其中 cmdData 定义如下：

```
"cmdData":{
  "cmdIndex":具体值由服务器生成序列码填写，
  "needWaitResult":false,
  "apduList":[
    {
      "index": 0,
      "apdu": DELETE,
      "result": "",
      "resultSW":0,
      "expectSW":[9000]
    }]
}
```

DELETE 详细定义见 DELETE 命令定义。

TEEAPPDeleteResponse 结构见“通用 Response 结构”，其中 cmdData 定义如下：

```
"cmdData":{
  "cmdIndex":与 request 命令中保持一致，
```

```
  "apduList":[
    {
      "index": 0,
      "apdu": "" ,
      "result":具体值由 SE 根据处理情况填写,
      "resultSW":具体值由 SE 根据处理情况填写,
      "expectSW":[9000]
    }]
}
```

B.3.27 TEEAPPLock

TEEAPPLockRequest 结构见“通用 Request 结构”,其中 cmdData 定义如下:

```
"cmdData":{
  "cmdIndex":具体值由服务器生成序列码填写,
  "needWaitResult":false,
  "apduList":[
    {
      "index": 0,
      "apdu": SET APP STATUS,
      "result": "",
      "resultSW":0,
      "expectSW":[9000]
    }]
}
```

SET APP STATUS 详细定义见 SET APP STATUS 命令定义。

TEEAPPLockResponse 结构见“通用 Response 结构”,其中 cmdData 定义如下:

```
"cmdData":{
  "cmdIndex":与 request 命令中保持一致,
  "apduList":[
    {
      "index": 0,
      "apdu": "" ,
      "result": 具体值由 SE 根据处理情况填写,
      "resultSW":具体值由 SE 根据处理情况填写,
      "expectSW":[9000]
    }]
}
```

B.3.28 TEEAPPUnlock

TEEAPPUnlockRequest 结构见“通用 Request 结构”,其中 cmdData 定义如下:

```
"cmdData":{
  "cmdIndex":具体值由服务器生成序列码填写,
  "needWaitResult":false,
```

"apduList":[
{
"index": 0,
"apdu": SET APP STATUS,
"result": "",
"resultSW":0,
"expectSW":[9000]
}]
}

SET APP STATUS 详细定义见 SET APP STATUS 命令定义。

TEEAPPUnlockResponse 结构见“通用 Response 结构”，其中 cmdData 定义如下：

"cmdData":{
"cmdIndex":与 request 命令中保持一致，
"apduList":[
{
"index": 0,
"apdu": "" ,
"result": 具体值由 SE 根据处理情况填写，
"resultSW":具体值由 SE 根据处理情况填写，
"expectSW":[9000]
}]
}

B.3.29 TEEAPPSuspend

TEEAPPSuspendRequest 结构见“通用 Request 结构”，其中 cmdData 定义如下：

"cmdData":{
"cmdIndex":具体值由服务器生成序列码填写，
"needWaitResult":false,
"apduList":[
{
"index": 0,
"apdu": SET APP STATUS,
"result": "",
"resultSW":0,
"expectSW":[9000]
}]
}

SET APP STATUS 详细定义见 SET APP STATUS 命令定义。

TEEAPPSuspendResponse 结构见“通用 Response 结构”，其中 cmdData 定义如下：

"cmdData":{
"cmdIndex":与 request 命令中保持一致，
"apduList":[
{

```
        "index": 0,
        "apdu": "" ,
        "result": 具体值由 SE 根据处理情况填写,
        "resultSW":具体值由 SE 根据处理情况填写,
        "expectSW":[9000]
      }]
}
```

B.3.30 TEEAPPUnSuspend

TEEAPPUnSuspendRequest 结构见“通用 Request 结构”,其中 cmdData 定义如下:

```
"cmdData":{
  "cmdIndex":具体值由服务器生成序列码填写,
  "needWaitResult":false,
  "apduList":[
    {
      "index": 0,
      "apdu":  SET APP STATUS,
      "result": "",
      "resultSW":0,
      "expectSW":[9000]
    }]
}
```

SET APP STATUS 详细定义见 SET APP STATUS 命令定义。

TEEAPPUnSuspendResponse 结构见“通用 Response 结构”,其中 cmdData 定义如下:

```
"cmdData":{
  "cmdIndex":与 request 命令中保持一致,
  "apduList":[
    {
      "index": 0,
      "apdu": "" ,
      "result": 具体值由 SE 根据处理情况填写,
      "resultSW":具体值由 SE 根据处理情况填写,
      "expectSW":[9000]
    }]
}
```

B.3.31 TEEAPPExit

TEEAPPExitRequest 结构见“通用 Request 结构”,其中 cmdData 定义如下:

```
"cmdData":{
  "cmdIndex":具体值由服务器生成序列码填写,
  "needWaitResult":false,
  "apduList":[
    {
```

```
        "index": 0,
        "apdu": SET APP STATUS,
        "result": "",
        "resultSW":0,
        "expectSW":[9000]
      }]
}
```

SET APP STATUS 详细定义见 SET APP STATUS 命令定义。

TEEAPPExitRequest 结构见“通用 Response 结构”，其中 cmdData 定义如下：

```
"cmdData":{
  "cmdIndex":与 request 命令中保持一致,
  "apduList":[
    {
      "index": 0,
      "apdu": "" ,
      "result":具体值由 SE 根据处理情况填写,
      "resultSW":具体值由 SE 根据处理情况填写,
      "expectSW":[9000]
    }]
}
```

B.3.32 TEEAPPFactoryNotify

```
{
  " TEEAPPFactoryNotify ":{
    "seId":"String",
    "appId":"String",
  }
}
```

seId：密码模块的唯一标识。

appId：密码模块内安全域的唯一标识。

B.3.33 TEEAPPFactoryReset

TEEAPPFactoryResetRequest 结构见“通用 Request 结构”，其中 cmdData 定义如下：

```
"cmdData":{
  "cmdIndex":具体值由服务器生成序列码填写,
  "needWaitResult":false,
  "apduList":[
    {
      "index": 0,
      "apdu": APP FACTORY RESET,
      "result": "",
      "resultSW":0,
      "expectSW":[9000]
```

```
    }]
}
```

APP FACTORY RESET 详细定义见 APP FACTORY RESET 命令定义。

TEEAPPFactoryResetResponse 结构见“通用 Response 结构”，其中 cmdData 定义如下：

```
"cmdData":{
  "cmdIndex":与 request 命令中保持一致，
  "apduList":[
    {
      "index": 0，
      "apdu": "" ，
      "result": 具体值由 SE 根据处理情况填写，
      "resultSW":具体值由 SE 根据处理情况填写，
      "expectSW":[9000]
    }]
}
```

B.3.34 TEEAPPGetStatus

TEEAPPGetStatusRequest 结构见“通用 Request 结构”，其中 cmdData 定义如下：

```
"cmdData":{
  "cmdIndex":具体值由服务器生成序列码填写，
  "needWaitResult":false，
  "apduList":[
    {
      "index": 0，
      "apdu": GET APP STATUS，
      "result": ""，
      "resultSW":0，
      "expectSW":[9000]
    }]
}
```

GET APP STATUS 详细定义见 GET APP STATUS 命令定义。

TEEAPPGetStatusResponse 结构见“通用 Response 结构”，其中 cmdData 定义如下：

```
"cmdData":{
  "cmdIndex":与 request 命令中保持一致，
  "apduList":[
    {
      "index": 0，
      "apdu": "" ，
      "result":具体值由 SE 根据处理情况填写，
      "resultSW":具体值由 SE 根据处理情况填写，
      "expectSW":[9000]
    }]
}
```

B.3.35 TEEAPPStoreStatus

TEEAPPStoreStatusRequest 结构见“通用 Request 结构”,其中 cmdData 定义如下:

```
"cmdData":{
  "cmdIndex":具体值由服务器生成序列码填写,
  "needWaitResult":false,
  "apduList":[
    {
      "index": 0,
      "apdu": SET APP STATUS,
      "result": "",
      "resultSW":0,
      "expectSW":[9000]
    }]
}
```

SET APP STATUS 详细定义见 SET APP STATUS 命令定义。

TEEAPPStoreStatusResponse 结构见“通用 Response 结构”,其中 cmdData 定义如下:

```
"cmdData":{
  "cmdIndex":与 request 命令中保持一致,
  "apduList":[
    {
      "index": 0,
      "apdu": "" ,
      "result":具体值由 SE 根据处理情况填写,
      "resultSW":具体值由 SE 根据处理情况填写,
      "expectSW":[9000]
    }]
}
```

B.3.36 TEEAPPGenKeyPair

TEEAPPGenKeyPairRequest 结构见“通用 Request 结构”,其中 cmdData 定义如下:

```
"cmdData":{
  "cmdIndex":具体值由服务器生成序列码填写,
  "needWaitResult":false,
  "apduList":[
    {
      "index": 0,
      "apdu": GEN KEYPAIR,
      "result": "",
      "resultSW":0,
      "expectSW":[9000]
    }]
}
```

GEN KEYPAIR 详细定义见 GEN KEYPAIR 命令定义。

TEEAPPGenKeyPairResponse 结构见“通用 Response 结构”，其中 cmdData 定义如下：

```
"cmdData":{
  "cmdIndex":与 request 命令中保持一致,
  "apduList":[
    {
      "index": 0,
      "apdu": "" ,
      "result":具体值由 SE 根据处理情况填写,
      "resultSW":具体值由 SE 根据处理情况填写,
      "expectSW":[9000]
    }]
}
```

B.3.37 TEEAPPPutKey

TEEAPPPutKeyRequest 结构见“通用 Request 结构”,其中 cmdData 定义如下：

```
"cmdData":{
  "cmdIndex":具体值由服务器生成序列码填写,
  "needWaitResult":false,
  "apduList":[
    {
      "index": 0,
      "apdu": PUT KEY,
      "result": "",
      "resultSW":0,
      "expectSW":[9000]
    }]
}
```

PUT KEY 详细定义见 PUT KEY 命令定义。

TEEAPPPutKeyResponse 结构见“通用 Response 结构”，其中 cmdData 定义如下：

```
"cmdData":{
  "cmdIndex":与 request 命令中保持一致,
  "apduList":[
    {
      "index": 0,
      "apdu": "" ,
      "result": 具体值由 SE 根据处理情况填写,
      "resultSW":具体值由 SE 根据处理情况填写,
      "expectSW":[9000]
    }]
}
```

B.3.38 TEEAPPGetProperty

TEEAPPGetPropertyRequest 结构见“通用 Request 结构”,其中 cmdData 定义如下：

```
"cmdData":{
  "cmdIndex":具体值由服务器生成序列码填写,
  "needWaitResult":false,
  "apduList":[
    {
      "index": 0,
      "apdu": GET APP PROPERTY,
      "result": "",
      "resultSW":0,
      "expectSW":[9000]
    }]
}
```

GET APP PROPERTY 详细定义见 GET APP PROPERTY 命令定义。

TEEAPPGetPropertyResponse 结构见“通用 Response 结构”，其中 cmdData 定义如下：

```
"cmdData":{
  "cmdIndex":与 request 命令中保持一致,
  "apduList":[
    {
      "index": 0,
      "apdu": "" ,
      "result":具体值由 SE 根据处理情况填写,
      "resultSW":具体值由 SE 根据处理情况填写,
      "expectSW":[9000]
    }]
}
```

B.3.39 TEEAPPStoreProperty

TEEAPPStorePropertyRequest 结构见“通用 Request 结构”,其中 cmdData 定义如下：

```
"cmdData":{
  "cmdIndex":具体值由服务器生成序列码填写,
  "needWaitResult":false,
  "apduList":[
    {
      "index": 0,
      "apdu": SET APP PROPERTY,
      "result": "",
      "resultSW":0,
      "expectSW":[9000]
    }]
}
```

SET APP PROPERTY 详细定义见 SET APP PROPERTY 命令定义。

TEEAPPStorePropertyResponse 结构见“通用 Response 结构”，其中 cmdData 定义如下：

```
"cmdData":{
```

```
  "cmdIndex":与 request 命令中保持一致,
  "apduList":[
    {
      "index": 0,
      "apdu": "" ,
      "result":具体值由 SE 根据处理情况填写,
      "resultSW":具体值由 SE 根据处理情况填写,
      "expectSW":[9000]
    }]
}
```

B.3.40 TEEAPPCreateFile

TEEAPPCreateFileRequest 结构见“通用 Request 结构”,其中 cmdData 定义如下:

```
"cmdData":{
  "cmdIndex":具体值由服务器生成序列码填写,
  "needWaitResult":false,
  "apduList":[
    {
      "index": 0,
      "apdu": CREATE FILE,
      "result": "",
      "resultSW":0,
      "expectSW":[9000]
    }]
}
```

CREATE FILE 详细定义见 CREATE FILE 命令定义。

TEEAPPCreateFileResponse 结构见“通用 Response 结构”,其中 cmdData 定义如下:

```
"cmdData":{
  "cmdIndex":与 request 命令中保持一致,
  "apduList":[
    {
      "index": 0,
      "apdu": "" ,
      "result":具体值由 SE 根据处理情况填写,
      "resultSW":具体值由 SE 根据处理情况填写,
      "expectSW":[9000]
    }]
}
```

B.3.41 TEEAPPDeleteFile

TEEAPPDeleteFileRequest 结构见“通用 Request 结构”,其中 cmdData 定义如下:

```
"cmdData":{
  "cmdIndex":具体值由服务器生成序列码填写,
```

"needWaitResult":false,
"apduList":[
{
"index": 0,
"apdu": DELETE FILE,
"result": "",
"resultSW":0,
"expectSW":[9000]
}]
}

DELETE FILE 详细定义见 A.4.10。

TEEAPPDeleteFileResponse 结构见“通用 Response 结构”，其中 cmdData 定义如下：

"cmdData":{
"cmdIndex":与 request 命令中保持一致,
"apduList":[
{
"index": 0,
"apdu": "" ,
"result":具体值由 SE 根据处理情况填写,
"resultSW":具体值由 SE 根据处理情况填写,
"expectSW":[9000]
}]
}

B.3.42 TEEAPPReadBinary

TEEAPPReadBinaryRequest 结构见“通用 Request 结构”,其中 cmdData 定义如下：

"cmdData":{
"cmdIndex":具体值由服务器生成序列码填写,
"needWaitResult":false,
"apduList":[
{
"index": 0,
"apdu": READ BINARY,
"result": "",
"resultSW":0,
"expectSW":[9000]
}]
}

READ BINARY 详细定义见 READ BINARY 命令定义。

TEEAPPReadBinaryRequest 结构见“通用 Response 结构”，其中 cmdData 定义如下：

"cmdData":{
"cmdIndex":与 request 命令中保持一致,
"apduList":[

```
      {
        "index": 0,
        "apdu": "" ,
        "result": 具体值由 SE 根据处理情况填写,
        "resultSW":具体值由 SE 根据处理情况填写,
        "expectSW":[9000]
      }]
  }
```

B.3.43 TEEAPPUpateBinary

TEEAPPUpateBinaryRequest 结构见“通用 Request 结构”,其中 cmdData 定义如下:

```
"cmdData":{
  "cmdIndex":具体值由服务器生成序列码填写,
  "needWaitResult":false,
  "apduList":[
    {
      "index": 0,
      "apdu": UPDATE BINARY,
      "result": "",
      "resultSW":0,
      "expectSW":[9000]
    }]
}
```

UPDATE BINARY 详细定义见 UPDATE BINARY 命令定义。

TEEAPPUpateBinaryResponse 结构见“通用 Response 结构”,其中 cmdData 定义如下:

```
"cmdData":{
  "cmdIndex":与 request 命令中保持一致,
  "apduList":[
    {
      "index": 0,
      "apdu": "" ,
      "result": 具体值由 SE 根据处理情况填写,
      "resultSW":具体值由 SE 根据处理情况填写,
      "expectSW":[9000]
    }]
}
```

B.3.44 TEEAPPAuthInit

TEEAPPAuthInitRequest 结构见“通用 Request 结构”,其中 cmdData 定义如下:

```
"cmdData":{
  "cmdIndex":具体值由服务器生成序列码填写,
  "needWaitResult":false,
  "apduList":[
```

```
    {
      "index": 0,
      "apdu": INITIALIZE UPDATE,
      "result": "",
      "resultSW":0,
      "expectSW":[9000]
    }]
}
```

INITIALIZE UPDATE 详细定义见 INITIALIZE UPDATE 命令定义。

TEEAPPAuthInitResponse 结构见“通用 Response 结构”，其中 cmdData 定义如下：

```
"cmdData":{
  "cmdIndex":与 request 命令中保持一致,
  "apduList":[
    {
      "index": 0,
      "apdu": "" ,
      "result": 具体值由 SE 根据处理情况填写,
      "resultSW":具体值由 SE 根据处理情况填写,
      "expectSW":[9000]
    }]
}
```

B.3.45 TEEAPPAuthFinal

TEEAPPAuthFinalRequest 结构见“通用 Request 结构”，其中 cmdData 定义如下：

```
"cmdData":{
  "cmdIndex":具体值由服务器生成序列码填写,
  "needWaitResult":false,
  "apduList":[
    {
      "index": 0,
      "apdu": EXTERNAL AUTHENTICATION,
      "result": "",
      "resultSW":0,
      "expectSW":[9000]
    }]
}
```

EXTERNAL AUTHENTICATION 详细定义见 EXTERNAL AUTHENTICATION 命令定义。

TEEAPPAuthFinalResponse 结构见“通用 Response 结构”，其中 cmdData 定义如下：

```
"cmdData":{
  "cmdIndex":与 request 命令中保持一致,
  "apduList":[
    {
      "index": 0,
```

```
        "apdu": "" ,
        "result": 具体值由 SE 根据处理情况填写,
        "resultSW":具体值由 SE 根据处理情况填写,
        "expectSW":[9000]
      }]
}
```

B.3.46 TEEP2PAccessNotify

```
{
  " TEEP2PAccessNotify ":{
    "seId":"String",
    "appId":"String",
  }
}
```

seId:密码模块的唯一标识。

appId: 密码模块内安全域的唯一标识。

B.3.47 TEEP2PAccess

TEEP2PAccessRequest 结构见“通用 Request 结构”,其中 cmdData 定义如下:

```
"cmdData":{
  "cmdIndex":具体值由服务器生成序列码填写,
  "needWaitResult":false,
  "apduList":[
    {
      "index": 0,
      "apdu": P2P ACCESS VERIFY,
      "result": "",
      "resultSW":0,
      "expectSW":[9000]
    }]
}
```

P2P ACCESS VERIFY 详细定义见 P2P ACCESS VERIFY 命令定义。

TEEP2PAccessResponse 结构见“通用 Response 结构”,其中 cmdData 定义如下。

```
"cmdData":{
  "cmdIndex":与 request 命令中保持一致,
  "apduList":[
    {
      "index": 0,
      "apdu": "" ,
      "result": 具体值由 SE 根据处理情况填写,
      "resultSW":具体值由 SE 根据处理情况填写,
      "expectSW":[9000]
    }]
```

}

B.4 APDU 命令定义

B.4.1 SELECT 命令

B.4.1.1 功能说明

根据应用 AID 或文件 FID 选择文件。

当选择应用目录成功后，应用切换到初始状态。

B.4.1.2 命令报文

SELECT 命令报文见表 B.2。

表 B.2 SELECT 命令

代码	值	含义
CLA	‘00’	
INS	‘A4’	SELECT
P1	‘04/00’	
P2	‘00’	
Lc	‘xx’	AID 的长度
DATA	‘xxxx…’	将被选定的应用的 AID
Le	‘00’	

B.4.1.3 命令数据域

数据域包含应用 AID 或文件 FID。

应用 AID 长度要求‘05’～‘10’。

设备上默认主安全域对象标识为：A000000003000000。

使用命令：00A4040008A000000003000000 选择安全域。

B.4.1.4 响应报文

该命令执行成功的报文状态码是 0x9000。

文件没有找到返回：0x6A82

SELECT 命令中响应的数据字段是由被选定的应用指定的信息组成的。

B.4.2 INITIALIZE UPDATE(SCP02)命令

B.4.2.1 功能说明

在安全通道的显式发起期间，INITIALIZE UPDATE 命令用于在 SE 和主机之间传送 SE 和会话数据。这个命令开始一个安全通道会话的发起。

在当前安全通道的任何时候，都可以将 INITIALIZE UPDATE 命令发送给 SE 以发起一个新的安全通道会话。

B.4.2.2 安全性要求

密钥没有被锁定。

B.4.2.3 命令报文

INITIALIZE UPDATE 命令按照表 B.3 进行编码。

表 B.3 INITIALIZE UPDATE 命令

代码	值	含义
CLA	'80'	
INS	'50'	INITIALIZE UPDATE
P1	'xx'	密钥类型号
P2	'00'	引用控制参数 P2
Lc	'10'	host challenge 的长度
Data	'xx xx…'	host challenge
Le	'00'	

B.4.2.4 命令数据域

命令消息的数据域包含了 16 字节的 host challenge。这个数据由 SE 外实体选择,对这个会话来说应该是唯一的。

B.4.2.5 响应报文

该命令执行成功的报文状态码 0x9000。响应消息中包含表 B.4 中数据的串联(没有分隔符)。

表 B.4 INITIALIZE UPDATE 响应

名称	长度
密钥派生数据	10 字节
密钥信息	2 字节
序列计数器	2 字节
SEchallenge	6 字节
SE 认证码	8 字节

密钥派生数据一般由后端系统来派生 SE 静态密钥。

密钥信息包含了密钥的版本号和安全通道的协议标识符,这里为'02',用于发起安全通道会话。

序列计数器是 SE 内一个递增的计数器,用于创建会话密钥。

SEChallenge 是内部生成的随机数。

SE 认证码是一个认证密码。

B.4.3 EXTERNAL AUTHENTICATION(SCP02)命令

B.4.3.1 功能说明

EXTERNAL AUTHENTICATION 命令用于 SE 内对 SE 外实体进行认证,同时确定后续命令所要求的安全级别。

在执行 EXTERNAL AUTHENTICATION 命令之前,应该先成功的执行 INITIALIZE UPDATE 命令。

B.4.3.2 安全性要求

密钥没有被锁定。

B.4.3.3 命令报文

EXTERNAL AUTHENTICATE 命令按照表 B.5 进行编码。

表 B.5 EXTERNAL AUTHENTICATE 命令

代码	值	含义
CLA	'84'	
INS	'82'	EXTERNAL AUTHENTICATE
P1	'xx'	安全级别
P2	'00'	引用控制参数 P2
Lc	'10'	主机认证码及 MAC 的长度
Data	'xx xx…'	主机认证码及 MAC
Le		不存在

引用控制参数 P1 定义了在这个安全会话范围内,EXTERNAL AUTHENTICATE 命令(不包括这个命令)后所有安全消息命令所需要的安全级别。引用控制参数 P1 定义见表 B.6。

表 B.6 EXTERNAL AUTHENTICATE 命令控制参数 P1

b8	b7	b6	b5	b4	b3	b2	b1	描述
0	0	1	1	0	0	1	1	RFU
0	0	1	1	0	0	0	1	RFU
0	0	1	1	0	0	0	0	RFU
0	0	0	1	0	0	1	1	DECRYPTION、请求 MAC 和响应 MAC
0	0	0	1	0	0	0	1	请求 MAC 和响应 MAC
0	0	0	1	0	0	0	0	响应 MAC
0	0	0	0	0	0	1	1	DECRYPTION 和响应 MAC
0	0	0	0	0	0	0	1	请求 MAC
0	0	0	0	0	0	0	0	不期待安全消息

B.4.3.4 命令数据域

命令消息的数据域包含了主机认证码和请求 MAC。

B.4.3.5 响应报文

这个命令的响应消息中没有数据段。

这个命令成功执行后，它的状态字应该为 0x9000。

B.4.4 GET CHALLENGE 命令

B.4.4.1 功能说明

获取随机数以便用于安全相关的过程。每次获取的随机数只对下一条指令有效，无论下一条指令是否使用该随机数，下一条指令结束后该随机数都失效。

B.4.4.2 命令报文

GET CHANLLENGE 命令按照表 B.7 进行编码。

表 B.7 GET CHALLENGE 命令

代码	值	含义
CLA	'80'	
INS	'40'	Get Challenge
P1	'00'	
P2	'00'	
Le	'10'	challenge 的长度

B.4.4.3 命令数据域

命令无数据域

B.4.4.4 响应报文

返回指定长度的随机数。

成功返回状态字：0x9000。

B.4.5 PUT KEY 命令

B.4.5.1 功能说明

PUT KEY 命令可以用来：

——用一个新的密钥替换一个已经存在的密钥：新密钥可以有与被替换的密钥相同的或不同的密钥类型号，但是必须与被替换的密钥有相同的密钥标识符；

——用新密钥替换多个已经存在的密钥：新密钥可以有与被替换的密钥相同的或不同的密钥类型号（对于所有的新密钥是同样的），但是必须与被替换的密钥有相同的密钥标识符；

——增加一个新的单密钥：新密钥有一个与其他已存在的密钥不同的密钥标识符/密钥类型号；

——增加新的多密钥：新密钥有一个与其他已存在的密钥不同的密钥标识符/密钥类型号（对于所

有的新密钥是同样的)。

当密钥管理操作要求有多个 PUT KEY 命令时,推荐将多个 PUT KEY 命令连接起来,以保证操作的完整性。

B.4.5.2 安全性要求

应建立 SCP02 安全通道,密钥没有被锁定。

B.4.5.3 命令报文

PUT KEY 命令按照表 B.8 进行编码。

表 B.8 PUT KEY 命令

代码	值	含义
CLA	'80'或'84'	
INS	'D8'	PUT KEY
P1	'xx'	引用控制参数 P1
P2	'xx'	引用控制参数 P2
Lc	'xx'	数据字段的长度
Data	'xxxx…'	密钥数据和 MAC(如果 MAC 存在)
Le	'00'	

引用控制参数 P1 定义了一个密钥类型和这个命令后是否还有更多 PUT KEY 命令。

密钥类型号标识了已经存在于 SE 上的一个密钥或一组密钥。值'00'表示一个新的密钥或一组密钥将被加入(新的密钥类型号是在命令消息的数据字段中指示的)。

密钥类型号的编码是从'01'到'7F'。

PUT KEY 命令消息的引用控制参数 P1 按照表 B.9 进行编码。

表 B.9 PUT KEY 控制参数 P1

b8	b7	b6	b5	b4	b3	b2	b1	含义
0								最后(或仅有)命令
1								更多的 PUT KEY 命令
	*	*	*	*	*	*	*	密钥类型号

引用控制参数 P2 定义了一个密钥标识符和指明数据字段中包含一个密钥还是多个密钥。

当命令消息的数据字段中包含一个密钥时,引用控制参数 P2 指出了这个密钥的密钥标识符。当命令消息的数据字段中包含多个密钥时,引用控制参数 P2 指出了数据字段中第一个密钥的密钥标识符。命令消息数据字段中后续的每个密钥都暗含一个密钥标识符,这个密钥标识符开始于第一个密钥标识符,每次增 1。

密钥标识符的编码是从'00'到'7F'。

PUT KEY 命令消息的引用控制参数 P2 按照表 B.10 进行编码。

表 B.10　PUT KEY 控制参数 P2

b8	b7	b6	b5	b4	b3	b2	b1	含义
0								单个密钥
1								多个密钥
	*	*	*	*	*	*	*	密钥标识符

B.4.5.4　命令数据域

命令消息的数据字段包含一个新的密钥类型号，后面跟着是一个或多个密钥的数据字段，按照表 B.11 进行编码。

表 B.11　PUT KEY 命令数据域

长度	含义
1	密钥类型
可变的：1～n	密钥数据
可变的：1～n	密钥数据

新的密钥类型号定义了：

——将在 SE 上创建的一个新密钥或一组密钥的版本号（P1 所指示的密钥类型号设置为 0）；

——将要替换已存在的一个密钥或一组密钥的一个新密钥或一组密钥的版本号（P1 所指示的密钥类型号不是 0）。

如果数据字段包含多个密钥，所有的密钥将共享一个相同的密钥类型号，并且在命令数据字段中的顺序将反映密钥标识符的递增的顺序。

密钥数据字段的结构按照表 B.12 进行编码。

表 B.12　密钥数据字段

长度	含义
1	密钥类型
1	密钥或密钥元素的长度
可变的：1～n	密钥或密钥元素的数据值
1	密钥校验值的长度
可变的：0～n	密钥校验值（如果存在）

密钥校验值为 3 个字节，只针对对称密钥有效，使用对称密钥对 16 个 0 进行 ECB 加密后取前面 3 个字节。

B.4.5.5　响应报文

操作成功返回 0x9000。

B.4.6 INSTALL 命令

B.4.6.1 功能说明

INSTALL 命令用来安装一个辅助安全域。

B.4.6.2 安全性要求

应建立 SCP02 安全通道，密钥没有被锁定。

B.4.6.3 命令报文

INSTALL 命令按照表 B.13 进行编码。

表 B.13 INSTALL 命令

代码	值	含义
CLA	‘80’或‘84’	
INS	‘E6’	INSTALL
P1	‘0C’	引用控制参数 P1
P2	‘00’	引用控制参数 P2
Lc	‘xx’	数据字段的长度
Data	‘xxxx…’	安装的数据
Le	‘00’	

B.4.6.4 命令数据域

INSTALL 命令的数据字段按照表 B.14 编码。

表 B.14 INSTALL 命令数据字段

名称	字节长度	值
模块 AID 长度	1	07
模块 AID	7	A0000000035350
执行模块 AID 长度	1	00
应用 AID 长度	1	
应用权限长度	1	01
应用权限	1	80
应用安装参数	1	04
应用密钥文件大小	2	
应用属性文件大小	2	
安装授权 TOKEN 长度	1	00
授权 TOKEN	xx	

B.4.6.5 响应报文

操作成功返回 0x9000。

B.4.7 APP DOWNLOAD 命令

B.4.7.1 功能说明

APP DOWNLOAD 命令用来下载应用，通过这条命令将数据传入到设备进行解密。

B.4.7.2 安全性要求

应建立 SCP02 安全通道，密钥没有被锁定。

B.4.7.3 命令报文

APP DOWNLOAD 命令按照表 B.15 进行编码。

表 B.15 APP DOWNLOAD 命令

代码	值	含义
CLA	‘84’	
INS	‘E8’	APP DOWNLOAD
P1	‘00’	引用控制参数 P1
P2	‘xx’	分包下载的包号低 8 位
Lc	‘xx’	数据字段的长度
Data	‘xxxx…’	数据
Le	‘00’	

APP DOWNLOAD 命令消息的引用控制参数 P1 按照表 B.16 进行编码。

表 B.16 APP DOWNLOAD 命令控制参数 P1

b8	b7	b6	b5	b4	b3	b2	b1	含义
0								非最后包
1								最后一包
	*	*	*	*	*	*	*	分包下载的包号高 7 位

B.4.7.4 命令数据域

APP DOWNLOAD 命令的第一包数据格式按照表 B.17 进行编码。

表 B.17　APP DOWNLOAD 命令数据域第一包

名称	字节长度	值
头部数据标识	1	0xE2
头部数据长度	1	
应用 AID 标识	1	0x4F
应用 AID 长度	1	
应用 AID	5～16	
应用 SM2 签名标识	1	0xC3
应用 SM2 签名长度	1	0x40
应用 SM2 签名	64	
应用数据长度标识	1	0xC4
应用数据长度描述的长度	1	0x82 长度用两个字节表示 0x83 长度用两个字节表示
应用数据长度	2 或 3	

非第一包数据按照表 B.18 进行编码。

表 B.18　APP DOWNLOAD 命令数据域非第一包

名称	字节长度	值
应用分包数据	1～255	

B.4.7.5　响应报文

包含数据的包，返回解密后的数据。

操作成功返回 0x9000。

B.4.8　DELETE 命令

B.4.8.1　功能说明

DELETE 命令用来删除一个辅助安全域。

B.4.8.2　安全性要求

必须建立 SCP02 安全通道，密钥没有被锁定。

B.4.8.3　命令报文

DELETE 命令按照表 B.19 进行编码。

表 B.19　DELETE 命令

代码	值	含义
CLA	‘80’或‘84’	
INS	‘E4’	DELETE
P1	‘00’	引用控制参数 P1
P2	‘xx’	引用控制参数 P2
Lc	‘xx’	数据字段的长度
Data	‘xxxx…’	TLV 格式编码的对象
Le	‘00’	

引用控制参数 P2 表示数据字段中的对象被删除还是数据字段中的对象及相关对象被删除，按照表 B.20 进行编码。

表 B.20　DELETE 命令控制参数 P2

b8	b7	b6	b5	b4	b3	b2	b1	含义
0	0	0	0	0	0	0	0	删除对象
1	0	0	0	0	0	0	0	删除对象及相关对象

B.4.8.4　命令数据域

DELETE 命令消息的数据字段将包含被删除的对象的名字，按照表 B.21 进行编码。

表 B.21　DELETE 命令数据字段

名称	字节长度	值
对象标识 tag	1	0x4F
AID 长度	1	
AID	5～16	

B.4.8.5　响应报文

操作成功返回 0x9000。

B.4.9　CREATE FILE 命令

B.4.9.1　功能说明

CREATE FILE 命令用于在 SE 内建立各种类型的文件。执行一次 CREATE FILE 命令只能建立一个文件，创建文件后，系统默认选中该文件，可以进行读写操作。

B.4.9.2　安全性要求

必须建立 SCP02 安全通道或者校验 PIN，密钥没有被锁定。

B.4.9.3 命令报文

CREATE FILE 命令按照表 B.22 进行编码。

表 B.22 CREATE FILE 命令

代码	值	含义
CLA	‘00’	
INS	‘E0’	CREATE FILE
P1	‘00’	引用控制参数 P1
P2	‘00’	引用控制参数 P2
Lc	‘xx’	数据字段的长度
Data	‘xxxx…’	
Le	‘00’	

B.4.9.4 命令数据域

CREATE FILE 命令的数据字段，普通文件按照表 B.23 进行编码。

表 B.23 CREATE FILE 命令数据字段：普通文件

名称	字节长度
文件 ID	2
文件大小	2
访问属性	2
文件读认证 KEY 标识	2
文件写认证 KEY 标识	2

CREATE FILE 命令的数据字段，应用目录文件(ADF)按照表 B.24 进行编码。

表 B.24 CREATE FILE 命令数据字段：ADF 文件

名称	字节长度
文件 ID	2
文件大小	2
访问属性	2
文件读认证 KEY 标识	2
文件写认证 KEY 标识	2
应用安全文件 ID	2
应用属性文件 ID	2
应用 AID 长度	1
应用 AID	5～n

应用访问属性定义：0 bit～3 bit 说明写要求，4 bit～7 bit 说明读要求，取值：

0：任意读或者写；

1：禁止读写；

2：指定 KEY ID SCP02 认证后读写；

3：指定 PIN ID 认证后读写，PIN ID＝0 表示默认；

4：任意 KEY SCP02 认证后读写；

5：任意 PIN 认证后读写；

6：SCP02 或者 PIN 认证之一认证后读写。

如属性为 0x05：表示任意读，PIN 认证后可写。

如属性为 0x21：指定 KEY ID 认证后读，禁止写，读认证 KEY ID 由创建文件的文件读认证 KEY 标识数据域指示。

每个应用都有一个安全文件，用于存放应用的密钥信息，创建应用时应用安全文件 ID 数据域指定对应的安全文件，必须在创建文件的时候指定安全文件 ID，否则读写文件认证无法找到密钥。

应用属性文件是可选文件，保存应用的一些属性信息，不使用可以填 FFFF。

B.4.9.5 响应报文

操作成功返回 0x9000。

B.4.10 DELETE FILE 命令

B.4.10.1 功能说明

删除指定的文件。

B.4.10.2 安全性要求

必须建立 SCP02 安全通道或者 PIN 认证，密钥没有被锁定。

B.4.10.3 命令报文

DELETE FILE 命令按照表 B.25 进行编码。

表 B.25 DELETE FILE 命令

代码	值	含义
CLA	‘00’	
INS	‘E1’	命令编码
P1	‘00’	引用控制参数 P1
P2	‘00’	引用控制参数 P2
Lc	‘02’	数据字段的长度
Data	‘xxxx…’	
Le	‘00’	

B.4.10.4 命令数据域

用 2 个字节表示删除文件的 ID。删除文件是 DF 或者 ADF，下面的文件全部删除。

B.4.10.5 响应报文

操作成功返回 0x9000。

B.4.11 READ BINARY 命令

B.4.11.1 功能说明

读取指定的文件。

B.4.11.2 安全性要求

必须满足文件读权限,密钥没有被锁定。

B.4.11.3 命令报文

READ BINARY 命令按照表 B.26 进行编码。

表 B.26 READ BINARY 命令

代码	值	含义
CLA	'00'	
INS	'B0'	命令编码
P1	'xx'	文件读偏移高字节
P2	'xx'	文件读偏移低字节
Lc	'00'	数据字段的长度
Data	无	
Le	'xx'	读取数据的长度

B.4.11.4 命令数据域

无。

B.4.11.5 响应报文

操作成功返回读取到的数据。

B.4.12 UPDATE BINARY 命令

B.4.12.1 功能说明

文件更新命令。

B.4.12.2 安全性要求

应满足文件写权限,密钥没有被锁定。

B.4.12.3 命令报文

UPDATE BINARY 命令按照表 B.27 进行编码。

表 B.27 UPDATE BINARY 命令

代码	值	含义
CLA	‘00’	
INS	‘D6’	命令编码
P1	‘xx’	文件写偏移高字节
P2	‘xx’	文件写偏移低字节
Lc	‘xx’	数据字段的长度
Data	‘xxxx…’	写入文件的数据
Le	‘00’	读取数据的长度

B.4.12.4 命令数据域

写入文件的数据。

B.4.12.5 响应报文

操作成功返回 0x9000。

B.4.13 VERIFY PIN 命令

B.4.13.1 功能说明

验证用户 PIN 码正确性。

B.4.13.2 安全性要求

密钥没有被锁定。

B.4.13.3 命令报文

VERIFY PIN 命令按照表 B.28 进行编码。

表 B.28 VERIFY PIN 命令

代码	值	含义
CLA	‘00’	
INS	‘20’	命令编码
P1	‘00’	引用控制参数 P1
P2	‘00’	引用控制参数 P2
Lc	‘xx’	PIN 认证数据长度
Data	‘xxxx…’	PIN 认证数据
Le	‘00’	

校验 PIN 之前，请求设备产生 16 字节的随机数，用随机数计算 PIN 认证数据。步骤如下：对 PIN 使用 SM3 算法计算的 HASH 值，取前面 16 字节作为加密密钥，使用计算的加密密钥对随机数加密得

到 PIN 认证数据。

B.4.13.4 命令数据域

用户 PIN。

B.4.13.5 响应报文

操作成功返回 0x9000。

B.4.14 CHANGE PIN 命令

B.4.14.1 功能说明

修改用户 PIN。

B.4.14.2 安全性要求

密钥没有被锁定。

B.4.14.3 命令报文

CHANGE PIN 命令按照表 B.29 进行编码。

表 B.29 CHANGE PIN 命令

代码	值	含义
CLA	‘00’	
INS	‘21’	命令编码
P1	‘00’	引用控制参数 P1
P2	‘00’	引用控制参数 P2
Lc	‘xx’	数据长度
Data	‘xxxx…’	数据
Le	‘00’	

B.4.14.4 命令数据域

CHANGE PIN 命令的数据字段按照表 B.30 进行编码。

表 B.30 CHANGE PIN 命令数据字段

名称	字节长度
旧的 PIN 码计算的 PIN 认证数据	16
加密后新的 PIN 码长度	1
加密后新的 PIN 码	
MAC	4

请求设备产生 16 字节的随机数，用随机数计算 PIN 认证数据。步骤如下：对 PIN 使用 SM3 算法计算的 HASH 值，取前面 16 字节作为加密密钥，使用计算的加密密钥对随机数加密得到 PIN 认证数

据，使用计算的加密密钥对 PIN 码进行加密，如果 PIN 码不足 16 字节，填充 FF。

将前面所有的数据和命令头(P3 加 4 个 Byte)一起用加密密钥计算 MAC。

MAC 计算方法：首先将数据填充到 16 的倍数(在数据末尾添加 0x80，然后添加 00，直到 16 的倍数)，然后使用 SM4 算法的 CBC 模式对数据进行加密，初始 IV 全 0，取最后 4 个字节作为 MAC。

B.4.14.5 响应报文

操作成功返回 0x9000。

B.4.15 RELOAD PIN 命令

B.4.15.1 功能说明

使用重载 PIN 密钥重载 PIN 码。

B.4.15.2 安全性要求

密钥没有被锁定。

B.4.15.3 命令报文

RELOAD PIN 命令按照表 B.31 进行编码。

表 B.31 RELOAD PIN 命令

代码	值	含义
CLA	'80'	
INS	'22'	命令编码
P1	'00'	引用控制参数 P1
P2	'00'	引用控制参数 P2
Lc	'14'	数据长度
Data	'xxxx…'	加密 PIN 数据和 MAC
Le	'00'	

B.4.15.4 命令数据域

RELOAD PIN 命令的数据字段按照表 B.32 进行编码。

表 B.32 RELOAD PIN 命令数据字段

名称	字节长度
用过程密钥加密后的 PIN 码	*n*
用过程密钥计算 MAC	4

RELOAD PIN 之前先获取 SE 随机数，然后使用重载 PIN 密钥加密随机数，SM4 ECB 加密，得到过程密钥，然后用过程密钥加密 PIN 码，如果 PIN 码不足 16 字节，填充 FF。

将加密后的 PIN 数据和命令头(P3 加 4 个 Byte)一起用过程密钥计算 MAC，命令头、PIN 加密数

据、4 Byte MAC 发送到 SE。

B.4.15.5 响应报文

操作成功返回 0x9000。

B.4.16 GEN KEYPAIR 命令

B.4.16.1 功能说明

产生 SM2 密钥对，私钥保存到 SE 内部，公钥可以导出。

B.4.16.2 安全性要求

密钥没有被锁定。

B.4.16.3 命令报文

GEN KEYPAIR 命令按照表 B.33 进行编码。

表 B.33 GET KEYPAIR 命令

代码	值	含义
CLA	'00'	
INS	'13'	命令编码
P1	'xx'	密钥用途
P2	'xx'	密钥标识
Lc	'00'	数据长度
Data		无
Le	'00'	

B.4.16.4 命令数据域

无。

B.4.16.5 响应报文

操作成功返回 0x9000。

B.4.17 GET PUBKEY 命令

B.4.17.1 功能说明

获取 SE 公钥。

B.4.17.2 安全性要求

密钥没有被锁定。

B.4.17.3 命令报文

GET PUBKEY 命令按照表 B.34 进行编码。

表 B.34 GET PUBKEY 命令

代码	值	含义
CLA	‘00’	
INS	‘14’	命令编码
P1	‘xx’	密钥用途
P2	‘xx’	密钥标识
Lc	‘00’	数据长度
Data		无
Le	‘00’	

B.4.17.4 命令数据域

无。

B.4.17.5 响应报文

操作成功返回公钥。

B.4.18 SM2 ZACAL 命令

B.4.18.1 功能说明

用指定的公钥计算 SM2 签名用 ZA。

B.4.18.2 安全性要求

应建立 SCP02 安全通道或者校验 PIN,密钥没有被锁定。

B.4.18.3 命令报文

SM2 ZACAL 命令按照表 B.35 进行编码。

表 B.35 SM2 ZACAL 命令

代码	值	含义
CLA	‘00’	
INS	‘B4’	命令编码
P1	‘xx’	密钥用途
P2	‘xx’	密钥标识
Lc	‘00’	数据长度
Data		无
Le	‘00’	

B.4.18.4 命令数据域

无。

B.4.18.5 响应报文

操作成功返回指定公钥计算后的 ZA。

B.4.19 SM2 CAL 命令

B.4.19.1 功能说明

SM2 功能调用，包括加密、加密、签名、验签。

B.4.19.2 安全性要求

应建立 SCP02 安全通道或者校验 PIN，密钥没有被锁定。

B.4.19.3 命令报文

SM2 CAL 命令按照表 B.36 进行编码。

表 B.36 SM2 CAL 命令

代码	值	含义
CLA	‘00’	
INS	‘B3’	命令编码
P1	‘xx’	密钥用途
P2	‘xx’	密钥标识
Lc	‘xx’	数据长度
Data	‘xx…’	数据
Le	‘00’	

B.4.19.4 命令数据域

SM2 CAL 命令的数据字段按照表 B.37 进行编码。

表 B.37 SM2 CAL 命令数据字段

名称	字节长度
模式	1
加密解密签名验签数据	*n*

模式取值：

1：SM2 加密，输入数据长度 len，返回 96＋len 个字节；

2：SM2 解密，输入 96＋len 个字节，返回 len 个字节；

4：SM2 签名，输入 32 个字节 SM3 算法计算的 HASH 值，返回 64 个字节签名；

8：SM2 验签，输入 32 个字节 SM3 算法计算的 HASH 值、64 个字节的签名，返回签名验证是否成功。

B.4.19.5 响应报文

操作成功返回 0x9000。

B.4.20 SM3 CAL 命令

B.4.20.1 功能说明

用 SM3 算法计算 HASH 值。

B.4.20.2 安全性要求

应建立 SCP02/SCP11 安全通道或者校验 PIN,密钥没有被锁定。

B.4.20.3 命令报文

SM3 CAL 命令按照表 B.38 进行编码。

表 B.38 SM3 CAL 命令

代码	值	含义
CLA	'00'	
INS	'B2'	命令编码
P1	'xx'	分块标识
P2	'00'	
Lc	'xx'	数据长度
Data	'xx…'	数据
Le	'00'	

P1 取值:

0:数据的第一块;

1:中间块数据;

2:数据结束块,进行 HASH 计算,返回 32 字节 HASH。

B.4.20.4 命令数据域

SM3 CAL 命令的数据字段按照表 B.39 进行编码。

表 B.39 SM3 CAL 命令数据字段

名称	字节长度
HASH 计算数据	*n*

B.4.20.5 响应报文

第一块和中间块操作成功返回 0x9000。

最后一块成功返回 HASH 值。

B.4.21 SM4 CAL 命令

B.4.21.1 功能说明

SM4 加密解密计算。

B.4.21.2 安全性要求

应建立 SCP02/SCP11 安全通道或者校验 PIN,密钥没有被锁定。

B.4.21.3 命令报文

SM4 CAL 命令按照表 B.40 进行编码。

表 B.40 SM4 CAL 命令

代码	值	含义
CLA	‘00’	
INS	‘B7’	命令编码
P1	‘xx’	密钥用途
P2	‘xx’	密钥标识
Lc	‘xx’	数据长度
Data	‘xx…’	数据
Le	‘00’	

B.4.21.4 命令数据域

SM4 CAL 命令的数据字段按照表 B.41 进行编码。

表 B.41 SM4 CAL 命令数据字段

名称	字节长度
SM4 计算模式	1
SM4 计算数据	*n*

计算模式取值:

Bit0——0:加密;1: 解密;

Bit1——0:ECB;1: CBC;

Bit2——0:非第一块;1: 第一块。

当进行 CBC 计算时,第一块数据的开始部分包含 16 字节的 IV。

B.4.21.5 响应报文

操作成功加密或者解密的数据。

B.4.22 IMPORT SM2 PUBKEY 命令

B.4.22.1 功能说明

导入 SM2 公钥。

B.4.22.2 安全性要求

应建立 SCP02/SCP11 安全通道或者 PIN 校验，密钥没有被锁定。

B.4.22.3 命令报文

IMPORT SM2 PUBKEY 命令按照表 B.42 进行编码。

表 B.42 IMPORT SM2 PUBKEY 命令

代码	值	含义
CLA	‘80’	
INS	‘1B’	命令编码
P1	‘xx’	密钥用途
P2	‘xx’	密钥标识
Lc	‘64’	数据长度
Data	‘xx…’	数据
Le	‘00’	

B.4.22.4 命令数据域

IMPORT SM2 PUBKEY 命令的数据字段按照表 B.43 进行编码。

表 B.43 IMPORT SM2 PUBKEY 命令数据字段

名称	字节长度
SM2 公钥	64

IMPORT SM2 PUBKEY 之前先获取 SE 随机数，然后使用主控密钥加密随机数，SM4 ECB 加密，得到过程密钥，然后用过程密钥加密 SM2 密钥。

将加密后的数据和命令头(P3 加 4 个 Byte)一起用过程密钥计算 MAC，命令头、加密数据、4 Byte MAC 发送到 SE。

B.4.22.5 响应报文

操作成功返回 0x9000。

B.4.23 GET SE UUID

B.4.23.1 功能说明

获取芯片唯一 ID。

B.4.23.2 命令报文

GET SE UUID 命令按照表 B.44 进行编码。

表 B.44　GET SE UUID 命令

代码	值	含义
CLA	‘00’	
INS	‘A0’	命令编码
P1	‘00’	引用控制参数 P1
P2	‘00’	引用控制参数 P2
Le	‘0’	芯片 UUID 长度

B.4.23.3　命令数据域

无。

B.4.23.4　响应报文

操作成功返回 16 字节的芯片唯一 ID 和 0x9000 状态字。

B.4.24　GET APP STATUS

B.4.24.1　功能说明

获取应用状态，每个应用对应一个安全域，有唯一的 AID 标识，主安全域代表了设备，AID 为 A000000003000000。

B.4.24.2　命令报文

GET APP STATUS 命令按照表 B.45 进行编码。

表 B.45　GET APP STATUS 命令

代码	值	含义
CLA	‘80’	
INS	‘34’	命令编码
P1	‘00’	
P2	‘00’	
Lc	‘xx’	AID 长度
Data	‘xx…’	AID 数据
Le	‘00’	

B.4.24.3　命令数据域

GET APP STATUS 命令的数据字段按照表 B.46 进行编码。

表 B.46　GET APP STATUS 命令数据字段

名称	字节长度
应用 AID 标识	5～16

B.4.24.4　响应报文

返回一个字节的应用状态标识。

成功返回状态字：0x9000。

应用状态定义按照表 B.47 进行编码。

表 B.47　GET APP STATUS 响应

值	说明
0x00	应用未初始化
0x01	应用已初始化
0x03	应用安全状态
0x04	应用锁定状态
0x05	应用挂起状态
0x06	应用注销状态

B.4.25　SET APP STATUS

B.4.25.1　功能说明

设置应用状态，每个应用对应一个安全域，有唯一的 AID 标识，主安全域代表了设备，AID 为 A000000003000000。

B.4.25.2　安全性要求

应建立 SCP02 安全通道或者 PIN 校验，密钥没有被锁定。

B.4.25.3　命令报文

SET APP STATUS 命令按照表 B.48 进行编码。

表 B.48　SET APP STATUS 命令

代码	值	含义
CLA	‘80’	
INS	‘33’	命令编码
P1	‘00’	
P2	‘xx’	应用状态
Lc	‘xx’	AID 长度
Data	‘xx…’	AID 数据
Le	‘00’	

B.4.25.4 命令数据域

SET APP STATUS 命令的数据字段按照表 B.49 进行编码。

表 B.49 SET APP STATUS 命令数据字段

名称	字节长度
应用 AID 标识	5～16

B.4.25.5 响应报文

成功返回状态字:0x9000。

B.4.26 GET APP PROPERTY

B.4.26.1 功能说明

获取应用属性。

B.4.26.2 安全性要求

应建立 SCP02 安全通道或者 PIN 校验,密钥没有被锁定。

B.4.26.3 命令报文

GET APP PROPERTY 命令按照表 B.50 进行编码。

表 B.50 GET APP PROPERTY 命令

代码	值	含义
CLA	‘80’	
INS	‘24’	命令编码
P1	‘00’	
P2	‘00’	
Lc	‘xx’	数据长度
Data	‘xx…’	数据
Le	‘00’	

B.4.26.4 命令数据域

GET APP PROPERTY 命令的数据字段按照表 B.51 进行编码。

表 B.51　GET APP PROPERTY 命令数据字段

名称	字节长度
应用 AID 长度	1
应用 AID	5～16
属性名称长度	1
属性名称	1～32

B.4.26.5　响应报文

返回应用属性值。

成功返回状态字：0x9000。

B.4.27　SET APP PROPERTY

B.4.27.1　功能说明

设置应用属性。

B.4.27.2　安全性要求

应建立 SCP02 安全通道或者 PIN 校验，密钥没有被锁定。

B.4.27.3　命令报文

SET APP PROPERTY 命令按照表 B.52 进行编码。

表 B.52　SET APP PROPERTY 命令

代码	值	含义
CLA	‘80’	
INS	‘23’	命令编码
P1	‘00’	
P2	‘00’	
Lc	‘xx’	数据长度
Data	‘xx…’	数据
Le	‘00’	

B.4.27.4　命令数据域

SET APP PROPERTY 命令的数据字段按照表 B.53 进行编码。

表 B.53　SET APP PROPERTY 命令数据字段

名称	字节长度
应用 AID 长度	1
应用 AID	5～16
属性名称长度	1
属性名称	1～32
属性值长度	1
属性值	

B.4.27.5　响应报文

成功返回状态字：0x9000。

B.4.28　STORE CERT

B.4.28.1　功能说明

保存证书到设备里面，证书链要依次上传，先上传自签名根证书，后续证书由根证书验证。

B.4.28.2　安全性要求

应建立 SCP02 安全通道或者 PIN 校验，密钥没有被锁定。

B.4.28.3　命令报文

STORE CERT 命令按照表 B.54 进行编码。

表 B.54　STORE CERT 命令

代码	值	含义
CLA	‘84’	
INS	‘E2’	命令编码
P1	‘00|01’	是否是最后一包数据
P2	‘xx’	数据包索引，包编号，0～n
Lc	‘xx’	数据长度
Data	‘xx…’	数据
Le	‘00’	

B.4.28.4　命令数据域

STORE CERT 命令的数据字段按表 B.55 进行编码。

表 B.55　STORE CERT 命令数据字段

名称	字节长度	值
密钥对象标识 tag	4	固定：A6048302
密钥标识	1	
密钥类型	1	
数据对象标识 tag	3	固定：BF2181
证书数据长度	1	
证书数据	1～245	

B.4.28.5　响应报文

成功返回状态字：0x9000。

B.4.29　GET CERT

B.4.29.1　功能说明

GET CERT 命令用来查询指定的证书。

B.4.29.2　命令报文

GET CERT 命令按照表 B.56 进行编码。

表 B.56　GET CERT 命令

代码	值	含义
CLA	‘80’	
INS	‘CA’	GET CERT
P1	‘00’	引用控制参数 P1
P2	‘00’	引用控制参数 P2
Lc	‘06 00’	数据字段的长度
Data	‘xxxx…’	数据
Le	‘00’	

B.4.29.3　命令数据域

GET CERT 命令的数据字段按表 B.57 进行编码。

表 B.57　GET CERT 命令数据字段

名称	字节长度	值
密钥对象标识 tag	4	固定：A6048302
密钥标识	1	
密钥类型	1	

第一条命令数据长度为6。

第二条命令获取后续的证书数据，数据长度为0。

B.4.29.4 响应报文

证书数据分包返回，返回状态字0x6310，表示证书数据还没有下载完成，继续下发第二条命令。

最后一条命令返回状态字0x9000。

B.4.30 FACTORY RESET命令

B.4.30.1 功能说明

清除设备上的全部数据，恢复出厂状态，相当于空白设备。

B.4.30.2 命令报文

FACTORY RESET命令按照表B.58进行编码。

表B.58 FACTORY RESET命令

代码	值	含义
CLA	'00'	
INS	'17'	FACTORY RESET
P1	'00'	
P2	'00'	
Le	'00'	

B.4.30.3 命令数据域

命令无数据域。

B.4.30.4 响应报文

成功返回状态字：0x9000。

B.4.31 P2P ACCESS VERIFY

B.4.31.1 功能说明

设备间进行端到端的接入访问时，需要从服务器获取访问授权。执行该命令前先获取随机数，请求服务器对随机数签名，然后调用本命令验证签名。

B.4.31.2 命令报文

P2P ACCESS VERIFY命令按照表B.59进行编码。

表 B.59　P2P ACCESS VERIFY 命令

代码	值	含义
CLA	‘80’	
INS	‘26’	命令编码
P1	‘xx’	服务器公钥的密钥类型
P2	‘xx’	服务器公钥的密钥标识
Lc	‘40’	签名长度
Data	‘xx…’	签名数据
Le	‘00’	

B.4.31.3　命令数据域

P2P ACCESS VERIFY 的数据字段按表 B.60 进行编码。

表 B.60　P2P ACCESS VERIFY 命令数据字段

名称	字节长度
签名数据	0x40

B.4.31.4　响应报文

成功返回状态字：0x9000。

附 录 C
（资料性）
密码安全服务接口参考

C.1 密码安全服务接口概述

本附录给出了使用 API 形式的安全服务接口参考。本附录在 GM/T 0016 的基础上进行了删减和扩展。

C.2 密码安全服务接口列表

PLC 控制系统中，密码安全服务接口列表见表 C.1。

表 C.1 密码安全服务接口列表

	API 函数	说明
密码服务接口	SKF_WaitForDevEvent	等待设备插拔事件
	SKF_CancelWaitForDevEvent	取消等待设备插拔事件
	SKF_EnumDev	枚举设备
	SKF_ConnectDev	连接设备
	SKF_DisconnectDev	断开连接
	SKF_GetDevState	获取设备状态
	SKF_GetUUID	获取设备唯一 ID
	SKF_SetLabel	设置设备标签
	SKF_GetDevInfo	获取设备信息
	SKF_Transmit	设备命令传输
	SKF_GetPINInfo	获得 PIN 码信息
	SKF_VerifyPIN	校验 PIN
	SKF_OpenApplication	打开应用
	SKF_CloseApplication	关闭应用
	SKF_EnumFiles	枚举文件
	SKF_GetFileInfo	获取文件信息
	SKF_ReadFile	读文件
	SKF_WriteFile	写文件
	SKF_EnumContainer	枚举容器
	SKF_OpenContainer	打开容器
	SKF_CloseContainer	关闭容器
	SKF_GetContainerType	获得容器类型
	SKF_ImportCertificate	导入数字证书

表 C.1　密码安全服务接口列表（续）

	API 函数	说明
密码服务接口	SKF_ExportCertificate	导出数字证书
	SKF_GenRandom	生成随机数
	SKF_GenExtRSAKey	生成外部 RSA 密钥对
	SKF_GenRSAKeyPair	生成 RSA 签名密钥对
	SKF_ImportRSAKeyPair	导入 RSA 加密密钥对
	SKF_RSASignData	RSA 签名
	SKF_RSAVerify	RSA 验签
	SKF_ExtRSAPubKeyOperation	RSA 外来公钥运算
	SKF_GenECCKeyPair	生成 ECC 签名密钥对
	SKF_ImportECCKeyPair	导入 ECC 加密密钥对
	SKF_ECCSignData	ECC 签名
	SKF_ECCVerify	ECC 验签
	SKF_ExtECCEncrypt	ECC 外来公钥加密
	SKF_ExtECCVerify	ECC 外来公钥验签
	SKF_GenerateAgreementDataWithECC	ECC 生成密钥协商参数并输出
	SKF_GenerateKeyWithECC	ECC 计算会话密钥
	SKF_GenerateAgreementDataAndKeyWithECC	ECC 产生协商数据并计算会话密钥
	SKF_GenerateKeywithIKE	计算 IKE 工作密钥
	SKF_GenerateKeywithEPK_IKE	计算 IKE 工作密钥并用外部公钥加密导出
	SKF_GenerateKeywithIPSEC	计算 IPSEC 会话密钥
	SKF_GenerateKeywithEPK_IPSEC	计算 IPSEC 会话密钥并用外部公钥加密导出
	SKF_GenerateKeywithSSL	计算 SSL 工作密钥
	SKF_GenerateKeywithEPK_SSL	计算 SSL 工作密钥并用外部公钥加密导出
	SKF_ExportPublicKey	导出公钥
	SKF_EncryptInit	加密初始化
	SKF_Encrypt	单组数据加密
	SKF_EncryptUpdate	多组数据加密
	SKF_EncryptFinal	结束加密
	SKF_DecryptInit	解密初始化
	SKF_Decrypt	单组数据解密
	SKF_DecryptUpdate	多组数据解密
	SKF_DecryptFinal	结束解密
	SKF_DigestInit	密码杂凑初始化
	SKF_Digest	单组数据密码杂凑

表 C.1　密码安全服务接口列表（续）

	API 函数	说明
密码服务接口	SKF_DigestUpdate	多组数据密码杂凑
	SKF_DigestFinal	结束密码杂凑
	SKF_MacInit	消息鉴别码运算初始化
	SKF_Mac	单组数据消息鉴别码运算
	SKF_MacUpdate	多组数据消息鉴别码运算
	SKF_MacFinal	结束消息鉴别码运算
	SKF_CloseHandle	关闭密码对象句柄
	SKF_SetAuthKey	设置认证使用的密钥
	SKF_AuthInit	安全通道初始化
	SKF_ExtAuth	安全通道外部认证

C.3　密码安全服务接口详细定义

C.3.1　SKF_WaitForDevEvent

见 GM/T 0016 中第 7 章的接口函数定义。

C.3.2　SKF_CancelWaitForDevEvent

见 GM/T 0016 中第 7 章的接口函数定义。

C.3.3　SKF_EnumDev

见 GM/T 0016 中第 7 章的接口函数定义。

C.3.4　SKF_ConnectDev

见 GM/T 0016 中第 7 章的接口函数定义。

C.3.5　SKF_DisconnectDev

见 GM/T 0016 中第 7 章的接口函数定义。

C.3.6　SKF_GetDevState

见 GM/T 0016 中第 7 章的接口函数定义。

C.3.7　SKF_GetUUID

原型　ULONG DEVAPI SKF_GetUUID(LPSTR szDevName, LPSTR *szDevUUID)

功能描述　该函数使用设备名称返回设备唯一 ID。

参数　szDevName　[IN]设备名称。

　　szDevUUID　[OUT] 输出参数,设备唯一 ID。

返回值　SAR_OK:成功。

　　其他:错误码。

本函数为阻塞函数。

C.3.8 SKF_GetName

原 型　　ULONG DEVAPI SKF _ WaitForDevEvent (LPSTR szDevUUID, LPSTR * szDevName)

功能描述　该函数使用设备的唯一 ID,返回设备名称。

参数　　szDevUUID　　[IN] 设备唯一 ID。

　　szDevName　　[OUT] 输出参数,设备名称。

返回值　　SAR_OK:成功。

　　其他:错误码。

本函数为阻塞函数。

C.3.9 SKF_SetLabel

见 GM/T 0016 中第 7 章的接口函数定义。

C.3.10 SKF_Transmit

见 GM/T 0016 中第 7 章的接口函数定义。本接口为调试验证接口,产品出厂时禁用。

C.3.11 SKF_GetPINInfo

见 GM/T 0016 中第 7 章的接口函数定义。

C.3.12 SKF_VerifyPIN

见 GM/T 0016 中第 7 章的接口函数定义。

C.3.13 SKF_OpenApplication

见 GM/T 0016 中第 7 章的接口函数定义。

C.3.14 SKF_CloseApplication

见 GM/T 0016 中第 7 章的接口函数定义。

C.3.15 SKF_CloseApplication

见 GM/T 0016 中第 7 章的接口函数定义。

C.3.16 SKF_GetFileInfo

见 GM/T 0016 中第 7 章的接口函数定义。

C.3.17 SKF_WriteFile

见 GM/T 0016 中第 7 章的接口函数定义。

C.3.18 SKF_ReadFile

见 GM/T 0016 中第 7 章的接口函数定义。

C.3.19 SKF_EnumContainer

见 GM/T 0016 中第 7 章的接口函数定义。

C.3.20 SKF_OpenContainer

见 GM/T 0016 中第 7 章的接口函数定义。

C.3.21 SKF_CloseContainer

见 GM/T 0016 中第 7 章的接口函数定义。

C.3.22 SKF_GetContainerType

见 GM/T 0016 中第 7 章的接口函数定义。

C.3.23 SKF_ImportCertificate

见 GM/T 0016 中第 7 章的接口函数定义。

C.3.24 SKF_ExportCertificate

见 GM/T 0016 中第 7 章的接口函数定义。

C.3.25 SKF_GenRandom

见 GM/T 0016 中第 7 章的接口函数定义。

C.3.26 SKF_GenRSAKeyPair

见 GM/T 0016 中第 7 章的接口函数定义。

C.3.27 SKF_ImportRSAKeyPair

见 GM/T 0016 中第 7 章的接口函数定义。

C.3.28 SKF_RSASignData

见 GM/T 0016 中第 7 章的接口函数定义。

C.3.29 SKF_RSAVerify

见 GM/T 0016 中第 7 章的接口函数定义。

C.3.30 SKF_ExtRSAPubKeyOperation

见 GM/T 0016 中第 7 章的接口函数定义。

C.3.31 SKF_GenECCKeyPair

见 GM/T 0016 中第 7 章的接口函数定义。

C.3.32 SKF_ImportECCKeyPair

见 GM/T 0016 中第 7 章的接口函数定义。

C.3.33 SKF_ECCSignData

见 GM/T 0016 中第 7 章的接口函数定义。

C.3.34 SKF_ECCVerify

见 GM/T 0016 中第 7 章的接口函数定义。

C.3.35 SKF_ExtECCEncrypt

见 GM/T 0016 中第 7 章的接口函数定义。

C.3.36 SKF_ExtECCVerify

见 GM/T 0016 中第 7 章的接口函数定义。

C.3.37 SKF_GenerateAgreementDataWithECC

见 GM/T 0016 中第 7 章的接口函数定义。

C.3.38 SKF_GenerateKeyWithECC

见 GM/T 0016 中第 7 章的接口函数定义。

C.3.39 SKF_GenerateAgreementDataAndKeyWithECC

见 GM/T 0016 中第 7 章的接口函数定义。

C.3.40 SKF_GenerateKeywithIKE

见 GM/T 0016 中第 7 章的接口函数定义。

C.3.41 SKF_GenerateKeywithEPK_IKE

见 GM/T 0016 中第 7 章的接口函数定义。

C.3.42 SKF_GenerateKeywithIPSEC

见 GM/T 0016 中第 7 章的接口函数定义。

C.3.43 SKF_GenerateKeywithEPK_IPSEC

见 GM/T 0016 中第 7 章的接口函数定义。

C.3.44 SKF_GenerateKeywithSSL

见 GM/T 0016 中第 7 章的接口函数定义。

C.3.45 SKF_GenerateKeywithEPK_SSL

见 GM/T 0016 中第 7 章的接口函数定义。

C.3.46 SKF_ExportPublicKey

见 GM/T 0016 中第 7 章的接口函数定义。

C.3.47 SKF_EncryptInit

见 GM/T 0016 中第 7 章的接口函数定义。

C.3.48 SKF_Encrypt

见 GM/T 0016 中第 7 章的接口函数定义。

C.3.49 SKF_EncryptUpdate

见 GM/T 0016 中第 7 章的接口函数定义。

C.3.50 SKF_EncryptFinal

见 GM/T 0016 中第 7 章的接口函数定义。

C.3.51 SKF_DecryptInit

见 GM/T 0016 中第 7 章的接口函数定义。

C.3.52 SKF_Decrypt

见 GM/T 0016 中第 7 章的接口函数定义。

C.3.53 SKF_DecryptUpdate

见 GM/T 0016 中第 7 章的接口函数定义。

C.3.54 SKF_DecryptFinal

见 GM/T 0016 中第 7 章的接口函数定义。

C.3.55 SKF_DigestInit

见 GM/T 0016 中第 7 章的接口函数定义。

C.3.56 SKF_Digest

见 GM/T 0016 中第 7 章的接口函数定义。

C.3.57 SKF_DigestUpdate

见 GM/T 0016 中第 7 章的接口函数定义。

C.3.58 SKF_DigestFinal

见 GM/T 0016 中第 7 章的接口函数定义。

C.3.59 SKF_MacInit

见 GM/T 0016 中第 7 章的接口函数定义。

C.3.60 SKF_Mac

见 GM/T 0016 中第 7 章的接口函数定义。

C.3.61 SKF_MacUpdate

见 GM/T 0016 中第 7 章的接口函数定义。

C.3.62 SKF_MacFinal

见 GM/T 0016 中第 7 章的接口函数定义。

C.3.63 SKF_SetAuthKey

原型　ULONG DEVAPI SKF_SetAuthKey（ULONG keyId，LPSTR key1，LPSTR key2，LPSTR key3）

功能描述　设置安全通道加密密钥，用于 Auth 命令计算认证数据。

参数

keyId	[IN] 用于认证的密钥标识。
key1	[IN]认证密钥 1，建立 SCP02 时为通道加密密钥；建立 SCP11，为用户密钥对。
key2	[IN] 认证密钥 2，建立 SCP02 时为 MAC 计算密钥；SCP11 为 NULL。
key3	[IN] 认证密钥 3，建立 SCP02 时为敏感数据加密密钥；SCP11 为 NULL。

返回值　SAR_OK：成功。
其他：错误码。

C.3.64 SKF_AuthInit

原型　ULONG DEVAPI SKF_AuthInit（ULONG scpType，ULONG keyId，ULONG mode，BYTE* certData，ULONG certDataLen）

功能描述　初始化安全通道建立，调用之前，应先调用 SKF_SetAuthKey 设置密钥。

参数

scpType	[IN] 认证类型：02：SCP02，11 SCP11。
keyId	[IN] 用于认证的密钥标识。该密钥标识由 SKF_SetAuthKey 设置。
mode	[IN] 认证模式：0 数据不计算 MAC，不加密； 1 数据应计算 MAC，不加密； 2 数据应计算 MAC，应加密。
certData	[IN] SCP11 认证时，客户端证书。SCP02 为 NULL。
certDataLen	[IN] 证书长度。

返回值　SAR_OK：成功。
其他：错误码。

C.3.65 SKF_ExtAuth

原型　ULONG DEVAPI SKF_ExtAuth（BYTE* caCertData，ULONG caCertDataLen）

功能描述　用于建立安全通道的认证，调用之前，应先执行 SKF_AuthInit。

参数

caCertData	[IN] SCP11 认证传人验证设备证书的 CA 证书，SCP02 为 NULL。
caCertDataLen	[IN] 证书数据长度。

返回值　SAR_OK：成功。
其他：错误码。

C.3.66 SKF_CloseHandle

见 GM/T 0016 中第 7 章的接口函数定义。