ICS 35.040
CCS L 80

LD

中华人民共和国劳动和劳动安全行业标准

LD/T 02.5—2022
代替 LD/T 30.5—2009

人力资源社会保障电子认证体系规范 第5部分：数字证书载体规范

Specifications for human resources and social security electronic authentication system—
Part 5: Specification for digital certificate storage carriers

2022-06-22 发布　　2022-07-01 实施

中华人民共和国人力资源和社会保障部　发布

目　　次

前言.....III
引言.....IV
1 范围.....1
2 规范性引用文件.....1
3 术语和定义.....1
4 缩略语.....2
5 数字证书载体分类.....2
6 数字证书载体硬件要求.....2
6.1 基本技术要求.....2
6.2 管理要求.....4
6.3 安全要求.....4
7 数字证书载体软件要求.....4
7.1 应用接口.....4
7.2 安装与卸载.....7
附录 A (资料性) 数字证书载体外观.....10
附录 B (资料性) 数字证书载体接口函数描述.....10

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。

LD/T 02人力资源社会保障电子认证体系系列规范，已经发布了以下五个部分：

——第1部分：框架规范

——第2部分：电子认证系统技术规范

——第3部分：数字证书格式规范

——第4部分：数字证书应用接口规范

——第5部分：数字证书载体规范

本文件为LD/T 02的第5部分。

本文件代替LD/T 30.5—2009《人力资源社会保障电子认证体系 第5部分：证书载体规范》，与LD/T 30.5—2009相比，除结构调整和编辑性改动外，主要技术变化如下：

a) 更改了本部分规范的名称为《人力资源社会保障电子认证体系 第5部分：数字证书载体规范》；

b) 增加了部分规范性引用文件（见第2章，2009版第2章）；

c) 删除了“证书载体”“证书撤销列表（CRL）”“数字证书”等术语和定义（见第3章，2009版第3章）；

d) 更改了部分缩略语，并删除了“CSP”缩略语（见第4章，2009版第4章）；

e) 增加了数字证书载体分类的描述（见第5章）；

f) 更改了证书载体硬件规范，增加了支持国产算法、支持国产操作系统相关内容描述（见第6章，2009版第5章）；

g) 更改了证书载体软件规范，删除了关于“CSP”的相关描述，并增加了接口规范描述，以及支持国产算法、支持国产操作系统相关内容描述（见第7章，2009版第6章）；

h) 更改了数字证书载体接口函数规范要求，并根据在正文中被提及的先后顺序调整附录编号（见附录B，2009版附录A）；

i) 更改了数字证书载体外观要求，根据证书类型，结合业务需求，对“智能密码钥匙”形式的数字证书载体类型、载体序列号及外观颜色做出新的要求，并根据在正文中被提及的先后顺序调整附录编号（见附录A，2009版附录B）。

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。

本文件由中华人民共和国人力资源社会保障部信息中心提出并归口。

本文件起草单位：中华人民共和国人力资源社会保障部信息中心、普华诚信信息技术有限公司、北京数字认证股份有限公司。

本文件主要起草人马丹蕾、张嵩、王岩、耿建军、唐淑静、韩晓颖、成勇、王祥宇、李娜、王智飞、郭丽芳、高五星、李述胜。

本文件所代替的历次版本发布情况为：

----LD/T 30.5—2009《人力资源社会保障电子认证体系 第5部分：证书载体规范》；

----本次为第一次修订。

引　言

为适应人力资源社会保障信息化发展要求，满足人力资源社会保障网络信任体系建设和管理的需要，人力资源社会保障部组织并制定了人力资源社会保障电子认证体系系列规范。随着我国商用密码技术的发展、国产密码算法的标准发布，以及人力资源社会保障行业的业务发展，需要对行业标准 LD/T 30—2009《人力资源社会保障电子认证体系规范》进行修改和完善。

本次修订，是在充分借鉴原标准的框架和结构的基础上，根据人力资源社会保障行业特点和电子认证业务发展需求，对电子认证体系总体结构和电子认证系统整体建设规划进行扩充完善，以符合国家及国家密码主管部门相关标准规范要求，满足人力资源社会保障业务和管理需求，推进 SM2 算法在人社信息系统中的应用，另一方面，也可有效配合《中华人民共和国密码法》、《中华人民共和国网络安全法》、密码管理及密码应用安全测评工作、等级保护工作的落实与实施。

LD/T 02描述了人力资源社会保障电子认证体系总体结构和电子认证系统整体建设规划，规定了各级人力资源社会保障部门电子认证系统建设和应用要求，由以下五个部分构成。

----第1部分：框架规范

----第2部分：电子认证系统技术规范

----第3部分：数字证书格式规范

----第4部分：数字证书应用接口规范

----第5部分：数字证书载体规范

LD/T 02的第1部分，是人力资源社会保障电子认证体系系列规范的总纲，规定了电子认证体系规范的总体框架。LD/T 02的第2部分~第5部分分别从电子认证系统技术、数字证书格式、数字证书应用接口、数字证书载体四个方面提出具体规范要求。

本部分重点引用了GB/T 35291-2017，并在此基础上，扩展了数字证书载体基本技术要求、数字证书载体管理要求、软件的安装卸载以及数字证书载体外观设计要求等相关内容，从满足人力资源社会保障业务需求的角度，对本行业发放的数字证书载体的软硬件和外观提出规范和要求。

人力资源社会保障电子认证体系规范
第5部分：数字证书载体规范

1 范围

本文件给出了数字证书载体分类，规定了数字证书载体硬件和软件要求。

注：本标准主要规范智能密码钥匙的技术指标，第三代社会保障卡应符合LD/T 32。

本文件适用于人力资源社会保障数字证书载体（智能密码钥匙）的设计、应用开发、使用和检测。

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 25056-2018 信息安全技术 证书认证系统密码及其相关安全技术规范

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 35276-2017 信息安全技术 SM2密码算法使用规范

GB/T 35291-2017 信息安全技术 智能密码钥匙应用接口规范

GB/T 36322-2018 信息安全技术 密码设备应用接口规范

GB/T 37092-2018 信息安全技术 密码模块安全要求

LD/T 32 社会保障卡规范

ISO7816-4 接触式卡智能卡与外界交互的介面

3 术语和定义

GB/T 35291、GB/T 36322界定的和下列术语和定义适用于本文件。

3.1

加密 encrypt

对数据进行密码变换以产生密文的过程。

[来源：GB/T 36322-2018，3.5]

3.2

解密 decrypt

加密过程对应的逆过程。

[来源：GB/T 36322-2018，3.3]

3.3

设备认证 device authentication

智能密码钥匙对应用程序的认证。

[来源：GB/T 35291-2017，3.2]

3.4

设备认证密钥 device authentication key

用于设备认证的密钥。

[来源：GB/T 35291-2017，3.3]

3.5

会话密钥 session key

处于层次化密钥结构中的最底层，尽在一次会话中使用的密钥。

[来源：GB/T 36322-2018，3.10]

3.6

用户密钥 user key

存储在设备内部的用于应用密码运算的非对称密钥，包含签名密钥对和加密密钥对。

[来源：GB/T 36322-2018，3.11]

4 缩略语

下列缩略语适用于本文件：

API：应用程序接口，简称应用接口（Application Program Interface）

CA：证书认证机构（Certification Authority）

CRL：证书撤销列表（Certificate Revocation List）

PKCS：公钥密码标准（the Public-Key Cryptography Standard)

PIN：个人身份识别码(Personal Identification Number)

Admin PIN：管理员PIN

User PIN：用户PIN

5 数字证书载体分类

数字证书载体的安全性须符合 GB/T 37092-2018 中安全等级第二级及以上相关要求。目前，人力资源社会保障电子认证系统所采用的数字证书载体分为智能密码钥匙和第三代社会保障卡两类。

6 数字证书载体硬件要求

6.1 基本技术要求

数字证书载体的基本技术要求见表 1。

表 1 数字证书载体基本技术要求

名称	要求	备注
外形及尺寸	符合人力资源社会保障部规定的数字证	

	书载体外观要求，见附录 A	
存储容量	>=128K Bytes（32K 型）	
CPU 芯片位数	>=32 位	
功耗	<400 毫瓦	
国际标准	符合 ISO7816-4	
国家标准	符合 GB/T 35291-2017	
编程接口	符合 GB/T 35291-2017 规定的要求	
证书格式	符合 GB/T 25056-2018 规定的要求	
硬件接口	符合 USB 接口规范，不需额外插电。	USB2.0 以上的规格
读写次数(次)	>=10 万	
存储有效期(年)	>=10	
存放温度	-40～70℃	
工作温度	0～60℃	
湿度要求	10%~90%	
工作电压	4.5V~5.5V	
适用浏览器	国产浏览器以及 IE、火狐、Chrome 等主流浏览器及相应版本	
适用操作系统	国产 Linux 桌面操作系统、WindowsXP、Windows7、Windows8、Windows10 等主流操作系统	支持简体中文、繁体中文、英文
非对称算法	符合 GB/T 35276-2017，密钥对应在芯片中生成，且私钥不能导出。	如 SM2 算法
对称算法	符合 GB/T 32907	如 SM4 算法
杂凑算法	符合 GB/T 32905	如 SM3 算法
公钥私钥对生成时间	<=30 秒	
数字签名和验证时间	<1 秒/次	密钥的签名验证速度
SM2 公私钥对生成速度	<50 ms/次	
SM2 签名速度	<10 ms/次	
SM2 验签速度	<20 ms/次	
SM1 加解密速度	>1.5Mbps	
SM4 加解密速度	>1Mbps	
存储要求	a) 公私钥对：>=2 个， b) 数字证书：>=2 张。	证书载体容器至少可同时存储 2 张数字证书和 2 个密钥对，以支持双证书。
安全性要求	a) 管理员登录认证后，方可解锁用户 PIN； b) 用户登录认证后，方可产生密钥对、导入密钥和证书，使用密钥和证书； c) 用户口令连续 10 次输错后应自动锁死。	
硬件真随机数发生器	支持	

6.2 管理要求

6.2.1 数字证书载体初始化

数字证书载体的初始化就是对数字证书载体进行区间划分，使其按照相关规范进行初始化。

初始化工具由各数字证书载体供应商提供。

初始化工具应有两种形式，一类是可执行的初始化工具，另一类是动态库dll接口文件，并可根据动态库文件开发通用的初始化工具。

6.2.2 口令管理

数字证书载体的客户端管理工具应具备校验口令和修改口令的功能。

口令长度应为6-16位，应是字母、数字和特殊字符组成的混合体，口令不得采用有特殊意义的（如姓名、生日、电话号码等）数字和词组。

6.2.3 锁死与解锁

数字证书载体连续 10 次输错口令应自动锁死。密码锁死后，即使输入正确的口令也不能使用证书，必须由管理员口令解锁后才能继续使用。管理员口令需随机生成，解锁操作应在安全可控的前提下执行。

解锁口令的长度应为6-16位。

6.2.4 其他

同一个终端可以同时使用多个数字证书载体，以标准接口调用数字证书载体设备时，系统会自动弹出设备选择框（列出设备的卷标名称），由用户选择设备。

6.3 安全要求

6.3.1 基本安全要求

数字证书载体的安全机制要求保障系统运行稳定可靠，数据访问安全可控，数据传输安全保密，可抵御外部攻击。

数字证书载体必须能产生SM2等非对称密钥对，私钥不能被读取，使用前应经过访问权限认证。

6.3.2 密钥和密码的存放

数字证书载体应该能保证非对称密钥和对称密钥的安全性。对称密钥在导出时必须加密保护，非对称密钥的私钥不允许导出，非对称密钥的公钥支持查看、导出。

7 数字证书载体软件要求

7.1 应用接口

数字证书载体软件应用接口应符合GB/T 35291规定的要求。

7.1.1 设备管理

设备管理主要完成设备的插拔响应、枚举、连接、断开、获取设备状态、设置设备信息、获取设备信息、锁定设备、解锁设备和设备命令传输等操作。使用者不必知道设备类型和具体的驱动模式，只需要通过本文件提供的接口，即可完成设备管理功能。

设备管理函数如表 2 所示。

表 2　设备管理函数

函数名称	功能
SKF_WaitForDevEvent	等待设备插拔事件
SKF_CancelWaiForDevEvent	取消等待设备插拔事件
SKF_EnumDev	枚举设备
SKF_ConnectDev	连接设备
SKF_DisconnectDev	断开设备
SKF_GetDevState	获取设备状态
SKF_SetLabel	设置设备标签
SKF_GetDevInfo	获取设备信息
SKF_LockDev	锁定设备
SKF_UnlockDev	解锁设备
SKF_Transmit	设备命令传输

7.1.2　访问控制

访问控制分为设备级访问控制和应用级访问控制。

- 设备级访问控制：包括内部认证和外部认证，用来进行设备之间的相互认证；
- 应用级访问控制：分为管理员和用户权限二级权限控制。管理员负责为用户提供 PIN 的初始化和解锁等服务。用户拥有设备使用权，使用设备提供的功能，存储自己的私有数据。

对于每一个设备而言，可以同时存在一个或多个应用，每个应用之间的访问控制相互独立。访问控制函数如表 3 所示。

表 3　访问控制函数

函数名称	功能
SKF_ChangeDevAuthKey	修改设备认证密钥
SKF_DevAuth	设备认证
SKF_ChangePIN	修改 PIN
SKF_GetPINInfo	获得 PIN 码信息
SKF_VerifyPIN	校验 PIN
SKF_UnblockPIN	解锁 PIN
SKF_ClearSecureState	清除应用安全状态

7.1.3　应用管理

一个设备可以建立一个或多个应用，每个应用之间的权限管理和密码服务彼此独立。

在每一个应用中都有相对应的文件体系和加密服务体系。应用管理主要完成应用的创建、枚举、删除、打开、关闭操作。应用管理函数如表4所示。

表 4　应用管理函数

函数名称	功能
SKF_CreateApplication	创建应用
SKF_EnumApplication	枚举应用
SKF_DeleteApplication	删除应用

SKF_OpenApplication	打开应用
SKF_CloseApplication	关闭应用

7.1.4 文件管理

文件管理函数用于满足用户扩展开发的需要，包括对文件的创建、删除、枚举、获取设备信息、读写操作，如表 5 所示。

表 5 文件管理函数

函数名称	功能
SKF_CreateFile	创建文件
SKF_DeleteFile	删除文件
SKF_EnumFiles	枚举文件
SKF_GetFileInfo	获取文件信息
SKF_ReadFile	读文件
SKF_WriteFile	写文件

7.1.5 容器管理

容器管理函数用于满足各种不同应用的管理，包括对容器的创建、删除、枚举、打开和关闭等操作，如表6所示。

表 6 容器管理系列函数

函数名称	功能
SKF_CreateContainer	创建容器
SKF_DeleteContainer	删除容器
SKF_EnumContainer	枚举容器
SKF_OpenContainer	打开容器
SKF_CloseContainer	关闭容器
SKF_GetContainerType	获取容器类型
SKF_ImportCertificate	导入数字证书
SKF_ExportCertificate	导出数字证书

7.1.6 密码服务

密码服务函数用于密码运算服务，包括密钥的生成、导入、导出、加密解密、签名验证等，如表 7 所示。会话密钥支持国产算法如 SM1、SM4，非对称密钥目前支持 256 位 SM2、2048 位 RSA。

表 7 密码服务函数

函数名称	功能
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_RSAExportSessionKey	RSA 生成并导出会话密钥
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_ECCExportSessionKey	ECC 生成并导出会话密钥
SKF_ExtECCEncrypt	ECC 外来公钥加密
SKF_ExtECCVerify	ECC 外来公钥验签
SKF_GenerateAgrementDataWithECC	ECC 生成密钥协商参数并输出
SKF_GenerateKeyWithECC	ECC 计算会话密钥
SKF_GenerateAgrementDataAndKeyWithECC	ECC 产生协商数据并计算会话密钥
SKF_ImportSessionKey	导入会话密钥
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
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

数字证书载体的接口函数的描述见附录B。

7.2 安装与卸载

7.2.1 安装程序要求

a) 基本要求：
   1) 同一型号的智能密码钥匙安装程序应将驱动程序、数字证书载体开发接口、管理工具封装在一起，共用一个安装程序；
   2) 安装程序要能够自动识别用户的国产 Linux 桌面操作系统或 Windows 操作系统版本并自动安装相应的兼容性插件；

3) 智能密码钥匙在安装过程中应给予用户足够的提示信息，但要尽量减少与用户的交互，简化安装过程；
4) 智能密码钥匙安装程序应兼容国产 Linux 桌面操作系统或 Windows XP Windows 7、Windows8、Windows 10 等操作系统，并包含必要的系统补丁；
5) 安装程序自动检测客户机操作系统语言环境，如简体中文、繁体中文或英文，将对应的语言环境自动安装到客户机；
6) 安装程序提示文字要简洁易懂、便于理解。菜单设计应清晰合理、方便查找。安装界面中提示信息的字体采用中文宋体 9 号字，英文采用 Arial 12 号字体。

b) 可选要求：
1) 安装成功后在开始菜单中提供卸载子菜单；
2) 安装成功后在控制面板中提供卸载接口；
3) 安装成功后在系统应用商店或系统自带的软件中心提供卸载接口；
4) 管理工具应具备校验口令的功能。

7.2.2 卸载程序要求

a) 基本要求：
1) 卸载程序应兼容国产 Linux 桌面操作系统或 WindowsXP、Windows 7、Windows8、Windows 10 等系列操作系统；
2) 卸载程序不需要客户干预，能自动完成卸载，包括：
i. 执行完卸载过程后，要求能正确清除掉开始菜单中的相关子菜单、控制面板、系统应用商店或系统自带的软件中心中的相应卸载接口，
ii. 执行完卸载过程后，要求能正确清除掉系统中相应的安装目录，
iii. 卸载驱动程序时不要删除系统本身自带的库和注册表信息中原有的键值；
3) 卸载程序不自动执行重新启动计算机，卸载完成后可提示客户重新启动计算机。

b) 可选要求：
1) 卸载完成后不能在文件系统或注册表中留下残余组件；
2) 通过开始菜单中提供卸载子菜单，能将驱动程序卸载；
3) 通过控制面板中的“添加/删除程序”，或系统应用商店、系统自带的软件中心中相应软件程序的“卸载”，能将驱动程序卸载；
4) 再次运行安装程序能自动卸载。

7.2.3 驱动程序的兼容性要求

驱动程序应满足下列要求。

a) 驱动程序的编写应符合 PC/SC 标准。
b) 驱动程序的编写应符合微软的驱动编写标准。
c) 安装后添加的文件路径和注册表信息中应带有产品的特有信息或标记。
d) 驱动程序不能拷贝系统本身自带的库到系统的目录下，如果使用了操作系统的动态库，建议尽量使用静态绑定。
e) 不同的智能密码钥匙必须相互兼容，安装在同一台机器上都能正常工作，不会相互影响。

7.2.4 接口动态库命名

接口动态库命名规则如下：

a) 数字证书载体动态库的文件名命名规则为“hrssxxxxyy. dll”，其中 xxxx 为厂商代码，yy 为产品代码；

b） 动态库导出的接口函数应为C语言函数；

c） 厂商代码和产品代码由人力资源社会保障部统一管理。

附 录 A
（资料性）
数字证书载体外观

人力资源社会保障数字证书载体（智能密码钥匙）的外形及尺寸如图 A.1 所示。

图 A.1 数字证书载体的外形及尺寸示意图

如图 A.1 所示，智能密码钥匙正面应标识证书类型，智能密码钥匙背面应标识序列号。

以智能密码钥匙作为数字证书载体，有以下几种类型：面向人力资源社会保障部门的政府工作人员发放的人员证书，以及面向人力资源社会保障部门机构和社会企事业单位的发放的机构证书。其外形名称和颜色的分类如下：

a) 人员：名称为“人员证书”，颜色为金属银白色；
b) 机构：名称为“机构证书”， 颜色为金属银白色。

数字证书载体外形上的序列号由 16 位字母或数字组成，按以下规则编码：

a) 第 1 位为生产厂商编号；
b) 第 2 位为产品型号，可为字母或数字，与生产厂商定义的产品型号相对应；
c) 第 3 位为证书类型，分别与机构证书、人员证书等相对应；
d) 第 4 位至第 7 位为市级行政区划（4 位）；
e) 第 8 位至第 9 位为生产年份（2 位）；
f) 第 10 位至第 11 位为产品生产批次（2 位）；
g) 第 12 位至第 16 位为产品流水号（5 位）。

附 录 B
（资料性）
数字证书载体接口函数描述

B.1 数据类型定义

数据类型定义如表B.1所示。

表 B.1 数据类型

类型名称	描述	定义
BOOL	布尔类型，取值为 TRUE 或 FALSE	
BYTE	字节类型，无符号 8 位整数	typedef UINT8 BYTE
CHAR	字符类型，无符号 8 位整数	typedef UINT8 CHAR
SHORT	短整数，有符号 16 位	typedef INT16 SHORT
USHORT	无符号 16 位整数	typedef UINT16 USHORT
LONG	长整数，有符号 32 位整数	typedef INT32 LONG
ULONG	长整数，无符号 32 位整数	typedef UINT32 ULONG
UINT	无符号 32 位整数	typedef UINT32 UINT
WORD	字类型，无符号 16 位整数	typedef UINT16 WORD
DWORD	双字类型，无符号 32 位整数	typedef UINT32 DWORD
FLAGS	标志类型，无符号 32 位整数	typedef UINT32 FLAGS
LPSTR	8 位字符串指针，按照 UTF8 格式存储及交换	typedef CHAR * LPSTR
HANDLE	句柄，指向任意数据对象的起始地址	typedef void * HANDLE
DEVHANDLE	设备句柄	typedef HANDLE DEVHANDLE
HAPPLICATION	应用句柄	Typedef HANDLE HAPPLICATION
HCONTAINER	容器句柄	typedef HANDLE HCONTAINER

a） 常量定义

数据常标识的定义如表B.2所示。

表 B.2 常量定义

类型名称	描述	定义
TRUE	0x00000001	布尔值为真
FALSE	0x00000000	布尔值为假
DEVAPI	_stdcall	_stdcall 函数调用方式
ADMIN_TYPE	0	管理员 PIN 类型
USER_TYPE	1	用户 PIN 类型

b） 版本数据

1） 类型定义

```
typedef struct Struct_Version{
    BYTE major;
    BYTE minor;
    }VERSION;
```

2） 数据项描述

版本数据类型数据项描述如表 B.3 所示。

表 B.3 版本数据类型数据项描述

数据项	类型	意义	备 注
major	BYTE	主版本号	主版本号和次版本号以“.”分隔，例如 Version 1.0，主版本号为 1，次版本号为 0；Version 2.10，主版本号为 2，次版本号为 10。
minor	BYTE	次版本号	

c） 设备信息

1） 类型定义

```
typedef struct Struct_DEVINFO{
    VERSION Version
    CHAR Manufacturer[64];
    CHAR Issuer[64];
    CHAR Label[64];
    CHAR SerialNumber[32];
    VERSION HWVersion;
    VERSION FirmwareVersion;
    ULONG    AlgSymCap;
    ULONG    AlgAsymCap ;
    ULONG    AlgHashCap ;
    ULONG    DevAuthAlgId ;
    ULONG    TotalSpace ;
    ULONG    FreeSpace ;
    ULONG    MaxECCBufferSize ;
    ULONG    MaxBufferSize ;
    BYTE Reserved[64] ;
}DEVINFO_SKF,*PDEVINFO_SKF;
```

2） 数据项描述

设备信息数据类型数据项描述如表 B.4 所示。

表 B.4 设备信息数据类型数据项描述

数据项	类型	意义	备注
Version	VERSION	版本号	数据结构版本号，本结构的版本号为 1.0
Manufacturer	CHAR 数组	设备厂商信息	以‘\0’为结束符的 ASCII 字符串
Issuer	CHAR 数组	发行厂商信息	以‘\0’为结束符的 ASCII 字符串
Label	CHAR 数组	设备标签	以‘\0’为结束符的 ASCII 字符串
SerialNumber	CHAR 数组	序列号	以‘\0’为结束符的 ASCII 字符串
HWVersion	VERSION	设备硬件版本	
FirmwareVersion	VERSION	设备本身固件版本	
AlgSymCap	ULONG	分组密码算法标识	
AlgAsymCap	ULONG	非对称密码算法标识	
AlgHashCap	ULONG	密码杂凑算法标识	
DevAuthAlgId	ULONG	设备认证使用的分组密码算法标识	
TotalSpace	ULONG	设备总空间大小	
FreeSpace	ULONG	用户可用空间的大小	
MaxECCBufferSize	ULONG	能够处理的 ECC 加密数据大小	
MaxBufferSize	ULONG	能够处理的分组运算和杂凑运算的数据大小	
Reserved	ULONG	保留扩展	

d） RSA公钥数据结构

1） 类型定义

```
typedef struct Struct_RSAPUBLICKEYBLOB{
    ULONG  AlgID;
    ULONG  BitLen;
    BYTE    Modulus[MAX_RSA_MODULUS_LEN];
    BYTE    PublicExponent[MAX_RSA_EXPONENT_LEN];
}RSAPUBLICKEYBLOB, *PRSAPUBLICKEYBLOB;
```

MAX_RSA_MODULUS_LEN 为算法模数的最大长度；

MAX_RSA_EXPONENT_LEN 为算法指数的最大长度。

2） 数据项描述

RSA 公钥数据结构数据类型数据项描述如表 B.5 所示。

表 B.5　RSA 公钥数据结构数据类型数据项描述

数据项	类型	意义	备注
AlgID	ULONG	算法标识号	
BitLen	ULONG	模数的实际位长度	必须是 8 的倍数
Modulus	BYTE 数组	模数 n = p * q	实际长度为 BitLen/8 字节 #define MAX_RSA_MODULUS_LEN 256 #define MAX_RSA_EXPONENT_LEN 4
PublicExponent	BYTE 数组	公开密钥 e	一般为 00010001

e） RSA私钥数据结构

1） 类型定义

```
typedef struct Struct_RSAPRIVATEKEYBLOB {
    ULONG  AlgID;
    ULONG  BitLen;
    BYTE    Modulus[MAX_RSA_MODULUS_LEN];
    BYTE    PublicExponent[MAX_RSA_MODULUS_LEN];
    BYTE    PrivateExponent[MAX_RSA_EXPONENT _LEN];
    BYTE    Prime1[MAX_RSA_MODULUS_LEN/2];
    BYTE    Prime2[MAX_RSA_MODULUS_LEN/2];
    BYTE    Prime1Exponent[MAX_RSA_MODULUS_LEN/2];
    BYTE    Prime2Exponent[MAX_RSA_MODULUS_LEN/2];
    BYTE    Coefficient[MAX_RSA_MODULUS_LEN/2];
}RSAPRIVATEKEYBLOB, *PRSAPRIVATEKEYBLOB;
```

MAX_RSA_MODULUS_LEN 为 RSA 算法模数的最大长度；

MAX_RSA_EXPONENT_LEN 为算法指数的最大长度。

2） 数据项描述

RSA 私钥数据结构数据类型数据项描述如表 B.6 所示。

表 B.6　RSA 私钥数据结构数据类型数据项描述

数据项	类型	意义	备注
AlgID	ULONG	算法标识号	
BitLen	ULONG	模数的实际位长度	必须是 8 的倍数
Modulus	BYTE 数组	模数 n = p * q	实际长度为 BitLen/8 字节
PublicExponent	ULONG	公开密钥 e	一般为 00010001
PrivateExponent	BYTE 数组	私有密钥 d	实际长度为 BitLen/8 字节
Prime1	BYTE 数组	素数 p	实际长度为 BitLen/16 字节
Prime2	BYTE 数组	素数 q	实际长度为 BitLen/16 字节
Prime1Exponent	BYTE 数组	d mod (p-1)的值	实际长度为 BitLen/16 字节
Prime2Exponent	BYTE 数组	d mod (q -1)的值	实际长度为 BitLen/16 字节

Coefficient	BYTE 数组	q 模 p 的乘法逆元	实际长度为 BitLen/16 字节

f） ECC公钥数据结构

1） 类型定义

```
typedef struct Struct_ECCPUBLICKEYBLOB{
    ULONG  BitLen;
    BYTE    XCoordinate[ECC_MAX _XCOORDINATE_BITS_LEN/8];
    BYTE    YCoordinate[ECC_MAX _YCOORDINATE_BITS_LEN/8];
}ECCPUBLICKEYBLOB,*PECCPUBLICKEYBLOB ;
```

ECC_MAX_XCOORDINATE_LEN 为 ECC 算法 X 坐标的最大长度

ECC_MAX_YCOORDINATE_LEN 为 ECC 算法 Y 坐标的最大长度。

2） 数据项描述

ECC 公钥数据结构数据类型数据项描述如 B.7 所示。

表 B.7 ECC 公钥数据结构数据类型数据项描述

数据项	类型	意义	备注
BitLen	ULONG	模数的实际位长度	必须是 8 的倍数
XCoordinate	BYTE 数组	曲线上点的 X 坐标	有限域上的整数 #define ECC_MAX_XCOORDINATE_BITS_ LEN 512
YCoordinate	BYTE 数组	曲线上点的 Y 坐标	有限域上的整数 #define ECC_MAX_YCOORDINATE_BITS_ LEN 512

g） ECC私钥数据结构

1） 类型定义

```
typedef struct Struct_ECCPRIVATEKEYBLOB{
    ULONG  BitLen;
    BYTE    PrivateKey[ECC_MAX _MODULUS_BITS_LEN/8];
}ECCPRIVATEKEYBLOB,*PECCPRIVATEKEYBLOB ;
```

ECC_MAX _MODULUS_BITS_LEN 为 ECC 算法模数的最大长度

2） 数据项描述

ECC 私钥数据结构数据类型数据项描述如表 B.8 所示。

表 B.8 ECC 私钥数据结构数据类型数据项描述

数据项	类型	意义	备注
BitLen	ULONG	模数的实际位长度	必须是 8 的倍数

PrivateKey	BYTE 数组	私有密钥	有限域上的整数 #define ECC_MAX_MODULUS_BITS_LEN 512

h） ECC密文数据结构

1） 类型定义

```
typedef struet Struet_ECCCIPHERBLOB{
    BYTE    Xcoordinate[ECC_MAX_XCOORDINATE_BITS_LEN/8];
    BYTE    YCoordinate[ECC_MAX_XCOORDINATE_BITS_LEN/8];
    BYTE    HASH[32] ;
    ULONG    CipherLen ;
    BYTE    Cipher[1] ;
}ECCCIPHERBLOB,*PECCCIPHERBLOB
```

2） 数据项描述

ECC 密文数据结构数据类型数据项描述如表 B.9 所示。

表 B.9 ECC 密文数据结构数据类型数据项描述

数据项	类型	意义	备注
XCoordinate	BYTE 数组	与 y 组成椭圆曲线上的点（x,y）	
YCoordinate	BYTE 数组	与 x 组成椭圆曲线上的点（x,y）	
HASH	BYTE 数组	明文的杂凑值	
CipherLen	ULONG	密文数据长度	
Cipher	BYTE 数组	密文数据	实际长度为 CipherLen

i） ECC签名数据结构

1） 类型定义

```
typedef struet Struet_ECCSIGNATUREBLOB{
    BYTE r[ECC_MAX_XCOORDINATE_BITS_LEN/8];
    BYTE s[ECC_MAX_XCOORDINATE_BITS_LEN/8];
}ECCSIGNATUREBLOB,*PECCSIGNATUREBLOB
```

ECC_MAX_MODULUS_BITS_LEN为ECC算法X坐标的最大比特长度。

2） 数据项描述

ECC 签名数据结构数据类型数据项描述如表 B.10 所示。

表 B.10 ECC 签名数据结构数据类型数据项描述

数据项	类型	意义	备注
r	BYTE 数组	签名结果的 r 部分	
s	BYTE 数组	签名结果的 s 部分	

j） 分组密码参数

1） 类型定义

```
typedef struet Struet_BLOCKCIPHERPARAM{
    BYTE   IV[MAX_IV_LEN] ;
    ULONG   IVLen;
    ULONG   PaddingType ;
    ULONG   FeedBitLen;
}BLOCKCIPHERPARAM,*PBLOCKCIPHERPARAM;
```

2） 数据项描述

ECC 签名数据结构数据类型数据项描述如表 B.11 所示。

表 B.11 ECC 签名数据结构数据类型数据项描述

数据项	类型	意义	备注
IV	BYTE 数组	初始向量，MAX_IV_LEN 为初始化向量的最大长度 #define MAX_IV_LEN 32	
IVLen	ULONG	初始向量的实际长度（按字节计算）	
PaddingType	ULONG	填充方式，0 表示不填充，1 表示按照 PKCS#5 方式进行填充	
FeedBitLen	ULONG	反馈值的位长度（按位计算）	只针对 OFB，CFB 模式

k） ECC加密密钥对保护结构

1） 类型定义

```
typedef struet Struet_ ENVELOPEDKEYBLOB{
    ULONG   Version;
    ULONG   ulSymmAlgID ;
    ULONG   ulBits;
    BYTE   cbEncryptedPriKey[64];
    ECCPUBLICKEYBLOB PubKey;
    ECCCIPHERBLOB ECCCipherBlob;
}ENVELOPEDKEYBLOB,*PENVELOPEDKEYBLOB;
```

2） 数据项描述

ECC 加密密钥对保护结构数据类型数据项描述如表 B.12 所示。

表 B.12 ECC 加密密钥对保护结构数据类型数据项描述

数据项	类型	意义	备注
Version	ULONG	版本号，本版本为 1	
ulSymmAlgI	ULONG	对称算法标识	应为ECB 模式

ulBits	ULONG	加密密钥对的密钥位长	
cbEncryptedPriKey	BYTE 数组	对称算法加密的加密私钥，加密私钥的原文为ECCPRIVATEKEYBLOB 结构中的 PrivateKey	其有效长度为原文的（ulBits+7）/8
PubKey	ECCPUBLICKEYBLOB	加密密钥对的公钥	
ECCCipherBlob	ECCCIPHERBLOB	用保护公钥加密过的对称密钥密文	

l） 文件属性

1) 类型定义

```
typedef struct Struct_FILEATTRIBUTE{
    CHAR    FileName[32];
    ULONG  FileSize;
    ULONG  ReadRights;
    ULONG  WriteRights;
} FILEATTRIBUTE, *PFILEATTRIBUTE;
```

2) 数据项描述

文件属性数据类型数据项描述如表 B.13 所示。

表 B.13 文件属性数据类型数据项描述

数据项	类型	意义	备注
FileName	CHAR 数组	文件名	ASCIIZ 字符串，最大长度为 32
FileSize	ULONG	文件大小	创建文件时定义的文件大小
ReadRights	ULONG	读取权限	读取文件需要的权限，见表 A-14。
WriteRights	ULONG	写入权限	写入文件需要的权限

m） 设备权限类型

设备权限类型如表B. 14所示。

表 B.14 设备权限类型

权限类型	值	说明
SECURE_NEVER_ACCOUNT	0x00000000	禁用
SECURE_ADM_ACCOUNT	0x00000001	管理员权限
SECURE_USER_ACCOUNT	0x00000010	用户权限
SECURE_ANYONE_ACCOUNT	0x000000FF	任何人

n） 设备状态

表B.15给出了设备状态的常量标识、值以及相应说明。

表 B.15　设备状态

常量标识	值	说明
DEV_ABSENT_STATE	0x00000000	设备不存在
DEV_PRESENT_STATE	0x00000001	设备存在
DEV_UNKNOW_STATE	0x00000002	设备状态未知

B.2 数字证书载体函数定义

a） 设备管理类函数

表 B.16 给出设备管理类具体函数。

表 B. 16　设备管理类函数

序号	函数名称	函数定义
1	等待设备插拔事件函数	SKF_WaitForDevEvent
2	取消等待设备插拔事件函数	SKF_CancelWaiForDevEvent
3	枚举设备函数	SKF_EnumDev
4	连接设备函数	SKF_ConnectDev
5	断开设备函数	SKF_DisconnectDev
6	获取设备状态函数	SKF_GetDevState
7	设置设备标签函数	SKF_SetLabel
8	获取设备信息函数	SKF_GetDevInfo
9	锁定设备函数	SKF_LockDev
10	解锁设备函数	SKF_UnlockDev
11	设备命令传输函数	SKF_Transmit

1）等待设备插拔事件函数

函数原型　ULONG DEVAPI SKF_WaitForDevEvent(LPSTR szDevName,
ULONG *pulDevNameLen, ULONG *pulEvent)

功能描述　等待设备插拔事件：该函数等待设备插入或者拔除事件。szDevName 返回发生事件的设备名称。

参数

	szDevName	[OUT] 发生事件的设备名称
	pulDevNameLen	[IN/OUT] 输入/输出参数，当输入时表示缓冲区长度，输出时表示设备名称的有效长度,长度包含字符串结束符。
	pulEvent	[OUT]事件类型。1 表示插入，2 表示拔出。

返回值

SAR_OK	表示成功
其他	返回错误码，见表 A-22

2）取消等待设备插拔事件

函数原型　ULONG DEVAPI SKF_CancelWaitForDevEvent()

功能描述　取消等待设备插拔事件：该函数取消等待设备插入或者拔除事件。

参数

返回值

SAR_OK	表示成功
其他	返回错误码，见表 A-22

备注　使本进程正在执行的 SKF_WaitForDevEvent 函数立即返回。

3）枚举设备函数

函数原型　ULONG DEVAPI SKF_EnumDev(BOOL bPresent, LPSTR szNameList,
ULONG *pulSize)

功能描述　枚举设备：获得当前系统中的设备列表

参数

	bPresent	[IN] 为 TRUE 表示取当前设备状态为存在的设备列表。为 FALSE 表示取当前驱动支持的设备列表。

	szNameList	[OUT] 设备名称列表。该参数为 NULL，将由 pulSize 返回所需要的内存空间大小。每个设备的名称以单个‘\0’结束，以双‘\0’表示列表的结束。
	pulSize	[IN, OUT] 输入参数，输入设备名称列表的缓冲区长度，输出参数，返回 szNameList 所需要的的空间大小。
返回值	SAR_OK	表示成功
	其他	返回错误码，见表 A-22
备注	使用设备应该在访问设备前首先调用该函数。	

4）连接设备函数

函数原型　ULONG DEVAPI SKF_ConnectDev (LPSTR szName, DEVHANDLE *phDev)

功能描述　连接设备：

通过设备名称连接设备，返回设备的句柄。

连接设备函数，是对设备进行其他操作前的第一步，连接设备成功后，可以对设备发送其他指令，例如创建文件，读写文件等。

为了节省和设备之间的通讯速度，一般在连接完一次设备后，尽可能多的进行该次连接的其他操作，然后再断开设备。

该函数以共享的方式连接设备，一个应用连接后其他应用还可以连接设备，如果一个应用想对设备进行独占操作，在连接设备后，调用锁定设备函数 SKF_LockDev。

参数	szName	[IN] 设备名称
	phDev	[OUT] 设备操作句柄
返回值	SAR_OK	表示成功
	其他	返回错误码，见表 A-22
备注	使用算法服务或者文件管理函数之前调用。	

5）断开连接函数

函数原型　ULONG DEVAPI SKF_DisConnectDev (DEVHANDLE hDev)

功能描述　断开设备：

断开一个已经连接的设备，并释放句柄。

调用断开设备函数成功后，该设备此次连接的句柄失效，如果再对设备进行操作，需要重新连接设备。

断开连接操作并不影响设备的权限状态，也不会释放应用对设备的同步状态。

参数	hDev	[IN] 连接设备时返回的设备句柄
返回值	SAR_OK	表示成功
	其他	返回错误码，见表 A-22

6）获取设备状态函数

函数原型　ULONG DEVAPI SKF_GetDevState(LPSTR szName，ULONG*pulDevState)

功能描述　获取设备状态：获取设备是否存在的状态

参数	szDevName	[IN] 设备名称
	pulDevState	[OUT] 返回设备状态
返回值	SAR_OK	表示成功
	其他	返回错误码，见表 A-22

7）设置设备标签函数

函数原型　ULONG DEVAPI SKF_SetLabel (DEVHANDLE hDev,
LPSTR szLable)

功能描述　设置设备标签：设置设备标签。

参数　　hDev　　[IN] 连接设备时返回的设备句柄
　　szLabel　[IN] 设备标签字符串。该字符串应小于 32 字节。
返回值　　SAR_OK　　表示成功
　　其他　　返回错误码，见表 A-22

8）获取设备信息函数

函数原型　ULONG DEVAPI SKF_GetDevInfo (DEVHANDLE hDev,
　　DEVINFO *pDevInfo)
功能描述　获取设备信息：
　　获取设备的一些特征信息，包括设备的标识、厂商信息、口令的长度范围、支持的算法等。
参数　　hDev　　[IN] 连接设备时返回的设备句柄
　　pDevInfo　[OUT] 返回设备信息
返回值　　SAR_OK　　表示成功
　　其他　　返回错误码，见表 A-22

9） 锁定设备函数

函数原型　ULONG DEVAPI SKF_LockDev (DEVHANDLE hDev,
　　ULONG ulTimeOut)
功能描述　锁定设备：
　　获得设备的独占使用权。
参数　　hDev　　[IN] 连接设备时返回的设备句柄
　　ulTimeOut　[IN] 超时时间，单位为毫秒。如果为 0xFFFFFFFF 表示无限等待。
返回值　　SAR_OK　　表示成功
　　其他　　返回错误码，见表 A-22

10）解锁设备函数

函数原型　ULONG DEVAPI SKF_UnLockDev (DEVHANDLE hDev)
功能描述　解锁设备：
　　释放设备的独占使用权。
参数　　hDev　　[IN] 连接设备时返回的设备句柄
返回值　　SAR_OK　　表示成功
　　其他　　返回错误码，见表 A-22

11）设备命令传输函数

函数原型　ULONG DEVAPI SKF_Transmit (DEVHANDLE hDev,BYTE*pbCommand,
　　ULONG ulCommandLen,BYTE*pbDate,ULONG*pulDateLen)
功能描述　设备命令传输：
　　将命令直接发给设备，并返回结果。
　　本函数仅用于设备检测。
参数　　hDev　　[IN] 设备句柄
　　PbCommand　　[IN] 设备命令。
　　ulCommandLen　[IN] 命令长度。
　　pbDate　　[OUT] 返回结果数据。
　　pulDateLen　　[IN,OUT] 输入时表示结果数据缓冲区长度，输出时表示结果数据实际长度。
返回值　　SAR_OK　　表示成功
　　其他　　返回错误码，见表 A-22

b） 访问控制类函数

表 B.17 给出访问控制类的具体函数。

表 B. 17　访问控制类函数

序号	函数名称	函数定义
1	修改设备认证密钥	SKF_ChangeDevAuthKey
2	设备认证	SKF_DevAuth
3	修改 PIN	SKF_ChangePIN
4	获得 PIN 码信息	SKF_GetPINInfo
5	校验 PIN	SKF_VerifyPIN
6	解锁 PIN	SKF_UnblockPIN
7	清除应用安全状态	SKF_ClearSecureState

1）修改设备认证密钥函数

函数原型　ULONG DEVAPI SKF_ChangeDevAuthKey(DEVHANDLE hDev,BYTE*pbKeyVa1ue, ULONG ulKeyLen)

功能描述　更改设备认证密钥。

参数　hDev　[IN] 连接时返回的设备句柄。

pbKeyVa1ue　[IN] 密钥值。

ulKeyLen　[IN] 密钥长度。

返回值　SAR_OK　表示成功

其他　返回错误码，见表 A-22

备注　设备认证成功后才能使用。

2）设备认证函数

函数原型　ULONG DEVAPI SKF_DevAuth(DEVHANDLE hDev,BYTE*pbAuthData, ULONG ulLen)

功能描述　设备认证：设备对应用程序的认证。

参数　hDev　[IN] 连接时返回的设备句柄。

pbAuthData　[IN] 认证数据。

ulLen　[IN] 认证数据的长度。

返回值　SAR_OK　表示成功

其他　返回错误码，见表 A-22

3）修改PIN函数

函数原型　ULONG DEVAPI SKF_ChangePIN (HAPPLICATION hApplication, ULONG ulPINType,LPSTR szOldPin, LPSTR szNewPin, ULONG *pulRetryCount)

功能描述　修改 PIN：

调用该函数可以修改 Admin PIN 和 User PIN 的值。

只有知道原 Admin PIN 或者 User PIN 才能修改。

如果原 PIN 错误，该函数会返回 Admin PIN 或者 User PIN 的剩余重试次数，当剩余次数为 0 时，表示 PIN 已经被锁死。

参数　hApplication　[IN] 应用句柄。

ulPINType　[IN] PIN 类型，可以为 ADMIN_TYPE 或 USER_TYPE。

szOldPin　[IN] 原 PIN 值。

	szNewPin	[IN] 新 PIN 值。
	pulRetryCount	[OUT] 出错后重试次数。
返回值	SAR_OK	表示成功
	其他	返回错误码，见表 A-22

4）获取PIN信息

函数原型	ULONG DEVAPI SKF_GetPINInfo (HAPPLICATION hApplication, ULONG ulPINType,ULONG*pulMaxRetryCount,ULONG*pulRemainRetryCount, BOOL *pbDefaultPin)
功能描述	获取 PIN： 获取 PIN 码信息，包括最大重试次数、当前剩余重试次数，以及当前 PIN 码是否为出厂默认 PIN 码。

参数	hApplication	[IN] 应用句柄
	ulPINType	[IN] PIN 类型。
	pulMaxRetryCount	[OUT] 最大重试次数
	pulRemainRetryCount	[OUT] 当前剩余重试次数，当为 0 时表示已锁死。
	pbDefaultPin	[OUT] 是否为出厂默认 PIN 码。
返回值	SAR_OK	表示成功
	其他	返回错误码，见表 A-22

5）校验PIN函数

函数原型	ULONG DEVAPI SKF_VerifyPIN (HAPPLICATION hApplication, ULONG ulUserType,LPSTR szPIN, ULONG *pulRetryCount)
功能描述	校验 PIN，即 Login： 校验 Admin PIN 或者 User PIN。 校验成功后，会获得相应的权限，如果 PIN 码错误，会返回 PIN 码的重试次数，当重试次数为 0 时表示口令已经锁死。

参数	hApplication	[IN] 应用句柄
	ulPINType	[IN] PIN 类型
	szPIN	[IN] PIN 值
	pulRetryCount	[OUT] 出错后返回的重试次数
返回值	SAR_OK	表示成功
	其他	返回错误码，见表 A-22

6）解锁PIN函数

函数原型	ULONG DEVAPI SKF_UnblockPIN (HAPPLICATION hApplication, LPSTR szAdminPIN, LPSTR szNewUserPIN, ULONG *pulRetryCount)
功能描述	解锁 User PIN： 当用户的 PIN 码锁死后，通过调用该函数来解锁用户 PIN 码。 只有知道 Admin PIN 才能够解锁用户口令，如果输入的 Admin PIN 不正确或者已经锁死，会调用失败，并返回 Admin PIN 的重试次数。 解锁后，用户口令被设置成新值，用户口令的重试次数也恢复到原值。

参数	hApplication	[IN] 应用句柄。
	szAdminPIN	[IN] 管理员 PIN 码。
	szNewUserPIN	[IN] 新的用户 PIN 码。
	pulRetryCount	[OUT] 管理员 PIN 码错误时，返回剩余重试次数。
返回值	SAR_OK	表示成功

其他　　返回错误码，见表 A-22

7）清除应用安全状态函数

函数原型　ULONG DEVAPI SKF_ClearSecureState (HAPPLICATION hApplication)

功能描述　清除应用安全状态，即 Logout：

清楚应应用当前的安全状态。

参数　hApplication　[IN] 应用句柄

返回值　SAR_OK　表示成功

其他　返回错误码，见表 A-22

c） 应用管理类函数

表 B.18 给出应用管理类具体函数。

表 B. 18　应用管理类函数

序号	函数名称	函数定义
1	创建应用函数	SKF_CreateApplication
2	枚举应用函数	SKF_EnumApplication
3	删除应用函数	SKF_DeleteApplication
4	打开应用函数	SKF_OpenApplication
5	关闭应用函数	SKF_CloseApplication

1）创建应用函数

函数原型　ULONG DEVAPI SKF_CreateApplication(DEVHANDLE hDev,

LPSTR szAppName, LPSTR szAdminPin, DWORD dwAdminPinRetryCount,

LPSTR szUserPin, DWORD dwUserPinRetryCount,

DWORD dwCreateFileRights, HAPPLICATION *phApplication)

功能描述　创建一个应用，一个设备可以创建多个应用。

参数　hDev　[IN] 连接设备时返回的设备句柄

szAppName　[IN] 应用名称

szAdminPin　[IN] 管理员 PIN

dwAdminPinRetryCount　[IN] 管理员 PIN 最大重试次数

szUserPin　[IN] 用户 PIN

dwUserPinRetryCount　[IN] 用户 PIN 最大重试次数

dwCreateFileRights　[IN] 在该应用下创建文件和容器的权限。

phApplication　[OUT] 应用的句柄

返回值　SAR_OK　成功

其他　返回错误码，见表 A-22

备注　需要用户权限。

2）枚举应用函数

函数原型　ULONG DEVAPI SKF_EnumApplication(DEVHANDLE hDev,

LPSTR szAppName,ULONG *pulSize, ULONG *pulAppCount)

功能描述　枚举应用：

枚举设备中存在的所有应用。

参数　hDev　[IN] 连接设备时返回的设备句柄

	szAppName	[OUT] 返回应用名称，该参数为空，将由 pulSize 返回所需要的内存空间大小。每个应用的名称以单个‘\0’结束，以双‘\0’表示列表的结束。
	pulSize	[IN,OUT] 输入参数，输入应用名称的缓冲区长度，输出参数，返回 szAppName 所需要的的空间大小。
返回值	SAR_OK	成功
	其他	返回错误码，见表A-22

3）删除应用函数

函数原型	ULONG DEVAPI SKF_DeleteApplication(DEVHANDLE hDev, LPSTR szAppName)	
功能描述	删除应用： 删除一个应用，需要满足安全权限，才能够删除。	
参数	hDev	[IN] 连接设备时返回的设备句柄
	szAppName	[IN] 应用名称
返回值	SAR_OK	成功
	其他	返回错误码，见表 A-22

4）打开应用函数

函数原型	ULONG DEVAPI SKF_OpenApplication(DEVHANDLE hDev, LPSTR szAppName, HAPPLICATION *phApplication)	
功能描述	打开指定的应用。	
参数	hDev	[IN] 连接设备时返回的设备句柄
	szAppName	[IN] 应用名称
	phApplication	[OUT] 应用的句柄
返回值	SAR_OK	成功
	其他	返回错误码，见表 A-22

5）关闭应用函数

函数原型	ULONG DEVAPI SKF_CloseApplication(HAPPLICATION hApplication)	
功能描述	关闭应用并释放应用句柄。此函数不影响应用安全状态。	
参数	hApplication	[IN]应用句柄
返回值	SAR_OK	成功
	其他	返回错误码，见表 A-22

d） 文件管理类函数

表 B.19 给出文件管理类具体函数。

表 B. 19 文件管理类函数

序号	函数名称	函数定义
1	创建文件函数	SKF_CreateFile
2	删除文件函数	SKF_DeleteFile
3	枚举文件函数	SKF_EnumFiles
4	获取文件信息	SKF_GetFileInfo
5	读文件函数	SKF_ReadFile
6	写文件函数	SKF_WriteFile

1）创建文件函数

函数原型　ULONG DEVAPI SKF_CreateFile (HAPPLICATION hApplication,
LPSTR szFileName, ULONG ulFileSize, ULONG ulReadRights，
ULONG ulWriteRights)

功能描述　创建文件：
创建文件时要指定文件的名称，大小，以及文件的读写权限。

参数

	hApplication	[IN] 应用句柄。
	szFileName	[IN] 文件名称，长度不得大于 32 个字节。
	ulFileSize	[IN] 文件大小。
	ulReadRights	[IN] 文件读权限。
	ulWriteRights	[IN] 文件写权限。

返回值

SAR_OK	表示成功
其他	返回错误码，见表 A-22

2）删除文件函数

函数原型　ULONG DEVAPI SKF_DeleteFile (HAPPLICATION hApplication,
LPSTR szFileName)

功能描述　删除文件：
文件删除后，文件中写入的所有信息将丢失。
文件在设备中的占用的空间将被释放。
删除一个已经创建的文件。

参数

	hApplication	[IN] 要删除文件所在的应用句柄
	szFileName	[IN] 要删除文件的名称

返回值

SAR_OK	表示成功
其他	返回错误码，见表 A-22

3）枚举文件函数

函数原型　ULONG DEVAPI SKF_EnumFiles (HAPPLICATION hApplication, LPSTR szFileList,
ULONG *pulSize)

功能描述　枚举文件：
枚举一个应用下存在的所有文件。

参数

	hApplication	[IN] 应用句柄
	szFileList	[OUT] 返回文件名称列表，该参数为空，由 pulSize 返回文件信息所需要的空间大小。每个文件的名称以单个‘\0’结束，以双‘\0’表示列表的结束
	pulSize	[IN, OUT] 输入为数据缓冲区的大小，输出为实际文件名称列表的长度

返回值

SAR_OK	表示成功
其他	返回错误码，见表 A-22

4）获取文件属性

函数原型　ULONG DEVAPI SKF_GetFileInfo (HAPPLICATION hApplication,
LPSTR szFileName, FILEATTRIBUTE*pFileInfo)

功能描述　获取文件：
获取应用文件的属性信息，例如文件的大小、权限等。

参数

	hApplication	[IN] 文件所在应用的句柄
	szFileName	[IN]文件名称。

	pFileInfo	[OUT] 文件信息，指向文件属性结构的指针。
返回值	SAR_OK	表示成功
	其他	返回错误码，见表 A-22

5）读文件函数

函数原型 ULONG DEVAPI SKF_ReadFile (HAPPLICATION hApplication，
LPSTR szFileName, ULONG ulOffset, ULONG ulSize, BYTE * pbOutData,
ULONG *pulOutLen)

功能描述 读文件：
读应用文件中的数据。
对于刚创建的文件，该函数也能够调用成功，读出的数据是文件原有的默认数据值。
对于一个应用文件，首先要对它进行写操作，然后才能够读出正确的数据。

参数	hApplication	[IN] 应用句柄
	szFileName	[IN] 文件名
	ulOffset	[IN] 文件读取偏移位置
	ulSize	[IN] 要读取的数据量
	pbOutData	[OUT] 返回的数据缓冲区
	pulOutLen	[IN,OUT] 输入输出参数：输入表示给出的缓冲区大小；输出表示实际读取返回的数据大小
返回值	SAR_OK	表示成功
	其他	返回错误码，见表 A-22

备注 需具备对该文件的读权限。

6）写文件函数

函数原型 ULONG DEVAPI SKF_WriteFile (HAPPLICATION hApplication,
LPSTR szFileName, ULONG ulOffset, BYTE *pbData, ULONG ulSize)

功能描述 写文件：
写数据到应用文件中。
对于数据在文件中开始偏移量和有效数据的长度由写入者自己记录，这样在对数据进行读操作的时候才能够读出有效的数据。

参数	hApplication	[IN] 应用句柄
	szFileName	[IN] 文件名
	ulOffset	[IN] 写入文件的偏移量
	pbData	[IN] 写入数据缓冲区
	ulSize	[IN] 写入数据的大小
返回值	SAR_OK	表示成功
	其他	返回错误码，见表 A-22

备注 需具备对该文件写权限。

e） 容器管理函数

表B.20给出容器管理类具体函数：

表 B. 20 容器管理类函数

序号	函数名称	函数定义
1	创建容器函数	SKF_CreateContainer
2	删除容器函数	SKF_DeleteContainer
3	枚举容器函数	SKF_EnumContainer

4	打开容器函数	SKF_OpenContainer
5	关闭容器函数	SKF_CloseContainer
6	获得容器类型	SKF_GetContainerType
7	导入数字证书	SKF_ImportCertificate
8	导出数字证书	SKF_ExportCertificate

1）创建容器函数

函数原型　ULONG DEVAPI SKF_CreateContainer (HAPPLICATION hApplication,
LPSTR szContainerName, HCONTAINER *phContainer)

功能描述　创建容器：
在应用下建立指定名称的容器并返回容器句柄。
创建容器时，需要用户权限。

参数　hApplication　[IN] 应用句柄
szContainerName　[IN] ASCII 字符串，表示所建立容器的名称，容器名称的最大长度不能超过 64 字节。
phContainer　[OUT] 返回所建立容器的容器句柄

返回值　SAR_OK　表示成功
其他　返回错误码，见表 A-22

2）删除容器函数

函数原型　ULONG DEVAPI SKF_DeleteContainer (HAPPLICATION hApplication,
LPSTR szContainerName)

功能描述　删除容器：
在应用下删除指定名称的容器并释放容器相关的资源。
删除容器时，需要用户权限。

参数　hApplication　[IN] 应用句柄
szContainerName　[IN] 指向删除容器的名称

返回值　SAR_OK　表示成功
其他　返回错误码，见表A-22

3） 枚举容器函数

函数原型　ULONG DEVAPI SKF_EnumContainer (HAPPLICATION hApplication,
LPSTR szContainerName,ULONG *pulSize)

功能描述　列举密钥容器：列举出指定设备中已存在的密钥容器。

参数　hApplication　[IN] 应用句柄
szContainerName　[OUT] 返回容器名称列表
pulSize　[IN,OUT] 返回容器名的长度

返回值　SAR_OK　表示成功
其他　返回错误码，见表A-22

4） 打开容器函数

函数原型　ULONG DEVAPI SKF_OpenContainer (HAPPLICATION hApplication,
LPSTR szContainerName,HCONTAINER*phContainer)

功能描述　打开容器：获取容器句柄。

参数　hApplication　[IN]应用句柄
hContainerName　[IN]容器的名称

	phContainer	[OUT]返回所打开容器的句柄
返回值	SAR_OK	表示成功
	其他	返回错误码，见表A-22

5） 关闭容器函数

函数原型	ULONG DEVAPI SKF_CloseContainer (HCONTAINER hContainer)	
功能描述	关闭容器句柄，并释放容器句柄相关资源。	
参数	hContainer	[IN] 容器句柄
返回值	SAR_OK	表示成功
	其他	返回错误码，见表A-22

6） 获取容器类型函数

函数原型	ULONG DEVAPI SKF_GetContainerType (HCONTAINER *hContainer, ULONG* pulContainerType)	
功能描述	获取容器类型：获取容器的类型。	
参数	hContainer	[IN] 容器句柄
	pulContainerType	[OUT] 返回容器的类型。指针指向的值为 0 表示未定、尚未分配类型或者为空容器，为 1 表示 RSA 容器，为 2 表示为 SM2 容器。
返回值	SAR_OK	表示成功
	其他返	返回错误码，见表A-22

7） 导入数字证书函数

函数原型	ULONG DEVAPI SKF_ ImportCertificate (HCONTAINER hContainer, BOOL bSignFlag, BYTE*pbCert,ULONG ulCertLen)	
功能描述	导入数字证书：向容器内导入数字证书。	
参数	hContainer	[IN] 容器句柄
	bSignFlag	[IN] TRUE 表示签名证书，FALSE 表示加密证书
	pbCert	[IN] 指向证书内容缓冲区
	ulCertLen	[IN] 证书长度
返回值	SAR_OK	表示成功
	其他	返回错误码，见表A-22

8） 导出数字证书函数

函数原型	ULONG DEVAPI SKF_ ExportCertificate (HCONTAINER hContainer, BOOL bSignFlag, BYTE*pbCert,ULONG*pulCertLen)	
功能描述	导出数字证书：从容器内导出数字证书。	
参数	hContainer	[IN] 容器句柄
	bSignFlag	[IN] TRUE 表示签名证书，FALSE 表示加密证书
	pbCert	[IN] 指向证书内容缓冲区，此参数为 NULL 时，pulcertLen 表示返回数据所需要缓冲区的长度；此参数不为 NULL 时，返回数字证书内容。
	pulCertLen	[IN,OUT] 输入时表示 pbCert 缓冲区的长度，输出时表示证书内容的长度
返回值	SAR_OK	表示成功
	其他	返回错误码，见表A-22

f） 密码服务类函数

表 B.21 给出密码服务类具体函数。

表 B.21　密码服务类函数

序号	函数名称	函数定义
1	生成随机数	SKF_GenRandom
2	生成外部 RSA 密钥对	SKF_GenExtRSAKey
3	生成 RSA 签名密钥对	SKF_GenRSAKeyPair
4	导入 RSA 加密密钥对	SKF_ImportRSAKeyPair
5	RSA 签名	SKF_RSASignData
6	RSA 验签	SKF_RSAVerify
7	RSA 生成并导出会话密钥	SKF_RSAExportSessionKey
8	RSA 外来公钥运算	SKF_ExtRSAPubKeyOperation
9	生成 ECC 签名密钥对	SKF_GenECCKeyPair
10	导入 ECC 加密密钥对	SKF_ImportECCKeyPair
11	ECC 签名	SKF_ECCSignData
12	ECC 验签	SKF_ECCVerify
13	ECC 生成并导出会话密钥	SKF_ECCExportSessionKey
14	ECC 外来公钥加密	SKF_ExtECCEncrypt
15	ECC 外来公钥验签	SKF_ExtECCVerify
16	ECC 生成密钥协商参数并输出	SKF_GenerateAgrementDataWithECC
17	ECC 计算会话密钥	SKF_GenerateKeyWithECC
18	ECC 产生协商数据并计算会话密钥	SKF_GenerateAgrementDataAndKeyWithECC
19	导入会话密钥	SKF_ImportSessionKey
20	导出公钥	SKF_ExportPublicKey
21	加密初始化	SKF_EncryptInit
22	单组数据加密	SKF_Encrypt
23	多组数据加密	SKF_EncryptUpdate
24	结束加密	SKF_EncryptFinal
25	解密初始化	SKF_DecryptInit
26	单组数据解密	SKF_Decrypt
27	多组数据解密	SKF_DecryptUpdate
28	结束解密	SKF_DecryptFinal
29	密码杂凑初始化	SKF_DigestInit
30	单组数据密码杂凑	SKF_Digest
31	多组数据密码杂凑	SKF_DigestUpdate
32	结束密码杂凑	SKF_DigestFinal
33	消息鉴别码运算初始化	SKF_MacInit
34	单组数据消息鉴别码运算	SKF_Mac
35	多组数据消息鉴别码运算	SKF_MacUpdate
36	结束消息鉴别码运算	SKF_MacFinal
37	关闭密码对象句柄	SKF_CloseHandle

1） 生成随机数函数

函数原型 ULONG DEVAPI SKF_GenRandom (DEVHANDLE hDev,
BYTE *pbRandom,ULONG ulRandomLen)

功能描述 产生随机数。由 hDev 句柄指向的设备的随机数发生器产生随机数，随机数长度为 ulRandomLen，得到的随机数保存到 pbRandom 指向的缓冲区。

参数

	hDev	[IN] 连接设备时返回的设备句柄
	pbRandom	[OUT] 返回的随机数缓冲区
	ulRandomLen	[IN] 需要返回的随机数长度

返回值

SAR_OK	表示成功
其他	返回错误码，见表A-22

2）生成外部RSA密钥对函数

函数原型 ULONG DEVAPI SKF_GenExtRSAKey (DEVHANDLE hDev,
ULONG ulBitsLen, RSAPRIVATEKEYBLOB *pBlog)

功能描述 由设备生成 RSA 密钥对并明文输出。

参数

	hDev	[IN] 连接设备时返回的设备句柄
	ulBitsLen	[IN] 密钥模长
	pBlog	[IN] 返回的私钥数据结构

返回值

SAR_OK	表示成功
其他	返回错误码，见表A-22

备注 生成的私钥只用于输出，接口内不做保留和计算。

3） 生成RSA签名密钥对函数

函数原型 ULONG DEVAPI SKF_GenRSAKeyPair (HCONTAINER hContainer,
ULONG ulBitsLen, RSAPRIVATEKEYBLOB *pBlog)

功能描述 生成 RSA 密钥对。

参数

	hContainer	[IN] 容器句柄
	ulBitsLen	[IN] 密钥模长
	pBlog	[IN] 返回的 RSA 公钥数据结构

返回值

SAR_OK	表示成功
其他	返回错误码，见表 A-22

备注 需要用户权限。

4） 导入RSA加密密钥对函数

函数原型 ULONG DEVAPI SKF_ImportRSAKeyPair (HCONTAINER hContainer,
ULONG ulSymAlgId,BYTE*pbWrappedKey,ULONG ulWrappedKeyLen,
BYTE*pbEncryptedData,ULONG ulEncryptedDataLen)

功能描述 导入 RSA 加密公私钥对。

参数

	hContainer	[IN] 密钥容器句柄
	ulSymAlgId	[IN] 对称算法密钥标识
	pbWrapedKey	[IN] 使用该容器内签名公钥保护的对称算法密钥。
	ulWrapedKeyLen	[IN] 保护的对称算法密钥长度。
	pbEncryptedData	[IN] 对称算法密钥保护的 RSA 加密私钥。私钥的格式遵循 PKCS#1v2.1 RSACryptographyStandard 中的私钥格式定义。
	ulEncryptedDataLen	[IN] 对称算法密钥保护的 RSA 加密公私钥对长度。

返回值

SAR_OK	表示成功
其他	返回错误码，见表A-22

备注　　需要用户权限

5） RSA签名函数

函数原型　ULONG DEVAPI SKF_RSASignData (HCONTAINER hContainer, BYTE *pbData, ULONG　ulDataLen, BYTE *pbSignature, ULONG *pulSigLen)

功能描述　RSA 数字签名。使用 hContainer 指定容器的签名私钥，对指定数据 pbData 进行数字签名。签名后的结果存放到 pbSignature 缓冲区，设置 pulSigLen 为签名的长度。

参数

hContainer	[IN] 用来签名的私钥所在容器句柄
pbData	[IN] 被签名的数据。
ulDataLen	[IN] 签名数据长度，应不大于 RSA 密钥模长-11.
pbSignature	[OUT] 存放签名结果的缓冲区指针，如果值为 NULL，用于取得签名结果长度。
pulSignLen	[IN,OUT] 输入时表示签名结果缓冲区大小，输出时表示签名数据长度。

返回值

SAR_OK	表示成功
其他	返回错误码，见表A-22

备注　　需要用户权限。

6） RSA验签函数

函数原型　ULONG DEVAPI SKF_RSAVerify (DEVHANDLE hDev, RSAPRIVATEKEYBLOB *pRSAPubKeyBlog, BYTE *pbData, ULONG　ulDataLen, BYTE *pbSignature, ULONG ulSigLen)

功能描述　验证 RSA 签名。用 pRSAPubKeyBlog 内的 RSA 公钥对数据签名进行验签，与原始数据对比，若验签后的数据与原始数据相同，则说明签名是有效的，否则签名无效。

参数

hDev	[IN] 连接设备时返回的设备句柄。
pRSAPubKeyBlog	[IN] RSA 公钥数据结构。
pbData	[IN] 验证签名的数据。
ulDataLen	[IN] 数据长度，应不大于公钥模长-11
pbSignature	[IN] 待验证的签名值。
ulSignLen	[IN] 数据签名长度，必须为公钥模长。

返回值

SAR_OK	表示成功
其他	返回错误码，见表A-22

7） RSA生成并导出会话密钥函数

函数原型　ULONG DEVAPI SKF_ExportSessionKey (HCONTAINER hContainer, ULONG ulAlgId,RSAPRIVATEKEYBLOB *pRSAPubKey, BYTE *pbData, ULONG *pulDataLen,HANDLE*phSessionKey)

功能描述　导出会话密钥。

参数

hContainer	[IN]容器句柄
ulAlgId	[IN]会话密钥算法标识
pRSAPubKey	[IN]加密会话密钥的 RSA 公钥数据结构
pbData	[OUT] 导出的加密会话密钥密文，按照 PKCS#1v1.5 要求封装
pulDataLen	[IN,OUT] 输入时表示会话密钥密文数据缓冲区长度，输出时表示会话密钥密文的实际长度
PhSessionKey	[OUT] 导出的密钥句柄

返回值

SAR_OK	表示成功
其他	返回错误码，见表 A-22

8）RSA外来公钥运算函数

函数原型	ULONG DEVAPI SKF_ExtRSAPubKeyOperation(DEVHANDLE hDev, RSAPUBLICKEYBLOB*pRSAPubKeyBlob, BYTE * pbInput,ULONG ulInputLen, BYTE*pbOutput, ULONG*pulOutputLen)	
功能描述	使用外部传入的 RSA 公钥对输入数据做公钥运算并输出结果。	
参数	hDev	[IN] 连接设备时返回的设备句柄
	pRSAPubKeyBlob	[IN] RSA 公钥数据结构。
	pbInput	[IN] 指向待运算的原始数据缓冲区。
	ulInputLen	[IN] 待运算原始数据的长度,必须为公钥模长。
	pbOutput	[OUT] 指向 RSA 公钥运算结果缓冲区,如果该参数为 NULL，则由 pulOutputLen 返回运算结果的实际长度
	pulOutputLen	[IN,OUT] 输入时表示 pbOutput 缓冲区的长度，输出时表示 RSA 公钥运算结果的实际长度。
返回值	SAR_OK	表示成功
	其他	返回错误码，见表 A-22

9） 生成ECC签名密钥对函数

函数原型	ULONG DEVAPI SKF_GenECCKeyPair(HCONTAINER hContainer,ULONG ulAlgId, ECPUBLICKEYBLOB*pBlob)	
功能描述	生成 ECC 签名密钥对并输出签名公钥。	
参数	hContainer	[IN] 密钥容器句柄。
	ulAlgId	[IN] 算法标识,只支持 SGD_SM2_1 算法。
	pBlob	[OUT] 返回 ECC 公钥数据结构。
返回值	SAR_OK	表示成功
	其他	返回错误码，见表A-22
备注	需要用户权限	

10）导入ECC加密密钥对函数

函数原型	ULONG DEVAPI SKF_ImportECCKeyPair(HCONTAINER hContainer, PENVELOPEDKEYBLOB pEnvelopedKeyBlob)	
功能描述	导入 ECC 公私钥对。	
参数	hContainer	[IN] 密钥容器句柄。
	pEnvelopedKeyBlob	[IN] 受保护的加密密钥对
返回值	SAR_OK	表示成功
	其他	返回错误码，见表A-22
备注	需要用户权限	

11）ECC签名函数

函数原型	ULONG DEVAPI SKF_ECCSignData(HCONTAINER hContainer,BYTE*pbData, ULONG ulDataLen,PECCSIGNATUREBLOB pSignature)	
功能描述	ECC 数字签名。采用 ECC 算法和指定私钥 hKey，对指定数据 pbData 进行数字签名。签名后的结果存放到 pSignature 中。	
参数	hContainer	[IN] 密钥容器句柄。
	pbData	[IN] 待签名的数据
	ulDataLen	[IN] 待签名数据长度，必须小于密钥模长。
	pSignature	[OUT] 签名值。
返回值	SAR_OK	表示成功

	其他	返回错误码，见表A-22
备注	需要用户权限。 输入数据为待签数据的杂凑值。当使用SM2算法时，该输入数据为待签数据经过SM2签名预处理的结果，预处理过程符合GB/T 35276-2017规定的要求。	

12）ECC验签函数

函数原型	ULONG DEVAPI SKF_ECCVerify(DEVHANDLE hDev, ECCPUBLICKEYBLOB*pECCPubKeyBlob,BYTE*pbData,ULONG ulDataLen, PECCSIGNATUREBLOB pSignature)	
功能描述	用 ECC 公钥对数据进行验签。	
参数	hDev	[IN] 设备句柄
	pECCPubKeyBlob	[IN] ECC 公钥数据结构。
	pbData	[IN] 待验证签名的数据
	ulDataLen	[IN] 数据长度
	pSignature	[IN] 待验证签名值
返回值	SAR_OK	表示成功
	其他	返回错误码，见表A-22
备注	输入数据为待签数据的杂凑值。当使用SM2算法时，该输入数据为待签数据经过SM2签名预处理的结果，预处理过程符合GB/T 35276-2017规定的要求。	

13）ECC生成并导出会话密钥函数

函数原型	ULONG DEVAPI SKF_ECCExportSessionKey(HCONTAINER hContainer, ULONG ulAlgId,ECCPUBLICKEYBLOB*pPubKey,PECCCIPHERBLOB pData, HANDLE*phSessionKey)	
功能描述	生成会话密钥并用外部公钥加密导出。	
参数	hContainer	[IN] 密钥容器句柄。
	ulAlgId	[IN] 会话密钥算法标识
	pPubKey	[IN] 外部输入的公钥结构
	pData	[OUT]会话密钥密文。
	phSessionKey	[OUT] 会话密钥句柄
返回值	SAR_OK	表示成功
	其他	返回错误码，见表A-22

14）ECC外来公钥加密函数

函数原型	ULONG DEVAPI SKF_ExtECCEncrypt(DEVHANDLE hDev, ECCPUBLICKEYBLOB*pECCPubKeyBlob,BYTE* pbPlainText, ULONG ulPlainTextLen,PECCIPHERBLOB pCipherText)	
功能描述	使用外部传入的 ECC 公钥对输入数据做加密运算并输出结果。	
参数	hDev	[IN] 设备句柄
	pECCPubKeyBlob	[IN] ECC 公钥数据结构。
	pbPlainText	[IN] 待加密的明文数据
	ulPlainTextLen	[IN] 待加密明文数据的长度
	pCipherTex	[OUT] 密文数据
返回值	SAR_OK	表示成功
	其他	返回错误码，见表A-22

15）ECC外来公钥验签函数

函数原型	ULONG DEVAPI SKF_ExtECCVerify(DEVHANDLE hDev,	

ECCPUBLICKEYBLOB*pECCPubKeyBlob,BYTE*pbData,ULONG ulDataLen,
PECCSIGNATUREBLOB pSignature)

功能描述 外部使用传入的 ECC 公钥做签名验证。

参数

	hDev	[IN] 设备句柄
	pECCPubKeyBlob	[IN] ECC 公钥数据结构。
	pbData	[IN] 待验证数据。
	ulDataLen	[IN] 待验证数据的长度。
	pSignature	[IN] 签名值。

返回值

SAR_OK	表示成功
其他	返回错误码，见表A-22

备注 输入数据为待签数据的杂凑值。当使用SM2算法时，该输入数据为待签数据经过SM2签名预处理的结果，预处理过程符合GB/T 35276-2017规定的要求。

16）ECC生成密钥协商参数并输出函数

函数原型 ULONG DEVAPI SKF_GenerateAgrementDataWithECC(HCONTAINER hContainer,
ULONG ulAlgId,ECCPUBLICKEYBLOB* pTempECCPubKeyBlob,BYTE*pbID,
ULONG ulIDLen,HANDLE*phAgrementHandle)

功能描述 使用 ECC 密钥协商算法，为计算会话密钥而产生协商参数，返回临时 ECC 密钥对的公钥及协商句柄。

参数

	hContainer	[IN] 密钥容器句柄。
	ulAlgId	[IN] 会话密钥算法标识
	pTempECCPubKeyBlob	[OUT] 发起方临时 ECC 公钥
	pbID	[IN] 发起方的 ID
	ulIDLen	[IN] 发起方 ID 的长度，不大于 32
	phAgrementHandle	[OUT] 返回的密钥协商句柄

返回值

SAR_OK	表示成功
其他	返回错误码，见表A-22

备注 为协商会话密钥,协商的发起方应首先询用本函数。

17）ECC产生协商数据并计算会话密钥函数

函数原型 ULONG DEVAPI SKF_GenerateAgrementDataAndKeyWithECC(
HANDLE hContainer,ULONG ulAlgId,
ECCPUBLICKEYBLOB* pSponsorECCPubKeyBlob,
ECCPUBLICKEYBLOB* pSponsorTempECCPubKeyBlob,
ECCPUBLICKEYBLOB* pTempECCPubKeBlob,
BYTE*pbID,ULONG ulIDLen,BYTE*pbSponsorID,ULON GulSponsorIDLen,
HANDLE*phKeyHandle)

功能描述 使用 ECC 密钥协商算法，产生协商参数并计算会话密钥,输出临时 ECC 密钥对公钥,并返回产生的密钥句柄。

参数

	hContainer	[IN] 密钥容器句柄。
	ulAlgId	[IN] 会话密钥算法标识
	pSponsorECCPubKeyBlob	[IN] 发起方的 ECC 公钥。
	pSponsorTempECCPubKeyBlob	[IN] 发起方的临时 ECC 公钥
	pTempECCPubKeyBlob	[OUT]响应方的临时 ECC 公钥
	pbID	[IN]响应方的 ID。
	ulIDLen	[IN]响应方 ID 的长度，不大于 32。

	pbSponsorID	[IN]发起方的 ID。
	ulSponsorIDLen	[IN] 发起方 ID 的长度，不大于 32。
	phKeyHandle	[OUT]返回的对称算法密钥句柄
返回值	SAR_OK	表示成功
	其他	返回错误码，见表A-22
备注	本函数由响应方询用。	

18）ECC计算会话密钥函数

函数原型 ULONG DEVAPI SKF_GenerateKeyWithECC(HANDLE hAgrementHandle, ECCPUBLICKEYBLOB* pECCPubKeyBlob, ECCPUBLICKEYBLOB* pTempECCPubKeyBlob, BYTE*pbID,ULONG ulIDLen, HANDLE*phKeyHandle)

功能描述 使用 ECC 密钥协商算法,使用自身协商句柄和响应方的协商参数计算会话密钥，同时返回会话密钥句柄。

参数	hAgrementHandle	[IN] 密钥协商句柄。
	pECCPubKeyB1ob	[IN] 外部输入的响应方 ECC 公钥
	pTempECCPubKeyB1ob	[IN] 外部输入的响应方临时 ECC 公钥。
	pbID	[IN]响应方的 ID。
	ulIDLen	[IN]响应方 ID 的长度，不大于 32。
	phKeyHandle	[OUT]返回的对称算法密钥句柄
返回值	SAR_OK	表示成功
	其他	返回错误码，见表A-22
备注	协商的发起方获得响应方的协商参数后询用本函数，计算会话密钥。计算过程符合 GB/T 35276-2017 规定的要求。	

19）导入会话密钥函数

函数原型 ULONG DEVAPI SKF_ImportSessionKey (HCONTAINER hContainer, ULONG ulAlgId, BYTE*pbWrapedData,ULONG ulWrapedLen,HANDLE*phKey)

功能描述 导入会话密钥密文，使用容器中的加密私钥解密得到会话密钥。

参数	hContainer	[IN] 容器句柄。
	ulAlgID	[IN] 会话密钥算法标识
	pbWrapedData	[IN] 要导入的会话密钥密文。当容器为 ECC 类型时，此参数为 ECCCIPHERBLOB 密文数据，当容器为 RSA 类型时，此参数为 RSA 公钥加密后的数据。
	ulWrapedLen	[IN] 会话密钥密文长度
	phKey	[OUT] 返回的会话密钥句柄
返回值	SAR_OK	表示成功
	其他	返回错误码，见表A-22
备注	需要用户权限	

20）导出公钥函数

函数原型 ULONG DEVAPI SKF_ExportPublicKey(HCONTAINER hContainer, BOOL bSignFlag, BYTE*pbBlob,ULONG*pulBlobLen)

功能描述 导出容器中的签名公钥或者加密公钥。

参数	hContainer	[IN] 密钥容器句柄。
	bSignFlag	[IN]。TRUE 表示导出签名公钥，FALSE 表示导出加密公钥。

	pbBlob	[OUT]指向 RSA 公钥结构（RSAPUBLICKEYBLOB）或者 ECC 公钥结构（ECCPUBLICKEYBLOB），如果此参数为 NULL 时，由 pulBlobLen 返回 pbBlob 的长度。
	pulBlobLen	[IN,OUT] 输入时表示 pbBlob 缓冲区的长度，输出时表示导出公钥结构的长度。
返回值	SAR_OK	表示成功
	其他	返回错误码，见表A-22

21）加密初始化函数

函数原型	ULONG DEVAPI SKF_EncryptInit (HANDLE hKey, BLOCKCIPHERPARAM EncryptParam)
功能描述	数据加密初始化。设置数据加密的算法相关参数。

参数	hKey	[IN] 加密密钥句柄
	EncryptParam	[IN] 分组密码算法相关参数：初始向量、初始向量长度、填充方法、反馈值的位长度
返回值	SAR_OK	表示成功
	其他	返回错误码，见表 A-22

22）单组数据加密函数

函数原型	ULONG DEVAPI SKF_Encrypt(HANDLE hKey, BYTE * pbData, ULONG ulDataLen, BYTE *pbEncryptedData, ULONG *pulEncryptedLen)
功能描述	单一分组数据的加密操作。用指定加密密钥对指定数据进行加密，被加密的数据只包含一个分组，加密后的密文保存到指定的缓冲区中。SKF_Encrypt 只对单个分组数据进行加密，在调用 SKF_Encrypt 之前，必须调用 SKF_EncryptInit 初始化加密操作。SKF_Encypt 等价于先调用 SKF_EncryptUpdate 再调用 SKF_EncryptFinal，SKF_Encrypt 不能与 SKF_EncryptUpdate 间隔调用。

参数	hKey	[IN] 加密密钥句柄
	pbData	[IN] 待加密数据
	ulDataLen	[IN] 待加密数据长度
	pbEncryptedData	[OUT] 加密后的数据缓冲区指针，可以为 NULL，用于获得加密后数据长度。
	pulEncryptedLen	[IN,OUT] 输入时表示结果数据缓冲区大小;输出时表示加密后的结果数据实际长度
返回值	SAR_OK	表示成功
	其他	返回错误码，见表 A-22

23）多组数据加密函数

函数原型	ULONG DEVAPI SKF_EncryptUpdate(HANDLE hKey, BYTE * pbData, ULONG ulDataLen, BYTE *pbEncryptedData, ULONG *pulEncryptedLen)
功能描述	多个分组数据的加密操作。用指定加密密钥对指定数据进行加密，被加密的数据包含多个分组，加密后的密文保存到指定的缓冲区中。SKF_EncryptUpdate 对多个分组数据进行加密，在调用 SKF_EncryptUpdate 之前，必须调用 SKF_EncryptInit 初始化加密操作；在调用 SKF_EncryptUpdate 之后，必须调用 SKF_EncryptFinal 结束加密操作。

参数	hKey	[IN] 加密密钥句柄
	pbData	[IN] 待加密数据

	ulDataLen	[IN] 待加密数据长度
	pbEncryptedData	[OUT] 加密后的数据缓冲区指针
	pulEncryptedLen	[OUT] 返回加密后的数据长度
返回值	SAR_OK	表示成功
	其他	返回错误码，见表 A-22

24）结束加密函数

函数原型 ULONG DEVAPI SKF_EncryptFinal (HANDLE hKey, BYTE *pbEncryptedData, ULONG *ulEncryptedDataLen)

功能描述 结束多个分组数据的加密。先调用 SKF_EncryptInit 初始化加密操作，再调用 SKF_EncryptUpdate 对多个分组数据进行加密，最后调用 SKF_EncryptFinal 结束多个分组数据的加密。

参数	hKey	[IN] 加密密钥句柄
	pbEncyptedData	[OUT] 加密结果的缓冲区
	ulEncyptedDataLen	[OUT] 加密结果的长度
返回值	SAR_OK	表示成功
	其他	返回错误码，见表 A-22

25）解密初始化函数

函数原型 ULONG DEVAPI SKF_DecryptInit (HANDLE hKey,BLOCKCIPHERPARAM DecryptParam)

功能描述 数据解密初始化，设置解密密钥相关参数。调用 SKF_DecryptInit 之后，可以调用 SKF_Decrypt 对单个分组数据进行解密，也可以多次调用 SKF_DecryptUpdate 之后再调用 SKF_DecryptFinal 完成对多个分组数据的解密。

参数	hKey	[IN] 解密密钥句柄
	DecryptParam	[IN] 分组密码算法相关参数：初始向量、初始向量长度、填充方法、反馈值的位长度
返回值	SAR_OK	表示成功
	其他	返回错误码，见表 A-22

26）单组数据解密函数

函数原型 ULONG DEVAPI SKF_Decrypt(HANDLE hKey, BYTE * pbEncryptedData, ULONG ulEncryptedLen, BYTE * pbData, ULONG * pulDataLen)

功能描述 单个分组数据的解密操作。用指定解密密钥对指定数据进行解密，被解密的数据只包含一个分组，解密后的明文保存到指定的缓冲区中。SKF_Decrypt 只对单个分组数据进行解密，在调用 SKF_Decrypt 之前，必须调用 SKF_DecryptInit 初始化解密操作。SKF_Decypt 等价于先调用 SKF_DecryptUpdate 再调用 SKF_DecryptFinal，SKF_Decrypt 不能与 SKF_DecryptUpdate 间隔调用。

参数	hKey	[IN] 解密密钥句柄
	pbEncryptedData	[IN] 待解密数据
	ulEncryptedLen	[IN] 待解密数据长度
	pbData	[OUT] 指向解密后的数据缓冲区指针，当为 NULL 可获得解密后的数据长度。
	pulDataLen	[IN，OUT] 输入时表示结果数据缓冲区长度，输出时表示解密后的数据实际长度。
返回值	SAR_OK	表示成功

其他　　返回错误码，见表 A-22

27）多组数据解密函数

函数原型　ULONG DEVAPI SKF_DecryptUpdate(HANDLE hKey, BYTE * pbEncryptedData, ULONG ulEncryptedLen, BYTE * pbData, ULONG * pulDataLen)

功能描述　多个分组数据的解密操作。用指定解密密钥对指定数据进行解密，被解密的数据包含多个分组，解密后的明文保存到指定的缓冲区中。SKF_DecryptUpdate 对多个分组数据进行解密，在调用 SKF_DecryptUpdate 之前，必须调用 SKF_DecryptInit 初始化解密操作；在调用 SKF_DecryptUpdate 之后，必须调用 SKF_DecryptFinal 结束解密操作。

参数　hKey　[IN] 解密密钥句柄

pbEncryptedData　[IN] 待解密数据

ulEncryptedLen　[IN] 待解密数据长度

pbData　[OUT] 指向解密后的数据缓冲区指针

pulDataLen　[IN，OUT] 输入时表示结果数据缓冲区长度，输出时表示解密后的数据实际长度。

返回值　SAR_OK　表示成功

其他　返回错误码，见表 A-22

28）结束解密函数

函数原型　ULONG DEVAPI SKF_DecryptFinal (HANDLE hKey, BYTE *pbDecryptedData, ULONG *pulDecryptedDataLen)

功能描述　结束多个分组数据的解密。先调用 SKF_DecryptInit 初始化解密操作，再调用 SKF_DecryptUpdate 对多个分组数据进行解密，最后调用 SKF_DecryptFinal 结束多个分组数据的解密。

参数　hKey　[IN] 解密密钥句柄

pbDecyptedData　[OUT] 解密结果的缓冲区，如果此参数为 NULL 时，由 pulDecryptedDataLen 返回解密结果的长度。

pulDecyptedDataLen　[IN，OUT] 输入时表示 pbDecyptedData 缓冲区长度，输出时表示解密结果的长度。

返回值　SAR_OK　表示成功

其他　返回错误码，见表 A-22

29）密码杂凑初始化函数

函数原型　ULONG DEVAPI SKF_DigestInit (DEVHANDLE hDev, ULONG ulAlgID, ECCPUBLICKEYBLOB*pPubKey, unSigned char*pucID,ULONG ulIDLen, HANDLE *phHash)

功能描述　初始化消息杂凑计算操作，指定计算消息杂凑的算法。调用 SKF_DigestInit 之后，可以调用 SKF_Digest 对单一分组数据计算消息杂凑，也可以多次调用 SKF_DigestUpdate 之后再调用 SKF_DigestFinal 对多个分组数据计算消息杂凑。

参数　hDev　[IN] 连接设备时返回的设备句柄

ulAlgID　[IN] 杂凑算法标识

pPubKey　[IN] 签名者公钥。当 ulAlgID 为 SGD_SM3 时有效。

pucID　[IN] 签名者的 ID 值。当 ulAlgID 为 SGD_SM3 时有效。

ulIDLen　[IN] 签名者 ID 的长度。当 ulAlgID 为 SGD_SM3 时有效。

phHash　[OUT] 杂凑对象句柄

返回值　SAR_OK　表示成功

其他　　返回错误码，见表 A-22

30）单组数据杂凑函数

函数原型　ULONG DEVAPI SKF_Digest (HANDLE hHash, BYTE *pbData, ULONG ulDataLen, BYTE *pbHashData, ULONG *pulHashLen)

功能描述　SKF_Digest 对单一分组的消息进行杂凑计算。调用 SKF_Digest 之前，必须调用 SKF_DigestInit 初始化密码杂凑计算操作。SKF_Digest 等价于多次调用 SKF_DigestUpdate 之后再调用 SKF_DigestFinal。

参数　hHash　[IN] 杂凑哈希对象句柄

pbData　[IN] 消息数据

ulDataLen　[IN] 消息数据长度

pbHashData　[OUT] 杂凑数据缓冲区指针，当此参数为 NULL 时，由 pulHashLen 返回密码杂凑结果的长度。

pulHashLen　[IN，OUT] 输入时表示结果数据缓冲区长度，输出时表示结果数据实际长度。

返回值　SAR_OK　表示成功

其他　返回错误码，见表 A-22

31）多组数据杂凑函数

函数原型　ULONG DEVAPI SKF_DigestUpdate (HANDLE hHash, BYTE *pbData, ULONG ulDataLen)

功能描述　SKF_DigestUpdate 对多个分组的消息进行密码杂凑计算。调用 SKF_DigestUpdate 之前，必须调用 SKF_DigestInit 初始化密码杂凑计算操作；调用 SKF_DigestUpdate 之后，必须调用 SKF_DigestFinal 结束密码杂凑计算操作。

参数　hHash　[IN] 密码杂凑对象句柄

pbData　[IN] 指向消息数据的缓冲区

ulDataLen　[IN] 消息数据长度

返回值　SAR_OK　表示成功

其他　返回错误码，见表 A-22

32）结束密码杂凑函数

函数原型　ULONG DEVAPI SKF_DigestFinal (HANDLE hHash, BYTE *pHashData, ULONG *pulHashLen)

功能描述　结束多个分组消息的杂凑计算操作，将杂凑保存到指定的缓冲区。SKF_DigestFinal 必须用于 SKF_DigestUpdate 之后。

参数　hHash　[IN] 密码杂凑对象句柄

pHashData　[OUT] 返回的杂凑数据缓冲区指针，如果此参数为 NULL，由 pulHashLen 返回杂凑结果的长度。

pulHashLen　[IN，OUT] 输入时表示杂凑结果缓冲区的长度，输出时表示杂凑数据长度

返回值　SAR_OK　表示成功

其他　返回错误码，见表 A-22

33）消息鉴别码运算初始化函数

函数原型　ULONG DEVAPI SKF_MacInit(HANDLE hKey, BLOCKCIPHERPARAM*pMacParam, HANDLE*phMac)

功能描述　初始化消息鉴别码计算操作，设置计算消息鉴别码的所需参数，并返回消息鉴别码句柄。

消息鉴别码计算采用分组加密算法的 CBC 模式，将加密结果的最后一块作为计算结果。待计算数据的长度必须是分组加密算法块长的倍数，接口内部不作数据填充。

参数

	hKey	[IN] 计算消息鉴别码的密钥句柄
	pMacParam	[IN] 消息认证计算相关参数,包括初始向量、初始向量长度、填充方法等。
	phMac	[OUT] 消息鉴别码对象句柄

返回值

SAR_OK	表示成功
其他	返回错误码，见表 A-22

34）单组数据消息鉴别码运算函数

函数原型　ULONG DEVAPI SKF_Mac(HANDLE hMac,
BYTE*pbData,ULONG ulDataLen, BYTE*pbMacData,ULONG*pu1MacLen)

功能描述　计算单一分组数据的消息鉴别码。

调用 SKF_Mac 之前,必须调用 SKF_MacInit 初始化消息鉴别码计算操作。SKF_Mac 等价于多次调用 SKF_MacUpdate 之后再调用 SKF_MacFinal。

参数

	hMac	[IN] 消息鉴别码的句柄
	pbData	[IN] 指向待计算数据的缓冲区
	ulDataLen	[IN] 待计算数据的长度
	pbMacData	[OUT] 指向计算后的 Mac 结果，如果此参数为 NULL 时，由 pu1MacLen 返回计算后 Mac 结果的长度。
	pu1MacLen	[IN ,OUT] 输入时表示 pbMacData 缓冲区的长度，输出时表示 Mac 结果的长度。

返回值

SAR_OK	表示成功
其他	返回错误码，见表 A-22

35）多组数据消息鉴别码运算函数

函数原型　ULONG DEVAPI SKF_MacUpdate (HANDLE hMac,BYTE*pbData,
ULONG ulDataLen)

功能描述　计算多个分组数据的消息鉴别码。

调用 SKF_MacUpdate 之前,必须调用 SKF_MacInit 初始化消息鉴别码计算操作；调用 SKF_MacUpdate 之后，必须调用 SKF_MacFinal 结束多个分组数据的消息鉴别码计算操作。

参数

	hMac	[IN] 消息鉴别码的句柄
	pbData	[IN] 指向待计算数据的缓冲区
	ulDataLen	[IN] 待计算数据的长度

返回值

SAR_OK	表示成功
其他	返回错误码，见表 A-22

36）结束消息鉴别码运算函数

函数原型　ULONG DEVAPI SKF_MacFinal (HANDLE hMac,BYTE*pbMacData,
ULONG pulMacDataLen)

功能描述　结束多个分组数据的消息鉴别码计算操作。

SKF_MacFinal 必须用于 SKF_MacUpdate 之后。

参数

	hMac	[IN] 消息鉴别码的句柄

	pbMacData	[OUT] 指向消息鉴别码的缓冲区，当此参数为 NULL 时，由 pulMacDataLen 返回消息鉴别码返回的长度。
	pulMacDataLen	[OUT] 调用时表示消息鉴别码缓冲区的最大长度，返回消息鉴别码的长度。
返回值	SAR_OK	表示成功
	其他	返回错误码，见表 A-22

37）关闭密码对象句柄函数

函数原型	ULONG DEVAPI SKF_C1oseHand1e(HANDLE hHand1e)	
功能描述	关闭会话密钥、密码杂凑对象、消息鉴别码对象、ECC 密钥协商等句柄。	
参数	hHandle	[IN] 要关闭的对象句柄
返回值	SAR_OK	表示成功
	其他	返回错误码，见表 A-22

g） 接口错误代码

接口函数返回的错误代码定义如表B.22所示。

表 B. 22 接口错误代码表

宏描述	预定义值	说明
SAR_OK	0X00000000	成功
SAR_FAIL	0X0A000001	失败
SAR_UNKNOWNERR	0X0A000002	异常错误
SAR_NOTSUPPORTYETERR	0X0A000003	不支持的服务
SAR_FILEERR	0X0A000004	文件操作错误
SAR_INVALIDHANDLEERR	0X0A000005	无效的句柄
SAR_INVALIDPARAMERR	0X0A000006	无效的参数
SAR_READFILEERR	0X0A000007	读文件错误
SAR_WRITEFILEERR	0X0A000008	写文件错误
SAR_NAMELENERR	0X0A000009	名称长度错误
SAR_KEYUSAGEERR	0X0A00000A	密钥用途错误
SAR_MODULUSLENERR	0X0A00000B	模的长度错误
SAR_NOTINITIALIZEERR	0X0A00000C	未初始化
SAR_OBJERR	0X0A00000D	对象错误
SAR_MEMORYERR	0X0A00000E	内存错误
SAR_TIMEOUTERR	0X0A00000F	超时
SAR_INDATALENERR	0X0A000010	输入数据长度错误
SAR_INDATAERR	0X0A000011	输入数据错误
SAR_GENRANDERR	0X0A000012	生成随机数错误
SAR_HASHOBJERR	0X0A000013	HASH 对象错
SAR_HASHERR	0X0A000014	HASH 运算错误
SAR_GENRSAKEYERR	0X0A000015	产生 RSA 密钥错
SAR_RSAMODULUSLENERR	0X0A000016	RSA 密钥模长错误
SAR_CSPIMPRTPUBKEYERR	0X0A000017	CSP 服务导入公钥错误
SAR_RSAENCERR	0X0A000018	RSA 加密错误

SAR_RSADECERR	0X0A000019	RSA 解密错误
SAR_HASHNOTEQUALERR	0X0A00001A	HASH 值不相等
SAR_KEYNOTFOUNTERR	0X0A00001B	密钥未发现
SAR_CERTNOTFOUNTERR	0X0A00001C	证书未发现
SAR_NOTEXPORTERR	0X0A00001D	对象未导出
SAR_DECRYPTPADERR	0X0A00001E	解密时做补丁错误
SAR_MACLENERR	0X0A00001F	MAC 长度错误
SAR_BUFFER_TOO_SMALL	0x0A000020	缓冲区不足
SAR_KEYINFOTYPEERR	0X0A000021	密钥类型错误
SAR_NOT_EVENTERR	0X0A000022	无事件错误
SAR_DEVICE_REMOVED	0X0A000023	设备已移除
SAR_PIN_INCORRECT	0X0A000024	PIN 不正确
SAR_PIN_LOCKED	0X0A000025	PIN 被锁死
SAR_PIN_INVALID	0X0A000026	PIN 无效
SAR_PIN_LEN_RANGE	0X0A000027	PIN 长度错误
SAR_USER_ALREADY_LOGGED_IN	0X0A000028	用户已经登录
SAR_USER_PIN_NOT_INITIALIZED	0X0A000029	没有初始化用户口令
SAR_USER_TYPE_INVALID	0X0A00002A	PIN 类型错误
SAR_APPLICATION_NAME_INVALID	0X0A00002B	应用名称无效
SAR_APPLICATION_EXISTS	0X0A00002C	应用已经存在
SAR_USER_NOT_LOGGED_IN	0X0A00002D	用户没有登录
SAR_APPLICATION_NOT_EXISTS	0X0A00002E	应用不存在
SAR_FILE_ALREADY_EXIST	0X0A00002F	文件已经存在
SAR_NO_ROOM	0X0A000030	空间不足
SAR_FILE_NOT_EXIST	0X0A000031	文件不存在
SAR_REACH_MAX_CONTAINER_COUNT	0X0A000031	已达到最大可管理容器数