



中华人民共和国劳动和劳动安全行业标准

LD/T 30.5—2009

人力资源和社会保障电子认证体系 第 5 部分:证书载体规范

Human resources and social security electronic authentication system—
Part 5: Specification of digital certificate storage medium

2009-12-14 发布

2010-03-01 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 I

1 范围 1

2 规范性引用文件 1

3 术语和定义 1

4 缩略语 3

5 证书载体硬件规范 3

 5.1 基本技术要求 3

 5.2 管理要求 4

 5.3 安全机制 5

6 证书载体软件规范 5

 6.1 应用接口 5

 6.2 安装与卸载 9

附录 A（资料性附录） 证书载体接口函数规范 11

附录 B（资料性附录） 证书载体外观 43

前 言

为适应人力资源和社会保障信息化发展要求,满足人力资源和社会保障网络信任体系建设和管理的需要,人力资源和社会保障部组织并制定了 LD/T 30—2009《人力资源和社会保障电子认证体系》。

网络信任体系包括电子认证体系、授权管理体系和责任认定体系,本标准主要描述了人力资源和社会保障电子认证体系相关内容,包括以下五个部分:

- 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:证书及证书撤销列表格式规范;
- 第 4 部分:证书应用管理规范;
- 第 5 部分:证书载体规范。

本部分为 LD/T 30—2009 的第 5 部分。

本部分主要描述了证书载体的技术指标,包括硬件规范和软件规范,并规定了证书载体的相关接口和外观规范。

本部分重点引用了《智能 IC 卡及智能密码钥匙密码应用接口规范》,并在此基础上,扩展了证书载体基本技术要求、证书载体管理要求、软件的安装卸载以及证书载体外观设计等相关内容,从满足人力资源社会保障业务需求的角度,对本行业发放的证书载体的软硬件和外观提出规范和要求。

本部分由中华人民共和国人力资源和社会保障部信息中心提出并归口。

本部分主要起草单位:中华人民共和国人力资源和社会保障部信息中心、上海市人力资源和社会保障局信息中心、北京数字证书认证中心、维豪信息技术有限公司。

本部分主要起草人:赵锡铭、戴瑞敏、贾怀斌、翟燕立、李丽虹、吴问滨、黄勇、吕丽娟、许华光、罗震、张加会、靳朝晖、陆春生、李永亮、宋京燕、李冰松、耿建军、杜守国、欧阳晋、林雪焰、李述胜、顾青、宋成。

本部分凡涉及密码相关内容,均按国家有关法规实施。

人力资源和社会保障电子认证体系

第 5 部分：证书载体规范

1 范围

LD/T 30 的本部分描述了在人力资源和社会保障系统中使用的证书载体的各项要求,包括硬件要求、软件要求、管理要求、安全机制以及相关接口标准等内容。

本部分适用于人力资源社会保障证书载体的设计、应用开发、使用和检测。

2 规范性引用文件

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

GM 0001—2005 证书认证系统密码及其相关安全技术规范

信息技术 安全技术 密码术语(国家密码管理局)

智能 IC 卡及智能密码钥匙密码应用接口规范(国家密码管理局)

3 术语和定义

以下术语和定义适用于本规范。

3.1

证书载体 certificate entity

用于存储密钥和数字证书并具有密码运算功能的载体,包括智能密码钥匙(USBKey)和 IC 卡等。

3.2

容器 container

特指密钥容器,是一个用于存放非对称密钥对和证书的逻辑对象。每个用户对应一个密钥容器,与用户相关的非对称密钥对和证书存放于该密钥容器,每个容器中最多可以存放一对加密密钥对和一对签名密钥对以及一张加密证书和一张签名证书。

3.3

证书撤销列表 certificate revocation list

CRL

标记一系列不再被证书发布者认为有效的证书的签名列表。

3.4

数字证书 digital certificate

由权威认证机构进行数字签名的包含公开密钥拥有者信息、公开密钥、签发者信息、有效期以及一些扩展信息的数字文件。

3.5

加密 encrypt

通过密码算法对数据进行变换来产生密文,以便隐藏数据的信息内容。

3.6

解密 decrypt

与一个可逆的加密过程相对应的反过程。该过程使用适当的密钥,将已加密的文本转换成明文。

3.7

数字签名 digital signature

附加在数据上的签名数据,或是对数据所作的密码变换,用以确认数据来源及其完整性,防止被人(例如接受者)进行伪造。

3.8

加密密钥对 exchange key pair

用来对会话密钥进行加密和解密的公/私密钥对,使用加密密钥对的公钥加密会话密钥,加密后的会话密钥传给接收者,接收者使用加密密钥对的私钥解密出会话密钥。

3.9

密钥交换 key exchange

通信实体间交换密钥的过程。

3.10

消息鉴别码 message authentication code

MAC

又称消息认证码,是消息鉴别算法的输出。

3.11

公钥 public key

在公钥密码体制中,用户密钥对中公布给其他用户的密钥。

3.12

私钥 private key

在公钥密码体制中,用户密钥对中仅为该用户持有的密钥。

3.13

RSA 算法 rivest-shamir-adleman algorithm

一种基于大整数因子分解问题的公钥密码算法。

3.14

会话密钥 session key

处于层次化密钥结构中的最低层,仅在一次会话中使用的密钥。

3.15

签名密钥对 signature key pair

用来对消息进行数字签名的公/私密钥对,用于消息鉴别。

3.16

公开密钥基础设施 public key infrastructure

PKI

用公钥密码技术建立的普遍适用的基础设施,为用户提供证书管理和密钥管理等安全服务。

3.17

信任 trust

通常,当一个实体(第一个实体)假设另一个实体(第二个实体)完全按照第一个实体的期望行动时,则称第一个实体“信任”第二个实体。这种“信任”可能只适用于某些特定功能。本框架中“信任”的关键作用是描述鉴别实体和认证机构之间的关系;鉴别实体应确信它能够“信任”认证机构仅创建有效且可靠的证书。

4 缩略语

下列缩略语适用于本部分：

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)
CSP	加密服务提供者(Cryptographic Service Provider)
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
PKCS# 11	公钥密码使用标准系列规范中的第 11 部分,为执行密码函数的设备确定了一种程序设计接口

5 证书载体硬件规范

5.1 基本技术要求

证书载体的基本技术要求如表 1 所示：

表 1 证书载体基本技术要求

名 称	要 求	备 注
存储容量	≥32 K Bytes (32 K 型)	
CPU 芯片位数	≥8 位	
功耗	<400 mW	
国际标准	a) ISO 7816-4 b) PKCS# 11	
微软标准	a) Microsoft CryptoAPI b) Cryptographic Service Provider	
编程接口	a) Microsoft CryptoAPI b) PKCS# 11 API	
证书格式	符合 GB/T 20518—2006《信息安全技术 公钥基础设施 数字证书格式》	
硬件接口	符合 USB 接口规范,不需额外插电	USB1.1 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 ℃	

表 1 证书载体基本技术要求（续）

名 称	要 求	备 注
湿度要求	10%~90%	
室温下数据保存时间	至少 50 年	
工作电压	2.7 V~5.5 V	
适用浏览器	IE6.0 以上各种版本	
适用操作系统	Windows XP, window2000, windows2003, Linux, Vista 及以上版本	支持简体中文、繁体中文、英文
非对称算法	遵循国家相关规定,密钥对必须在芯片中生成,且私钥不可导出	如 RSA、SM2 等
对称算法	遵循国家相关规定	如 SSF33、SM1 等
杂凑算法	遵循国家相关规定	如 SHA-1、SHA256 等
公钥私钥对生成时间	≤30 s	
数字签名和验证时间	<1 s/次	密钥的签名验证速度
RSA 加密速度	>50 kbit/s	
RSA 解密速度	>30 kbit/s	
存储要求	a) 公私钥对:≥2 个 b) 数字证书:≥2 张 c) 扩展区文件:≥10 K	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	

5.2 管理要求

5.2.1 证书载体初始化

证书载体的初始化就是对证书载体进行区间划分,使证书载体按照相关规范进行初始化。
初始化工具由各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。

5.2.2 证书载体的安装注册

证书载体内的证书可实现自动注册,即当证书载体插入电脑后,载体内证书自动注册到操作系统“MY”区,拔出证书载体以后,数字证书自动从“MY”区中删除。

5.2.3 口令管理

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。

口令长度为 6~16 位。

5.2.4 锁死与解锁

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。

解锁口令的长度为 6~16 位。

5.2.5 扩展区要求

证书载体内可创建用户文件区,要求可以在证书载体内保存用户文件,以实现某些扩展应用。接口应包括:对用户私有文件区的创建、删除和读写操作等功能接口。用户在对扩展区操作时,应输入证书载体的用户 PIN 码验证。扩展区内的数据是否需要加密,根据应用需求而定。

扩展区的文件长度不小于 10 K。

5.2.6 其他

同一个终端可以同时使用多个证书载体,以微软标准 CSP 接口调用证书载体设备时,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。

5.3 安全机制

5.3.1 基本安全要求

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。

证书载体必须能产生 RSA 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。

5.3.2 密钥和密码的存放

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,非对称密钥的公钥必须在校验用户密码后方可导出。

6 证书载体软件规范

6.1 应用接口

6.1.1 CSP 接口

CSP 主要函数如下:

a) 服务提供者函数,用于连接或断开 CSP:

```
CryptAcquireContext
CryptContextAddRe
CryptEnumProviders
CryptEnumProviderTypes
CryptGetProviderParam
CryptGetDefaultProvider
```

- CryptGetProvParam
- CryptInstallDefaultContext
- CryptReleaseContext
- CryptSetProvider
- CryptSetProviderEx
- CryptSetProvParam
- CryptUninstallDefaultContext
- b) 密钥产生和交换函数,用于密钥的产生和交换,包括产生、配置和销毁:
 - CPDeriveKey
 - CPDestroyKey
 - CPDuplicateKey
 - CPEExportKey
 - CPGenKey
 - CPGenRandom
 - CPGetKeyParam
 - CPGetUserKey
 - CPImportKey
 - CPSetKeyParam
- c) 对象编码和解码函数,用于对证书、CRL 等进行编码和解码工作:
 - CryptDecodeObject
 - CryptDecodeObjectEx
 - CryptEncodeObject
 - CryptEncodeObjectEx
- d) 数据加密、解密函数,用于数据的加密、解密操作:
 - CryptDecrypt
 - CryptEncrypt
 - CryptProtectData
 - CryptProtectMemory
 - CryptUnprotectData
 - CryptUnprotectMemory
- e) 哈希和数字签名函数,用于计算数据的哈希值、创建和验证数字签名:
 - CryptCreateHash
 - CryptDestroyHash
 - CryptDuplicateHash
 - CryptGetHashParam
 - CryptHashData
 - CryptHashSessionKey
 - CryptSetHashParam
 - CryptSignHash
 - CryptUIWizDigitalSign
 - CryptUIWizFreeDigitalSignContext
 - CryptVerifySignature

6.1.2 扩展接口

6.1.2.1 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、设置设备信息、获取设备信息、锁定设备、释放设备操作。使用者不必知道设备类型和具体的驱动模式,只需要通过本规范提供的接口,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	备 注
HRSS_KEY_WaitForDevEvent	等待设备插拔事件	插拔响应
HRSS_KEY_EnumDev	枚举设备	枚举支持的设备
HRSS_KEY_ConnectDev	连接设备	共享打开返回设备句柄
HRSS_KEY_DisconnectDev	断开设备	断开设备连接
HRSS_KEY_InitDevInfo	设置设备信息	设置设备信息
HRSS_KEY_GetDevState	判断设备状态	判断设备的当前状态
HRSS_KEY_GetDevInfo	获取设备信息	获取设备信息

6.1.2.2 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。

访问管理函数如表 3 所示。

表 3 访问控制函数

函数名称	功 能
HRSS_KEY_ChangePIN	修改 PIN
HRSS_KEY_VerifyPIN	校验 PIN
HRSS_KEY_UnblockPIN	解锁 PIN
HRSS_KEY_ClearSecureState	清除安全状态

6.1.2.3 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
HRSS_CreateApplication	创建应用
HRSS_EnumApplication	枚举应用
HRSS_DeleteApplication	删除应用
HRSS_OpenApplication	打开应用
HRSS_CloseApplication	关闭应用

6.1.2.4 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。

文件管理函数如表 5 所示。

表 5 文件管理函数

函数名称	功 能
HRSS_KEY_CreateFile	创建文件函数
HRSS_KEY_DeleteFile	删除文件函数
HRSS_KEY_EnumFiles	枚举文件函数
HRSS_KEY_ReadFile	读文件函数
HRSS_KEY_WriteFile	写文件函数

6.1.2.5 密码服务

密码服务函数用于密码运算服务,包括密钥容器的创建、销毁;密钥的生成、导入、导出、加密解密、签名验证等,会话密钥支持国产算法如 SSF33、SM1,非对称密钥目前支持 1024 和 2048 位 RSA。

密码服务函数如表 6 所示。

表 6 密码服务函数

函数名称	功 能
HRSS_KEY_CreateContainer	创建容器函数
HRSS_KEY_DestroyContainer	销毁容器函数
HRSS_KEY_EnumContainer	枚举容器函数
HRSS_KEY_GetContainerHandle	获取容器句柄函数
HRSS_KEY_GenRandom	生成随机数函数
HRSS_KEY_GenSessionKey	生成会话密钥函数
HRSS_KEY_GenRSAKeyPair	生成 RSA 公私钥对函数

表 6 密码服务函数（续）

函数名称	功 能
HRSS_KEY_GetAsymmetricKeyHandle	获取非对称密钥对句柄函数
HRSS_KEY_ReleaseAsymmetricKeyHandle	释放非对称密钥对句柄函数
HRSS_KEY_ExportSessionKey	导出会话密钥函数
HRSS_KEY_ExportRSAPublicKey	导出 RSA 公钥函数
HRSS_KEY_ImportSessionKey	导入会话密钥函数
HRSS_KEY_ImportRSAKeyPair	导入 RSA 公私钥对函数
HRSS_KEY_ImportRSAPublicKey	导入 RSA 公钥函数
HRSS_KEY_ImportRSAPrivateKey	导入 RSA 私钥函数
HRSS_KEY_GetKeyParam	取密钥参数函数
HRSS_KEY_SetKeyParam	设置密钥参数函数
HRSS_KEY_RSASignData	RSA 数据签名函数
HRSS_KEY_RSASVerify	RSA 验签函数
HRSS_KEY_RSASEncrypt	RSA 加密函数
HRSS_KEY_GenTempRSAKeyPair	生成临时 RSA 公私钥对函数
HRSS_KEY_TempRSASEncrypt	临时 RSA 密钥加密函数
HRSS_KEY_TempRSADecrypt	临时 RSA 密钥解密函数
HRSS_KEY_EncryptInit	加密初始化函数
HRSS_KEY_Encrypt	单组数据加密函数
HRSS_KEY_EncryptUpdate	多组数据加密函数
HRSS_KEY_EncryptFinal	加密结束函数
HRSS_KEY_DecryptInit	解密初始化函数
HRSS_KEY_Decrypt	单组数据解密函数
HRSS_KEY_DecryptUpdate	多组数据解密函数
HRSS_KEY_DecryptFinal	解密结束函数
HRSS_KEY_DigestInit	杂凑初始化函数
HRSS_KEY_Digest	单组数据杂凑函数
HRSS_KEY_DigestUpdate	多组数据杂凑函数

证书载体的扩展接口的描述见附录 A。

6.2 安装与卸载

6.2.1 安装程序要求

a) 基本要求

- 1) 同一型号的 USBKey 安装程序应将驱动程序、CSP、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 Windows 操作系统版本并自动安装相应的兼容性插件。
 - 3) USBKey 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) USBKey 安装程序应兼容 Windows 2000、Windows XP、Vista 及其以上的系列操作系统,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) 管理工具应具备校验口令的功能。

6.2.2 卸载程序要求

- a) 基本要求
- 1) 卸载程序应兼容 Windows 2000、Windows XP、Vista 及其以上的系列操作系统。
 - 2) 卸载程序不需要客户干预,能自动完成卸载。
 - 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 CSP 和驱动程序卸载。
 - 3) 通过控制面板中的“添加/删除程序”,能将 CSP 和驱动程序卸载。
 - 4) 再次运行安装程序能自动卸载。

6.2.3 驱动程序的兼容性要求

- 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 USBKey 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。

6.2.4 接口动态库命名

- a) 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和社会保障部统一管理。

附录 A
(资料性附录)
证书载体接口函数规范

A.1 数据类型定义

表 A.1 数据类型

类型名称	描 述	定 义
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) 版本数据类型定义:

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

表 A.2 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	

b) 设备初始化信息设备类型:

```
typedef struct Struct_DEVINITINFO{
```

```
VERSION AppVersion;  
CHAR Manufacturer[64];  
CHAR Label[64];  
}DEVINITINFO, * PDEVINITINFO;
```

表 A.3 设备初始化信息数据类型数据项描述

数据项	类 型	意 义	备 注
AppVersion	VERSION	SIC/SZD 接口规范版本	
Manufacturer	CHAR 数组	设备厂商信息	最长 64 个字符,不足 64 个字符以空白字符(ASCII 码为 0xFF)填充,不能以 null(0x00)结束。
Label	CHAR 数组	SIC/SZD 标签	最长 64 个字符,不足 64 个字符以空白字符(ASCII 码为 0xFF)填充,不能以 null(0x00)结束。

c) 设备信息

1) 类型定义

```
typedef struct Struct_DEVINFO{  
    CHAR Manufacturer[64];  
    CHAR Label[64];  
    CHAR SerialNumber[32];  
    VERSION HWVersion;  
    VERSION FirmwareVersion;  
    VERSION AppVersion;  
    ULONG Type;  
    BYTE MinPINLen;  
    BYTE MaxPINLen;  
    ULONG AlgID;  
    ULONG Reserved;  
}DEVINFO, * PDEVINFO;
```

2) 数据项描述

表 A.4 设备信息数据类型数据项描述

数据项	类 型	意 义	备 注
Manufacturer	CHAR 数组	设备厂商信息	
Label	CHAR 数组	设备标签	
SerialNumber	CHAR 数组	序列号	
HWVersion	VERSION	设备硬件版本	
FirmwareVersion	VERSION	设备本身固件版本	
AppVersion	VERSION	支持设备的应用版本	
Type	ULONG	设备类型	0:USBKey,1:IC CARD
MinPINLen	BYTE	最小 PIN 长度	

表 A.4 设备信息数据类型数据项描述 (续)

数据项	类 型	意 义	备 注
MaxPINLen	BYTE	最大 PIN 长度	
AlgID	ULONG	支持的国产算法标识	见表 A.11
Reserved	ULONG	保留扩展	

d) 双向认证密钥的密钥头

1) 类型定义

```
typedef struct Struct_KEYHEAD{
    UINT8 ApplicationType;
    UINT8 KeyLen;
    UINT8 ErrorCounter;
}KEYHEAD, * PKEYHEAD;
```

2) 数据项描述

表 A.5 双向认证密钥的密钥头数据类型数据项描述

数据项	类 型	意 义	备 注
ApplicationType	UINT8	认证密钥的应用类型：外部认证密钥，内部认证密钥	内部认证密钥： AT_INTERNAL_AUTHENTICATE_KEY 外部认证密钥： AT_EXTERNAL_AUTHENTICATE_KEY
KeyLen	UINT8	密钥值长度，按字节计算	
ErrorCounter	UINT8	错误计数器，高 4 位为最大错误次数，低 4 位为剩余尝试次数	

e) 算法信息

1) 类型定义

```
typedef struct Struct_ALGINFO{
    ULONG MinKeyLen;
    ULONG MaxKeyLen;
    ULONG AlgFlags;
}ALGINFO, * PALGINFO;
```

2) 类型定义

表 A.6 算法信息数据类型数据项描述

数据项	类 型	意 义	备 注
KeyMinLen	ULONG	密钥最小长度	
KeyMaxLen	ULONG	密钥最大长度	
AlgFlags	ULONG	算法特性标志	见表 A.7 定义

表 A.7 算法特性标志定义

算法特性标志(AlgFlags)			
位	标 识	掩 码	意 义
1	AF_HW	0X00000001	1:算法由硬件执行 0:算法由软件执行
9	AF_ENCRYPT	0X00000100	1:算法能被用于 HRSS_KEY_EncryptInit 函数 0:算法不能被用于 HRSS_KEY_EncryptInit 函数
10	AF_DECRYPT	0X00000200	1:算法能被用于 HRSS_KEY_DecryptInit 函数 0:算法不能被用于 HRSS_KEY_DecryptInit 函数
11	AF_DIGEST	0X00000400	1:算法能被用于 HRSS_KEY_DigestInit 函数 0:算法不能被用于 HRSS_KEY_DigestInit 函数
12~32			扩展用

f) RSA 公钥交换数据块

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

表 A.8 RSA 公钥交换数据块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 $n=p \cdot q$	实际长度为 BitLen/8 字节 # define MAX_RSA_MODULUS_LEN 256 # define MAX_RSA_EXPONENT_LEN 6
PublicExponent	ULONG	公开密钥 e	一般为 65537

g) RSA 私钥交换数据块

1) 类型定义

```
typedef struct Struct_RSAPRIVATEKEYBLOB {
    ULONG AlgID;
    ULONG BitLen;
    BYTE  Modulus[MAX_RSA_MODULUS_LEN];
```

```
ULONG PublicExponent;
BYTE PrivateExponent[MAX_RSA_MODULUS_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。
```

2) 数据项描述

表 A.9 RSA 私钥交换数据块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 $n=p \cdot q$	实际长度为 BitLen/8 字节
PublicExponent	ULONG	公开密钥 e	一般为 65537
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

h) 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

表 A.10 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.19
WriteRights	ULONG	写入权限	写入文件需要的权限

i) 算法标识号

算法标识号 AlgID 的取值可以是以下常量：

表 A. 11 算法标识号

对称算法标识		
宏描述	预定义值	说 明
ALG_SSF33	0X00000001	SSF33 算法
ALG_SM1	0X00000002	SM1 通用算法
ALG_SM1_S	0X00000004	SM1 专用算法
非对称算法标识		
宏描述	预定义值	说 明
ALG_RSA1024	0X00000008	指定使用 1 024 位的 RSA,模长在函数的长度中指定
ALG_RSA2048	0X00000010	指定使用 2 048 位的 RSA,模长在函数的长度中指定
杂凑算法标识		
宏描述	预定义值	说 明
ALG_SCH	0X00000040	SCH 杂凑算法
ALG_SHA1	0X00000080	SHA1 杂凑算法
ALG_SHA256	0X00000100	SHA256 杂凑算法

j) 密钥参数类型

密钥参数类型(Key Parameter Type)的取值可以是以下常量,在获取密钥参数(HRSS_KEY_GetKeyParam)或设置密钥参数(HRSS_KEY_SetKeyParam)时指定要获取或设置的密钥参数：

表 A. 12 密钥参数类型

密钥参数类型	值	描 述
KP_ALGID	0X00000001	密钥所适应算法的标识号
KP_BLOCKLEN	0X00000002	对于会话密钥(对称密钥),是指块(分组)长度; 对于 RSA 密钥,即是模数的长度
KP_KEYLEN	0X00000003	密钥实际长度(按位计算)
KP_SALT	0X00000004	该参数类型只针对会话密钥,指会话密钥的调剂值(SaltValue)
KP_PERMISSIONS	0X00000005	密钥使用许可状态
KP_IV	0X00000006	密钥初始向量,该参数类型只针对分组加密密钥
KP_PADDING	0X00000007	填充方法,目前定义了 PKCS5_Padding 的填充方法
KP_MODE	0X00000008	加密模式,该参数类型只针对分组加密密钥;
KP_MODE_BITS	0X00000009	反馈值长度(按位计算),该参数类型只针对 OFB(outputfeedback)和 CFB(cipherfeedback)两种分组加密模式

k) 密钥使用许可

密钥使用许可(Key Permissions)可以是以下常量的组合：

表 A. 13 密钥使用许可

密钥使用许可	值	描 述
CRYPT_ENCRYPT	0X00000001	可用于加密
CRYPT_DECRYPT	0X00000002	可用于解密
CRYPT_EXPORT	0X00000004	允许被导出
CRYPT_READ	0X00000008	允许读取密钥参数
CRYPT_WRITE	0X00000010	允许写入密钥参数
CRYPT_MAC	0X00000020	允许被用于计算 MAC 值

l) 加密模式
加密模式(Cipher Modes)有以下几种：

表 A. 14 加密模式

加密模式	值	描 述
CRYPT_MODE_ECB	0X00000001	电子密码本加密模式
CRYPT_MODE_CBC	0X00000002	分组链接加密模式
CRYPT_MODE_OFB	0X00000003	输出反馈加密模式
CRYPT_MODE_CFB	0X00000004	密文反馈加密模式
CRYPT_MODE_MAX	0X00000005	MAC 加密模式

m) 分组密码填充方法

表 A. 15 分组密码填充方法

填充方法	常量标识	值	描 述
PKCS#5 填充	PKCS5_PADDING	0X00000001	补码的值为需要补码的位数。例如数据的分组长度为 16, 现在有 11 字节数据, 需要 5 个字节的补码, 则补 5 个字节的“5”

n) 密钥类型

表 A. 16 密钥类型

密钥类型	常量标识	值
公开密钥	KT_PUBLIC_KEY	0X00000001
私有密钥	KT_PRIVATE_KEY	0X00000002
对称密钥	KT_SECRET_KEY	0X00000003

o) 密钥应用类型

表 A. 17 密钥应用类型

密钥应用类型	常量标识	值	描 述
加密密钥对(非对称密钥对)	AT_KEYEXCHANGE	0X00000001	用于加密会话密钥(对称密钥),使得会话密钥可以安全地传输和存储。用公钥对会话密钥进行加密,用私钥对会话密钥进行解密
签名密钥对(非对称密钥对)	AT_SIGNATURE	0X00000002	用于数字签名,私钥用于签名,公钥用于验证签名
会话密钥(对称密钥)	AT_SESSION	0X00000003	会话时用于对信息进行加密和解密,由加密密钥对的公钥对会话密钥加密后,加密的会话密钥和加密信息一起传输
内部认证密钥(对称密钥)	AT_INTERNAL_AUTHENTICATE_KEY	0X00000004	用于终端对设备的认证
外部认证密钥(对称密钥)	AT_EXTERNAL_AUTHENTICATE_KEY	0X00000005	用于设备对终端的认证

p) 导出密钥格式

按照 TLV 格式进行输出,T:固定为 1 字节,L 固定为 2 字节,见表 A. 18:

表 A. 18 导出密钥格式

	RSA	SM2
公钥	n,e	x,y,a,b,p,q
私钥	n,e,d,p,q,P,Q,I	k,x,y,a,b,p,q

q) 设备权限类型

表 A. 19 设备权限类型

权限类型	值	说 明
SECURE_NEVER_ACCOUNT	0X00000000	不允许
SECURE_ADM_ACCOUNT	0X00000001	管理员权限
SECURE_USER_ACCOUNT	0X00000010	用户权限
SECURE_EVERYONE_ACCOUNT	0X000000FF	任何人

r) PIN 口令长度

表 A. 20 PIN 口令长度常量定义

常 量	常量标识	值	描 述
PIN 长度的最小值	MIN_PIN_LEN	0X000006	口令的最小长度,包括对管理员口令和用户口令
PIN 长度的最大值	MAX_PIN_LEN	0X000010	口令的最大长度,包括对管理员口令和用户口令

s) 文件名称长度

表 A.21 文件名称长度常量定义

常 量	常量标识	值	描 述
文件名称长度的最小值	MIN_FILE_NAME_LEN	0X000001	文件名称的最小长度,可以为 1~32 个字符
文件名称长度的最大值	MAX_FILE_NAME_LEN	0X000020	

t) 设备状态

表 A.22 设备状态常量定义

常 量	常量标识	值	描 述
设备不存在	DEV_EMPTY_STATE	0X000000	
设备存在	DEV_PRESENT_STATE	0X000001	

u) 前缀

表 A.23 变量常量前缀定义

名 称	描 述
HRSS_KEY_	API 函数接口前缀
H	句柄类型
P	指针
Sz	字符串指针
Ul	ULONG 类型
Ph	句柄指针
Pul	ULONG 指针
B	BYTE 类型
Ph	BYTE 指针

A.2 证书载体函数定义

a) 设备管理类函数

设备管理包括以下具体函数：

表 A.24 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	HRSS_KEY_WaitForDevEvent
2	枚举设备函数	HRSS_KEY_EnumDev
3	连接设备函数	HRSS_KEY_ConnectDev

表 A.24 设备管理类函数 (续)

序号	函数名称	函数定义
4	断开设备函数	HRSS_KEY_DisconnectDev
5	设置设备信息函数	HRSS_KEY_InitDevInfo
6	判断设备状态函数	HRSS_KEY_GetDevState
7	获取设备信息函数	HRSS_KEY_GetDevInfo

1) 等待设备插拔事件函数

函数原型	ULONG WINAPI HRSS_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.32

2) 枚举设备函数

函数原型	ULONG WINAPI HRSS_KEY_EnumDev (BOOL bPresent, LPSTR szNameList, ULONG * pulSize)	
功能描述	枚举设备: 该函数枚举当前驱动支持的所有设备列表或者当前状态为存在的设备,通过该函数来管理,获得当前系统中的设备列表情况。在进行连接设备前,可以通过该函数判断设备是否存在。	
参数	bPresent	[IN] 为 1 表示取当前设备状态为存在的设备列表。为 0 表示取当前驱动支持的设备列表。
	szNameList	[OUT] 设备名称列表。该参数为空,将由 pulSize 返回所需要的内存空间大小。
	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 所需要的空间大小。每个设备的名称以单个 NULL 结束,以双 NULL 表示列表的结束。
返回值	SAR_OK	表示成功
	其他	返回错误码,见表 A.32
备注	使用设备应该在访问设备前首先调用该函数。	

3) 连接设备函数

函数原型	ULONG WINAPI HRSS_KEY_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 HRSS_KEY_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. 32
备注	使用算法服务或者文件管理函数之前调用。	

4) 断开连接函数

函数原型	ULONG DEVAPI HRSS_KEY_DisConnectDev (DEVHANDLE hDev)	
功能描述	断开设备: 断开一个已经连接的设备。 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. 32

5) 设置设备信息函数

函数原型	ULONG DEVAPI HRSS_KEY_InitDevInfo (DEVHANDLE hDev, DEVINITINFO * pInfo)	
功能描述	设置初始化设备信息: 经过初始化后,设备中原来创建的对象都将被删除。	
参数	hDev	[IN] 连接设备时返回的设备句柄
	pInfo	[IN] 设备的初始化信息
返回值	SAR_OK	表示成功
	其他	返回错误码,见表 A. 32
备注	建议:对设备进行初始化之前必须执行设备的认证,认证通过才进行设备的初始化操作。	

6) 判断设备状态函数

函数原型	ULONG DEVAPI HRSS_KEY_GetDevState (LPSTR szDevName, ULONG * pulDevState)	
功能描述	判断设备状态: 判断设备是否存在于系统中,可以返回 DEV_PRESENT_STATE 或者 DEV_EMPTY_STATE。	
参数	szDevName	[IN] 连接名称
	pulDevState	[OUT] 返回的设备状态
返回值	SAR_OK	表示成功
	其他	返回错误码,见表 A. 32
	SAR_UNKNOWNERR	发生未知错误
	SAR_MEMORYERR	发生内存错误

7) 获取设备信息函数

函数原型 ULONG DEVAPI HRSS_KEY_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. 32

b) 访问控制类函数

访问控制包括以下具体函数：

表 A. 25 访问控制类函数

序号	函数名称	函数定义
1	修改 PIN 函数	HRSS_KEY_ChangePIN
2	校验 PIN 函数	HRSS_KEY_VerifyPIN
3	解锁 PIN 函数	HRSS_KEY_UnblockPIN
4	清除安全状态函数	HRSS_KEY_ClearSecureState

1) 修改 PIN 函数

函数原型 ULONG DEVAPI HRSS_KEY_ChangePIN (DEVHANDLE hDev, 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 hDev [IN] 连接时返回的设备句柄

 hApplication [IN] 应用句柄

 ulPINType [IN] PIN 类型，可以为 ADMIN_TYPE = 0，或 USER_TYPE = 1

 szOldPin [IN] 原 PIN 值

 szNewPin [IN] 新 PIN 值

 pulRetry- [OUT] 出错后重试次数

 Count

返回值 SAR_OK 表示成功

 其他 返回错误码，见表 A. 32

2) 校验 PIN 函数

函数原型 ULONG DEVAPI HRSS_KEY_VerifyPIN (DEVHANDLE hDev, HAPPLICATION hApplication, ULONG ulUserType, LPSTR szPIN, ULONG * pulRetryCount)

功能描述 校验口令，即 Login：
 校验 Admin PIN 或者 User PIN。
 校验成功后，会获得相应的权限，如果调用失败，会返回口令的重试次数，当重试次数为 0 时表示口令已经锁死。

参数	hDev	[IN] 连接设备时返回的设备句柄
	hApplication	[IN] 应用句柄
	ulUserType	[IN] 用户类型
	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. 32

3) 解锁 PIN 函数

函数原型 `ULONG DEVAPI HRSS_KEY_UnblockPIN (DEVHANDLE hDev, HAPPLICATION hApplication, LPSTR szAdminPIN, LPSTR szNewUserPIN, ULONG * pulRetryCount)`

功能描述	<p>解锁 User PIN:</p> <p>当用户的口令锁死后,通过调用该函数来解锁用户口令。</p> <p>只知道 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。</p> <p>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。</p>
------	---

参数	hDev	[IN] 连接设备时返回的设备句柄
	hApplication	[IN] 应用句柄
	szAdminPIN	[IN] 管理员 PIN
	szNewUserPIN	[IN] 普通用户新 PIN
	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. 32

4) 清除安全状态函数

函数原型 `ULONG DEVAPI HRSS_KEY_ClearSecureState (DEVHANDLE hDev, HAPPLICA-
TION hApplication)`

功能描述	清除安全状态,即 Logout; 通过调用该函数,把应用之前获得所有权限,包括校验 Admin PIN, User PIN 获得的权限全部清除掉。
------	--

参数	hDev	[IN] 连接设备时返回的设备句柄
	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.32

c) 应用管理类函数

应用管理包括以下具体函数：

表 A.26 应用管理类函数

序号	函数名称	函数定义
1	创建应用函数	HRSS_KEY_CreateApplication
2	枚举应用函数	HRSS_KEY_EnumApplication
3	删除应用函数	HRSS_KEY_DeleteApplication
4	打开应用函数	HRSS_KEY_OpenApplication
5	关闭应用函数	HRSS_KEY_CloseApplication

1) 创建应用函数

函数原型	ULONG DEVAPI HRSS_KEY_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. 32

2) 枚举应用函数

函数原型	ULONG DEVAPI HRSS_KEY_EnumApplication(DEVHANDLE hDev, LPSTR szAppName, ULONG * pulSize, ULONG * pulAppCount)	
功能描述	枚举应用: 枚举存在的所有应用。	
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
	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 所需要的空间大小。每个应用的名称以单个 NULL 结束,以双 NULL 表示列表的结束
	pulAppCount	[OUT] 应用的个数
返回值	SAR_OK	成功
	其他	返回错误码,见表 A. 32

3) 删除应用函数

函数原型	ULONG DEVAPI HRSS_KEY_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. 32

4) 打开应用函数

函数原型	ULONG DEVAPI HRSS_KEY_OpenApplication(DEVHANDLE hDev, LPSTR szAppName, HAPPLICATION * phApplication)	
功能描述	打开应用	

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. 32

5) 关闭应用函数

函数原型 ULONG DEVAPI HRSS_KEY_CloseApplication (DEVHANDLE hDev, HAPPLICA-
 TION hApplication)
功能描述 关闭应用
参数 hDev [IN]连接设备时返回的设备句柄
 hApplication [IN]应用句柄
返回值 SAR_OK 成功
 其他 返回错误码,见表 A. 32

d) 文件管理类函数
文件管理包括以下具体函数:

表 A. 27 文件管理类函数

序号	函数名称	函数定义
1	创建文件函数	HRSS_KEY_CreateFile
2	删除文件函数	HRSS_KEY_DeleteFile
3	枚举文件函数	HRSS_KEY_FnumFiles
4	读文件函数	HRSS_KEY_ReadFile
5	写文件函数	HRSS_KEY_WriteFile

1) 创建文件函数

函数原型 ULONG DEVAPI HRSS_KEY_CreateFile (HAPPLICATION hApplication, LPSTR sz-
 FileName, ULONG ulFileSize, ULONG ulReadRights, ULONG ulWriteRights)
功能描述 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. 32

2) 删除文件函数

函数原型 ULONG DEVAPI HRSS_KEY_DeleteFile (DEVHANDLE hDev, 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	hDev	[IN] 连接设备时返回的设备句柄
	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. 32
3) 枚举文件函数		
函数原型	ULONG DEVAPI HRSS_KEY_EnumFiles (DEVHANDLE hDev, 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	hDev	[IN] 连接设备时返回的设备句柄
	hApplication	[IN] 应用句柄
	szFileList	[OUT] 返回文件名称,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 NULL 结束,以双 NULL 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的大小
返回值	SAR_OK	表示成功
	其他	返回错误码,见表 A. 32
4) 读文件函数		
函数原型	ULONG DEVAPI HRSS_KEY_ReadFile (DEVHANDLE hDev, HAPPLICATION hApplication, LPSTR szFile, 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	hDev	[IN] 连接设备时返回的设备句柄
	hApplication	[IN] 应用句柄
	szFile	[IN] 文件名
	ulOffset	[IN] 读文件需要的偏移量
	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. 32
5) 写文件函数		
函数原型	ULONG DEVAPI HRSS_KEY_WriteFile (DEVHANDLE hDev, HAPPLICATION hApplication, LPSTR szFile, ULONG ulOffset, BYTE * pbData, ULONG ulSize, ULONG * pulLen)	
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	hDev	[IN] 连接设备时返回的设备句柄
	hApplication	[IN] 应用句柄
	szFile	[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
	pulLen	[OUT] 实际写入数据大小
返回值	SAR_OK	表示成功
	其他	返回错误码,见表 A. 32

e) 密码服务类函数
密码服务包括以下具体函数：

表 A. 28 密码服务类函数

序号	函数名称	函数定义
1	创建容器函数	HRSS_KEY_CreateContainer
2	销毁容器函数	HRSS_KEY_DestroyContainer
3	枚举容器函数	HRSS_KEY_EnumContainer
4	获取容器句柄函数	HRSS_KEY_GetContianerHandle
5	生成随机数函数	HRSS_KEY_GenRandom
6	生成会话密钥函数	HRSS_KEY_GenSessionKey
7	生成 RSA 公私钥对函数	HRSS_KEY_GenRSAKeyPair
8	获取非对称密钥对句柄函数	HRSS_KEY_GetAsymmetricKeyHandle
9	释放非对称密钥对句柄函数	HRSS_KEY_ReleaseAsymmetricKeyHandle
10	导出会话密钥函数	HRSS_KEY_ExportSessionKey
11	导出 RSA 公钥函数	HRSS_KEY_ExportRSAPublicKey
12	导入会话密钥函数	HRSS_KEY_ImportSessionKey
13	导入 RSA 公私钥对函数	HRSS_KEY_ImportRSAKeyPair
14	导入 RSA 公钥函数	HRSS_KEY_ImportRSAPublicKey
15	导入 RSA 私钥函数	HRSS_KEY_ImportRSAPrivateKey
16	取密钥参数函数	HRSS_KEY_GetKeyParam
17	设置密钥参数函数	HRSS_KEY_SetKeyParam
18	RSA 数据签名函数	HRSS_KEY_RSASignData
19	RSA 验签函数	HRSS_KEY_RSAVerify
20	RSA 加密函数	HRSS_KEY_RSAEncrypt
21	生成临时 RSA 公私钥对函数	HRSS_KEY_GenTempRSAKeyPair
22	临时 RSA 密钥加密函数	HRSS_KEY_TempRSAEncrypt
23	临时 RSA 密钥解密函数	HRSS_KEY_TempRSADecrypt
24	加密初始化函数	HRSS_KEY_EncryptInit

表 A.28 密码服务类函数 (续)

序号	函数名称	函数定义
25	单组数据加密函数	HRSS_KEY_Encrypt
26	多组数据加密函数	HRSS_KEY_EncryptUpdate
27	加密结束函数	HRSS_KEY_EncryptFinal
28	解密初始化函数	HRSS_KEY_DecryptInit
29	单组数据解密函数	HRSS_KEY_Decrypt
30	多组数据解密函数	HRSS_KEY_DecryptUpdate
31	解密结束函数	HRSS_KEY_DecryptFinal
32	杂凑初始化函数	HRSS_KEY_DigestInit
33	单组数据杂凑函数	HRSS_KEY_Digest
34	多组数据杂凑函数	HRSS_KEY_DigestUpdate
35	杂凑结束函数	HRSS_KEY_DigestFinal

1) 创建容器函数

函数原型 ULONG DEVAPI HRSS_KEY_CreateContainer (DEVHANDLE hDev, HAPPLICA-
TION hApplication, LPSTR szContainer, HCONTAINER * phContainer)

创建容器 创建密钥容器。密钥容器是密钥库的一部分, 一个密钥容器包含某个特定用户的所有非
对称密钥对; 签名密钥对、加密密钥对。签名密钥对用于数字签名和验证数字签名; 加密
密钥对用于加密和解密, 通常用于加密和解密会话密钥。

创建密钥容器时, 需要为每个密钥容器指定唯一的名称, 创建成功后, 返回指向密钥
容器的指针。

参数 hDev [IN] 连接设备时返回的设备句柄
 hApplication [IN] 应用句柄
 szContainer [IN] 容器名称
 phContainer [OUT] 返回的容器句柄

返回值 SAR_OK 表示成功
 其他 返回错误码, 见表 A. 32

2) 销毁容器函数

函数原型 ULONG DEVAPI HRSS_KEY_DestroyContainer (DEVHANDLE hDev, HAPPLICA-
TION hApplication, HCONTAINER hContainer)

功能描述 销毁密钥容器。

参数 hDev [IN] 连接设备时返回的设备句柄
 hApplication [IN] 应用句柄
 hContainer [IN] 要销毁容器的句柄

返回值 SAR_OK 表示成功
 其他 返回错误码, 见表 A. 32

3) 枚举容器函数

函数原型 ULONG DEVAPI HRSS_KEY_EnumContainer (DEVHANDLE hDev, HAPPLICA-
TION hApplication, LPSTR szContainer, ULONG * pulSize)

功能描述 枚举密钥容器: 枚举出指定设备中已存在的密钥容器。

参数	hDev	[IN] 连接设备时返回的设备句柄
	hApplication	[IN] 应用句柄
	szContainer	[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. 32
4) 获取容器句柄函数		
函数原型	ULONG DEVAPI HRSS_KEY_GetContainerHandle (DEVHANDLE hDev, HAPPLICATION hApplication, LPSTR szContainerName, HCONTAINER * phContainer)	
功能描述	获取密钥容器句柄。	
参数	hDev	[IN] 连接设备时返回的设备句柄
	hApplication	[IN] 应用句柄
	szContainerName	[IN] 容器名称
	phContainer	[OUT] 返回容器的句柄
返回值	SAR_OK	表示成功
	其他	返回错误码,见表 A. 32
5) 生成随机数函数		
函数原型	ULONG DEVAPI HRSS_KEY_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. 32
6) 生成会话密钥函数		
函数原型	ULONG DEVAPI HRSS_KEY_GenSessionKey (DEVHANDLE hDev, HAPPLICATION hApplication, ULONG ulKeyLen, ULONG ulAlgID, ULONG ulGenKeyFlags, HANDLE * phKey)	
功能描述	生成会话密钥。	
参数	hDev	[IN] 连接设备时返回的设备句柄
	hApplication	[IN] 应用句柄
	ulKeyLen	[IN] 密钥长度,如果密钥长度被算法指定,则此参数无效。如果算法的密钥长度是可变的,此长度值又不符合算法密钥长度规则时,返回错误码,见表 A. 32
	ulAlgID	[IN] 密钥适用算法的标识号
	ulGenKeyFlags	[IN] 生成密钥的控制标志
	phKey	[OUT] 返回会话密钥的句柄
返回值	SAR_OK	表示成功
	其他	返回错误码,见表 A. 32
备注	不允许以明文方式导出会话密钥。	

会话密钥生成控制标志可以是以下标志位的组合(或运算):

表 A. 29 会话密钥生成控制标志定义

会话密钥生成控制标志 (ulGenKeyFlags)			
位	标 识	掩 码	意 义
1	CRYPT_EXPORTABLE	0X00000001	1:生成的密钥允许被导出; 0:生成的密钥不允许导出
2	CRYPT_CREATE_SALT	0X00000002	1:给生成的密钥加上一个随机的调剂值(Salt Value); 0:给生成的密钥加上为 0 的调剂值
3	CRYPT_NO_SALT	0X00000004	1:对于 40 位的对称密钥,不加调剂值; 0:可以加调剂值
4	CRYPT_USER_PROTECTED	0X00000008	1:对生成的密钥设置了用户保护,违反规则的密钥操作会失败,并提示用户; 0:对生成的密钥不设置用户保护
5	CRYPT_PREGEN	0X00000000	0:不指定初始密钥生成
6	CRYPT_ONDEVICE	0X00000020	1:生成在设备中固定的会话密钥; 0:生成临时会话密钥
7~32	保留		

7) 生成 RSA 公私钥对函数

函数原型	ULONG DEVAPI HRSS_KEY_GenRSAKeyPair (DEVHANDLE hDev, HAPPLICATION hApplication, HCONTAINER hContainer, ULONG ulKeyUsage, ULONG ulBitLen, ULONG ulGenKeyFlags, HANDLE * phRSAKey)		
功能描述	生成 RSA 密钥对。		
参数	hDev	[IN]	连接设备时返回的设备句柄
	hApplication	[IN]	应用句柄
	hContainer	[IN]	密钥容器句柄
	ulKeyUsage	[IN]	密钥用途,取值为 AT_KEYEXCHANGE(加密密钥),或 AT_SIGNATURE(签名密钥)
	ulBitLen	[IN]	公私钥对的模数位长
	ulGenKeyFlags	[IN]	生成密钥对的控制标志
	phRSAKey	[OUT]	返回的 RSA 公私钥对句柄
返回值	SAR_OK		表示成功
	其他		返回错误码,见表 A. 32

非对称密钥生成控制标志可以是以下标志位的组合(或运算):

表 A. 30 非对称密钥生成控制标志定义

非对称密钥生成控制标志 (ulGenKeyFlags)			
位	标 识	掩 码	意 义
1	CRYPT_USER_PROTECTED	0X00000001	1:对生成的密钥设置用户使用权限,违反规则的密钥操作会失败,并提示用户; 0:对生成的密钥不设置用户使用权限
2~32	保留		

8) 获取非对称密钥对的句柄函数

函数原型 `ULONG DEVAPI HRSS_KEY_GetAsymmetricKeyHandle (DEVHANDLE hDev, HAPPLICATION hApplication, HCONTAINER hContainer, ULONG ulKeyUsage, HANDLE * phAsymmetricKey)`

功能描述 获取非对称密钥对的公钥或私钥句柄。

参数

<code>hDev</code>	[IN] 连接设备时返回的设备句柄
<code>hApplication</code>	[IN] 应用句柄
<code>hContainer</code>	[IN] 密钥容器句柄
<code>ulKeyUsage</code>	[IN] 密钥对类型, 签名/加密
<code>phAsymmetricKey</code>	[OUT] 返回的密钥句柄

返回值

<code>SAR_OK</code>	表示成功
其他	返回错误码, 见表 A. 32

9) 释放非对称密钥对句柄函数

函数原型 `ULONG DEVAPI HRSS_KEY_ReleaseAsymmetricKeyHandle (DEVHANDLE hDev, HAPPLICATION hApplication, HCONTAINER hContainer, HANDLE * phAsymmetricKey)`

功能描述 释放非对称密钥对的公钥或私钥句柄。

参数

<code>hDev</code>	[IN] 连接设备时返回的设备句柄
<code>hApplication</code>	[IN] 应用句柄
<code>hContainer</code>	[IN] 密钥容器句柄
<code>phAsymmetricKey</code>	[IN] 要释放的密钥句柄

返回值

<code>SAR_OK</code>	表示成功
其他	返回错误码, 见表 A. 32

10) 导出会话密钥函数

函数原型 `ULONG DEVAPI HRSS_KEY_ExportSessionKey (DEVHANDLE hDev, HAPPLICATION hApplication, HANDLE hKey, HANDLE hWrapKey, BYTE * pbData, ULONG * pulDataLen)`

功能描述 导出会话密钥。

参数

<code>hDev</code>	[IN] 连接设备时返回的设备句柄
<code>hApplication</code>	[IN] 应用句柄
<code>hkey</code>	[IN] 被导出的密钥
<code>hWrapKey</code>	[IN] 用来导出密钥的密钥句柄
<code>pbData</code>	[OUT] 密钥导出的数据地址
<code>pulDataLen</code>	[IN, OUT] 返回导出数据长度

返回值

<code>SAR_OK</code>	表示成功
其他	返回错误码, 见表 A. 32

11) 导出 RSA 公钥函数

函数原型 `ULONG DEVAPI HRSS_KEY_ExportRSAPublicKey (DEVHANDLE hDev, HAPPLICATION hApplication, HCONTAINER hContainer, HANDLE hKey, BYTE * pbData, ULONG * pulDataLen)`

功能描述 导出 RSA 公钥。

参数

<code>hDev</code>	[IN] 连接设备时返回的设备句柄
<code>hApplication</code>	[IN] 应用句柄

	hContainer	[IN] 容器句柄
	hKey	[IN] 被导出的密钥
	pbData	[OUT] 密钥导出的数据地址
	pulDataLen	[IN,OUT] 返回导出数据长度
返回值	SAR_OK	表示成功
	其他	返回错误码,见表 A. 32
12) 导入会话密钥函数		
函数原型	ULONG DEVAPI HRSS_KEY_ImportSessionKey (DEVHANDLE hDev, HAPPLICA- TION hApplication, HANDLE hWrapKey, ULONG ulAlgID, BYTE * pbWrappedData, ULONG ulWrappedLen, HANDLE * phKey)	
功能描述	导入会话密钥。	
参数	hDev	[IN] 连接时返回的设备句柄
	hApplication	[IN] 应用句柄
	hWrapKey	[IN] 用来导入密钥的密钥句柄
	ulAlgID	[IN] 密钥算法标识
	pbWrappedData	[IN] 要导入的数据
	ulWrappedLen	[IN] 数据长度
	phKey	[OUT] 返回的密钥句柄
返回值	SAR_OK	表示成功
	其他	返回错误码,见表 A. 32
13) 导入 RSA 公私钥对函数		
函数原型	ULONG DEVAPI HRSS_KEY_ImportRSAKeyPair (DEVHANDLE hDev, HAPPLI- CATION hApplication, HANDLE hWrapKey, HANDLE hContainer, ULONG ulAlgID, BYTE * pbWrappedData, ULONG ulWrappedLen, HANDLE * phRSAKey)	
功能描述	导入 RSA 公私钥对。	
参数	hDev	[IN] 连接设备时返回的设备句柄
	hApplication	[IN] 应用句柄
	hWrapKey	[IN] 用来导入密钥的密钥句柄
	hContainer	[IN] 密钥容器句柄
	ulAlgID	[IN] 密钥用途,加密密钥
	pbWrappedData	[IN] 要导入的数据内存地址
	ulWrappedLen	[IN] 数据长度
	phRSAKey	[OUT] 返回 RSA 公私钥对句柄
返回值	SAR_OK	表示成功
	其他	返回错误码,见表 A. 32
14) 导入 RSA 公钥函数		
函数原型	ULONG DEVAPI HRSS_KEY_ImportRSAPublicKey (DEVHANDLE hDev, HAPPLI- CATION hApplication, HANDLE hContainer, BYTE * pbWrappedData, ULONG ul- WrappedLen, HANDLE * phKey)	
功能描述	导入 RSA 公钥。	
参数	hDev	[IN] 连接设备时返回的设备句柄
	hApplication	[IN] 应用句柄
	hContainer	[IN] 密钥容器句柄

	pbWrappedData	[IN] 指向导入的数据缓冲区的指针
	ulWrappedLen	[IN] 导入数据的长度
	phKey	[OUT] 导入密钥返回的密钥句柄
返回值	SAR_OK	表示成功
	其他	返回错误码,见表 A. 32
15) 导入 RSA 私钥函数		
函数原型	ULONG DEVAPI HRSS_KEY_ImportRSAPrivateKey (DEVHANDLE hDev, HAPPLICATION hApplication, HANDLE hWrapKey, HANDLE hContainer, BYTE * pbWrappedData, ULONG ulWrappedLen, ULONG ulAlgID, HANDLE * phKey)	
功能描述	导入 RSA 私钥。	
参数	hDev	[IN] 设备句柄
	hApplication	[IN] 应用句柄
	hWrapKey	[IN] 用来导入密钥的密钥句柄
	hContainer	[IN] 密钥容器句柄
	pbWrappedData	[IN] 指向导入的数据缓冲区的指针
	ulWrappedLen	[IN] 导入数据的长度
	ulAlgID	[IN] 密钥用法
	phKey	[OUT] 导入密钥返回的密钥句柄
返回值	SAR_OK	表示成功
	其他	返回错误码,见表 A. 32
16) 获取密钥参数函数		
函数原型	ULONG DEVAPI HRSS_KEY_GetKeyParam (DEVHANDLE hDev, HAPPLICATION hApplication, HANDLE hContainer, HANDLE hKey, ULONG ulParam, BYTE * pbData, ULONG * pulDataLen)	
功能描述	获取密钥的参数。	
参数	hDev	[IN] 连接设备时返回的设备句柄
	hApplication	[IN] 应用句柄
	hContainer	[IN] 密钥容器句柄
	hKey	[IN] 密钥句柄
	ulParam	[IN] 密钥参数类型,指定要获取哪个密钥参数,取值参见表 A. 31 密钥参数类型定义
	pbData	[OUT] 返回的数据指针
	pulDataLen	[IN,OUT] 返回的数据长度
返回值	SAR_OK	表示成功
	其他	返回错误码,见表 A. 32

表 A. 31 密钥参数类型定义

密钥参数类型	值	描 述
KP_ALGID	0X00000001	密钥所适应算法的标识号
KP_BLOCKLEN	0X00000002	对于会话密钥(对称密钥),是指块(分组)长度; 对于 RSA 密钥,即是模数的长度
KP_KEYLEN	0X00000003	密钥实际长度(按位计算)

表 A. 31 密钥参数类型定义 (续)

密钥参数类型	值	描 述
KP_SALT	0X00000004	该参数类型只针对会话密钥,指会话密钥的调剂值(Salt Value)
KP_PERMISSIONS	0X00000005	密钥使用许可状态
KP_IV	0X00000006	密钥初始向量,该参数类型只针对分组加密密钥
KP_PADDING	0X00000007	填充方法,目前定义了 PKCS5_Padding 的填充方法
KP_MODE	0X00000008	加密模式,该参数类型只针对分组加密密钥
KP_MODE_BITS	0X00000009	反馈值长度(按位计算),该参数类型只针对 OFB(output feedback)和 CFB(cipher feedback)两种分组加密模式
KP_CERTIFICATE		

17) 设置密钥参数函数

函数原型	ULONG DEVAPI HRSS_KEY_SetKeyParam (DEVHANDLE hDev, HAPPLICATION hApplication, HANDLE hContainer, HANDLE hKey, ULONG ulParam, BYTE * pbData, ULONG ulDataLen)	
功能描述	设置密钥的参数。	
参数	hDev	[IN] 连接设备时返回的设备句柄
	hApplication	[IN] 应用句柄
	hContainer	[IN] 密钥容器句柄
	hKey	[IN] 密钥句柄
	ulParam	[IN] 密钥参数类型
	pbData	[IN] 指向数据缓冲区的指针
	ulDataLen	[IN] pbData 对应的数据长度
返回值	SAR_OK	表示成功
	其他	返回错误码,见表 A. 32

18) RSA 签名函数

函数原型	ULONG DEVAPI HRSS_KEY_RSASignData (DEVHANDLE hDev, HAPPLICATION hApplication, HANDLE hKey, ULONG ulHashAlgID, BYTE * pbData, ULONG ulDataLen, BYTE * pbSignature, ULONG * pulSigLen)	
功能描述	RSA 数字签名。采用 RSA 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 pbSignature 缓冲区,设置 pulSigLen 为签名的长度。	
参数	hDev	[IN] 连接设备时返回的设备句柄
	hApplication	[IN] 应用句柄
	hKey	[IN] 用来签名的私钥句柄
	ulHashAlgID	[IN] 杂凑算法标识
	pbData	[IN] 被签名的数据,该数据为已经杂凑运算后的杂凑值
	ulDataLen	[IN] 签名数据长度
	pbSignature	[OUT] 存放签名结果的缓冲区指针
	pulSigLen	[IN,OUT] 返回签名数据长度的指针
返回值	SAR_OK	表示成功
	其他	返回错误码,见表 A. 32

19) RSA 验签函数

函数原型	ULONG DEVAPI HRSS_KEY_RSASVerify (HANDLE hKey, ULONG ulHashAlgID, BYTE * pbData, ULONG ulDataLen, BYTE * pbSignature, ULONG ulSigLen)	
功能描述	验证 RSA 签名。用 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	hKey	[IN] 用来验证签名的公钥句柄
	ulHashAlgID	[IN] 杂凑算法标识
	pbData	[IN] 验证签名的数据
	ulDataLen	[IN] 数据长度
	pbSignature	[IN] 指向数据签名的指针
	ulSigLen	[IN] 数据签名长度
返回值	SAR_OK	表示成功
	其他	返回错误码,见表 A. 32
备注	无输入数据错误	

20) RSA 加密函数

函数原型	ULONG DEVAPI HRSS_KEY_RSAEncrypt (DEVHANDLE hDev, HAPPLICATION hApplication, HANDLE hKey, BYTE * pbData, ULONG ulDataLen, BYTE * pbEncrypt, ULONG * pulEncLen)	
功能描述	按照 PKCS#1 规范要求进行 RSA 数据加密。对输入参数中的数据进行加密,输出加密结果。	
参数	hDev	[IN] 连接设备时返回的设备句柄
	hApplication	[IN] 应用句柄
	hKey	[IN] RSA 公钥句柄
	pbData	[IN] 要加密的输入数据
	ulDataLen	[IN] 数据长度,应小于 117 字节
	pbEncrypt	[OUT] 存放加密结果的缓冲区指针
	pulEncLen	[OUT] 返回加密结果长度的指针
返回值	SAR_OK	表示成功
	其他	返回错误码,见表 A. 32

21) 生成临时 RSA 公私钥对

函数原型	ULONG DEVAPI HRSS_KEY_GenTempRSAkeyPair (DEVHANDLE hDev)	
功能描述	生成临时 RSA 密钥,作为临时加解密数据使用。	
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. 32

22) 临时 RSA 加密函数

函数原型	ULONG DEVAPI HRSS_KEY_TempRSAEncrypt (DEVHANDLE hDev, ULONG ulFlags, BYTE * pbData, ULONG ulDataLen, BYTE * pbEncrypt, ULONG * pulEncLen)	
功能描述	按照 PKCS#1 要求,用临时 RSA 密钥进行数据加密。对输入参数中的数据进行加密,输出加密结果。	
参数	hDev	[IN] 连接设备时返回的设备句柄
	ulFlags	[IN] 补码方式,目前可以为 PKCS5_PADDING
	pbData	[IN] 要加密的输入数据

	ulDataLen	[IN] 数据长度,长度应小于 117 字节
	pbEncrypt	[OUT] 存放加密结果的缓冲区指针
	pulEncLen	[OUT] 返回加密结果长度的指针
返回值	SAR_OK	表示成功
	其他	返回错误码,见表 A. 32

23) 临时 RSA 解密函数

函数原型	ULONG DEVAPI HRSS_KEY_TempRSADecrypt (DEVHANDLE hDev, BYTE * pbData, ULONG ulDataLen, BYTE * pbDecrypt, ULONG * pulDecLen)	
功能描述	按照 PKCS#1 要求,用临时 RSA 密钥进行数据解密。对输入参数中的数据进行解密,输出解密结果。	
参数	hDev	[IN] 连接设备时返回的设备句柄
	pbData	[IN] 要解密的输入数据
	ulDataLen	[IN] 数据长度
	pbDecrypt	[OUT] 存放解密结果的缓冲区指针
	pulDecLen	[OUT] 返回解密结果长度的指针
返回值	SAR_OK	表示成功
	其他	返回错误码,见表 A. 32

24) 加密初始化函数

函数原型	ULONG DEVAPI HRSS_KEY_EncryptInit (DEVHANDLE hDev, HAPPLICATION hApplication, 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	hDev	[IN] 连接设备时返回的设备句柄
	hApplication	[IN] 应用句柄
	hKey	[IN] 加密密钥句柄
	EncryptParam	[IN] 分组密码算法相关参数:算法标识号、密钥长度、初始向量、初始向量长度、填充方法、加密模式、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. 32

```
typedef struct Struct_BLOCKCIPHERPARAM{
    ULONG AlgID;
    BYTE KeyLen;
    BYTE IV[MAX_IV_LEN];
    BYTE IVLen;
    BYTE Padding;
    BYTE Mode;
    BYTE FeedBitLen;
} BLOCKCIPHERPARAM, * PBLOCKCIPHERPARAM;
```

25) 单组数据加密函数

函数原型	ULONG DEVAPI HRSS_KEY_Encrypt(DEVHANDLE hDev, HAPPLICATION hApplication, 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。HRSS_KEY_Encrypt 只对单个分组数据进行加密,在调用 HRSS_KEY_Encrypt 之前,必须调用 HRSS_KEY_EncryptInit 初始化加密操作。HRSS_KEY_Encrypt 等价于先调用 HRSS_KEY_EncryptUpdate 再调用 HRSS_KEY_EncryptFinal,HRSS_KEY_Encrypt 不能与 HRSS_KEY_EncryptUpdate 间隔调用。

参数

hDev	[IN] 连接设备时返回的设备句柄
hApplication	[IN] 应用句柄
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	[IN,OUT] 输入,给出的缓冲区大小;输出,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. 32

26) 多组数据加密函数

函数原型 ULONG DEVAPI HRSS_KEY_EncryptUpdate (DEVHANDLE hDev, HAPPLICATION hApplication, 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。HRSS_KEY_EncryptUpdate 对多个分组数据进行加密,在调用 HRSS_KEY_EncryptUpdate 之前,必须调用 HRSS_KEY_EncryptInit 初始化加密操作;在调用 HRSS_KEY_EncryptUpdate 之后,必须调用 HRSS_KEY_EncryptFinal 结束加密操作。

参数

hDev	[IN] 连接设备时返回的设备句柄
hApplication	[IN] 应用句柄
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. 32

27) 结束加密函数

函数原型 ULONG DEVAPI HRSS_KEY_EncryptFinal (DEVHANDLE hDev, HAPPLICATION hApplication, HANDLE hKey, BYTE * pbEncryptedData, ULONG * ulEncryptedDataLen)

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

参数

hDev	[IN] 连接设备时返回的设备句柄
hApplication	[IN] 应用句柄
hKey	[IN] 加密密钥句柄
pbEncryptedData	[OUT] 加密结果的缓冲区

	ulEncryptedDataLen	[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. 32
28) 解密初始化函数		
函数原型	ULONG DEVAPI HRSS_KEY_DecryptInit (DEVHANDLE hDev, HAPPLICATION hApplication, 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 HRSS_KEY_DecryptInit 之后,可以调用 HRSS_KEY_Decrypt 对单个分组数据进行解密,也可以多次调用 HRSS_KEY_DecryptUpdate 之后再调用 HRSS_KEY_DecryptFinal 完成对多个分组数据的解密。	
参数	hDev	[IN] 连接设备时返回的设备句柄
	hApplication	[IN] 应用句柄
	hKey	[IN] 解密密钥句柄
	DecryptParam	[IN] 分组密码算法相关参数:算法标识号、密钥长度、初始向量、初始向量长度、填充方法、加密模式、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. 32
29) 单组数据解密函数		
函数原型	ULONG DEVAPI HRSS_KEY_Decrypt(DEVHANDLE hDev, HAPPLICATION hApplication, 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。HRSS_KEY_Decrypt 只对单个分组数据进行解密,在调用 HRSS_KEY_Decrypt 之前,必须调用 HRSS_KEY_DecryptInit 初始化解密操作。HRSS_KEY_Decrypt 等价于先调用 HRSS_KEY_DecryptUpdate 再调用 HRSS_KEY_DecryptFinal,HRSS_KEY_Decrypt 不能与 HRSS_KEY_DecryptUpdate 间隔调用。	
参数	hDev	[IN] 连接设备时返回的设备句柄
	hApplication	[IN] 应用句柄
	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] 返回解密后的数据长度
返回值	SAR_OK	表示成功
	其他	返回错误码,见表 A. 32
30) 多组数据解密函数		
函数原型	ULONG DEVAPI HRSS_KEY_DecryptUpdate (DEVHANDLE hDev, HAPPLICATION hApplication, 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。HRSS_KEY_DecryptUpdate 对多个分组数据进行解密,在调用 HRSS_KEY_DecryptUpdate 之前,必须调用 HRSS_KEY_DecryptInit 初始化解密操作;在调用 HRSS_KEY_DecryptUpdate 之后,必须调用 HRSS_KEY_DecryptFinal 结束解密操作。	

参数	hDev	[IN] 连接设备时返回的设备句柄
	hApplication	[IN] 应用句柄
	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] 返回解密后的数据长度
返回值	SAR_OK	表示成功
	其他	返回错误码,见表 A. 32

31) 结束解密函数

函数原型	ULONG DEVAPI HRSS_KEY_DecryptFinal (DEVHANDLE hDev, HAPPLICATION hApplication, HANDLE hKey, BYTE * pbDecryptedData, ULONG * pulDecryptedDataLen)	
功能描述	结束多个分组数据的解密。先调用 HRSS_KEY_DecryptInit 初始化解密操作,再调用 HRSS_KEY_DecryptUpdate 对多个分组数据进行解密,最后调用 HRSS_KEY_DecryptFinal 结束多个分组数据的解密。	
参数	hDev	[IN] 连接设备时返回的设备句柄
	hApplication	[IN] 应用句柄
	hKey	[IN] 解密密钥句柄
	pbDecryptedData	[OUT] 解密结果的缓冲区
	pulDecryptedDataLen	[IN,OUT] 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. 32

32) 杂凑初始化函数

函数原型	ULONG DEVAPI HRSS_KEY_DigestInit (DEVHANDLE hDev, ULONG ulAlgID, 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 HRSS_KEY_DigestInit 之后,可以调用 HRSS_KEY_Digest 对单一分组数据计算消息杂凑,也可以多次调用 HRSS_KEY_DigestUpdate 之后再调用 HRSS_KEY_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
	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. 32

33) 单组数据杂凑函数

函数原型	ULONG DEVAPI HRSS_KEY_Digest (DEVHANDLE hDev, HANDLE hHash, BYTE * pbData, ULONG ulDataLen, BYTE * pbHashData, ULONG * pulHashLen)	
功能描述	HRSS_KEY_Digest 对单一分组的消息进行杂凑计算。调用 HRSS_KEY_Digest 之前,必须调用 HRSS_KEY_DigestInit 初始化杂凑计算操作。HRSS_KEY_Digest 等价于多次调用 HRSS_KEY_DigestUpdate 之后再调用 HRSS_KEY_DigestFinal。	
参数	hDev	[IN] 连接设备时返回的设备句柄
	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
	pulHashLen	[IN,OUT] 指向返回杂凑数据长度的指针
返回值	SAR_OK	表示成功
	其他	返回错误码,见表 A. 32

34) 多组数据杂凑函数

函数原型	ULONG DEVAPI HRSS_KEY_DigestUpdate (DEVHANDLE hDev, HANDLE hHash, BYTE * pbPart, ULONG ulPartLen)	
功能描述	HRSS_KEY_DigestUpdate 对多个分组的消息进行杂凑计算。调用 HRSS_KEY_DigestUpdate 之前,必须调用 HRSS_KEY_DigestInit 初始化杂凑计算操作;调用 HRSS_KEY_DigestUpdate 之后,必须调用 HRSS_KEY_DigestFinal 结束杂凑计算操作。	
参数	hDev	[IN] 连接设备时返回的设备句柄
	hHash	[IN] 哈希对象句柄
	pbPart	[IN] 消息数据指针
	ulPartLen	[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. 32

35) 结束杂凑函数

函数原型	ULONG DEVAPI HRSS_KEY_DigestFinal (DEVHANDLE hDev, 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。HRSS_KEY_DigestFinal 必须用于 HRSS_KEY_DigestUpdate 之后。	
参数	hDev	[IN] 连接设备时返回的设备句柄
	hHash	[IN] 哈希对象句柄
	pHashData	[OUT] 返回的杂凑数据缓冲区指针
	pulHashLen	[IN,OUT] 返回的杂凑数据长度
返回值	SAR_OK	表示成功
	其他	返回错误码,见表 A. 32

f) 接口错误代码

接口函数返回的错误代码定义如表 A. 32:

表 A. 32 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

表 A.32 接口错误代码表 (续)

宏 描 述	预定义值	说 明
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_CSPIMPRTTPUBKEYERR	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

表 A. 32 接口错误代码表 (续)

宏 描 述	预定义值	说 明
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

附录 B
(资料性附录)
证书载体外观

人力资源和社会保障数字证书载体的外形及尺寸如图 B.1 所示：

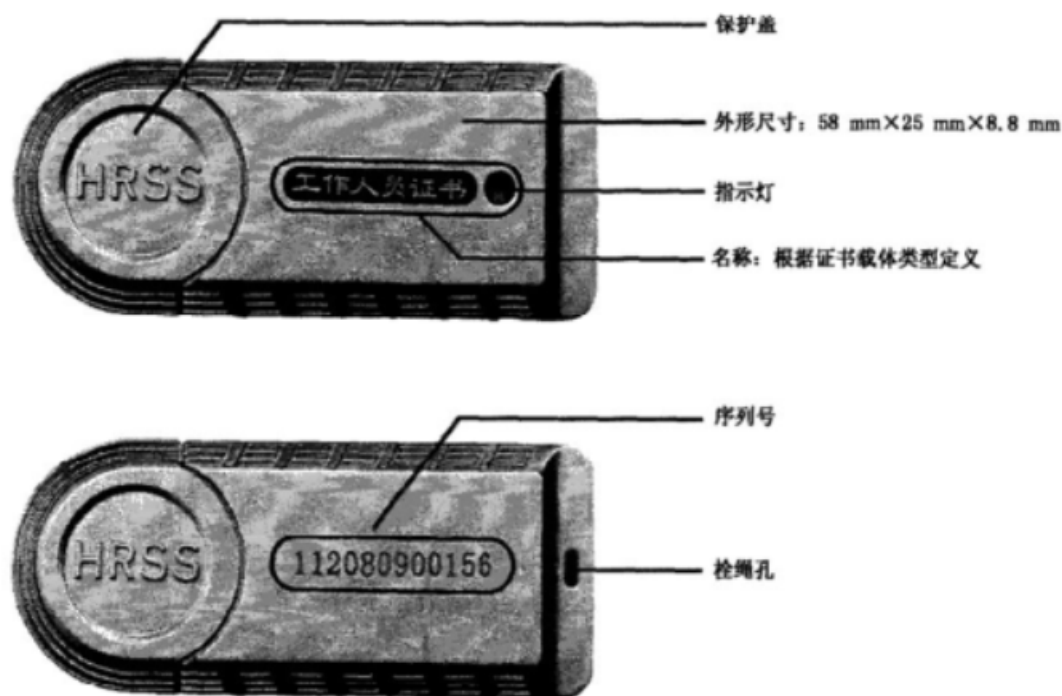


图 B.1 证书载体的外形及尺寸示意图

如图 B.1 所示,USBKey 正面应标识证书类型,USBKey 背面应标识序列号。

证书载体有以下几种类型:面向社会保障部门的政府工作人员、机构、设备的证书载体,面向社会企事业单位的证书载体。各类型证书载体的外形名称和颜色的分类如下:

- a) 工作人员:名称为“工作人员证书”,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。
- c) 设备:名称为“设备证书”,颜色为金色。
- d) 企事业单位:名称为“单位证书”,颜色为深灰色。

证书载体外形上的序列号由 12 位字母或数字组成,按以下规则编码:

- 第 1 位为生产厂商编号;
- 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;
- 第 3 位为证书类型,分别与机构证书、工作人员证书、设备证书、单位证书、个人证书等相对应;
- 第 4 位至第 7 位为生产年份(2 位)和生产月份(2 位);
- 第 8 位至第 12 位为产品流水号。

中华人民共和国劳动和劳动安全
行 业 标 准
人力资源和社会保障电子认证体系
第 5 部分:证书载体规范
LD/T 30.5—2009

*

中国标准出版社出版发行
北京复兴门外三里河北街 16 号
邮政编码:100045

网址 www.spc.net.cn

电话:68523946 68517548

中国标准出版社秦皇岛印刷厂印刷
各地新华书店经销

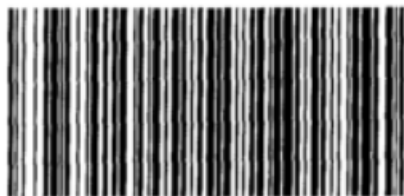
*

开本 880×1230 1/16 印张 3 字数 85 千字
2010 年 2 月第一版 2010 年 2 月第一次印刷

*

书号: 155066 · 2-20300

如有印装差错 由本社发行中心调换
版权专有 侵权必究
举报电话:(010)68533533



LD/T 30.5-2009