

中华人民共和国国家标准

GB/T 43694—2024

网络安全技术 证书应用综合服务接口规范

Cybersecurity technology—Certificate application
integrated service interface specification

2024-04-25发布

2024-11-01 实施

国家市场监督管理总局
国家标准化管理委员会

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 2

5 证书应用综合服务接口 2

 5.1 证书应用综合服务接口在公钥密码应用技术体系框架中的位置 2

 5.2 证书应用综合服务接口分类 2

 5.3 客户端服务接口 2

 5.4 服务器端服务接口 2

6 标识和数据结构 3

 6.1 标识定义 3

 6.2 数据结构定义 3

 6.3 数据格式要求 3

7 证书应用综合服务接口定义 3

 7.1 客户端COM 组件接口 3

 7.2 客户端JavaScript 脚本接口 16

 7.3 服务器端COM 组件接口 28

 7.4 服务器端Java 组件接口 42

8 接口验证方法 56

 8.1 验证环境 56

 8.2 验证原则 56

 8.3 验证场景 57

附录 A（规范性） 证书应用综合服务接口错误代码定义 61

附录B（资料性） 证书应用综合服务接口典型部署模型 64

附录 C（资料性） 证书应用综合服务接口集成示例 65

附录 D（资料性） 证书应用综合服务接口汇总 67

附录 E（资料性） 客户端JavaScript 脚本接口异步调用示例说明 73

参考文献 74

前 言

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

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别专利的责任。

本文件由全国网络安全标准化技术委员会(SAC/TC 260)提出并归口。

本文件起草单位：北京数字认证股份有限公司、博雅中科(北京)信息技术有限公司、北京奇虎科技有限公司、山东得安信息技术有限公司、中国电力科学研究院、北京信安世纪科技股份有限公司、无锡江南信息安全工程技术中心、中国电子技术标准化研究院、格尔软件股份有限公司、中电科网络安全科技股份有限公司、深圳市不动产登记中心、郑州信大捷安信息技术股份有限公司、阿里云计算有限公司、浙江九州量子信息技术股份有限公司、航天信息股份有限公司、数安时代科技股份有限公司、智巡密码(上海)检测技术有限公司、中科信息安全共性技术国家工程研究中心有限公司、中国汽车工程研究院股份有限公司。

本文件主要起草人：刘伟、赵永省、夏鲁宁、李述胜、刘中、程科伟、浦雨三、张屹、张志磊、马洪富、袁中林、李智虎、焦靖伟、刘平、黄晶晶、谭武征、寇建波、颜海龙、刘献伦、刘为华、肖淑婷、张文科、杨倩媚、董亮亮、周蔚林、韩玮、高振鹏、胡建勋、刘冲、牟洁。

网络安全技术 证书应用综合服务接口规范

1 范围

本文件规定了面向证书应用的综合服务接口要求和定义，描述了相应验证方法。
本文件适用于公钥密码基础设施应用技术体系下证书应用中间件和证书应用系统的开发，以及密码应用支撑平台的研制和检测。

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 20518	信息安全技术	公钥基础设施 数字证书格式
GB/T 25061	信息安全技术	XML数字签名语法与处理规范
GB/T 25069	信息安全技术	术语
GB/T 33560	信息安全技术	密码应用标识规范
GB/T 35275	信息安全技术	SM2 密码算法加密签名消息语法规范
GB/T 35276	信息安全技术	SM2 密码算法使用规范
GB/T 35291	信息安全技术	智能密码钥匙应用接口规范
GB/T 36322	信息安全技术	密码设备应用接口规范
GB/T 43578	信息安全技术	通用密码服务接口规范
GM/T 0094—2020	公钥密码应用技术体系框架规范	
GM/Z4001	密码术语	

3 术语和定义

GB/T 25069、GM/Z4001 界定的以及下列术语和定义适用于本文件。

3.1

数字证书 digital certificate

由 CA 签名的包含公开密钥拥有者信息、公开密钥、签发者信息、有效期以及扩展信息的一种数据结构。
注：数字证书也称公钥证书，按类别分为个人证书、机构证书和设备证书，按用途分为签名证书和加密证书。
[来源：GM/Z4001—2013, 2.115]

3.2

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对。
注：用户密钥包含签名密钥对和加密密钥对。

3.3

密钥容器 key container

密码设备中用于保存用户密钥的唯一性存储空间。

4 缩略语

- 下列缩略语适用于本文件。
- B/S: 浏览器/服务器(Browser/Server)
 - CA: 证书认证机构(Certification Authority)
 - COM: 组件对象模型(Component Object Model)
 - CRL: 证书撤销列表(Certificate Revocation List)
 - JAR:Java 归档文件(Java Archive)
 - LDAP: 轻量级目录访问协议(Lightweight Directory Access Protocol)
 - OID: 对象标识符(Object Identifier)
 - XML: 可扩展标记语言(Extensible Markup Language)

5 证书应用综合服务接口

5.1 证书应用综合服务接口在公钥密码应用技术体系框架中的位置

公钥密码应用技术体系框架由应用层、典型密码应用支撑层、通用密码应用支撑层、基础设施安全支撑平台、密码设备服务层组成。证书应用综合服务接口属于通用密码应用支撑层，是中间件，为典型密码应用支撑层和应用层提供密码服务。其功能主要包括：负责完成与密码设备的安全连接；实现基于数字证书的身份认证，从证书中获取有关信息，实现授权管理、访问控制等安全机制；负责具体与密码设备交互实现具体的密码运算；将数据按照GB/T35275 要求的格式进行封装，实现数据封装格式与应用系统无关性，实现应用系统互联互通和信息共享。证书应用综合服务接口规范在公钥密码应用技术框架内的位置应遵循 GM/T 0094—2020 的第4章。

5.2 证书应用综合服务接口分类

证书应用综合服务接口包括客户端服务接口和服务器端服务接口两类。它位于典型密码应用支撑层和密码设备服务层之间，既可被典型密码应用支撑层调用也可被应用层直接调用，向上层提供证书信息解析、基于数字证书身份认证、信息的机密性、完整性和不可否认性等密码服务。

5.3 客户端服务接口

客户端服务接口采用支持多种编程语言的COM 组件和客户端常使用的JavaScript 脚本语言为例描述，其适用于客户端程序调用，接口的形态包括动态库、控件、插件、扩展等。

客户端服务接口一般通过调用智能密码钥匙应用接口或通用密码服务接口来实现，此类接口应符合 GB/T35291 或 GB/T43578 中的规定，主要功能包括：配置管理、证书解析、签名与验证、加密与解密数字信封、XML 数据签名与验证等。

5.4 服务器端服务接口

服务器端服务接口采用支持多种编程语言的COM 组件和服务端常使用的Java 语言为例描述，其适用于服务器端程序调用，接口的形态包括COM 组件、JAR 包等，支持主流操作系统。

服务器端服务接口一般通过调用密码设备应用接口或通用密码服务接口来实现，此类接口应符合 GB/T36322 或 GB/T43578 中的规定，功能基本与客户端服务接口相对应，主要包括：配置管理、证书解析、签名与验证、加密与解密数字信封、XML 数据签名与验证、时间戳等。

6 标识和数据结构

6.1 标识定义

本文件所使用的常量、各类算法标识和证书解析标识应符合GB/T 33560的规定。

6.2 数据结构定义

本文件规定的接口处理的数据分为两种类型：

- 数据类型 A: 是一种基于公钥的加密或签名结构，当公钥算法为RSA 时，数据的结构参见PKCS#1 v1.5;当公钥算法为SM2 时，数据的结构应符合GB/T 35276的规定。
- 数据类型 B: 是一种基于证书的加密或签名结构，当证书的公钥算法为 RSA 时，消息的结构参见PKCS#7; 当证书的公钥算法为 SM2 时，消息的结构应符合 GB/T 35275的规定。

6.3 数据格式要求

- 本文件所涉及的数字证书格式应符合GB/T 20518的规定。
- 本文件所描述的 Base64 编码格式参见 RFC4648 中 Base 64 Encoding部分。
- 在没有特殊说明的情况下，本文件中字符串采用UTF-8 编码，其格式参见 RFC3629。
- 本文件所定义接口返回的错误码或抛出的异常信息应符合附录 A, 典型部署模型可参见附录 B, 集成示例可参见附录C。
- 本文件以 COM 组件为例进行接口描述时，所用到的数据类型说明见表1。

表 1 COM 组件数据类型说明

类型	说明
BSTR	字符串类型，不同的开发语言应采取对应的类型定义，如：char*、CString、java.lang.String等
LONG	32位整数
SHORT	16位整数
BOOL	布尔类型，其取值范围是TRUE和FALSE, 其中TRUE表示真值，FALSE表示假值
空串	长度等于0的字符串
非空	长度大于0的字符串

7 证书应用综合服务接口定义

7.1 客户端 COM 组件接口

7.1.1 客户端COM 组件接口综述

客户端 COM 组件接口提供配置管理、证书解析、签名与验证、加密与解密数字信封、XML 数据签名与验证等功能，共包含35个接口。客户端COM 组件接口列表见附录 D 的表 D.1, 7.1.2~7.1.36 给出了接口详细定义。

7.1.2 获取接口版本信息

获取接口版本信息接口定义应符合表2的规定。

表 2 获取接口版本信息接口定义

原型	BSTR SOf_GetVersion()
描述	获取接口的版本号
参数	无
返回值	接口版本号(成功)
	空串(失败, 可通过SOf_GetLastError获取符合表A. 1定义的错误代码)

7.1.3 设置签名算法

设置签名算法接口定义应符合表3的规定。

表 3 设置签名算法接口定义

原型	LONG SOf_SetSignMethod(LONG SignMethod)
描述	设置接口在签名和验签运算时使用的签名算法
参数	SignMethod(签名算法标识, 应符合GB/T 33560的规定)
返回值	SOR_OK(成功)
	其他(失败, 返回符合表A. 1定义的错误代码)

7.1.4 获得签名算法

获得签名算法接口定义应符合表4的规定。

表4 获得签名算法接口定义

原型	LONG SOf_GetSignMethod()
描述	获得接口当前使用的签名算法
参数	无
返回值	当前使用的签名算法标识(成功)
	0(当前没有设置签名算法)

7.1.5 设置加密算法

设置加密算法接口定义应符合表5的规定。

表 5 设置加密算法接口定义

原型	LONG SOf_SetEncryptMethod(LONG EncryptMethod)
描述	设置接口进行数据加密时使用的对称算法
参数	EncryptMethod(对称加密算法标识, 应符合GB/T33560的规定, 本接口可支持不带附加认证数据的加解密算法)
返回值	SOR_OK(成功)
	其他(失败, 返回表A. 1定义的错误代码)

7.1.6 获得加密算法

获得加密算法接口定义应符合表6的规定。

表 6 获得加密算法接口定义

原型	LONG SOF_GetEncryptMethod()
描述	获得接口当前使用的对称加解密算法
参数	无
返回值	当前使用的加密算法标识(成功)
	0(当前没有设置加密算法)

7.1.7 获得证书列表

获得证书列表接口定义应符合表7的规定。

表 7 获得证书列表接口定义

原型	BSTR SOF_GetUserList()
描述	取得当前已安装证书的用户列表
参数	无
返回值	证书列表[成功, 返回证书列表字符串数据, 格式为: 用户名1 CertID1&. &. 用户名2 CertID2&. &. &…., 根据证书应用的策略不同得到不同的证书列表返回值。在证书列表中, 用户名代表证书的通用名(Common Name), CertID是证书唯一标识, 其格式由实现者自定义, 宜包含容器名、应用名、设备序列号等项。通过CertID应找到唯一的签名证书、加密证书, 并使用对应的用户密钥]
	空串(失败或当前不存在证书用户列表, 可通过SOF_GetLastError获取符合表A. 1定义的错误代码)

7.1.8 导出用户签名证书

导出用户签名证书接口定义应符合表8的规定。

表 8 导出用户签名证书接口定义

原型	BSTR SOF_ExportUserCert(BSTR CertID)
描述	根据证书唯一标识, 获取Base64编码的签名证书字符串
参数	CertID(证书唯一标识)
返回值	Base64编码的签名证书字符串(成功)
	空串(失败, 可通过SOF_GetLastError获取符合表A. 1定义的错误代码)

7.1.9 证书登录

证书登录接口定义应符合表9的规定。

表 9 证书登录接口定义

原型	BOOL S0F_Login(BSTR CertID,BSTR PassWd)
描述	校证书口令，进行证书认证
参数	CertID(证书唯一标识)
	PassWd(证书口令)
返回值	TRUE(成功)
	FALSE(失败，可通过S0F_GetLastError获取符合表A. 1定义的错误代码)
安全规则	证书登录成功后，表示该证书用户已拥有私钥使用权限，即登录状态。登录状态下，可正常调用签名、解密等需要私钥使用权限的接口，获得计算结果

7.1.10 获取证书口令剩余重试次数

获取证书口令剩余重试次数接口定义应符合表10的规定。

表10 获取证书口令剩余重试次数接口定义

原型	LONG S0F_GetPinRetryCount(BSTR CertID)
描述	获取证书口令的剩余重试次数
参数	CertID(证书唯一标识)
返回值	剩余口令重试次数，当重试次数小于或等于0时表示证书口令已被锁定

7.1.11 修改证书口令

修改证书口令接口定义应符合表11的规定。

表11 修改证书口令接口定义

原型	BOOL S0F_ChangePassWd(BSTR CertID,BSTR OldPassWd,BSTR NewPassWd)
描述	修改设备的用户认证口令
参数	CertID(证书唯一标识)
	OldPassWd(旧证书口令)
	NewPassWd(新证书口令)
返回值	TRUE(成功)
	FALSE(失败，可通过S0F_GetLastError获取符合表A. 1定义的错误代码)

7.1.12 导出用户加密证书

导出用户加密证书接口定义应符合表12的规定。

表12 导出用户加密证书接口定义

原型	BSTR S0F_ExportExChangeUserCert(BSTR CertID)
描述	根据证书唯一标识，获取Base64编码的加密(交换)证书字符串
参数	CertID(证书唯一标识)
返回值	Base64编码的加密证书字符串(成功)
	空串(失败，可通过S0F_GetLastError获取符合表A.1定义的错误代码)

7.1.13 获得证书信息

获得证书信息接口定义应符合表13的规定。

表13 获得证书信息接口定义

原型	BSTR S0F_GetCertInfo(BSTR Base64Cert, SHORT Type)
描述	获取证书内指定类型的信息
参数	Base64Cert(Base64编码的证书字符串)
	Type(证书解析标识，应符合GB/T 33560的规定)
返回值	指定类型的证书信息(成功)
	空串(失败或证书中不存在该项内容)

7.1.14 获得证书扩展信息

获得证书扩展信息接口定义应符合表14的规定。

表14 获得证书扩展信息接口定义

原型	BSTR S0F_GetCertInfoByOid(BSTR Base64Cert, BSTR Oid)
描述	根据OID获取证书私有扩展项信息
参数	Base64Cert(Base64编码的证书字符串)
	Oid(私有扩展对象ID, 如“1.2.156.xxx”)
返回值	证书私有扩展项OID对应的信息(成功)
	空串(失败或证书中不存在该私有扩展项)

7.1.15 获得设备信息

获得设备信息接口定义应符合表15的规定。

表15 获得设备信息接口定义

原型	BSTR S0F_GetDeviceInfo(BSTR CertID, LONG Type)
描述	根据证书唯一标识和类型代码获得设备信息
参数	CertID(证书唯一标识)
	Type(设备信息的类型, 应符合GB/T 33560的规定)
返回值	对应的设备信息(成功)
	空串(失败, 可通过S0F_GetLastError获取符合表A. 1定义的错误代码)

7.1.16 验证证书有效性

验证证书有效性接口定义应符合表16的规定。

表16 验证证书有效性接口定义

原型	LONG S0F_ValidateCert(BSTR Base64Cert)
描述	验证证书有效性
参数	Base64Cert(Base64编码的证书字符串)
返回值	SAR_OK(验证成功)
	其他(验证失败, 失败原因应符合表A. 1中的错误代码范围0X0B000500~0X0B000505)
验证策略	基本的证书验证策略应包括: a) 验证CA信任列表, 各层都要进行签名验证; b) 各层证书的有效期验证; c) 各层证书的吊销状态。在特殊情况下(如: 网络条件不允许), 证书的吊销状态可采取灵活方式, 由应用系统各层内部维护一个吊销列表, 在证书登录认证时应用该吊销列表。验证证书有效性也可采取代理验证方式

7.1.17 数据签名

数据签名接口定义应符合表17的规定。

表17 数据签名接口定义

原型	BSTR S0F_SignData(BSTR CertID, BSTR InData)
描述	对字符串数据进行数字签名, 返回Base64编码的数据类型A签名结果
参数	CertID(证书唯一标识)
	InData(原文)
返回值	Base64编码的签名值(成功)
	空串(失败, 可通过S0F_GetLastError获取符合表A. 1定义的错误代码)
安全规则	本接口应在登录状态下才能成功返回签名值, 否则应返回空串

7.1.18 验证数据签名

验证数据签名接口定义应符合表18的规定。

表18 验证数据签名接口定义

原型	BOOL SOf_VerifySignedData(BSTR Base64Cert, BSTR InData, BSTR SignValue)
描述	验证数据签名，签名值格式为Base64编码的数据类型A
参数	Base64Cert (Base64编码的签名者证书字符串)
	InData (原文)
	SignValue (Base64编码的签名值)
返回值	TRUE (成功)
	FALSE (失败，可通过SOf_GetLastError获取符合表A.1定义的错误代码)

7.1.19 文件签名

文件签名接口定义应符合表19的规定。

表19 文件签名接口定义

原型	BSTR SOf_SignFile(BSTR CertID, BSTR InFile)
描述	根据文件全路径，对指定文件进行数字签名，返回Base64编码的数据类型A签名结果
参数	CertID (证书唯一标识)
	InFile (原文文件全路径)
返回值	Base64编码的签名值 (成功)
	空串 (失败，可通过SOf_GetLastError获取符合表A.1定义的错误代码)
安全规则	本接口应在登录状态下才能成功返回签名值，否则应返回空串

7.1.20 验证文件签名

验证文件签名接口定义应符合表20的规定。

表20 验证文件签名接口定义

原型	BOOL SOf_VerifySignedFile(BSTR Base64Cert, BSTR InFile, BSTR SignValue)
描述	验证文件的数字签名，签名值格式为Base64编码的数据类型A
参数	Base64Cert (Base64编码的签名者证书)
	InFile (原文文件全路径)
	SignValue (Base64编码的签名值)
返回值	TRUE (成功)
	FALSE (失败，可通过SOf_GetLastError获取符合表A.1定义的错误代码)

7.1.21 数据加密

数据加密接口定义应符合表21的规定。

表21 数据加密接口定义

原型	BSTR SOF_EncryptData(BSTR Base64Cert, BSTR InData)
描述	数字信封加密，加密过程为使用临时产生的对称密钥加密数据，然后使用数字证书的公钥加密对称密钥，返回Base64编码格式的数据类型B密文数据
参数	Base64Cert (Base64编码的数据接收者的加密证书，若使用多个证书对数据加密，证书之间用&. &. &. 作为分隔符连接)
	InData (待加密的明文)
返回值	Base64编码的密文数据 (成功)
	空串 (失败，可通过SOF_GetLastError获取符合表A. 1定义的错误代码)

7.1.22 数据解密

数据解密接口定义应符合表22的规定。

表22 数据解密接口定义

原型	BSTR SOF_DecryptData(BSTR CertID, BSTR InData)
描述	使用证书对应的私钥解密数字信封，密文数据格式为Base64编码的数据类型B
参数	CertID (证书唯一标识)
	InData (待解密的Base64编码的密文数据)
返回值	解密后的明文数据 (成功)
	空串 (失败，可通过SOF_GetLastError获取符合表A. 1定义的错误代码)
安全规则	本接口应在登录状态下才能解密得到明文，否则应返回空串

7.1.23 消息签名

消息签名接口定义应符合表23的规定。

表23 消息签名接口定义

原型	BSTR SOF_SignMessage(SHORT Flag, BSTR CertID, BSTR InData)
描述	对字符串数据进行消息签名，返回Base64编码的数据类型B签名结果
参数	Flag (是否为Detached的标识), 取值范围： a) 1: 表示Detached, 即不带原文； b) 0: 表示Attached, 即带原文
	CertID (证书唯一标识)
	InData (原文)
返回值	Base64编码的消息签名值 (成功)
	空串 (失败，可通过SOF_GetLastError获取符合表A. 1定义的错误代码)
安全规则	签名结果包含：原文 (可选)、签名者证书、签名算法和签名值。 本接口应在登录状态下才能返回签名值，否则应返回空串

7.1.24 验证消息签名

验证消息签名接口定义应符合表24的规定。

表24 验证消息签名接口定义

原型	BOOL SOF_VerifySignedMessage(BSTR SignedMessage, BSTR InData)
描述	验证消息签名，签名值格式为Base64编码的数据类型B
参数	SignedMessage (Base64编码的消息签名值)
	InData (原文，若签名结果中包含原文，忽略本参数)
返回值	TRUE (成功)
	FALSE (失败，可通过SOF_GetLastError获取符合表A.1定义的错误代码)

7.1.25 解析消息签名

解析消息签名接口定义应符合表25的规定。

表25 解析消息签名接口定义

原型	BSTR SOF_GetInfoFromSignedMessage(BSTR SignedMessage, SHORT Type)
描述	解析消息签名值中的信息，包括：原文、签名值、签名证书等，消息签名值格式为Base64编码的数据类型B
参数	SignedMessage (Base64编码的消息签名值)
	Type (获取的信息类型)，取值范围： a) 1: 解析出原文； b) 2: 解析出Base64编码的签名者证书； c) 3: 解析出Base64编码的签名值
返回值	解析结果 (成功)
	空串 (失败或不存在该项)

7.1.26 XML 数据签名

XML 数据签名接口定义应符合表26的规定。

表26 XML 数据签名接口定义

原型	BSTR SOF_SignDataXML(BSTR CertID, BSTR InData)
描述	对XML数据进行数字签名，证书为RSA算法时签名结果参见RFC3275, 证书为SM2算法时签名结果应符合GB/T 25061的规定
参数	CertID (证书唯一标识)
	InData (XML格式的签名原文)
返回值	XML签名结果 (成功)
	空串 (失败，可通过SOF_GetLastError获取符合表A.1定义的错误代码)

表26 XML 数据签名接口定义（续）

安全规则	本接口在登录状态下才能返回签名值，否则返回空串。 缺省配置和参数如下： a) 采用封皮签名，签名前应对签名原文做格式化； b) 格式化方法采用带注释的XML格式化1.1，证书为RSA算法时标识符为http://www.w3.org/2006/12/xml-c14n11#WithComment，证书为SM2算法时标识符为http://127.0.0.1/2006/12/xml-c14n11#WithComments
------	---

7.1.27 验证 XML 数据签名

验证 XML 数据签名接口定义应符合表27的规定。

表27 验证 XML 数据签名接口定义

原型	BOOL SOF_VerifySignedDataXML(BSTR XMLSignedData)
描述	验证XML签名
参数	XMLSignedData(XML签名结果)
返回值	TRUE(成功)
	FALSE(失败，可通过SOF_GetLastError获取符合表A.1定义的错误代码)

7.1.28 解析 XML 数据签名

解析 XML 数据签名接口定义应符合表28的规定。

表28 解析 XML 数据签名接口定义

原型	BSTR SOF_GetXMLSignatureInfo(BSTR XMLSignedData, SHORT Type)
描述	解析XML数据签名，获取签名值、XML原文、证书和相关算法等信息
参数	XMLSignedData(XML签名结果)
	Type(待解析的参数类型), 取值范围： a) 1:解析出XML. 原文； b) 2:解析出摘要值； c) 3:解析出签名值； d) 4:解析出签名证书； e) 5:解析出摘要算法； f) 6:解析出签名算法
返回值	解析结果(成功)
	空串(失败，可通过SOF_GetLastError获取符合表A.1定义的错误代码)

表 28 解析 XML数据签名接口定义（续）

安全规则	XML签名缺省配置和参数值样例： a) XML原文取自Object元素； b) 摘要值取自DigestValue元素，形如http://127.0.0.1/2001/04/xmldsig-more#sm3； c) 签名值取自SignatureValue元素，为Base64编码； d) 签名证书取自X509Data元素，为Base64编码； e) 摘要算法取自DigestMethod元素的Algorithm属性，形如http://127.0.0.1/2001/04/xmldsig-more#sm3； f) 签名算法取自SignatureMethod元素的Algorithm属性，形如http://127.0.0.1/2001/04/xmldsig-more#sm2-sm3
------	--

7.1.29 产生随机数

产生随机数接口定义应符合表29的规定。

表29 产生随机数接口定义

原型	BSTR SOF_GenRandom(LONG RandomLen)
描述	产生指定长度的随机数
参数	RandomLen(待产生随机数的字节长度)
返回值	Base64编码的随机数(成功) 空串(失败，可通过SOF_GetLastError获取符合表A.1定义的错误代码)

7.1.30 获取最新的错误信息

获取最新的错误信息接口定义应符合表30的规定。

表30 获取最新的错误信息接口定义

原型	LONG SOF_GetLastError()
描述	获取最新的错误代码
参数	无
返回值	错误代码，应符合表A.1错误代码表

7.1.31 计算数据摘要

计算数据摘要接口定义应符合表31的规定。

表31 计算数据摘要接口定义

原型	BSTR SOF_HashData(LONG HashAlg, BSTR InData, BSTR SignCert, BSTR UserID)
描述	计算数据摘要。若采用SM3算法，当SignCert和UsreID为空时，只计算数据的摘要值，当SignCert和UsreID值不为空时，应按照GB/T35276规定的预处理过程计算，摘要值可作为SM2签名的输入

表 31 计算数据摘要接口定义（续）

参数	HashAlg(摘要算法，应符合GB/T 33560的规定)
	Indata(原文)
	SignCert(Base64编码的签名者证书，当摘要算法为SM3时有效，如不需要可传空串)
	UserID(签名者用户ID, 当摘要算法为SM3时有效，若SignCert参数为空，本参数无意义)
返回值	Base64编码的数据摘要值(成功)
	空串(失败，可通过SOF_GetLastError获取符合表A. 1定义的错误代码)

7.1.32 计算文件摘要

计算文件摘要接口定义应符合表32的规定。

表32 计算文件摘要接口定义

原型	BSTR SOF_HashFile(LONG HashAlg, BSTR InFile, BSTR SignCert, BSTR UserID)
描述	计算文件数据摘要，若采用SM3算法，当SignCert和UsreID为空时，只计算数据的摘要值，当SignCert和UsreID值不为空时，应按照GB/T35276规定的预处理过程计算，摘要值可作为SM2签名的输入
参数	HashAlg(摘要算法，应符合GB/T33560的规定)
	InFile(原文文件全路径)
	SignCert(Base64编码的签名者证书，当摘要算法为SM3时有效，如不需要可传空串)
	UserID(签名者用户ID, 当摘要算法为SM3时有效，若SignCert参数为空，本参数无意义)
返回值	Base64编码的文件摘要值(成功)
	空串(失败，可通过SOF_GetLastError获取符合表A. 1定义的错误代码)

7.1.33 摘要数据签名

摘要数据签名接口定义应符合表33的规定。

表33 摘要数据签名接口定义

原型	BSTR SOF_SignHashData(BSTR CertID, BSTR Base64HashData, LONG HashAlg)
描述	对数据摘要值签名，返回Base64编码的数据类型A签名结果。 Base64HashData参数一般是SOF_HashData或SOF_HashFile函数的计算结果。当采用SM2算法签名时，SM3算法的摘要值应按照GB/T 35276规定的预处理过程计算
参数	CertID(证书唯一标识)
	Base64HashData(Base64编码的摘要值)
	HashAlg(摘要算法，应符合GB/T33560的规定)
返回值	Base64编码的签名值(成功)
	空串(失败，可通过SOF_GetLastError获取符合表A. 1定义的错误代码)
安全规则	本接口应在登录状态下才能成功返回签名值，否则应返回空串

7.1.34 验证摘要数据签名

验证摘要数据签名接口定义应符合表34的规定。

表34 验证摘要数据签名接口定义

原型	BOOL SOf_VerifySignedHashData(BSTR Base64Cert, BSTR Base64HashData, BSTR SignValue, LONG HashAlg)
描述	数据摘要签名验证，签名值格式为Base64编码的数据类型A。 若使用SM2算法验证签名，Base64HashData参数是SM3算法的摘要值，应按照GB/T35276规定的预处理过程计算
参数	Base64Cert (Base64编码的签名者证书)
	Base64HashData (Base64编码的摘要值)
	SignValue (Base64编码的签名值)
	HashAlg (摘要算法，应符合GB/T33560的规定)
返回值	TRUE (成功)
	FALSE (失败，可通过SOf_GetLastError获取符合表A.1定义的错误代码)

7.1.35 证书登出

证书登出接口定义应符合表35的规定。

表35 证书登出接口定义

原型	BOOL SOf_Logout(BSTR CertID)
描述	退出登录状态
参数	CertID(证书唯一标识)
返回值	TRUE (成功)
	FALSE (失败，可通过SOf_GetLastError获取符合表A.1定义的错误代码)
安全规则	证书登出表示释放该证书用户对应的私钥使用权限。证书登出后再调用签名、解密等需要拥有私钥使用权限才能正常操作的接口时，应再次成功调用SOf_Login接口获取私钥使用权限

7.1.36 证书登录状态检测

证书登录状态检测接口定义应符合表36的规定。

表36 证书登录状态检测接口定义

原型	BOOL SOf_IsLogin(BSTR CertID)
描述	判断证书用户是否为登录状态
参数	CertID(证书唯一标识)
返回值	TRUE (登录状态)
	FALSE (非登录状态)

7.2 客户端 JavaScript 脚本接口

7.2.1 客户端JavaScript 脚本接口综述

客户端JavaScript 脚本接口采用异步方式定义，所有接口的返回值都通过回调函数的第一个参数返回，提供配置管理、证书解析、签名与验证、加密与解密数字信封、XML 数据签名与验证等功能，共包含32个接口。客户端JavaScript 脚本接口列表见表 D. 2。7. 2. 2~7. 2. 33 给出了接口详细定义，其调用示例说明见附录 E。

7.2.2 获取接口版本信息

获取接口版本信息接口定义应符合表37的规定。

表37 获取接口版本信息接口定义

原型	function SOF_GetVersion(cb,ctx)
描述	获取接口的版本号
参数	cb(回调函数)
	ctx(回调函数所需参数)
返回值	接口版本号(成功)
	空串(失败，可通过SOF_GetLastError获取符合表A. 1定义的错误代码)

7.2.3 设置签名算法

设置签名算法接口定义应符合表38的规定。

表38 设置签名算法接口定义

原型	function SOF_SetSignMethod(SignMethod,cb,ctx)
描述	设置接口在签名和验签运算时使用的签名算法
参数	SignMethod(签名算法标识，应符合GB/T 33560的规定)
	cb(回调函数)
	ctx(回调函数所需参数)
返回值	SAR_OK(成功)
	其他(失败，返回表A. 1定义的错误代码)

7.2.4 获得签名算法

获得签名算法接口定义应符合表39的规定。

表39 获得签名算法接口定义

原型	function SOF_GetSignMethod(cb,ctx)
描述	获得接口当前使用的签名算法

表 39 获得签名算法接口定义（续）

参数	cb(回调函数)
	ctx(回调函数所需参数)
返回值	当前使用的签名算法标识(成功)
	0(当前没有设置签名算法)

7.2.5 设置加密算法

设置加密算法接口定义应符合表40的规定。

表40 设置加密算法接口定义

原型	[unction SOf_SetEncryptMethod(EncryptMethod, cb, ctx)
描述	设置接口进行数据加密时使用的对称算法
参数	EncryptMethod(对称密码算法标识，应符合GB/T33560的规定。本接口支持不带附加认证数据的加密算法)
	cb(回调函数)
	ctx(回调函数所需参数)
返回值	SOf_OK(成功)
	其他(失败，返回符合表A.1定义的错误代码)

7.2.6 获得加密算法

获得加密算法接口定义应符合表41的规定。

表41 获得加密算法接口定义

原型	function SOf_GetEncryptMethod(cb, ctx)
描述	获得接口当前使用的对称加密算法
参数	cb(回调函数)
	ctx(回调函数所需参数)
返回值	当前使用的加密算法标识(成功)
	0(当前没有设置加密算法)

7.2.7 获得证书用户列表

获得证书用户列表接口定义应符合表42的规定。

表42 获得证书用户列表接口定义

原型	function SOF_GetUserList(cb, ctx)
描述	获取证书用户列表
参数	cb(回调函数)
	ctx(回调函数所需参数)
返回值	证书列表(成功, 格式和7.1.7相同)
	空串(失败或当前不存在证书用户列表, 可通过SOF_GetLastError获取符合表A.1定义的错误代码)

7.2.8 导出用户签名证书

导出用户签名证书接口定义应符合表43的规定。

表43 导出用户签名证书接口定义

原型	function SOF_ExportUserCert(CertID, cb, ctx)
描述	根据证书唯一标识, 获取Base64编码的签名证书字符串
参数	CertID(证书唯一标识)
	cb(回调函数)
	ctx(回调函数所需参数)
返回值	Base64编码的签名证书字符串(成功)
	空串(失败, 可通过SOF_GetLastError获取符合表A.1定义的错误代码)

7.2.9 证书登录

证书登录接口定义应符合表44的规定。

表44 证书登录接口定义

原型	function SOF_Login(CertID, PassWd, cb, ctx)
描述	校证书口令, 进行用户认证
参数	CertID(证书唯一标识)
	PassWd(证书口令)
	cb(回调函数)
	ctx(回调函数所需参数)
返回值	true(成功)
	false(失败, 可通过SOF_GetLastError获取符合表A.1定义的错误代码)
安全规则	证书登录成功后, 表示该证书用户已拥有私钥使用权限, 即登录状态。登录状态下, 可正常调用签名、解密等需要私钥使用权限的接口, 获得计算结果

7.2.10 获取证书口令剩余重试次数

获取证书口令剩余重试次数接口定义应符合表45的规定。

表45 获取证书口令剩余重试次数接口定义

原型	function SOF_GetPinRetryCount(CertID, cb, ctx)
描述	获取证书口令剩余重试次数
参数	CertID(证书唯一标识)
	cb(回调函数)
	ctx(回调函数所需参数)
返回值	剩余口令重试次数，当重试次数小于或等于0时表示证书口令已被锁定

7.2.11 修改证书口令

修改证书口令接口定义应符合表46的规定。

表46 修改证书口令接口定义

原型	function ChangeUserPassword(CertID, OldPassWd, NewPassWd, cb, ctx)
描述	修改证书口令
参数	CertID(证书唯一标识)
	OldPassWd(旧证书口令)
	NewPassWd(新证书口令)
	cb(回调函数)
	ctx(回调函数所需参数)
返回值	true(修改口令成功)
	false(失败，可通过SOF_GetLastError获取符合表A.1定义的错误代码)

7.2.12 导出用户加密证书

导出用户加密证书接口定义应符合表47的规定。

表47 导出用户加密证书接口定义

原型	function SOF_ExportExChangeUserCert(CertID, cb, ctx)
描述	根据证书唯一标识，获取Base64编码的加密(交换)证书字符串
参数	CertID(证书唯一标识)
	cb(回调函数)
	ctx(回调函数所需参数)
返回值	Base64编码的加密证书字符串(成功)
	空串(失败，可通过SOF_GetLastError获取符合表A.1定义的错误代码)

7.2.13 获得证书基本信息

获得证书基本信息接口定义应符合表48的规定。

表48 获得证书基本信息接口定义

原型	function S0F_GetCertInfo(Base64Cert, Type, cb, ctx)
描述	获取证书基本信息
参数	Base64Cert (Base64编码的证书字符串)
	Type (证书解析标识, 应符合GB/T33560的规定)
	cb (回调函数)
	ctx (回调函数所需参数)
返回值	指定类型的证书信息 (成功)
	空串 (失败或证书中不存在该项内容)

7.2.14 获得证书扩展信息

获得证书扩展信息接口定义应符合表49的规定。

表49 获得证书扩展信息接口定义

原型	function S0F_GetCertInfoByOid(Base64Cert, Oid, cb, ctx)
描述	根据OID获取证书扩展项信息
参数	Base64Cert (Base64编码的证书字符串)
	Oid (私有扩展对象ID, 如 “1.2.156.xxx”)
	cb (回调函数)
	ctx (回调函数所需参数)
返回值	证书私有扩展项OID对应的信息 (成功)
	空串 (失败或证书中不存在该私有扩展项)

7.2.15 获得设备信息

获得设备信息接口定义应符合表50的规定。

表50 获得设备信息接口定义

原型	function S0F_GetDeviceInfo(CertID, Type, cb, ctx)
描述:	根据证书唯一标识和类型代码获得设备信息
参数	CertID (证书唯一标识)
	Type (设备信息的类型, 应符合GB/T33560的规定)
	cb (回调函数)
	ctx (回调函数所需参数)
返回值	对应的设备信息 (成功)
	空串 (失败, 可通过S0F_GetLastError获取符合表A.1定义的错误代码)

7.2.16 验证证书有效性

验证证书有效性接口定义应符合表51的规定。

表 5 1 验证证书有效性接口定义

原型	function SOF_ValidateCert(Base64Cert, cb, ctx)
描述	验证证书有效性
参数	Base64Cert (Base64编码格式的证书)
	cb (回调函数)
	ctx (回调函数所需参数)
返回值	SAR_ OK (验证成功)
	其他 (验证失败，失败原因应符合表A. 1中的错误代码范围0X0B000500~0X0B000505)
验证策略	<p>本接口传入的证书是Base64格式的串，一般通过获取签名(或加密)证书来取得。验证证书有效性在本地完成，可信根证书列表由实现者管理。</p> <p>基本的证书验证策略应包括：</p> <p>a) 验证CA信任列表，各层都要进行签名验证；</p> <p>b) 各层证书的有效期验证；</p> <p>c) 各层证书的吊销状态。在特殊情况下(如：网络条件不允许)，证书的吊销状态可采取灵活方式，由应用系统各层内部维护一个吊销列表，在证书登录认证时应用该吊销列表。验证证书有效性也可采取代理验证方式</p>

7.2.17 数据签名

数据签名接口定义应符合表52的规定。

表52 数据签名接口定义

原型	[unction SOF_SignData(CertID, InData, cb, ctx)
描述	对字符串数据进行数字签名，返回Base64编码的数据类型A签名结果
参数	CertID (证书唯一标识)
	InData (原文)
	cb (回调函数)
	ctx (回调函数所需参数)
返回值	Base64编码的签名值 (成功)
	空串 (失败，可通过SOF_GetLastError获取符合表A. 1定义的错误代码)
安全规则	本接口应在登录状态下才能成功返回签名值，否则应返回空串

7.2.18 验证数据签名

验证数据签名接口定义应符合表53的规定。

表53 验证数据签名接口定义

原型	function S0F_VerifySignedData(Base64Cert, InData, SignValue, cb, ctx)
描述	验证数据签名，签名值为Base64编码的数据类型A
参数	Base64Cert (Base64编码的签名者证书)
	InData (原文)
	SignValue (Base64编码的签名值)
	cb (回调函数)
	ctx (回调函数所需参数)
返回值	true (成功)
	false (失败，可通过S0F_GetLastError获取符合表A. 1定义的错误代码)

7.2.19 数据加密

数据加密接口定义应符合表54的规定。

表54 数据加密接口定义

原型	function S0F_EncryptData(Base64Cert, InData, cb, ctx)
描述	数字信封加密，加密过程为使用临时产生的对称密钥加密数据，然后使用数字证书的公钥加密对称密钥，返回Base64编码的数据类型B密文数据
参数	Base64Cert (Base64编码的数据接收者的加密证书，如使用多个证书对数据加密，证书之间用8.&.& 作为分隔符连接)
	InData (待加密的明文数据)
	cb (回调函数)
	ctx (回调函数所需参数)
返回值	Base64编码格式的密文数据 (成功)
	空串 (失败，可通过S0F_GetLastError获取符合表A. 1定义的错误代码)

7.2.20 数据解密

数据解密接口定义应符合表55的规定。

表55 数据解密接口定义

原型	function S0F_DecryptData(CertID, InData, cb, ctx)
描述	使用证书对应的私钥解密数字信封，密文数据格式为Base64编码的数据类型B
参数	CertID (证书唯一标识)
	Indata (待解密的Base64编码格式的密文数据)
	cb (回调函数)
	ctx (回调函数所需参数)

表 55 数据解密接口定义（续）

返回值	解密后的明文数据(成功)
	空串(失败，可通过SOF_GetLastError获取符合表A. 1定义的错误代码)
安全规则	本接口应在登录状态下才能解密得到明文，否则应返回空串

7.2.21 消息签名

消息签名接口定义应符合表56的规定。

表56 消息签名接口定义

原型	function SOF_SignMessage(Flag, CertID, InData, cb, ctx)
描述	对字符串数据进行消息签名，返回Base64编码的数据类型B签名结果
参数	Flag(是否为Detached的标识), 取值范围: a) 1: 表示Detached, 即不带原文; b) 0: 表示Attached, 即带原文
	CertID(证书唯一标识)
	InData(原文)
	cb(回调函数)
	ctx(回调函数所需参数)
返回值	Base64编码的消息签名值(成功)
	空串(失败，可通过SOF_GetLastError获取符合表A. 1定义的错误代码)
安全规则	本接口应在登录状态下才能成功返回签名值，否则应返回空串

7.2.22 验证消息签名

验证消息签名接口定义应符合表57的规定。

表57 验证消息签名接口定义

原型	function SOF_VerifySignedMessage(SignedMessage, InData, cb, ctx)
描述	验证消息签名，签名值的格式为Base64编码的数据类型B
参数	SignedMessage(Base64编码的消息签名值)
	InData(原文，若签名中包含原文，忽略本参数)
	cb(回调函数)
	ctx(回调函数所需参数)
返回值	true(成功)
	false(失败，可通过SOF_GetLastError获取符合表A. 1定义的错误代码)

7.2.23 解析消息签名

解析消息签名接口定义应符合表58的规定。

表58 解析消息签名接口定义

原型	function SOF_GetInfoFromSignedMessage(SignedMessage, Type, cb, ctx)
描述	解析消息签名值中的信息，包括：原文、签名值、签名证书等，消息签名值格式为Base64编码的数据类型B
参数	SignedMessage (Base64编码的消息签名值)
	Type (类型), 取值范围： a) 1: 解析出原文； b) 2: 解析出Base64编码的签名者证书； c) 3: 解析出Base64编码的签名值
	cb (回调函数)
	ctx (回调函数所需参数)
返回值	解析结果 (成功)
	空串 (失败或不存在该项)

7.2.24 XML 数据签名

XML 数据签名接口定义应符合表59的规定。

表59 XML 数据签名接口定义

原型	function SOF_SignDataXML (CertID, InXMLData, cb, ctx)
描述	对XML数据进行数字签名，证书为RSA算法时签名结果参见RFC3275, 证书为SM2算法时签名结果应符合GB/T 25061的规定
参数	CertID (证书唯一标识)
	InXMLData (待签名的XML数据)
	cb (回调函数)
	ctx (回调函数所需参数)
返回值	XML签名结果 (成功)
	空串 (失败，可通过SOF_GetLastError获取符合表A.1定义的错误代码)
安全规则	本接口应在登录状态下才能返回签名值，否则应返回空串。 缺省配置和参数： a) 采用封皮签名，签名前应对签名原文做格式化； b) 格式化方法采用带注释的XML格式化1.1, 证书为RSA算法时标识符为http://www.w3.org/2006/12/xml-c14n11#WithComment, 证书为SM2算法时标识符为http://127.0.0.1/2006/12/xml-c14n11#WithComments

7.2.25 验证 XML 数据签名

验证 XML 数据签名接口定义应符合表60的规定。

表60 验证 XML 数据签名接口定义

原型	function SOF_VerifySignedDataXML(SignedXMLData, cb, ctx)
描述	验证XML签名
参数	SignedXMLData(XML签名结果)
	cb(回调函数)
	ctx(回调函数所需参数)
返回值	true(成功)
	false(失败, 可通过SOF_GetLastError获取符合表A.1定义的错误代码)

7.2.26 解析 XML 数据签名

解析 XML 数据签名接口定义应符合表61的规定。

表61 解析 XML 数据签名接口定义

原型	function SOF_GetXMLSignatureInfo(XMLSignedData, Type, cb, ctx)
描述	解析XML数据签名, 获取签名值、XML原文、证书和相关算法等信息
参数	XMLSignedData(XML签名结果)
	Type(待解析的参数类型), 取值范围: a) 1: 解析出XML原文; b) 2: 解析出摘要值; c) 3: 解析出签名值; d) 4: 解析出签名证书 e) 5: 解析出摘要算法; f) 6: 解析出签名算法
	cb(回调函数)
	ctx(回调函数所需参数)
返回值	解析结果(成功)
	空串(失败, 可通过SOF_GetLastError获取符合表A.1定义的错误代码)
安全规则	XML签名缺省配置和参数值样例: a) XML原文取自Object元素; b) 摘要值取自DigestValue元素, 形如http://127.0.0.1/2001/04/xmldsig-more#sm3; c) 签名值取自SignatureValue元素, 为Base64编码; d) 签名证书取自X509Data元素, 为Base64编码; e) 摘要算法取自DigestMethod元素的Algorithm属性, 形如http://127.0.0.1/2001/04/xmldsig-more#sm3; f) 签名算法取自SignatureMethod元素的Algorithm属性, 形如http://127.0.0.1/2001/04/xmldsig-more#sm2-sm3

7.2.27 产生随机数

产生随机数接口定义应符合表62的规定。

表62 产生随机数接口定义

原型	function SOF_GenRandom(RandomLen, cb, ctx)
描述	产生指定长度的随机数
参数	RandomLen(待产生随机数的字节长度)
	cb(回调函数)
	ctx(回调函数所需参数)
返回值	Base64编码的随机数(成功)
	空串(失败, 可通过SOF_GetLastError获取符合表A. 1定义的错误代码)

7.2.28 获取最新的错误信息

获取最新的错误信息接口定义应符合表63的规定。

表63 获取最新的错误信息接口定义

原型	function SOF_GetLaseError(cb, ctx)
描述	获取最新的错误码
参数	cb(回调函数)
	ctx(回调函数所需参数)
返回值	错误代码, 应符合表A. 1错误代码表

7.2.29 计算数据摘要

计算数据摘要接口定义应符合表64的规定。

表64 计算数据摘要接口定义

原型	function SOF_HashData(HashAlg, InData, SignCert, UserID, cb, ctx)
描述	计算数据摘要, 若采用SM3算法, 当SignCert和UsreID为空时, 只计算数据的摘要值, 当SignCert和UsreID值不为空时, 应按照GB/T35276规定的预处理过程计算, 摘要值可作为SM2签名的输入
参数	HashAlg(摘要算法, 应符合GB/T 33560的规定)
	InData(原始数据)
	SignCert(Base64编码的签名者证书, 当摘要算法为SM3时有效, 如不需要传空串)
	UserID(签名者用户ID, 摘要算法为SM3时有效, 若SignCert参数为空, 本参数无意义)
	cb(回调函数)
	ctx(回调函数所需参数)
返回值	Base64编码的数据摘要值(成功)
	空串(失败, 可通过SOF_GetLastError获取符合表A. 1定义的错误代码)

7.2.30 摘要数据签名

摘要数据签名接口定义应符合表65的规定。

表65 摘要数据签名接口定义

原型	function SOf_SignHashData(CertID, Base64HashData, HashAlg, cb, ctx)
描述	对数据摘要签名，返回Base64编码的数据类型A签名结果。 Base64HashData值一般是SOf_HashData或SOf_HashFile函数的计算结果。当采用SM2算法签名时，SM3算法的摘要值应按照GB/T35276规定的预处理过程计算
参数	CertID(证书唯一标识)
	Base64HashData(Base64编码的摘要值)
	HashAlg(摘要算法，应符合GB/T 33560的规定)
	cb(回调函数)
	ctx(回调函数所需参数)
返回值	Base64编码的签名值(成功)
	空串(失败，可通过SOf_GetLastError获取符合表A.1定义的错误代码)
安全规则	本接口应在登录状态下才能成功返回签名值，否则应返回空串

7.2.31 验证摘要数据签名

验证摘要数据签名接口定义应符合表66的规定。

表66 验证摘要数据签名接口定义

原型	function SOf_VerifySigned HashData(Base64Cert, Base64HashData, SignValue, HashAlg, cb, ctx)
描述	数据摘要签名验证，签名值为Base64编码的数据类型A。 若使用SM2算法验证签名，Base64HashData参数是SM3算法的摘要值，应按照GB/T35276规定的预处理过程计算
参数	Base64Cert(Base64编码签名者证书)
参数	Base64HashData(Base64编码摘要值)
	SignValue(Base64编码的签名值)
	HashAlg(摘要算法，应符合GB/T 33560的规定)
	cb(回调函数)
	ctx(回调函数所需参数)
返回值	true(成功)
	false(失败，可通过SOf_GetLastError获取符合表A.1定义的错误代码)

7.2.32 证书登出

证书登出接口定义应符合表67的规定。

表67 证书登出接口定义

原型	function S0F_Logout(CertID, cb, ctx)
描述	退出登录状态
参数	CertID(证书唯一标识)
	cb(回调函数)
	ctx(回调函数所需参数)
返回值	true(成功)
	false(失败，可通过S0F_GetLastError获取符合表A.1定义的错误代码)
安全规则	证书登出表示释放该证书用户对应的私钥使用权限。证书登出后再调用签名、解密等需要拥有私钥使用权限才能正常操作的接口时，应再次成功调用S0F_Login接口获取私钥使用权限

7.2.33 证书登录状态检测

证书登录状态检测接口定义应符合表68的规定。

表68 证书登录状态检测接口定义

原型	function S0F_IsLogin(CertID, cb, ctx)
描述	判断证书用户是否为登录状态
参数	CertID(证书唯一标识)
	cb(回调函数)
	ctx(回调函数所需参数)
返回值	true(登录状态)
	false(非登录状态)

7.3 服务器端 COM 组件接口

7.3.1 服务器端 COM 组件接口综述

服务器端 COM 组件接口提供配置管理、证书解析、签名与验证、加密与解密数字信封、XML 数据签名与验证、时间戳等功能，共包含39个接口。服务器端COM 组件接口列表见表 D.3, 7.3.2~7.3.40 给出了接口详细定义。

7.3.2 设置证书信任列表

设置证书信任列表接口定义应符合表69的规定。

表69 设置证书信任列表接口定义

原型	LONG S0F_SetCertTrustList(BSTR CTLAltName, BSTR CTLContent, LONG CTLContentLen)
描述	设置证书信任列表
参数	CTLAltName(证书信任列表别名)
	CTLContent(Base64编码格式的证书信任列表内容)
	CTLContentLen(证书信任列表长度)
返回值	SAR_OK(成功)
	其他(失败, 返回表A.1定义的错误代码)

7.3.3 查询证书信任列表别名

查询证书信任列表别名接口定义应符合表70的规定。

表70 查询证书信任列表别名接口定义

原型	BSTR S0F_GetCertTrustListAltNames()
描述	查询证书信任列表别名
参数	无
返回值	信任列表别名(成功, 返回信任列表别名的字符串组合, 如“CA001@CA002@CA003”)
	空串(失败, 可通过S0F_GetLastError获取符合表A.1定义的错误代码)

7.3.4 查询证书信任列表

查询证书信任列表接口定义应符合表71的规定。

表 7 1 查询证书信任列表接口定义

原型	BSTR S0F_GetCertTrustList(BSTR CTLAltName)
描述	根据别名查询证书信任列表
参数	CTLAltName(证书信任列表别名)
返回值	信任列表(成功, 返回Base64编码的证书信任列表)
	空串(失败, 可通过S0F_GetLastError获取符合表A.1定义的错误代码)

7.3.5 删除证书信任列表

删除证书信任列表接口定义应符合表72的规定。

表72 删除证书信任列表接口定义

原型	LONG SOF_DelCertTrustList(BSTR CTLAltName)
描述	根据别名删除证书信任列表
参数	CTLAltName(证书信任列表别名)
返回值	SAR_OK(成功)
	其他(失败, 返回表A.1定义的错误代码)

7.3.6 初始化应用策略



初始化应用策略接口定义应符合表73的规定。

表73 初始化应用策略接口定义

原型	LONG SOF_InitCertAppPolicy(BSTR PolicyName)
描述	根据应用策略名称设置应用遵循的证书应用策略。该名称要和服务器配置文件对应。接口从配置文件中读取应用策略信息, 宜包括使用的密钥和证书、信任的根证书、证书验证的策略、验证方式。配置内容自行定义
参数	PolicyName(应用策略名称)
返回值	SAR_OK(成功)
	其他(失败, 返回表A.1定义的错误代码)

7.3.7 设置签名算法

设置签名算法接口定义应符合表74的规定。

表74 设置签名算法接口定义

原型	LONG SOF_SetSignMethod(LONG SignMethod)
描述	设置COM组件签名运算使用的签名算法
参数	SignMethod(签名算法标识, 应符合GB/T33560的规定)
返回值	SAR_OK(成功)
	其他(失败, 返回表A.1定义的错误代码)

7.3.8 获得签名算法

获得签名算法接口定义应符合表75的规定。

表75 获得签名算法接口定义

原型	LONG SOF_GetSignMethod()
描述	获得组件当前签名和验签运算使用的签名算法
参数	无
返回值	当前使用的签名算法标识(成功)
	0(当前没有设置签名算法)

7.3.9 设置加密算法

设置加密算法接口定义应符合表76的规定。

表76 设置加密算法接口定义

原型	LONG SOF_SetEncryptMethod(LONG EncryptMethod)
描述	设置组件对数据加密使用的对称算法
参数	EncryptMethod(对称密码算法标识, 应符合GB/T 33560的规定。本接口支持不带附加认证数据的加密算法)
返回值	SAR_OK(成功)
	其他(失败, 返回表A.1定义的错误代码)

7.3.10 获得加密算法

获得加密算法接口定义应符合表77的规定。

表77 获得加密算法接口定义

原型	LONG SOF_GetEncryptMethod()
描述	获得组件当前使用的对称加解密算法
参数	无
返回值	当前使用的加密算法标识(成功)
	0(当前没有设置加密算法)

7.3.11 获得服务器证书

获得服务器证书接口定义应符合表78的规定。

表78 获得服务器证书接口定义

原型	BSTR SOF_GetServerCertificate(SHORT CertUsage)
描述	读取当前应用指定的服务器证书
参数	CertUsage(证书用途), 取值范围: a) 1: 加密证书; b) 2: 签名证书
返回值	Base64编码的服务器证书(成功)
	空串(失败, 可通过SOF_GetLastError获取符合表A.1定义的错误代码)

7.3.12 产生随机数

产生随机数接口定义应符合表79的规定。

表79 产生随机数接口定义

原型	BSTR SOf_GenRandom(SHORT RandomLen)
描述	产生指定长度的随机数
参数	RandomLen(待产生随机数的字节长度)
返回值	Base64编码的随机数(成功)
	空串(失败, 可通过SOf_GetLastError获取符合表A. 1定义的错误代码)

7.3.13 获得证书信息

获得证书信息接口定义应符合表80的规定。

表80 获得证书信息接口定义

原型	BSTR SOf_GetCertInfo(BSTR Base64Cert, LONG Type)
描述	根据指定类型, 获取证书内的相关信息
参数	Base64Cert(Base64编码的数字证书)
	Type(证书解析标识, 应符合GB/T33560的规定)
返回值	指定类型的证书信息(成功)
	空串(失败或证书中不存在该项内容)

7.3.14 获得证书扩展信息

获得证书扩展信息接口定义应符合表81的规定。

表81 获得证书扩展信息接口定义

原型	BSTR SOf_GetCertInfoByOid(BSTR Base64Cert, BSTR Oid)
描述	根据OID获取证书扩展项信息
参数	Base64Cert(Base64编码的证书)
	Oid(私有扩展对象ID, 如“1.2.156.xxx”)
返回值	证书私有扩展项OID对应的信息(成功)
	空串(失败或证书中不存在该私有扩展项)

7.3.15 验证证书有效性

验证证书有效性接口定义应符合表82的规定。

表82 验证证书有效性接口定义

原型	LONG SOF_ValidateCert(BSTR Base64Cert)
描述	根据应用的策略根据验证证书有效性
参数	Base64Cert (待验证的Base64编码证书)
返回值	SAR_OK (验证成功)
	其他 (验证失败，失败原因应符合表A. 1中的错误代码范围0X0B000500~0X0B000505)
验证策略	基本的证书验证策略应包括： a) 验证CA信任列表，各层都要进行签名验证； b) 各层证书的有效期验证； c) 各层证书的吊销状态。在特殊情况下(如：网络条件不允许)，证书的吊销状态可采取灵活方式，由应用系统内部维护一个吊销列表，在证书登录认证时应用该吊销列表。验证证书有效性也可采取代理验证方式

7.3.16 数据签名

数据签名接口定义应符合表83的规定。

表83 数据签名接口定义

原型	BSTR SOF_SignData(BSTR InData)
描述	对字符串数据进行数字签名，返回Base64编码的数据类型A签名结果
参数	InData (原文)
返回值	Base64编码的签名值 (成功)
	空串 (失败，可通过SOF_GetLastError获取符合表A. 1定义的错误代码)

7.3.17 验证数据签名

验证数据签名接口定义应符合表84的规定。

表84 验证数据签名接口定义

原型	LONG SOF_VerifySignedData(BSTR Base64Cert, BSTR InData, BSTR SignValue)
描述	验证数字签名，签名值格式为Base64编码的数据类型A
参数	Base64Cert (Base64编码的签名证书)
	InData (原文)
	SignValue (Base64编码的签名值)
返回值	SAR_OK (成功)
	其他 (失败，返回表A. 1定义的错误代码)

7.3.18 文件签名

文件签名接口定义应符合表85的规定。

表85 文件签名接口定义

原型	BSTR SOF_SignFile(BSTR InFile)
描述	对文件进行数字签名，返回Base64编码的数据类型A签名结果
参数	InFile(待签名的文件全路径)
返回值	Base64编码的签名值(成功)
	空串(失败，可通过SOF_GetLastError获取符合表A. 1定义的错误代码)
文件规则	待签名的文件全路径指运行本接口的主机上的文件，非服务器端文件

7.3.19 验证文件签名

验证文件签名接口定义应符合表86的规定。

表86 验证文件签名接口定义

原型	LONG SOF_VerifySignedFile(BSTR Base64Cert,BSTR InFile,BSTR SignValue)
描述	验证文件数字签名，签名值格式为Base64编码的数据类型A
参数	Base64Cert(Base64编码的签名证书)
	InFile(待验证的文件全路径)
	SignValue(Base64编码的签名值)
返回值	SAR_OK(成功)
	其他(失败，返回表A. 1定义的错误代码)
文件规则	待验证的原文路径指运行本接口的主机上的文件，非服务器端文件

7.3.20 数据加密

数据加密接口定义应符合表87的规定。

表87 数据加密接口定义

原型	BSTR SOF_EncryptData(BSTR Cert,BSTR InData)
描述	数字信封加密，加密过程为使用临时产生的对称密钥加密数据，然后使用数字证书的公钥加密对称密钥，返回Base64编码的数据类型B密文数据
参数	Cert(Base64编码的数据接收者的加密证书，如有多个接收者，多个接收者加密证书之间用8.8.&.作为分隔符连接)
	InData(待加密的明文数据)
返回值	Base64编码格式的密文数据(成功)
	空串(失败，可通过SOF_GetLastError获取符合表A. 1定义的错误代码)

7.3.21 数据解密

数据解密接口定义应符合表88的规定。

表88 数据解密接口定义

原型	BSTR SOF_DecryptData(BSTR CertID, BSTR InData)
描述	解密格式为Base64编码的数据类型B密文数据
参数	CertID(解密密钥对应的证书唯一标识，如不需要可传空串)
	InData(Base64编码的密文数据)
返回值	解密后的明文数据(成功)
	空串(失败，可通过SOF_GetLastError获取符合表A.1定义的错误代码)

7.3.22 文件加密

文件加密接口定义应符合表89的规定。

表89 文件加密接口定义

原型	LONG SOF_EncryptFile(BSTR Cert, BSTR InFile, BSTR OutFile)
描述	使用证书加密文件，密文文件结果为数据类型B
参数	Cert(Base64编码的数据接收者的加密证书，如有多个接收者，多个接收者加密证书之间用&. &. &. 作为分隔符连接)
	InFile(待加密的明文文件全路径)
	OutFile(密文文件保存全路径)
返回值	SAR_OK(成功)
	其他(失败，返回表A.1定义的错误代码)
文件规则	待加密的明文文件全路径和密文文件保存全路径，指运行本接口的主机上的文件，非服务器端文件

7.3.23 文件解密

文件解密接口定义应符合表90的规定。

表90 文件解密接口定义

原型	LONG SOF_DecryptFile(BSTR CertID, BSTR InFile, BSTR OutFile)
描述	使用证书对应的私钥解密文件，密文文件格式为数据类型B
参数	CertID(解密密钥对应的证书唯一标识，如不需要可传空串)
	InFile(待解密的密文文件全路径)
	OutFile(明文文件保存全路径)
返回值	SAR_OK(成功)
	其他(失败，返回表A.1定义的错误代码)
文件规则	待解密的密文文件全路径和明文文件保存全路径，指运行本接口的主机上的文件，非服务器端文件

7.3.24 消息签名

消息签名接口定义应符合表91的规定。

表91 消息签名接口定义

原型	BSTR SOf_SignMessage(BSTR InData)
描述	对字符串数据进行消息签名，返回Base64编码的带原文的数据类型B签名结果
参数	InData(原文)
返回值	Base64编码的消息签名值(成功)
	空串(失败，可通过SOf_GetLastError获取符合表A.1定义的错误代码)

7.3.25 验证消息签名

验证消息签名接口定义应符合表92的规定。

表92 验证消息签名接口定义

原型	LONG SOf_VerifySignedMessage(BSTR SignedMessage)
描述	验证消息签名值，消息签名值格式为Base64编码的带原文的数据类型B
参数	SignedMessage(Base64编码的消息签名值)
返回值	SAR_OK(成功)
	其他(失败，返回表A.1定义的错误代码)

7.3.26 不带原文的消息签名

不带原文的消息签名接口定义应符合表93的规定。

表93 不带原文的消息签名接口定义

原型	BSTR SOf_SignMessageDetach(BSTR InData)
描述	对字符串数据进行数字签名，返回Base64编码的不带原文的数据类型B签名结果
参数	InData(原文)
返回值	Base64编码的消息签名值(成功)
	空串(失败，可通过SOf_GetLastError获取符合表A.1定义的错误代码)

7.3.27 验证不带原文的消息签名

验证不带原文的消息签名接口定义应符合表94的规定。

表94 验证不带原文的消息签名接口定义

原型	LONG SOF_VerifySignedMessageDetach(BSTR InData, BSTR SignedMessage)
描述	验证消息签名值，签名值格式为Base64编码的不带原文的数据类型B
参数	InData(原文)
	SignedMessage(Base64编码的消息签名值)
返回值	SAR_OK(成功)
	其他(失败，返回表A.1定义的错误代码)

7.3.28 解析消息签名

解析消息签名接口定义应符合表95的规定。

表95 解析消息签名接口定义

原型	BSTR SOF_GetInfoFromSignedMessage(BSTR SignedMessage, SHORT Type)
描述	解析消息签名值中的信息，包括：原文、签名值、签名证书等，消息签名值格式为Base64编码的数据类型B
参数	SignedMessage(Base64编码的消息签名值)
	Type(类型), 取值范围： a) 1: 解析出原文； b) 2: 解析出Base64编码的签名者证书； c) 3: 解析出Base64编码的签名值
返回值	解析结果(成功)
	空串(失败或不存在该项)

7.3.29 XML 数据签名

XML 数据签名接口定义应符合表96的规定。

表96 XML 数据签名接口定义

原型	BSTR SOF_SignDataXML(BSTR InData)
描述	对XML数据进行数字签名，证书为RSA算法时签名结果参见RFC3275, 证书为SM2算法时签名结果应符合GB/T 25061的规定
参数	InData(XML格式的签名原文)
返回值	XML签名结果(成功)
	空串(失败，可通过SOF_GetLastError获取符合表A.1定义的错误代码)
安全规则	缺省配置和参数： a) 采用封皮签名，签名前应对签名原文做格式化； b) 格式化方法采用带注释的XML格式化1.1, 证书为RSA算法时标识符为http://www.w3.org/2006/12/xml-c14n11#WithComment, 证书为SM2算法时标识符为http://127.0.0.1/2006/12/xml-c14n11#WithComments

7.3.30 验证 XML 数据签名

验证 XML 数据签名接口定义应符合表97的规定。

表97 验证 XML 数据签名接口定义

原型	LONG SOF_VerifySignedDataXML(BSTR XMLSignedData)
描述	验证XML数据签名
参数	XMLSignedData(XML签名结果)
返回值	SAR_OK(成功)
	其他(失败, 返回表A.1定义的错误代码)

7.3.31 解析 XML数据签名

解析 XML 数据签名接口定义应符合表98的规定。

表98 解析 XML 数据签名接口定义

原型	BSTR SOF_GetXMLSignatureInfo(BSTR XMLSignedData, SHORT Type)
描述	解析XML数据签名, 获取签名值、XML原文、证书等信息
参数	XMLSignedData XML签名结果
	Type(待解析的参数类型), 取值范围: a) 1:解析出XML原文; b) 2:解析出摘要值; c) 3:解析出签名值; d) 4:解析出签名证书; e) 5:解析出摘要算法; f) 6:解析出签名算法
返回值	解析结果(成功)
	空串(失败, 可通过SOF_GetLastError获取符合表A.1定义的错误代码)
安全规则	XML签名缺省配置和参数值样例: a) XML原文取自Object元素; b) 摘要值取自DigestValue元素, 形如http://127.0.0.1/2001/04/xmldsig-more#sm3; c) 签名值取自SignatureValue元素, 为Base64编码; d) 签名证书取自X509Data元素, 为Base64编码; e) 摘要算法取自DigestMethod元素的Algorithm属性, 形如http://127.0.0.1/2001/04/xmldsig-more#sm3; f) 签名算法取自SignatureMethod元素的Algorithm属性, 形如http://127.0.0.1/2001/04/xmldsig-more#sm2-sm3。

7.3.32 创建时间戳请求

创建时间戳请求接口定义应符合表99的规定。

表99 创建时间戳请求接口定义

原型	BSTR SOf_CreateTimeStampRequest(BSTR InData, LONG HashAlg, SHORT ReqType, BSTR Base64Ext)
描述	创建时间戳请求
参数	InData(待创建时间戳请求的原文)
	HashAlg(摘要算法，应符合GB/T 33560的规定)
	ReqType(请求的时间戳服务类型), 取值范围： a) 0: 表示时间戳响应应包含时间戳服务器证书； b) 1: 表示时间戳响应不包含时间戳服务器证书
	Base64Ext(Base64编码格式的扩展项)
返回值	Base64编码格式的时间戳请求(成功)
	空串(失败，可通过SOf_GetLastError获取符合表A. 1定义的错误代码)

7.3.33 创建时间戳响应

创建时间戳响应接口定义应符合表100的规定。

表100 创建时间戳响应接口定义

原型	BSTR SOf_CreateTimeStampResponse(BSTR TSRequest)
描述	创建时间戳响应，即签发时间戳
参数	TSRequest(Base64编码格式的时间戳请求)
返回值	Base64编码格式的时间戳响应(成功)
	空串(失败，可通过SOf_GetLastError获取符合表A. 1定义的错误代码)

7.3.34 验证时间戳

验证时间戳接口定义应符合表101的规定。

表101 验证时间戳接口定义

原型	LONG SOf_VerifyTimeStamp(BSTR InData, BSTR TSResData, BSTR Base64Cert)
描述	验证时间戳
参数	InData(待验证的原文)
	TSResData(Base64编码格式的时间戳)
	Base64Cert(Base64编码格式的时间戳服务器证书，当时间戳响应中不包含时间戳服务器证书时应传入本参数)
返回值	SAR_OK(成功)
	其他(失败，返回表A. 1定义的错误代码)

7.3.35 解析时间戳

解析时间戳接口定义应符合表102的规定。

表102 解析时间戳接口定义

原型	BSTR SOf_GetTimeStampInfo(BSTR TSResData, SHORT Type)
描述	解析时间戳，获得时间戳的信息，包括时间、时间戳服务器证书等
参数	TSResData (Base64编码格式的时间戳)
	Type (信息类型), 取值范围： a) 1: 返回时间； b) 3: 返回时间戳服务器签名证书
返回值	解析结果 (成功)
	空串 (失败，可通过SOf_GetLastError获取符合表A.1定义的错误代码)

7.3.36 获取最新的错误代码

获取最新的错误代码接口定义应符合表103的规定。

表103 获取最新的错误代码接口定义

原型	LONG SOf_GetLastError()
描述	获取接口最新的错误代码
参数	无
返回值	错误代码，应符合表A.1错误代码表

7.3.37 计算数据摘要

计算数据摘要接口定义应符合表104的规定。

表104 计算数据摘要接口定义

原型	BSTR SOf_HashData(LONG HashAlg, BSTR InData, BSTR SignCert, BSTR UserID)
描述	计算数据摘要，若采用SM3算法，当SignCert和UsreID为空时，只计算数据的摘要值，当SignCert和UsreID值不为空时，应按照GB/T35276规定的预处理过程计算，摘要值可作为SM2签名的输入
参数	HashAlg (摘要算法，应符合GB/T33560的规定)
	Indata (原始数据)
	SignCert (Base64编码的签名者证书，当摘要算法为SM3时有效，如不需要传空串)
	UserID (签名者用户ID, 摘要算法为SM3时有效，若SignCert参数为空串，本参数无意义)
返回值	Base64编码的数据摘要值 (成功)
	空串 (失败，可通过SOf_GetLastError获取符合表A.1定义的错误代码)

7.3.38 计算文件摘要

计算文件摘要接口定义应符合表105的规定。

表105 计算文件摘要接口定义

原型	BSTR SOf_HashFile (LONG HashAlg, BSTR InFile, BSTR SignCert, BSTR UserID)
描述	计算文件数据摘要，若采用SM3算法，当SignCert和UsreID为空时，只计算数据的摘要值，当SignCert和UsreID值不为空时，应按照GB/T 35276规定的预处理过程计算，摘要值可作为SM2签名的输入
参数	HashAlg (摘要算法，应符合GB/T 33560的规定)
	InFile (原文文件全路径)
	SignCert (Base64编码的签名者证书，当摘要算法为SM3时有效，如不需要传空串)
	UserID (签名者用户ID, 摘要算法为SM3时有效，若SignCert参数为空串，本参数无意义)
返回值	Base64编码的文件摘要值 (成功)
	空串 (失败，可通过SOf_GetLastError获取符合表A.1定义的错误代码)
文件规则	原文文件全路径，指运行本接口的主机上的文件，非服务器端文件

7.3.39 摘要数据签名

摘要数据签名接口定义应符合表106的规定。

表106 摘要数据签名接口定义

原型	BSTR SOf_SignHashData (BSTR Base64HashData, LONG HashAlg)
描述	对数据摘要签名，返回Base64编码的数据类型A签名结果。 Base64HashData值一般是SOf_HashData或SOf_HashFile函数的计算结果。当采用SM2算法签名时，SM3算法的摘要值应按照GB/T 35276规定的预处理过程计算
参数	Base64HashData (Base64编码的摘要值)
	HashAlg (摘要算法，应符合GB/T33560的规定)
返回值	Base64编码的签名值 (成功)
	空串 (失败，可通过SOf_GetLastError获取符合表A.1定义的错误代码)

7.3.40 验证摘要数据签名

验证摘要数据签名接口定义应符合表107的规定。

表107 验证摘要数据签名接口定义

原型	BOOL SOf_VerifySignedHashData (BSTR Base64Cert, BSTR Base64HashData, BSTR SignValue, LONG HashAlg)
描述	数据摘要签名验证，签名值为Base64编码的数据类型A。 若使用SM2算法验证签名，Base64HashData参数是SM3算法的摘要值，应按照GB/T35276规定的预处理过程计算

表107 验证摘要数据签名接口定义（续）

参数	Base64Cert (Base64编码的签名者证书)
	Base64HashData (Base64编码的摘要值)
	SignValue (Base64编码的签名值)
	HashAlg (摘要算法，应符合GB/T33560的规定)
返回值	TRUE (成功)
	FALSE (失败，可通过SOF_GetLastError获取符合表A. 1定义的错误代码)

7.4 服务器端 Java 组件接口

7.4.1 服务器端Java 组件接口综述

服务器端Java 组件接口提供配置管理、证书解析、签名与验证、加密与解密数字信封、XML 数据签名与验证、时间戳等功能，共包含40个接口。服务器端Java 组件接口列表见表D. 4, 7. 4. 2~7. 4. 41 给出了接口详细定义。

服务器端Java 组件接口通过两种方式获得错误信息， 一种是通过 SOF_getLastError 获取错误代码，另一种是通过Java 捕获异常方式获取错误信息。本文件仅对使用SOF_getLastError 方式进行说明。

7.4.2 设置证书信任列表

设置证书信任列表接口定义应符合表108的规定。

表108 设置证书信任列表接口定义

原型	boolean SOF_setCertTrustList (java. lang. String ctlAltName, java. lang. String ctlContent)
描述	设置证书信任列表
参数	ctlAltName (证书信任列表别名)
	ctlContent (Base64编码格式的证书信任列表内容)
返回值	true (成功)
	false (失败，可通过SOF_getLastError获取符合表A. 1定义的错误代码)

7.4.3 查询证书信任列表别名

查询证书信任列表别名接口定义应符合表109的规定。

表109 查询证书信任列表别名接口定义

原型	java. lang. String SOF_getCertTrustListAltNames ()
描述	查询证书信任列表别名
参数	无
返回值	信任列表别名 (成功，返回信任列表别名的字符串组合，如 “CA001@CA002@CA003”)
	空串 (失败，可通过SOF_getLastError获取符合表A. 1定义的错误代码)

7.4.4 根据别名查询证书信任列表

根据别名查询证书信任列表接口定义应符合表110的规定。

表110 根据别名查询证书信任列表接口定义

原型	java.lang.String SOF_getCertTrustList(java.lang.String ctlAltName)
描述	根据别名查询证书信任列表
参数	ctlAltName(证书信任列表别名)
返回值	Base64编码的证书信任列表(成功)
	空串(失败, 可通过SOF_getLastError获取符合表A.1定义的错误代码)

7.4.5 删除证书信任列表

删除证书信任列表接口定义应符合表111的规定。

表111 删除证书信任列表接口定义

原型	boolean SOF_delCertTrustList(java.lang.String ctlAltName)
描述	根据别名删除证书信任列表
参数	ctlAltName(证书信任列表别名)
返回值	true(成功)
	false(失败, 可通过SOF_getLastError获取符合表A.1定义的错误代码)

7.4.6 获取指定应用的实例

获取指定应用的实例接口定义应符合表112的规定。

表112 获取指定应用的实例接口定义

原型	static java.lang.Object SOF_getInstance(java.lang.String policyName)
描述	初始化接口, 通过应用别名获取实例, 应用别名关联所配置的证书、密钥、信任证书链、算法类型、CRL及证书验证策略等
参数	policyName(应用策略名称)
返回值	应用示例对象(成功)
	空对象(失败, 可通过SOF_getLastError获取符合表A.1定义的错误代码)

7.4.7 设置签名算法

设置签名算法接口定义应符合表113的规定。

表113 设置签名算法接口定义

原型	void SOf_setSignMethod(long signMethod)
描述	设置Java组件签名运算使用的签名算法
参数	signMethod(签名算法标识，应符合GB/T33560的规定)
返回值	无

7.4.8 获得签名算法

获得签名算法接口定义应符合表114的规定。

表114 获得签名算法接口定义

原型	long SOf_getSignMethod()
描述	获得Java组件当前签名和验签运算使用的签名算法标识
参数	无
返回值	当前使用的签名算法标识(成功)
	0(当前没有设置签名算法)

7.4.9 设置加密算法

设置加密算法接口定义应符合表115的规定。

表115 设置加密算法接口定义

原型	void SOf_setEncryptMethod(long encryptMethod)
描述	设置组件对数据加密使用的对称算法标识
参数	encryptMethod(对称密码算法标识，应符合GB/T33560的规定。本接口支持不带附加认证数据的加密算法)
返回值	无

7.4.10 获得加密算法

获得加密算法接口定义应符合表116的规定。

表116 获得加密算法接口定义

原型	long SOf_getEncryptMethod()
描述	获得组件当前使用的对称算法标识
参数	无
返回值	当前使用的加密算法标识(成功)
	0(当前没有设置加密算法)

7.4.11 获得服务器证书

获得服务器证书接口定义应符合表117的规定。

表117 获得服务器证书接口定义

原型	java.lang.String S0F_getServerCertificate()
描述	读取当前应用的服务器的签名证书。若有签名证书则得到签名证书，否则得到加密证书
参数	无
返回值	Base64编码的服务器证书(成功)
	空串(失败，可通过S0F_getLastError获取符合表A.1定义的错误代码)

7.4.12 获得指定密钥用途的服务器证书

获得指定密钥用途的服务器证书接口定义应符合表118的规定。

表118 获得指定密钥用途的服务器证书接口定义

原型	java.lang.String S0F_getServerCertificateByUsage(short certUsage)
描述	根据密钥用途，读取当前应用指定的服务器证书
参数	certUsage(证书用途),取值范围： a) 1:加密证书； b) 2:签名证书
返回值	Base64编码的服务器证书(成功)
	空串(失败，可通过S0F_getLastError获取符合表A.1定义的错误代码)

7.4.13 产生随机数

产生随机数接口定义应符合表119的规定。

表119 产生随机数接口定义

原型	java.lang.String S0F_genRandom(short randomLen)
描述	产生指定长度的随机数
参数	randomLen(待产生的随机数字节长度)
返回值	Base64编码的随机数(成功)
	空串(失败，可通过S0F_getLastError获取符合表A.1定义的错误代码)

7.4.14 获得证书信息

获得证书信息接口定义应符合表120的规定。

表120 获得证书信息接口定义

原型	java.lang.String SOF_getCertInfo(java.lang.String base64Cert,int type)
描述	根据type解析证书内的相关信息
参数	base64Cert(Base64编码的证书)
	type(证书解析标识, 应符合GB/T 33560的规定)
返回值	指定类型的证书信息(成功)
	空串(失败或证书中不存在该项内容)

7.4.15 获得证书扩展信息

获得证书扩展信息接口定义应符合表121的规定。

表121 获得证书扩展信息接口定义

原型	java.lang.String SOF_getCertInfoByOid(java.lang.String base64Cert, java.lang.String oid)
描述	根据OID获取证书私有扩展项信息
参数	base64Cert(Base64编码的证书)
	oid(私有扩展对象ID, 如“1.2.156.xxx”)
返回值	证书私有扩展项OID对应的信息(成功)
	空串(失败或证书中不存在该私有扩展项)

7.4.16 验证证书有效性

验证证书有效性接口定义应符合表122的规定。

表122 验证证书有效性接口定义

原型	int SOF_validateCert(java.lang.String base64Cert)
描述	根据应用的策略验证证书有效性
参数	base64Cert(待验证的Base64编码证书)
返回值	SAR_OK(验证成功, 证书有效)
	其他(验证失败, 失败原因应符合表A.1中的错误代码范围0X0B000500~0X0B000505)

7.4.17 数据签名

数据签名接口定义应符合表123的规定。

表123 数据签名接口定义

原型	java.lang.String SOF_signData(byte[] inData)
描述	对字符串数据进行数字签名，返回Base64编码的数据类型A签名结果
参数	inData(原文)
返回值	Base64编码的签名值(成功)
	空串(失败，可通过SOF_getLastError获取符合表A.1定义的错误代码)

7.4.18 验证数据签名

验证数据签名接口定义应符合表124的规定。

表124 验证数据签名接口定义

原型	boolean SOF_verifySignedData(java.lang.String base64Cert,byte[] inData, java.lang.String signValue)
描述	验证数字签名，签名值格式为Base64编码的数据类型A
参数	base64Cert(Base64编码的签名证书)
	inData(原文)
	signValue(Base64编码的签名值)
返回值	true(成功)
	alse(失败，可通过SOF_getLastError获取符合表A.1定义的错误代码)

7.4.19 文件签名

文件签名接口定义应符合表125的规定。

表125 文件签名接口定义

原型	java.lang.String SOF_signFile(java.lang.String inFile)
描述	对文件数字签名，返回Base64编码的数据类型A签名结果
参数	inFile(待签名的文件全路径)
返回值	Base64编码的签名值(成功)
	空串(失败，可通过SOF_getLastError获取符合表A.1定义的错误代码)
文件规则	待签名的文件全路径，指运行本接口的主机上的文件，非服务器端文件

7.4.20 验证文件签名

验证文件签名接口定义应符合表126的规定。

表126 验证文件签名接口定义

原型	boolean SOF_verifySignedFile(java.lang.String base64Cert java.lang.String inFile, java.lang.String signValue)
描述	验证文件数字签名，签名值格式为Base64编码的数据类型A
参数	base64Cert (Base64编码的签名证书)
	inFile (待验证的文件路径)
	signValue (Base64编码的签名值)
返回值	true (成功)
	false (失败，可通过SOF_getLastError获取符合表A. 1定义的错误代码)
文件规则	待签名的文件全路径，指运行本接口的主机上的文件，非服务器端文件

7.4.21 数据加密

数据加密接口定义应符合表127的规定。

表127 数据加密接口定义

原型	java.lang.String SOF_encryptData(java.lang.String baseCert,byte[]inData)
描述	数字信封加密，加密过程为使用临时产生的对称密钥加密数据，然后使用数字证书的公钥加密对称密钥，返回Base64编码的数据类型B的密文数据
参数	baseCert (Base64编码的数据接收者的加密证书，如有多个接收者，多个接收者证书之间用&. &. &. 作为分隔符连接)
	inData (待加密的明文数据)
返回值	Base64编码格式的密文数据 (成功)
	空串 (失败，可通过SOF_getLastError获取符合表A. 1定义的错误代码)

7.4.22 数据解密

数据解密接口定义应符合表128的规定。

表128 数据解密接口定义

原型	byte[]SOF_decryptData(java.lang.String certId,java.lang.String inData)
描述	解密Base64编码的数据类型B的密文数据
参数	certId (解密密钥对应的证书唯一标识，如不需要可传空串)
	inData (Base64编码的待解密的密文数据)
返回值	解密后的明文数据 (成功)
	空串 (失败，可通过SOF_getLastError获取符合表A. 1定义的错误代码)

7.4.23 文件加密

文件加密接口定义应符合表129的规定。

表129 文件加密接口定义

原型	<code>boolean SOf_encryptFile(java.lang.String base64Cert, java.lang.String inFile, java.lang.String outFile)</code>
描述	加密文件，得到数据类型B的密文文件
参数	base64Cert (Base64编码的数据接收者的加密证书，如有多个接收者，多个接收者证书之间用8.&.作为分隔符连接)
	inFile(待加密的明文文件全路径)
	outFile(密文文件保存全路径)
返回值	true(成功)
	false(失败，可通过SOf_getLastError获取符合表A.1定义的错误代码)
文件规则	待加密的明文文件全路径和密文文件保存全路径，指运行本接口的主机上的文件，非服务器端文件

7.4.24 文件解密

文件解密接口定义应符合表130的规定。

表130 文件解密接口定义

原型	<code>boolean SOf_decryptFile(java.lang.String certId. java.lang.String inFile, java.lang.String outFile)</code>
描述	解密密文文件，密文文件格式为数据类型B
参数	certId(解密密钥对应的证书唯一标识，如不需要可传空串)
	inFile(待解密的密文文件路径)
	outFile(明文文件保存路径)
返回值	true(成功)
	false(失败，可通过SOf_getLastError获取符合表A.1定义的错误代码)
文件规则	待解密的密文文件全路径和明文文件保存全路径，指运行本接口的主机上的文件，非服务器端文件

7.4.25 消息签名

消息签名接口定义应符合表131的规定。

表131 消息签名接口定义

原型	java.lang.String SOF_signMessage(byte[] inData)
描述	对字符串数据进行消息签名，返回Base64编码的带原文的数据类型B签名结果
参数	inData(原文)
返回值	Base64编码的消息签名值(成功)
	空串(失败，可通过SOF_getLastError获取符合表A.1定义的错误代码)

7.4.26 验证消息签名

验证消息签名接口定义应符合表132的规定。

表132 验证消息签名接口定义

原型	boolean SOF_verifySigned Message(java.lang.String signedMessage)
描述	验证消息签名，签名格式为Base64编码的带原文的数据类型B
参数	signedMessage(Base64编码的消息签名值)
返回值	true(成功
	false(失败，可通过SOF_getLastError获取符合表A.1定义的错误代码)

7.4.27 解析消息签名

解析消息签名接口定义应符合表133的规定。

表133 解析消息签名接口定义

原型	byte[]SOF_getInfoFromSignedMessage(java.lang.String signedMessage,short type)
描述	解析Base64编码的带原文的数据类型B签名值的信息，可获得原文、签名值、签名证书等信息
参数	signedMessage(Base64编码的消息签名值)
	Type(类型),取值范围： a) 1:解析出原文； b) 2:解析出签名者证书； c) 3:解析出签名值
返回值	解析结果(成功)
	空串(失败或不存在该项)

7.4.28 不带原文的消息签名

不带原文的消息签名接口定义应符合表134的规定。

表134 不带原文的消息签名接口定义

原型	java.lang.String SOF_signMessageDetach(byte[] inData)
描述	对字符串数据进行消息签名，返回Base64编码的不带原文的数据类型B的签名结果
参数	inData(原文)
返回值	Base64编码的消息签名值(成功)
	空串(失败，可通过SOF_getLastError获取符合表A.1定义的错误代码)

7.4.29 验证不带原文的消息签名

验证不带原文的消息签名接口定义应符合表135的规定。

表135 验证不带原文的消息签名接口定义

原型	boolean SOF_verifySignedMessageDetach(byte[] inData, java.lang.String signedMessage)
描述	验证签名格式为Base64编码的不带原文数据类型B的数字签名
参数	inData(原文)
	signedMessage(Base64编码的签名值)
返回值	true(成功)
	false(失败，可通过SOF_getLastError获取符合表A.1定义的错误代码)

7.4.30 XML 数据签名

XML 数据签名接口定义应符合表136的规定。

表136 XML 数据签名接口定义

原型	java.lang.String SOF_signDataXML(java.lang.String inData)
描述	对XML数据进行数字签名，证书为RSA算法时签名结果参见RFC3275, 证书为SM2算法时签名结果应符合GB/T 25061的规定
参数	InData(XML签名原文)
返回值	XML签名结果(成功)
	空串(失败，可通过SOF_getLastError获取符合表A.1定义的错误代码)
安全规则	缺省配置和参数： a) 采用封皮签名，签名前应对签名原文做格式化； b) 格式化方法采用带注释的XML格式化1.1, 证书为RSA算法时标识符为http://www.w3.org/2006/12/xml-c14n11#WithComment, 证书为SM2算法时标识符为http://127.0.0.1/2006/12/xml-c14n11#WithComments

7.4.31 验证 XML 数据签名

验证 XML 数据签名接口定义应符合表137的规定。

表137 验证 XML数据签名接口定义

原型	boolean SOF_verifySignedDataXML(java.lang.String xmlSignedData)
描述	验证XML数据签名
参数	xmlSignedData(XML签名结果)
返回值	true(成功)
	false(失败, 可通过SOF_getLastError获取符合表A. 1定义的错误代码)

7.4.32 解析 XML数据签名

解析 XML 数据签名接口定义应符合表138的规定。

表138 解析 XML数据签名接口定义

原型	java.lang.String SOF_getXMLSignatureInfo(java.lang.String xmlSignedData, short type)
描述	解析XML数据签名, 获取签名值、XML原文、证书等信息
参数	xmlSignedData(XML签名结果)
	type(待解析的参数类型), 取值范围: a) 1: 解析出XML原文; b) 2: 解析出摘要值; c) 3: 解析出签名值; d) 4: 解析出签名证书; e) 5: 解析出摘要算法; f) 6: 解析出签名算法
返回值	解析结果(成功)
	空串(失败, 可通过SOF_getLastError获取符合表A. 1定义的错误代码)
安全规则	XML签名缺省配置和参数值样例: a) XML原文取自Object元素; b) 摘要值取自DigestValue元素, 形如http://127.0.0.1/2001/04/xmldsig-more#sm3; c) 签名值取自SignatureValue元素, 为Base64编码; d) 签名证书取自X509Data元素, 为Base64编码; e) 摘要算法取自DigestMethod元素的Algorithm属性, 形如http://127.0.0.1/2001/04/xmldsig-more#sm3; f) 签名算法取自SignatureMethod元素的Algorithm属性, 形如http://127.0.0.1/2001/04/xmldsig-more#sm2-sm3

7.4.33 创建时间戳请求

创建时间戳请求接口定义应符合表139的规定。

表139 创建时间戳请求接口定义

原型	java.lang.String SOF_createTimeStampRequest(byte[]inData,long hashAlg,short reqType,byte[]extension)
描述	创建时间戳请求
参数	inData(待创建时间戳请求的原文)
	hashAlg(摘要算法，应符合GB/T33560的规定)
	reqType(请求的时间戳服务类型),取值范围： a)0:表示时间戳响应应包含时间戳服务器证书； b)1:表示时间戳响应不包含时间戳服务器证书
	extension(扩展项)
返回值	Base64编码格式的时间戳请求(成功)
	空串(失败，可通过SOF_getLastError获取符合表A.1定义的错误代码)

7.4.34 创建时间戳响应

创建时间戳响应接口定义应符合表140的规定。

表140 创建时间戳响应接口定义

原型	java.lang.String SOF_createTimeStampResponse(java.lang.String tsRequest)
描述	创建时间戳响应，即签发时间戳
参数	tsRequest(Base64编码的时间戳请求)
返回值	Base64编码格式的时间戳响应(成功)
	空串(失败，可通过SOF_getLastError获取符合表A.1定义的错误代码)

7.4.35 验证时间戳

验证时间戳接口定义应符合表141的规定。

表141 验证时间戳接口定义

原型	boolean SOF_verify TimeStamp(byte[]inData java.lang.String tsResData, java.lang.String base64Cert)
描述	验证时间戳
参数	inData(待验证的原文)
	tsResData(Base64编码格式的时间戳)
	base64Cert(Base64编码格式的时间戳服务器证书，在时间戳响应中不包含时间戳服务器证书时应传入本参数)
返回值	true(成功
	false(失败，可通过SOF_getLastError获取符合表A.1定义的错误代码)

7.4.36 解析时间戳

解析时间戳接口定义应符合表142的规定。

表142 解析时间戳接口定义

原型	java.lang.String SOf_getTimeStampInfo(java.lang.String tsResData, short type)
描述	解析时间戳，获得时间戳的信息，包括时间、时间戳服务器证书等
参数	tsResData (Base64编码的时间戳)
	type (信息类型), 取值范围: a) 1: 返回时间; b) 3: 返回时间戳服务器签名证书
返回值	解析结果 (成功)
	空串 (失败, 可通过SOf_getLastError获取符合表A.1定义的错误代码)

7.4.37 获取最新的错误代码

获取最新的错误代码接口定义应符合表143的规定。

表143 获取最新的错误代码接口定义

原型	long SOf_getLastError()
描述	获取接口最新的错误代码
参数	无
返回值	错误代码, 应符合表A.1错误代码表

7.4.38 计算数据摘要

计算数据摘要接口定义应符合表144的规定。

表144 计算数据摘要接口定义

原型	java.lang.String SOf_hashData(long hashAlg, byte[] inData, byte[] signCert, byte[] userID)
描述	计算数据摘要, 若采用SM3算法, 当signCert和userID为空时, 只计算数据的摘要值, 当signCert和userID值不为空时, 应按照GB/T35276规定的预处理过程计算, 摘要值可作为SM2签名的输入
参数	hashAlg (摘要算法, 应符合GB/T33560的规定)
	inData (原始数据)
	signCert (签名者证书, 当摘要算法为SM3时有效, 如不需要传null)
	userID (签名者用户ID, 当摘要算法为SM3时有效, 若signCert为null, 本参数无意义)
返回值	Base64编码的数据摘要值 (成功)
	空串 (失败, 可通过SOf_getLastError获取符合表A.1定义的错误代码)

7.4.39 计算文件摘要

计算文件摘要接口定义应符合表145的规定。

表145 计算文件摘要接口定义

原型	java.lang.String SOF_hashFile(long hashAlg, java.lang.String inFile,byte[] signCert,byte[]userID)
描述	计算文件数据摘要，若采用SM3算法，当signCert和userID为空时，只计算数据的摘要值，当signCert和userID值不为空时，应按照GB/T35276规定的预处理过程计算，摘要值可作为SM2签名的输入
参数	hashAlg(摘要算法，应符合GB/T 33560的规定)
	inFile(原文文件全路径)
	signCert(签名者证书，当摘要算法为SM3时有效，如不需要传null)
	userID(签名者用户ID,当摘要算法为SM3时有效，若signCert为null,本参数无意义)
返回值	Base64编码的文件摘要值(成功)
	空串(失败，可通过SOF_getLastError获取符合表A.1定义的错误代码)
文件规则	原文文件全路径，指运行本接口的主机上的文件，非服务器端文件

7.4.40 摘要数据签名

摘要数据签名接口定义应符合表146的规定。

表146 摘要数据签名接口定义

原型	java.lang.String SOF_signHashData(java.lang.String base64HashData,long hashAlg)
描述	对数据摘要签名，返回Base64编码的数据类型A的签名结果。 base64HashData参数一般是SOF_HashData或SOF_HashFile函数的计算结果。当采用SM2算法签名时，SM3算法的摘要值应按照GB/T 35276规定的预处理过程计算
参数	base64 HashData(Base64编码的签名摘要值)
	hashAlg(摘要算法，应符合GB/T 33560的规定)
返回值	Base64编码的签名值(成功)
	空串(失败，可通过SOF_getLastError获取符合表A.1定义的错误代码)

7.4.41 验证摘要数据签名

验证摘要数据签名接口定义应符合表147的规定。

表147 验证摘要数据签名接口定义

原型	boolean SOF_verifySigned HashData(java.lang.String base64Cert, java.lang.String base64HashData,java.lang.String signValue,long hashAlg)
描述	数据摘要签名验证，签名值为Base64编码的数据类型A。 若使用SM2算法验证签名，base64HashData参数是SM3算法的摘要值，应按照GB/T35276规定的预处理过程计算

表147 验证摘要数据签名接口定义（续）

参数	base64Cert (Base64编码的签名者证书)
	base64HashData (Base64编码的摘要值)
	signValue (Base64编码的签名值)
	hashAlg (摘要算法，应符合GB/T 33560的规定)
返回值	true (成功)
	false (失败，可通过SOF_getLastError获取符合表A.1定义的错误代码)

8 接口验证方法

8.1 验证环境

接口提供方实现第7章规定的接口，提供测试程序和测试程序源代码。接口需求方可选择使用接口提供方提供的测试程序验证接口，也可自编写测试程序来验证接口。接口提供方可根据设备类型选择性提供对应的接口，客户端产品仅提供第7章规定的客户端服务接口，服务器端产品仅提供第7章规定的服务器端服务接口。要验证接口，需具备以下验证环境：

- a) 运行客户端服务接口所需的操作系统及测试程序软件；
- b) 运行服务器端服务接口所需的操作系统及测试程序软件；
- c) 证书认证中心给客户端签发对应的客户端证书，证书和密钥需存放在符合要求的存储介质中（如智能密码钥匙）；
- d) 证书认证中心给服务器端签发对应的服务器证书，并导入到服务器端密码设备中。

8.2 验证原则

8.2.1 接口原型验证

符合性验证内容及原则要求如下：

- a) 接口名称应符合第7章的要求；
- b) 接口参数个数及参数类型应符合第7章的要求；
- c) 接口返回值类型应符合第7章的要求。

8.2.2 接口逻辑验证

符合性验证内容及原则要求如下：

- a) 输入正确类型及格式的参数的接口应返回符合本文件规定的的数据；
- b) 输入错误类型或格式的参数的接口应返回符合本文件规定的错误代码或抛出异常。

8.2.3 接口互通性验证

符合性验证内容及原则要求如下：

- a) 客户端接口返回的数据应被服务器端所对应的接口验证正确性；
- b) 服务器端接口返回的数据应被客户端所对应的接口验证正确性；
- c) 本文件规定的接口返回的数据应被其他设备接口或通用密码服务接口验证正确性。

8.3 验证场景

8.3.1 接口功能验证

8.3.1.1 验证目的

接口功能验证目的是检测接口提供方所提供的第7章规定接口的所有功能，验证前需具备8.1所描述的验证环境。

8.3.1.2 验证过程

对第7章规定的所有接口功能进行验证，此处以客户端COM 组件接口为例说明，其他三种类型的接口验证过程类似。验证过程可参见如下步骤：

- a) 调用 SOF_GetVersion 获取版本号；
- b) 调用 SOF_SetSignMethod 设置签名算法；
- c) 调用SOF_GetSignMethod 获得当前签名算法，检查是否和设置的签名算法一致；
- d) 调用SOF_SetEncryptMethod 设置加密算法；
- e) 调用 SOF_GetEncryptMethod 获得加密算法，检查是否和设置的签名算法一致；
- f) 调用 SOF_GetUserList 获得证书列表，检查是否和当前存在证书列表一致；
- g) 调用 SOF_ExportUserCert 导出用户证书，检查证书是否符合GB/T 20518的规定；
- h) 调用SOF_Login 校证书口令，检查是否正确的证书口令校验成功，错误的证书口令校验失败；
- i) 调用 SOF_IsLogin, 检查用户登录状态是否正确；
- j) 调用 SOF_Logout, 清空用户登录状态；
- k) 调用 SOF_GetPinRetryCount 获取用户认证口令剩余重试次数，检查是否和介质用户认证口令剩余重试次数一致；
- l) 调用 SOF_ChangePassWd 修改证书口令，修改成功后使用新口令再次调用SOF_Login；
- m) 调用SOF_ExportExChangeUserCert 导出用户加密证书，检查证书是否符合GB/T 20518的规定；
- n) 调用SOF_GetCertInfo 获得证书信息，检查获取到的证书信息是否正确；
- o) 调用SOF_GetCertInfoByOid 获得证书扩展信息，检查获取到的证书扩展信息是否正确；
- p) 调用 SOF_GetDeviceInfo 获得设备信息，检查获取到的设备信息是否正确；
- q) 调用 SOF_ValidateCert 验证证书有效性，检查返回的结果是否正确；
- r) 调用SOF_SignData 对数据签名，检查签名结果是否符合数据类型 A；
- s) 调用SOF_VerifySignedData 验证数据签名，验签使用的证书通过调用SOF_ExportUserCert 得到，检查返回结果是否正确；
- t) 调用SOF_SignFile 对文件签名，检查签名结果是否符合数据类型 A；
- u) 调用 SOF_VerifySignedFile 验证文件签名，验签使用的证书通过调用SOF_ExportUserCert 得到，检查返回结果是否正确；
- v) 调用SOF_EncryptData 加密数据，加密使用的证书通过调用SOF_ExportExChangeUserCert 得到，检查加密结果是否符合数据类型 B；
- w) 调用SOF_DecryptData 解密数据，检查解密结果是否和加密前的数据一致；
- x) 调用SOF_SignMessage 对消息签名，检查签名结果是否符合数据类型 B；
- y) 调用 SOF_VerifySignedMessage 验证消息签名，检查返回结果是否正确；
- z) 调用SOF_GetInfoFromSignedMessage 解析消息签名，检查返回结果是否正确；

- aa) 调用 SOF_SignDataXML 对 XML 数据签名, 检查结果是否符合 RFC3275 或 GB/T 25061 的规定;
- ab) 调用 SOF_VerifySignedDataXML 验证 XML 数字签名, 检查返回结果是否正确;
- ac) 调用 SOF_GetXMLSignatureInfo 解析 XML 签名数据, 检查返回结果是否正确;
- ad) 调用 SOF_GenRandom 产生随机数, 检查返回的随机数;
- ae) 调用 SOF_HashData 计算数据摘要, 检查返回结果是否正确;
- af) 调用 SOF_HashFile 计算文件摘要, 检查返回结果是否正确;
- ag) 调用 SOF_SignHashData 签名摘要值, 检查签名结果是否符合数据类型 A;
- ah) 调用 SOF_VerifySignedHashData 验证摘要值签名, 验签使用的证书通过调用 SOF_ExportUserCert 得到, 检查返回结果是否正确;
- ai) 上述调用过程任何一个接口出错时, 都调用 SOF_GetLastError 获取最新的错误代码, 检查获取到的错误码是否符合附录 A。

8.3.1.3 通过标准

每个接口输入正确的参数可返回正确结果, 输入错误的参数应返回符合附录 A 规定的错误代码。

8.3.2 证书登录认证场景

8.3.2.1 验证目的

检测服务器端接口和客户端接口配合使用证书进行业务系统登录认证的场景。

8.3.2.2 验证过程

证书登录认证场景验证过程如下。

- a) 客户端打开登录界面, 服务器端产生挑战随机数并发送给客户端, 步骤如下:
 - 1) 服务器端调用 SOF_GenRandom 产生挑战随机数 Rs;
 - 2) 将挑战随机数 Rs 和服务端认证标识 S 返回给客户端, 其中服务端认证标识 S 应是客户端可容易获取并认可的, 如域名;
 - 3) 将挑战随机数 Rs 存放在会话中。
- b) 客户端列举当前存在的证书, 步骤如下:
 - 1) 客户端调用 SOF_GetUserList 获取证书列表;
 - 2) 解析证书列表, 将证书名称显示在界面上供用户选择。
- c) 用户选择证书, 输入证书口令进行登录认证并提交给服务器端, 步骤如下:
 - 1) 用户选择对应的证书并输入证书口令;
 - 2) 客户端调用 SOF_Login 校证书口令;
 - 3) 若校证书口令失败, 调用 SOF_GetPinRetryCount 获取重试次数, 否则继续;
 - 4) 客户端调用 SOF_ExportUserCert 获取签名证书;
 - 5) 客户端调用 SOF_GenRandom 产生随机数 Rc
 - 6) 客户端调用 SOF_SignData 对“Rs||Rc|| 服务端认证标识 S”签名;
 - 7) 客户端将签名证书、客户端产生的随机数 Rc、客户端的签名作为客户端认证数据提交给服务器端。
- d) 服务器端验证客户端的认证数据, 步骤如下:
 - 1) 服务器端调用 SOF_VerifySignedData 验证客户端认证数据, 注意此处验签所使用的 Rs 应是服务器端在会话中保持的挑战随机数;

- 2) 服务器端调用SOF_ValidateCert 验证客户端证书有效性;
- 3) 服务器端调用SOF_GetCertInfo 解析客户端证书信息, 如即将过期则给用户提示;
- 4) 服务器端调用SOF_GetCertInfoByOid 获取证书唯一标识, 获取该用户的操作权限并展示。

8.3.2.3 通过标准

符合条件的证书用户可正常登录, 不符合条件的证书用户应给出正确提示。

8.3.3 业务数据签名场景

8.3.3.1 验证目的

检测服务器端接口和客户端接口配合使用证书进行业务数据签名验签的场景。

8.3.3.2 验证过程

本场景验证的前提是已按照8.3.2成功完成基于证书的登录认证过程。验证过程如下。

- a) 服务器端展示业务数据提交界面。
- b) 用户确认业务数据。
- c) 客户端对业务数据签名, 可业务不同调用不同的接口:
 - 若业务数据是普通数据, 调用 SOF_SignData 接口;
 - 若业务数据存放在文件中, 调用SOF_SignFile 接口;
 - 若业务数据是 XML 格式, 调用 SOF_SignDataXML 接口;
 - 若业务数据是消息, 调用SOF_SignMessage 接口;
 - 若业务数据仅以 Hash 方式提交, 先调用 SOF_HashData 或 SOF_HashFile, 再调用 SOF_SignHashData 接口;
- d) 将业务数据和签名值通过加密信道提交到服务器端。
- e) 服务器端验证业务数据签名, 采用和客户端签名对应的验证接口:
 - 若业务数据是普通数据, 调用SOF_VerifySignedData 接口;
 - 若业务数据存放在文件中, 调用SOF_VerifySignedFile 接口;
 - 若业务数据是 XML 格式, 调用 SOF_SignDataXML 接口;
 - 若业务数据是消息, 调用 SOF_VerifySignedMessage 或 SOF_VerifySignedMessageDetach 接口;
 - 若业务数据仅以 Hash 方式提交, 调用 SOF_VerifySignedHashData 接口。
- f) 服务器端将业务数据验证处理结果展示给用户。

8.3.3.3 通过标准

检测得到预期结果。

8.3.4 业务数据加密场景

8.3.4.1 验证目的

检测服务器端接口和客户端接口配合使用证书进行业务数据加密的场景。

8.3.4.2 验证过程

本场景验证的前提是已按照8.3.2成功完成基于证书的登录认证过程。验证过程如下:

- a) 服务器端展示业务数据提交界面，且将服务端加密证书发送给客户端；
- b) 客户端调用SOF_ValidateCert 接口验证服务端证书有效性，如验证失败则给用户提醒；
- c) 用户确认业务数据；
- d) 客户端调用SOF_EncryptData 对业务数据加密，此时使用服务器端加密证书；
- e) 将加密后的业务数据提交到服务器端；
- f) 服务器调用SOF_DecryptData 解密业务加密数据；
- g) 服务器端将业务数据验证处理结果展示给用户。

8.3.4.3 通过标准

检测得到预期结果。

8.3.5 数据加盖时间戳场景

8.3.5.1 验证目的

检测服务器端接口对业务数据加盖时间戳功能。

8.3.5.2 验证过程

本场景验证的前提是已按照8.3.2成功完成基于证书的登录认证过程。验证过程如下：

- a) 客户端将按照8.3.4的过程将待加盖时间戳的数据提交到服务端；
- b) 服务端调用 SOF_CreateTimeStampRequest 获取时间戳请求；
- c) 服务端调用 SOF_CreateTimeStampResponse 对数据加盖时间戳，其中输入参数是 SOF_CreateTimeStampRequest 的返回值；
- d) 服务端调用 SOF_VerifyTimeStamp 验证时间戳；
- e) 服务端调用 SOF_GetTimeStampInfo 解析时间戳，将解析结果展示给用户；
- f) 用户确认后，服务端将 SOF_CreateTimeStampResponse 的返回值返回给客户端，此数据是对原始业务数据加盖时间戳的结果。

8.3.5.3 通过标准

检测得到预期结果。

附 录 A
(规范性)
证书应用综合服务接口错误代码定义

本文件所定义的错误代码符合 GM/T 0094—2020 错误代码区间划分要求。COM 组件错误代码定义见表 A.1, JAVA 组件异常信息定义见表 A.2。

表 A.1 COM 组件错误代码定义

宏描述	预定义值	说明
SOR_OK	0	成功
SOR_UnknownErr	0X0B000001	异常错误
SOR_NotSupportYetErr	0X0B000002	不支持的服务
SOR_FileErr	0X0B000003	文件操作错误
SOR_ProviderTypeErr	0X0B000004	服务提供者参数类型错误
SOR_LoadProviderErr	0X0B000005	导入服务提供者接口错误
SOR_LoadDevMngApiErr	0X0B000006	导入设备管理接口错误
SOR_AlgoTypeErr	0X0B000007	算法类型错误
SOR_NameLenErr	0X0B000008	名称长度错误
SOR_KeyUsageErr	0X0B000009	密钥用途错误
SOR_ModulusLenErr	0X0B000010	模的长度错误
SOR_NotInitializeErr	0X0B000011	未初始化
SOR_ObjErr	0X0B000012	对象错误
SOR_MemoryErr	0X0B000100	内存错误
SOR_TimeoutErr	0X0B000101	服务超时
SOR_IndataLenErr	0X0B000200	输入数据长度错误
SOR_IndataErr	0X0B000201	输入数据错误
SOR_GenRandErr	0X0B000300	生成随机数错误
SOR_HashObjErr	0X0B000301	HASH对象错
SOR_HashErr	0X0B000302	HASH运算错误
SOR_GenRsaKeyErr	0X0B000303	产生RSA密钥错
SOR_RsaModulusLenErr	0X0B000304	RSA密钥模长错误
SOR_CsplmprtPubKeyErr	0X0B000305	CSP服务导入公钥错误
SOR_RsaEncErr	0X0B000306	RSA加密错误
SOR_RSGDecErr	0X0B000307	RSA解密错误
SOR_HashNotEqualErr	0X0B000308	HASH值不相等

表 A.1 COM 组件错误代码定义（续）

宏描述	预定义值	说明
SOR_KeyNotFountErr	0X0B000309	密钥未发现
SOR_CertNotFountErr	0X0B000310	证书未发现
SOR_NotExportErr	0X0B000311	对象未导出
SOR_VeryPolicyErr	0X0B000312	未能完全按照策略验证成功
SOR_DecryptPadErr	0X0B000400	解密时做补丁错误
SOR_MacLenErr	0X0B000401	MAC长度错误
SOR_KeyInfoTypeErr	0X0B000402	密钥类型错误
SOR_NULLPointerErr	0X0B000403	某一个参数为空指针
SOR_APPNotFoundErr	0X0B000404	没有找到该应用
SOR_CERTENCODERErr	0X0B000405	证书编码格式错误
SOR_CERTINVALIDErr	0X0B000406	证书无效，不是可信CA颁发的证书
SOR_CERTHASEXPIREDErr	0X0B000407	证书已过期
SOR_CERTREVOKEDErr	0X0B000408	证书已经被吊销
SOR_SIGNDATAErr	0X0B000409	签名失败
SOR_VERIFYSIGNDATAErr	0X0B000410	验证签名失败
SOR_READFILEErr	0X0B000411	读文件异常，可能文件不存在或没有读取权限等
SOR_WRITEFILEErr	0X0B000412	写文件异常，可能文件不存在或没有写权限等
SOR_SECRETSEGMENTErr	0X0B000413	门限算法密钥分割失败
SOR_SECERTRECOVERYErr	0X0B000414	门限恢复失败
SOR_ENCRYPTDATAErr	0X0B000415	对数据的对称加密失败
SOR_DECRYPTDATAErr	0X0B000416	对称算法的数据解密失败
SOR_PKCS7ENCODERErr	0X0B000417	PKCS7编码格式错误
SOR_XMLENCODERErr	0X0B000418	不是合法的xml编码数据
SOR_PARAMETERNOTSUPPORTErr	0X0B000419	不支持的参数
SOR_CTLNOTFOUND	0X0B000420	没有发现信任列表
SOR_APPNOTFOUND	0X0B000421	设置的应用名称没发现
SOR_CERTNOTTRUST	0X0B000500	验证证书失败，证书不被信任
SOR_CERTHASEXPIRED	0X0B000501	验证证书失败，证书超过有效期范围
SOR_CERTHASREVOKED	0X0B000502	验证证书失败，证书已作废
SOR_CERTHASHOLDED	0X0B000503	验证证书失败，证书已冻结
SOR_CERTNOTVALIDYET	0X0B000504	验证证书失败，证书未生效
SOR_CERTVERIFYOTHERERR	0X0B000505	验证证书失败，其他错误

表 A.2 JAVA 组件的异常信息表

异常描述	说明
java.lang.PointerException	某一个参数为空指针
SOR_InitException	初始化环境失败
SOR_AppNotFoundException	没有找的该应用
SOR_CertEncodeException	证书编码格式错误
SOR_CertInvalidException	证书无效，不是可信CA颁发的证书
SOR_CertNotYetValidException	证书未生效
SOR_CertHasExpiredException	证书已过期
SOR_CertRevokedException	证书已经被吊销
SOR_SignDataException	签名失败
SOR_VerifySignDataException	验证签名失败
SOR_ReadFileException	读文件异常，可能文件不存在或没有读取权限等
SOR_WriteFileException	写文件异常，可能文件不存在或没有写权限等
SOR_SecretSegmentException	门限分割算法失败
SOR_SecertRecoveryException	门限恢复失败
SOR_EncryptDataException	数据加密失败
SOR_DecryptDataException	数据解密失败
SOR_Pkes7EncodeException	PKCS#7 编 码 格 式 错 误
SOR_XmlEncodeException	不是合法的xml编码数据
SOR_ParameterNotSupportException	不支持的参数
SOR_EncryptDataException	数据加密失败
SOR_DecryptDataException	数据解密失败
SOR_MessageEncodeException	消息编码格式错误
SOR_XmlEncodeException	不是合法的xml编码数据

附录 B
(资料性)
证书应用综合服务接口典型部署模型

基于B/S 架构应用系统的证书应用综合服务接口的部署示意图如图 B.1 所示。

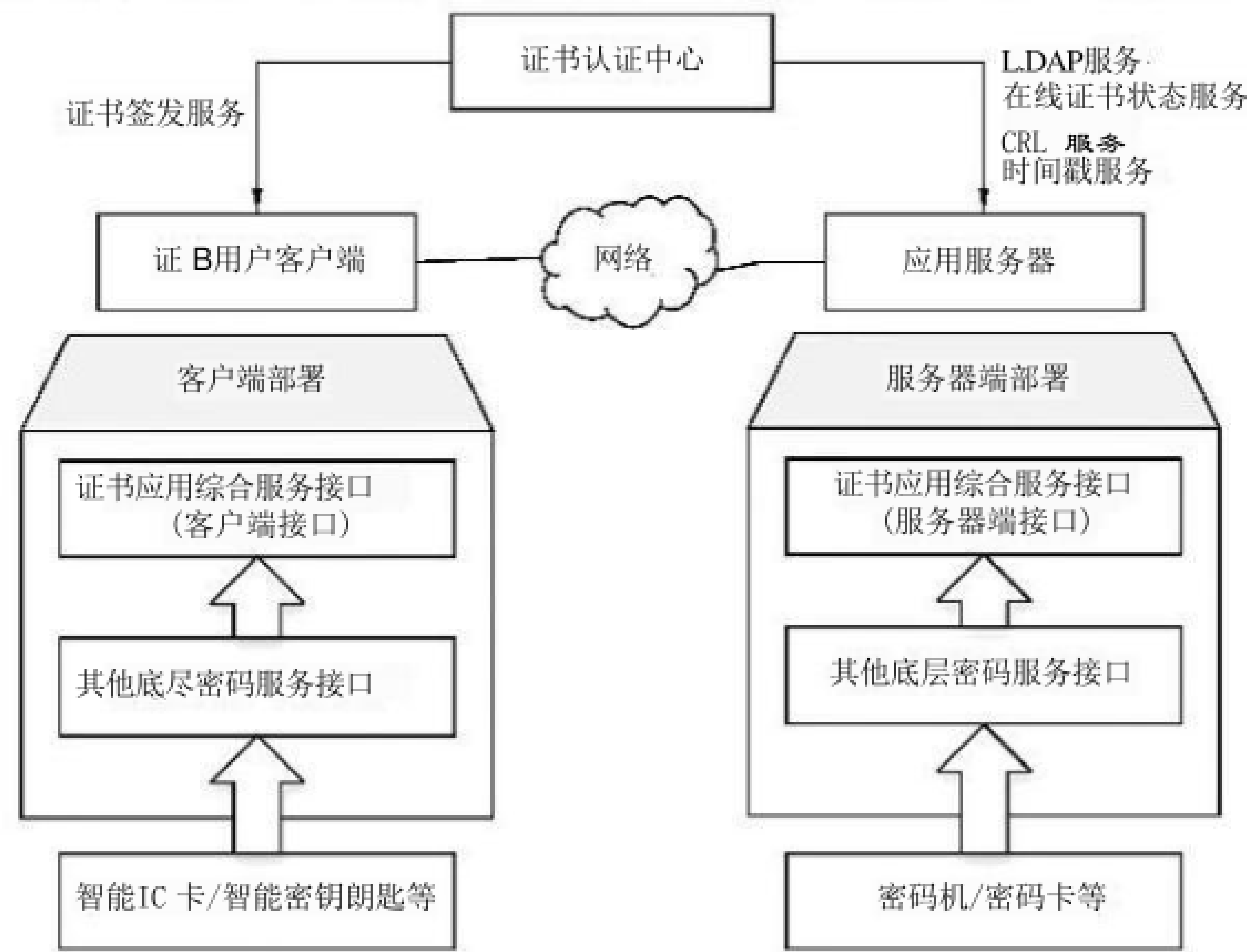


图 B.1 B/S 结构应用系统的典型证书应用接口部署示意图

在应用服务器端部署的软件包括：证书应用综合服务接口（服务器端接口，接口形态一般为COM 组件或JAVA 组件两类）和其他底层密码服务接口（包括密码设备接口和通用密码服务接口等），部署的硬件是密码设备，如密码机或密码卡，用于服务器端的签名、验证、加密、解密等密码运算。

证书用户客户端包括PC 机、手机或其他智能移动终端等，部署的软件包括：证书应用综合服务接口（客户端接口，接口形态一般为COM 组件、动态库或JavaScript 等形态，随着技术的发展接口形态可进行扩展）和其他底层密码服务接口（主要包括智能 IC 卡/智能密码钥匙应用接口和通用密码服务接口、证书载体驱动程序等），部署的硬件是密码设备，如智能 IC 卡或智能密码钥匙(USBKey)，用于客户端的签名、验证、加密、解密等密码运算。

附录 C
(资料性)
证书应用综合服务接口集成示例

典型的证书登录认证流程图如图C.1 所示。

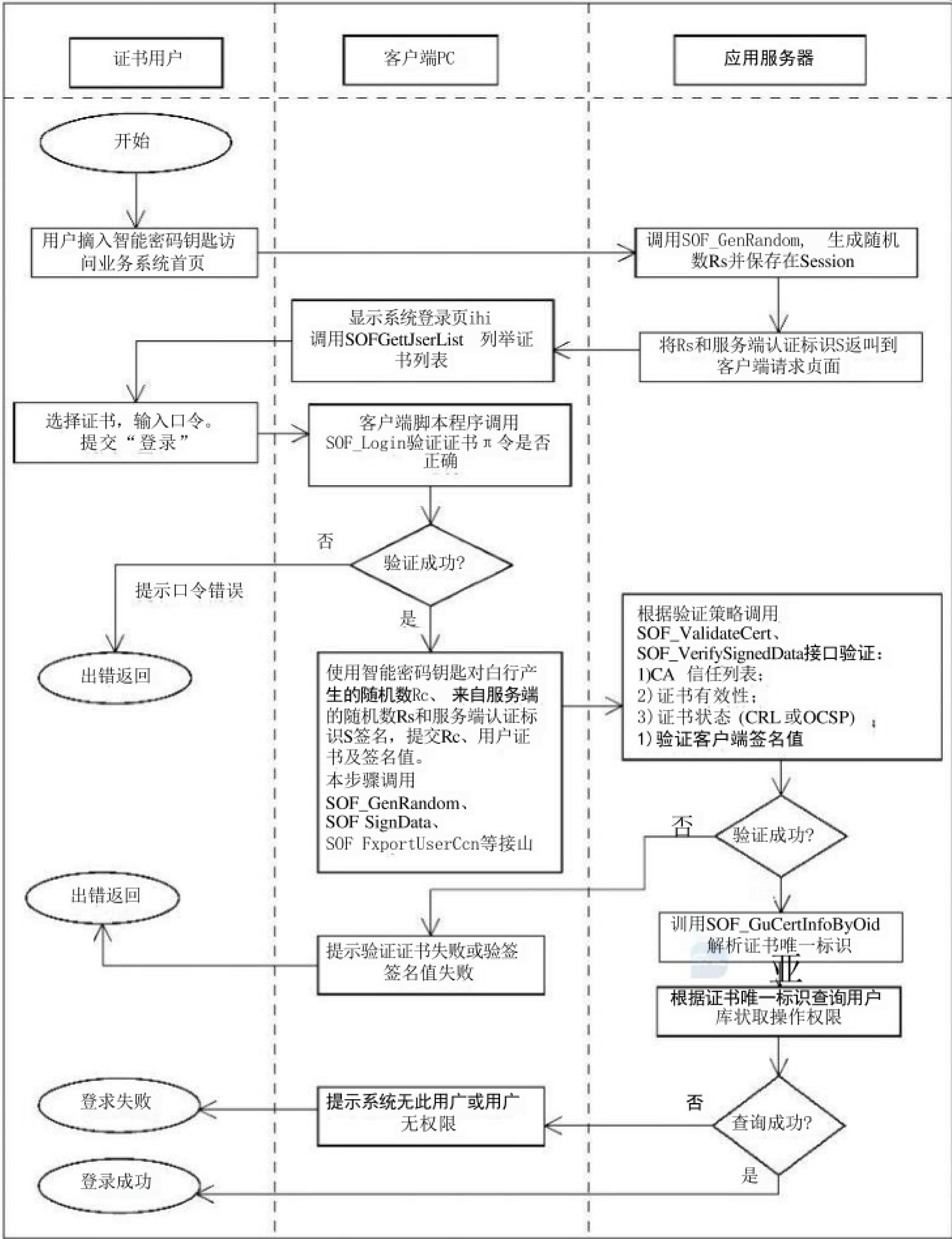


图 C.1 证书登录认证流程示意图

基于表单业务数据的签名和验签流程如图C.2 所示。

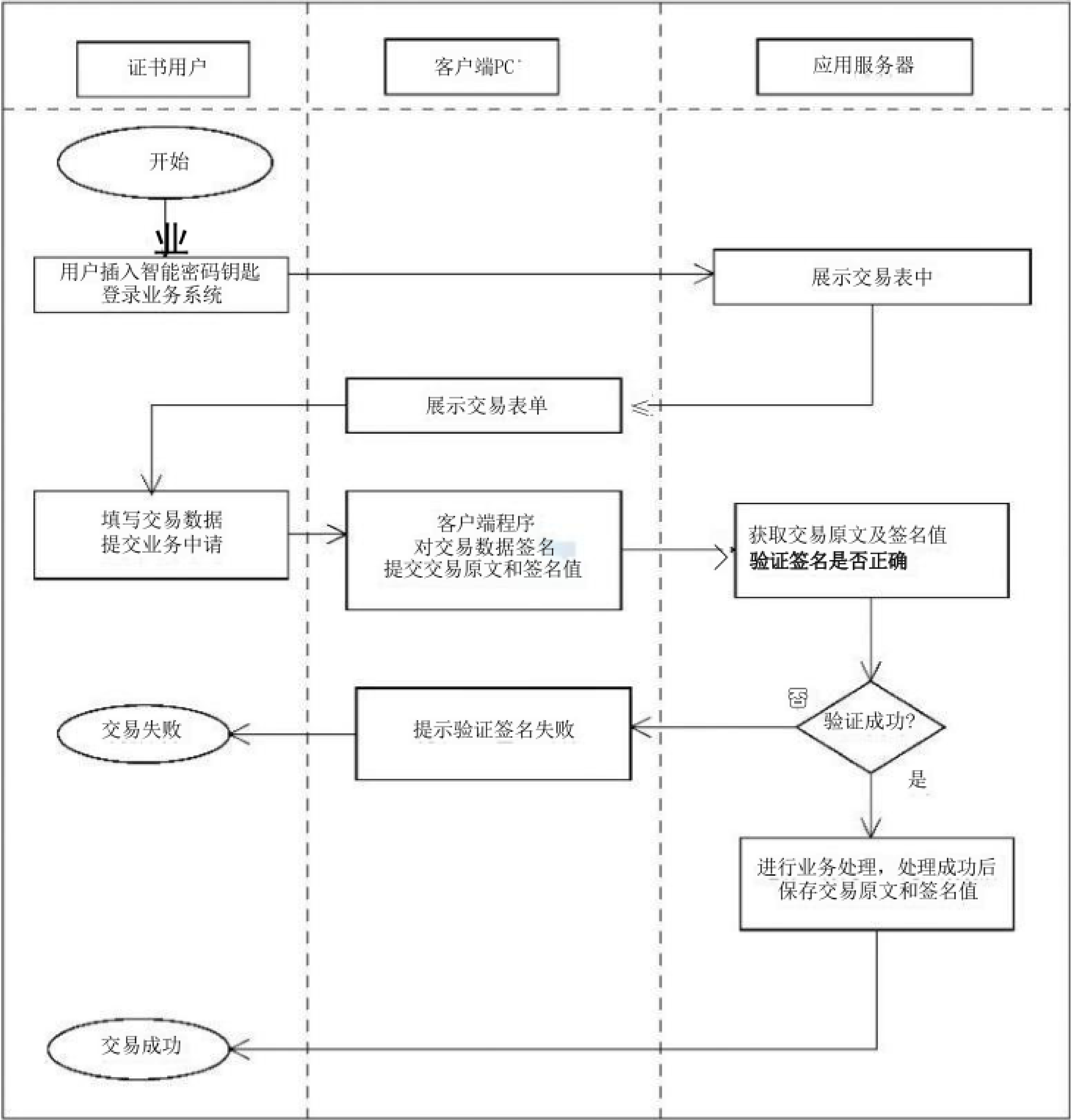


图 C.2 业务数据签名验证流程图

基于表单业务数据的签名和验签流程，首先按照图C.1 完成证书登录认证流程。登录认证成功后，业务系统展示交易表单，证书用户根据业务系统展示的表单填写业务交易数据，检查无误后提交业务申请。提交业务申请时需对业务交易数据做数字签名，根据业务数据不同选择调用不同的签名接口：

- 若业务数据是普通数据，调用SOF_SignData 接口；
- 若业务数据存放在文件中，调用SOF_SignFile 接口；
- 若业务数据是 XML 格式，调用 SOF_SignDataXML 接口；
- 若业务数据是消息，调用SOF_SignMessage 接口；
- 若业务数据仅以 Hash 方式提交，先调用 SOF_HashData 或 SOF_HashFile，再调用 SOF_SignHashData 接口。

业务交易数据和签名值一起提交到业务系统服务器。业务系统服务器调用对应的验证接口，验证正确后进行业务处理，完成后提示用户交易成功。

附 录 D
(资料性)
证书应用综合服务接口汇总

本附录是第7章定义的接口汇总，方便查阅。其中客户端COM 组件接口汇总见表 D.1，客户端脚本接口汇总见表 D.2，服务端 COM 组件接口汇总见表 D.3，服务端Java 组件接口汇总见表 D.4。

表 D.1 客户端 COM 组件接口分类汇总

序号	章条号	接口名称	功能
	7.1.2	SOF_GetVersion	获取接口版本号
2	7.1.9	SOF_Login	校证书口令
3	7.1.10	SOF_GetPinRetryCount	获取证书口令剩余重试次数
4	7.1.11	SOF_ChangePassWd	修改证书口令
5	7.1.15	SOF_GetDeviceInfo	获得设备信息
6	7.1.30	SOF_GetLastError	获取最新的错误代码
7	7.1.35	SOF_Logout	证书用户登出
8	7.1.36	SOF_IsLogin	证书用户登录状态检测
9	7.1.3	SOF_SetSignMethod	设置签名算法
10	7.1.4	SOF_GetSignMethod	获得当前签名算法
11	7.1.5	SOF_SetEncryptMethod	设置加密算法
12	7.1.6	SOF_GetEncryptMethod	获得加密算法
13	7.1.7	SOF_GetUserList	获得证书列表
14	7.1.8	SOF_ExportUserCert	导出用户签名证书
15	7.1.12	SOF_ExportExChangeUserCert	导出用户加密证书
16	7.1.13	SOF_GetCertInfo	获得证书信息
17	7.1.14	SOF_GetCertInfoByOid	获得证书扩展信息
18	7.1.16	SOF_ValidateCert	验证证书有效性
19	7.1.17	SOF_SignData	数据签名
20	7.1.19	SOF_SignFile	文件签名
21	7.1.23	SOF_SignMessage	消息签名
22	7.1.26	SOF_SignDataXML	XML数据签名
23	7.1.33	SOF_SignHashData	摘要数据签名
24	7.1.18	SOF_VerifySignedData	验证数据签名
25	7.1.20	SOF_VerifySignedFile	验证文件签名
26	7.1.24	SOF_VerifySignedMessage	验证消息签名

表 D.1 客户端 COM 组件接口分类汇总（续）

序号	章条号	接口名称	功能
27	7.1.27	SOF_VerifySignedDataXML.	验证XML数据签名
28	7.1.34	SOF_VerifySignedHashData	验证摘要数据签名
29	7.1.25	SOF_GetInfoFromSignedMessage	解析消息签名
30	7.1.28	SOF_GetXMLSignatureInfo	解析XML数据签名
31	7.1.21	SOF_EncryptData	数据加密
32	7.1.22	SOF_DecryptData	数据解密
33	7.1.31	SOF_HashData	计算数据摘要
34	7.1.32	SOF_HashFile	计算文件摘要
35	7.1.29	SOF_GenRandom	产生随机数

表 D.2 客户端脚本接口分类汇总

序号	章条号	接口名称	功能
1	7.2.2	SOF_GetVersion	获取接口版本号
2	7.2.9	SOF_Login	校证书口令
3	7.2.10	SOF_GetPinRetryCount	获取证书口令剩余重试次数
4	7.2.11	SOF_ChangePassWd	修改证书口令
5	7.2.15	SOF_GetDeviceInfo	获得设备信息
6	7.2.28	SOF_GetLastError	获取最新的错误代码
7	7.2.32	SOF_Logout	证书用户登出
8	7.2.33	SOF_IsLogin	证书登录状态检测
9	7.2.3	SOF_SetSignMethod	设置签名算法
10	7.2.4	SOF_GetSignMethod	获得当前签名算法
11	7.2.5	SOF_SetEncryptMethod	设置加密算法
12	7.2.6	SOF_GetEncryptMethod	获得加密算法
13	7.2.7	SOF_GetUserList	获得证书用户列表
14	7.2.8	SOF_ExportUserCert	导出用户签名证书
15	7.2.12	SOF_ExportExChangeUserCert	导出用户加密证书
16	7.2.13	SOF_GetCertInfo	获得证书基本信息
17	7.2.14	SOF_GetCertInfoByOid	获得证书扩展信息
18	7.2.16	SOF_ValidateCert	验证证书有效性
19	7.2.18	SOF_SignData	数据签名

表 D.2 客户端脚本接口分类汇总 (续)

序号	章条号	接口名称	功能
20	7.2.21	SOF_SignMessage	消息签名
21	7.2.24	SOF_SignDataXML	XML数据签名
22	7.2.30	SOF_SignHashData	摘要数据签名
23	7.2.18	SOF_VerifySignedData	验证数据签名
24	7.2.22	SOF_VerifySigned Message	验证消息签名
25	7.2.25	SOF_VerifySignedDataXML	验证XML数据签名
26	7.2.31	SOF_VerifySignedHashData	验证摘要数据签名
27	7.2.23	SOF_GetInfoFromSignedMessage	解析消息签名
28	7.2.26	SOF_GetXMLSignatureInfo	解析XML数据签名
29	7.2.19	SOF_EncryptData	数据加密
30	7.2.20	SOF_DecryptData	数据解密
31	7.2.29	SOF_HashData	计算数据摘要
32	7.2.27	SOF_GenRandom	产生随机数

表 D.3 服务端 COM 组件接口分类汇总

序号	章条号	接口名称	功能
1	7.3.2	SOF_SetCertTrustList	设置证书信任列表
2	7.3.3	SOF_GetCertTrustListAltNames	查询证书信任列表别名
3	7.3.4	SOF_GetCertTrustList	查询证书信任列表
4	7.3.5	SOF_DelCertTrustList	删除证书信任列表
5	7.3.6	SOF_InitCertAppPolicy	初始化应用策略
6	7.3.7	SOF_SetSignMethod	设置签名算法
7	7.3.8	SOF_GetSignMethod	获得当前签名算法
8	7.3.9	SOF_SetEncryptMethod	设置加密算法
9	7.3.10	SOF_GetEncryptMethod	获得加密算法
10	7.3.36	SOF_GetLastError	获取最新的错误代码
11	7.3.11	SOF_GetServerCertificate	获得服务器证书
12	7.3.13	SOF_GetCertInfo	获得证书信息
13	7.3.14	SOF_GetCertInfoByOid	获得证书扩展信息
14	7.3.15	SOF_ValidateCert	验证证书有效性
15	7.3.16	SOF_SignData	数据签名

表 D.3 服务端 COM 组件接口分类汇总（续）

序号	章条号	接口名称	功能
16	7.3.18	SOF_SignFile	文件签名
17	7.3.24	SOF_SignMessage	消息签名
18	7.3.26	SOF_SignMessageDetach	不带原文的消息签名
19	7.3.29	SOF_SignDataXML	XML数据签名
20	7.3.39	SOF_SignHashData	摘要数据签名
21	7.3.17	SOF_VerifySignedData	验证数据签名
22	7.3.19	SOF_VerifySignedFile	验证文件签名
23	7.3.25	SOF_VerifySignedMessage	验证消息签名
24	7.3.27	SOF_VerifySignedMessageDetach	验证不带原文的消息签名
25	7.3.30	SOF_VerifySignedDataXML	验证XML数据签名
26	7.3.40	SOF_VerifySigned HashData	验证摘要数据签名
27	7.3.28	SOF_GetInfoFromSignedMessage	解析消息签名
28	7.3.31	SOF_GetXMLSignatureInfo	解析XML数据签名
29	7.3.32	SOF_CreateTimeStampRequest	创建时间戳请求
30	7.3.33	SOF_CreateTimeStampResponse	创建时间戳响应
31	7.3.34	SOF_VerifyTimeStamp	验证时间戳
32	7.3.35	SOF_GetTimeStampInfo	解析时间戳
33	7.3.20	SOF_EncryptData	数据加密
34	7.3.21	SOF_DecryptData	数据解密
35	7.3.22	SOF_EncryptFile	文件加密
36	7.3.23	SOF_DecryptFile	文件解密
37	7.3.37	SOF_HashData	计算数据摘要
38	7.3.38	SOF_HashFile	计算文件摘要
39	7.3.12	SOF_GenRandom	产生随机数

表 D.4 服务端 Java 组件接口分类汇总

序号	章条号	接口名称	功能
	7.4.2	SOF_setCertTrustList	设置证书信任列表
2	7.4.3	SOF_getCertTrustListAltNames	查询证书信任列表别名
3	7.4.4	SOF_getCertTrustList	根据别名查询证书信任列表
4	7.4.5	SOF_delCertTrustList	删除证书信任列表

表 D.4 服务端Java 组件接口分类汇总（续）

序号	章条号	接口名称	功能
5	7.4.6	SOF_getInstance	获取指定应用的实例
6	7.4.7	SOF_setSignMethod	设置签名算法
7	7.4.8	SOF_getSignMethod	获得当前签名算法
8	7.4.9	SOF_setEncryptMethod	设置加密算法
9	7.4.10	SOF_getEncryptMethod	获得加密算法
10	7.4.37	SOF_getLastError	获取最新的错误代码
11	7.4.11	SOF_getServerCertificate	获得服务器证书
12	7.4.12	SOF_getServerCertificateByUsage	获得指定密钥用途的服务器证书
13	7.4.14	SOF_getCertInfo	获得证书信息
14	7.4.15	SOF_getCertInfoByOid	获得证书扩展信息
15	7.4.16	SOF_validateCert	验证证书有效性
16	7.4.17	SOF_signData	数据签名
17	7.4.19	SOF_signFile	文件签名
18	7.4.25	SOF_signMessage	消息签名
19	7.4.28	SOF_signMessageDetach	不带原文的消息签名
20	7.4.30	SOF_signDataXML	XML 数据 签 名
21	7.4.40	SOF_signHashData	摘要数据签名
22	7.4.18	SOF_verifySignedData	验证数据签名
23	7.4.20	SOF_verifySignedFile	验证文件签名
24	7.4.26	SOF_verifySigned Message	验证消息签名
25	7.4.29	SOF_verifySignedMessageDetach	验证不带原文的消息签名
26	7.4.31	SOF_verifySignedDataXML	验 证 XML 数 据 签 名
27	7.4.41	SOF_verifySignedHashData	验证摘要数据签名
28	7.4.27	SOF_getInfoFromSignedMessage	解析消息签名
29	7.4.32	SOF_getXMLSignatureInfo	解 析 XML 数 据 签 名
30	7.4.33	SOF_createTimeStampRequest	创建时间戳请求
31	7.4.34	SOF_createTimeStampResponse	创建时间戳响应
32	7.4.35	SOF_verifyTimeStamp	验证时间戳
33	7.4.36	SOF_getTimeStampInfo	解析时间戳
34	7.4.21	SOF_encryptData	数据加密
35	7.4.22	SOF_decryptData	数据解密

表 D.4 服务端Java 组件接口分类汇总（续）

序号	章条号	接口名称	功能
36	7.4.23	S0F_encryptFile	文件加密
37	7.4.24	S0F_decryptFile	文件解密
38	7.4.38	S0F_hashData	计算数据摘要
39	7.4.39	S0F_hashFile	计算文件摘要
40	7.4.13	S0F_genRandom	产生随机数

附 录 E
(资料性)
客户端JavaScript 脚本接口异步调用示例说明

为保证多种类型浏览器的兼容性，7.2客户端脚本接口采用异步调用方式定义。每个功能接口都对应一个回调函数和一个回调参数对象，回调函数的原型为：

```
function callback(result,ctx)
```

功能接口的返回值通过 result 参数传递给回调函数，对功能接口返回值的业务处理代码在回调函数内完成。若回调函数不需要ctx 参数，调用功能接口时可传递空值或忽略。

使用示例：

```
function SOF_GetVersion_cb(result,ctx){  
    alert(“接口版本号为： ” + result);  
    alert(“调用者传入的回调参数” + ctx);
```

```
SOF_GetVersion(SOF_GetVersion_cb);
```

所有功能接口的返回值都通过回调函数的 result 返回，功能接口本身没有返回值。如上述调用，在获取版本号时，在回调函数SOF_GetVersion_cb 中接收 SOF_GetVersion 接口的返回值并做相关业务端处理代码，如下是不正确的使用方式：

```
var strVersion=SOF_GetVersion();
```

因功能接口采用异步调用方式定义，strVersion 变量并不保证能正确获取到版本号。

参 考 文 献

- [1] GB/T19713—2005 信息技术 安全技术 公钥基础设施 在线证书状态协议
 - [2] ISO/IEC 8825-1:2021 Information technology—ASN.1 encoding rules—Part 1:Specification of Basic Encoding Rules(BER),Canonical Encoding Rules(CER)and Distinguished Encoding Rules(DER)
 - [3] IETF RFC 1777 Lightweight Directory Access Protocol
 - [4] IETF RFC 2459 Internet X.509 Public Key Infrastructure Certificate and CRL Profile
 - [5] IETF RFC 2560 X.509 Internet Public Key Infrastructure Online Certificate Status Protocol
 - [6] IETF RFC 2587 Internet X.509 Public Key Infrastructure LDAPv2 Schema
 - [7] IETF RFC 3275 (Extensible Markup Language)XML—Signature Syntax and Processing
 - [8] IETF RFC 3629 UTF-8,a transformation format of ISO 10646
 - [9] IETF RFC 3647 Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework
 - [10] IETF RFC4648 The Base16,Base32,and Base64 Data Encodings
 - [11] PKCS#1RSA Cryptography Standard
 - [12] PKCS#7 Cryptographic Message Syntax Standard
 - [13]PKCS#11 Cryptographic Token Interface Standard
 - [14] RSA Security:Public—Key Cryptography Standards(PKCS)
 - [15]The Component Object Model Specification
-

www.bzxz.net

免费标准下载网