ICS 35.040
L 80
备案号:38318—2013

GM

中华人民共和国密码行业标准

GM/T 0020—2012

证书应用综合服务接口规范

Certificate application integrated service interface specification

2012-11-22 发布　　2012-11-22 实施

国家密码管理局　　发布

目　次

前　言

本标准按照 GB/T 1.1—2009 的规则编写。

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

本标准由国家密码管理局提出并归口。

本标准中的附录 A 为规范性附录，附录 B 和附录 C 为资料性附录。

本标准起草单位：北京数字认证股份有限公司、上海格尔软件股份有限公司、北京海泰方圆科技有限公司、上海市数字证书认证中心有限公司、无锡江南信息安全工程技术中心、成都卫士通信息产业股份有限公司、长春吉大正元信息技术股份有限公司、兴唐通信科技有限公司、山东得安信息技术有限公司、国家信息安全工程技术研究中心、国家密码管理局商用密码检测中心。

本标准起草人：刘平、李述胜、谭武征、柳增寿、刘承、徐强、李元正、赵丽丽、王妮娜、孔凡玉、袁峰、李志伟。

本标准凡涉及密码算法相关内容，按照国家有关法规实施。

引　言

本标准依托于GM/T 0019—2012《通用密码服务接口规范》，向上为应用层规定了统一的高级密码服务接口。

证书应用综合服务接口为上层的应用系统提供简洁、易用的证书应用接口，屏蔽了各类密码设备（服务器密码机和智能密码钥匙等）的设备差异性，屏蔽了各类密码设备的密码应用接口之间的差异性，实现应用与密码设备无关性，可简化应用开发的复杂性。证书应用综合服务接口分成客户端服务接口和服务器端服务接口两类，可满足B/S和C/S等多种架构的应用系统的调用需求，有利于密码服务接口产品的开发，有利于应用系统在密码服务过程中的集成和实施，有利于实现各应用系统的互联互通。

证书应用综合服务接口规范

1 范围

本标准规定了面向证书应用的统一服务接口。

本标准适用于公钥密码应用技术体系下密码应用服务产品的开发，密码应用支撑平台的研制及检测，也可用于指导直接使用密码设备和密码服务的应用系统的集成和开发。

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/T 0006 密码应用标识规范

GM/T 0009 SM2 密码算法使用规范

GM/T 0010 SM2 密码算法加密签名消息语法规范

GM/T 0015 基于 SM2 密码算法的数字证书格式规范

GM/T 0019 通用密码服务接口规范

PKCS #7 Cryptographic Message Syntax

RFC3275 (Extensible Markup Language) XML-Signature Syntax and Processing

3 术语和定义

下列术语和定义适用于本文件。

3.1

数字证书 digital certificate

由认证权威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。

3.2

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，包含签名密钥对和加密密钥对。

3.3

容器 container

密码设备中用于保存密钥所划分的唯一性存储空间。

4 缩略语

下列缩略语适用于本文件：

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

CA Certification Authority 证书认证机构

CN Common Name 通用名

CRL Certificate Revocation List 证书撤销列表
CSP Cryptographic Service Provider 加密服务提供者
DER Distinguished Encoding Rules 可区分编码规则
DN Distinguished Name 可识别名
LDAP Lightweight Directory Access Protocol 轻量级目录访问协议
OID Object Identifier 对象标识符
PKCS the Public-Key Cryptography Standard 公钥密码标准

5 算法标识和数据结构

5.1 标识定义

本规范所使用常量、各类算法标识和证书解析标识的具体定义见 GM/T 0006。

5.2 数据结构定义

本规范对应的接口处理的数据分为两种类型：

数据类型 A：当公钥算法为 RSA 时，数据的结构遵循 PKCS#1；当公钥算法为 SM2 时，数据的结构遵循 GM/T 0009。

数据类型 B：当公钥算法为 RSA 时，消息的结构遵循 PKCS#7；当公钥算法为 SM2 时，消息的结构遵循 GM/T 0010。

6 证书应用综合服务接口概述

6.1 概述

证书应用综合服务接口位于应用系统和典型密码服务接口之间，向应用层直接提供证书信息解析、基于数字证书身份认证和信息的机密性、完整性、不可否认性等高级密码服务。该接口可直接供应用系统调用，将应用系统的密码服务请求转向通用密码服务接口，通过通用密码服务接口调用相应的密码设备实现具体的密码运算和密钥操作。通用密码服务接口应遵循 GM/T 0019。

本规范所定义的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。其中服务器端服务接口采用 COM 组件形式和 Java 形式两类描述。本文所涉及的数字证书格式应遵循 GM/T 0015。

6.2 客户端服务接口

本规范定义的客户端服务接口采用客户端控件方式。客户端控件适用于客户端程序调用，接口的形态包括 DLL 动态库、ActiveX 控件、Applet 插件等，接口应支持 WindowsXP、Windows2000、Windows2003、Vista、Windows7 等终端用户使用的主流操作系统。

客户端控件接口的主要函数功能应包括：配置管理、证书解析、签名与验证、加密与解密、数字信封、XML 数据的签名与验证等。

在定义客户端服务接口时，本规范以 ActiveX 控件为例进行描述，其中 BSTR 代表函数返回值或参数类型为 OLECHAR 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。

6.3 服务器端服务接口

服务器端服务接口适用于服务器端程序调用，接口的形态包括 COM 组件、JAR 包、WebService 等

形态，接口应支持 Windows、Linux、Unix、AIX、Solaris 等服务器使用的主流操作系统。服务器端服务接口的函数功能与客户端控件接口相对应，主要包括：配置管理、数字证书解析、签名与验证、加密与解密、数据信封、XML 数据的签名与验证、时间戳等。

7 证书应用综合服务接口函数定义

7.1 客户端控件接口函数

7.1.1 客户端接口函数定义

客户端控件接口包括以下函数：

a) 获取接口的版本号 SOF_GetVersion
b) 设置签名算法 SOF_SetSignMethod
c) 获得当前签名算法 SOF_GetSignMethod
d) 设置加密算法 SOF_SetEncryptMethod
e) 获得加密算法 SOF_GetEncryptMethod
f) 获得证书列表 SOF_GetUserList
g) 导出用户证书 SOF_ExportUserCert
h) 校验证书口令 SOF_Login
i) 获取用户认证口令剩余重试次数 SOF_GetPinRetryCount
j) 修改证书口令 SOF_ChangePassWd
k) 导出用户加密证书 SOF_ExportExChangeUserCert
l) 获得证书信息 SOF_GetCertInfo
m) 获得证书扩展信息 SOF_GetCertInfoByOid
n) 获得设备信息 SOF_GetDeviceInfo
o) 验证证书有效性 SOF_ValidateCert
p) 数字签名 SOF_SignData
q) 验证签名 SOF_VerifySignedData
r) 文件签名 SOF_SignFile
s) 验证文件签名 SOF_VerifySignedFile
t) 加密数据 SOF_EncryptData
u) 解密数据 SOF_DecryptData
v) 文件加密 SOF_EncryptFile
w) 文件解密 SOF_DecryptFile
x) 消息签名 SOF_SignMessage
y) 验证消息签名 SOF_VerifySignedMessage
z) 解析消息签名 SOF_GetInfoFromSignedMessage
aa) XML 数字签名 SOF_SignDataXML
bb) 验证 XML 数字签名 SOF_VerifySignedDataXML
cc) 解析 XML 签名数据 SOF_GetXMLSignatureInfo
dd) 产生随机数 SOF_GenRandom
ee) 获取最新的错误代码 SOF_GetLastError()

以 ActiveX 控件形态为例，下面对接口函数进行定义。

7.1.2 获取接口版本信息 SOF_GetVersion

原型：BSTR SOF_GetVersion()
描述：获取控件的版本号。
参数：无
返回值：非空　成功
空　失败

7.1.3 设置签名算法 SOF_SetSignMethod

原型：long SOF_SetSignMethod(long SignMethod)
描述：设置接口在签名运算时使用的签名算法。
参数：SignMethod[in]　签名算法标识，详见 GM/T 0006
返回值：0　成功
其他　失败，详见错误代码表

7.1.4 获得当前签名算法 SOF_GetSignMethod

原型：long SOF_GetSignMethod()
描述：获得控件签名使用的签名算法。
参数：无
返回值：非 0　当前接口使用的签名算法的预定义值
0　没有设置签名算法

7.1.5 设置加密算法 SOF_SetEncryptMethod

原型：long SOF_SetEncryptMethod(long EncryptMethod)
描述：设置组件进行数据加解密时使用的对称算法。
参数：EncryptMethod[in]　对称加解密算法标识，详见 GM/T 0006
返回值：0　成功
其他　失败，详见错误代码表

7.1.6 获得加密算法 SOF_GetEncryptMethod

原型：long SOF_GetEncryptMethod()
描述：获得控件使用的对称加解密算法。
参数：无
返回值：非 0　当前控件使用的加密算法的预定义值
0　没有设置加密算法

7.1.7 获得证书列表 SOF_GetUserList

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：非空　用户列表字符串数据格式为：用户名 1||ContainerName1&&& 用户名 2||ContainerName2&&&…
空　失败

备注：　根据证书应用的策略不同得到不同的证书列表。在证书列表中，用户名代表证书的CN项内容；ContainerName代表证书和密钥对应的容器名，也可以是代表证书实体身份的唯一的号码，通过ContainerName可以找到唯一的加密证书、签名证书，并使用对应的密钥。

7.1.8　导出用户证书 SOF_ExportUserCert

原型：　BSTR SOF_ExportUserCert(BSTR ContainerName)
描述：　根据证书容器名，获取base64编码的证书字符串。
参数：　ContainerName[in]　证书容器名
返回值：　非空　base64编码的证书字符串
　　空　失败
备注：　默认导出签名证书，无签名证书时导出加密证书。

7.1.9　校验证书口令 SOF_Login

原型：　BOOL SOF_Login(BSTR ContainerName,BSTR PassWd)
描述：　校验设备的用户认证口令，进行用户认证。
参数：　ContainerName[in]　证书容器名
　　PassWd [in]　设备的用户认证口令
返回值：　TRUE　成功
　　FALSE　失败

7.1.10　获取用户认证口令剩余重试次数 SOF_GetPinRetryCount

原型：　long SOF_GetPinRetryCount(BSTR ContainerName)
描述：　获取用户认证口令的剩余密码重试次数。
参数：　ContainerName[in]　证书容器名
返回值：　剩余口令重试次数，当重试次数小于或等于0时表示证书介质口令已被锁死

7.1.11　修改证书口令 SOF_ChangePassWd

原型：　BOOL SOF_ChangePassWd(BSTR ContainerName,BSTR OldPassWd,
　　BSTR NewPassWd)
描述：　修改设备的用户认证口令。
参数：　ContainerName[in]　证书容器名
　　OldPassWd [in]　旧口令
　　NewPassWd [in]　新口令
返回值：　TRUE　修改口令成功
　　FALSE　失败

7.1.12　导出用户加密证书 SOF_ExportExChangeUserCert

原型：　BSTR SOF_ExportExChangeUserCert(BSTR ContainerName)
描述：　根据证书容器名，获取base64编码的加密(交换)证书字符串。
参数：　ContainerName[in]　证书容器名
返回值：　非空　base64编码的证书字符串
　　空　失败

7.1.13 获得证书信息 SOF_GetCertInfo

原型：BSTR SOF_GetCertInfo(BSTR Base64EncodeCert,short Type)

描述：获取证书内指定类型的信息。

参数：

Base64EncodeCert[in]	base64 编码的证书
Type[in]	获取信息的类型，代码标识见 GM/T 0006

返回值：

非空	证书内指定类型的信息
空	失败

7.1.14 获得证书扩展信息 SOF_GetCertInfoByOid

原型：BSTR SOF_GetCertInfoByOid(BSTR Base64EncodeCert,BSTR Oid)

描述：根据 OID 获取证书私有扩展项信息。

参数：

Base64EncodeCert [in]	base64 编码的证书
oid [in]	私有扩展对象 ID，如“1.2.156.xxx”

返回值：

非空	证书私有扩展项 OID 对应的信息
空	失败

7.1.15 获得设备信息 SOF_GetDeviceInfo

原型：BSTR SOF_GetDeviceInfo(BSTR ContainerName,long type)

描述：根据容器和类型代码获得设备信息。

参数：

ContainerName[in]	证书容器名
type[in]	信息类别，定义详见 GM/T 0006 的设备信息标识表

返回值：

非空	type 对应的设备信息
空	失败

7.1.16 验证证书有效性 SOF_ValidateCert

原型：long SOF_ValidateCert(BSTR Base64EncodeCert)

描述：验证证书有效性。

参数：

Base64EncodeCert[in]	base64 编码的证书

返回值：

0	验证成功
其他	-1 证书不被信任
	-2 超过有效期范围
	-3 证书已作废
	-4 证书已冻结
	-5 证书未生效
	-6 其他错误

备注：基本的证书验证策略应包括

a) 验证 CA 信任列表，各层都要进行签名和有效期验证；

b) 各层证书的有效期；

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 数字签名 SOF_SignData

原型：BSTR SOF_SignData(BSTR ContainerName,BSTR InData)

描述：对字符串数据进行数字签名，签名值为数据类型 A。

参数：

ContainerName[in]	证书容器名
InData[in]	签名原文

返回值：

非空	base64 编码的签名结果
空	失败

7.1.18 验证签名 SOF_VerifySignedData

原型：BOOL SOF_VerifySignedData(
BSTR Base64EncodeCert,BSTR InData,BSTR SignValue)

描述：验证数字签名，签名值为数据类型 A。

参数：

Base64EncodeCert[in]	base64 编码签名者证书
InData[in]	签名原文
SignValue[in]	base64 编码的数据类型 A 签名值

返回值：

TRUE	成功
FALSE	失败

7.1.19 文件签名 SOF_SignFile

原型：BSTR SOF_SignFile(BSTR ContainerName,BSTR InFile)

描述：根据文件路径，对指定文件中的数据进行数字签名，签名值为数据类型 A。

参数：

ContainerName[in]	证书容器名
InFile[in]	原文文件路径，包含文件名

返回值：

非空	base64 编码的签名结果
空	失败

7.1.20 验证文件签名 SOF_VerifySignedFile

原型：BOOL SOF_VerifySignedFile(
BSTR Base64EncodeCert,BSTR InFile,BSTR SignValue)

描述：验证文件的数字签名，签名值为数据类型 A。

参数：

Base64EncodeCert [in]	base64 编码的签名者证书
InFile[in]	全路径文件名称
SignValue[in]	base64 编码的数据类型 A 签名值

返回值：

TRUE	成功
FALSE	失败

7.1.21 加密数据 SOF_EncryptData

原型：BSTR SOF_EncryptData(BSTR Base64EncodeCert,BSTR Indata)

描述：使用临时产生的对称密钥加密数据，然后使用数字证书加密对称密钥。密文数据格式为数据类型 B。

参数：

Base64EncodeCert[in]	base64 编码的加密用的数字证书
Indata[in]	待加密的明文

返回值： 非空 加密后的数据类型 B 的密文，采用 base64 编码
空 失败

7.1.22 解密数据 SOF_DecryptData

原型： BSTR SOF_DecryptData(BSTR ContainerName,BSTR Indata)
描述： 使用证书对应的私钥解密数字信封，密文数据格式为数据类型 B。
参数： ContainerName[in] 证书容器名
Indata[in] 待解密的 base64 编码的数据类型 B 的密文
返回值： 非空 解密后的明文
空 失败

7.1.23 文件加密 SOF_EncryptFile

原型： BOOL SOF_EncryptFile(BSTR Base64EncodeCert,BSTR InFile,BSTR OutFile)
描述： 使用证书加密文件，得到数据类型 B 的数字信封文件。
参数： Base64EncodeCert[in] base64 编码的加密证书
InFile[in] 待加密的全路径文件名称
OutFile[in] 待生成的密文文件全路径名称
返回值： TRUE 成功
FALSE 失败

7.1.24 文件解密 SOF_DecryptFile

原型： BOOL SOF_DecryptFile(BSTR ContainerName,BSTR InFile,BSTR OutFile)
描述： 使用证书对应的私钥解密数据类型 B 的数字信封文件。
参数： ContainerName[in] 证书容器名
InFile[in] 待解密的密文文件路径
OutFile[in] 明文文件保存路径
返回值： TRUE 成功
FALSE 失败

7.1.25 消息签名 SOF_SignMessage

原型： BSTR SOF_SignMessage(short flag,BSTR ContainerName,BSTR InData)
描述： 对字符串数据进行数字签名，签名结果的格式为数据类型 B。
参数： flag[in] 是否带原文的标识，1：不带原文；0：带原文。
ContainerName[in] 证书容器名
InData[in] 签名原文
返回值： 非空 base64 编码的签名结果
空 失败
备注： 签名结果包含：原文(可选)＋签名者证书＋签名值。

7.1.26 验证消息签名 SOF_VerifySignedMessage

原型： BOOL SOF_VerifySignedMessage(BSTR MessageData,BSTR InData)
描述： 验证数字签名，签名值为数据类型 B。
参数： MessageData[in] base64 编码的消息签名数据

	InData[in]	原文。如果签名结果带原文，则本参数为空。
返回值：	TRUE	成功
	FALSE	失败

7.1.27 解析消息签名 SOF_GetInfoFromSignedMessage

原型： BSTR SOF_GetInfoFromSignedMessage(BSTR SignedMessage,short type)

描述： 解析签名包内的信息，包括：原文、签名值、签名证书等信息。签名包为数据类型 B。

参数：	MessageData[in]	base64 编码的签名包
	type[in]	类型
返回值：	非空	解析出的信息
	空	失败

备注： Type 为 1 时解析出原文；Type 为 2 时解析出 base64 编码的签名者证书；Type 为 3 时解析出 base64 编码的签名值。

7.1.28 XML 数字签名 SOF_SignDataXML

原型： BSTR SOF_SignDataXML(BSTR ContainerName,BSTR InData)

描述： 对 XML 数据进行数字签名，输出符合 RFC3275 的 XML 签名结果。

参数：	ContainerName[in]	证书容器名
	InData[in]	XML 格式的签名原文
返回值：	非空	签名结果
	空	失败

备注： XML 签名标准遵循 RFC3275。

7.1.29 验证 XML 数字签名 SOF_VerifySignedDataXML

原型： BOOL SOF_VerifySignedDataXML(BSTR InData)

描述： 验证 xml 签名。

参数：	InData[in]	XML 签名结果
返回值：	TRUE	成功
	FALSE	失败

备注： XML 签名标准遵循 RFC3275。

7.1.30 解析 XML 签名数据 SOF_GetXMLSignatureInfo

原型： BSTR SOF_GetXMLSignatureInfo(BSTR XMLSignedData,short type)

描述： 解析 XML 签名数据，获取签名值、XML 原文、证书等信息。

参数：	XMLSignedData[in]	XML 格式的签名数据
	type [in]	待解析的参数类型
返回值：	非空	根据 type 解析出各项对应的信息
	空	失败

备注： type 参数意义如下：1：xml 原文；2：摘要；3：签名值；4：签名证书；5：摘要算法；6：签名算法。XML 签名标准遵循 RFC3275。

7.1.31 产生随机数 SOF_GenRandom

原型： BSTR SOF_GenRandom(short RandomLen)

描述：　产生指定长度的随机数。

参数：	RandomLen[in]	待产生的随机数长度
返回值：	非空	base64 编码的随机数值
	空	失败

7.1.32　获取最新的错误信息 SOF_GetLastError()

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，详见错误代码表

7.2　服务器端 COM 组件接口函数

7.2.1　COM 组件接口定义

COM 组件接口函数定义如下：

a）　设置证书信任列表 SOF_SetCertTrustList
b）　查询证书信任列表别名 SOF_GetCertTrustListAltNames
c）　查询证书信任列表 SOF_GetCertTrustList
d）　删除证书信任列表 SOF_DelCertTrustList
e）　初始化应用策略 SOF_InitCertAppPolicy
f）　设置签名算法 SOF_SetSignMethod
g）　获得当前签名算法 SOF_GetSignMethod
h）　设置加密算法 SOF_SetEncryptMethod
i）　获得加密算法 SOF_GetEncryptMethod
j）　获得服务器证书 SOF_GetServerCertificate
k）　产生随机数 SOF_GenRandom
l）　获得证书信息 SOF_GetCertInfo
m）　获得证书扩展信息 SOF_GetCertInfoByOid
n）　验证证书有效性 SOF_ValidateCert
o）　数字签名 SOF_SignData
p）　验证签名 SOF_VerifySignedData
q）　文件签名 SOF_SignFile
r）　验证文件签名 SOF_VerifySignedFile
s）　加密数据 SOF_EncryptData
t）　解密数据 SOF_DecryptData
u）　文件加密 SOF_EncryptFile
v）　文件解密 SOF_DecryptFile
w）　消息签名 SOF_signMessage
x）　验证消息签名 SOF_VerifySignedMessage
y）　不带原文的消息签名 SOF_SignMessageDetach
z）　验证不带原文的消息签名 SOF_VerifySignedMessageDetach
aa）　解析消息签名 SOF_GetInfoFromSignedMessage

bb) XML 数字签名 SOF_SignDataXML

cc) 验证 XML 数字签名 SOF_VerifySignedDataXML

dd) 解析 XML 签名数据 SOF_GetXMLSignatureInfo

ee) 创建时间戳请求 SOF_CreateTimeStampRequest

ff) 创建时间戳响应 SOF_CreateTimeStampResponse

gg) 验证时间戳 SOF_VerifyTimeStamp

hh) 解析时间戳 SOF_GetTimeStampInfo

ii) 获取最新的错误代码 SOF_GetLastError

7.2.2 设置证书信任列表 SOF_SetCertTrustList

原型：long SOF_SetCertTrustList(BSTR CTLAltName,BSTR CTLContent,
short CTLContentLen)

描述：设置证书信任列表。

参数：		
	CTLAltName[in]	证书信任列表别名
	CTLContent[in]	base64 编码格式的证书信任列表内容
	CTLContentLen[in]	证书信任列表长度
返回值：	0	成功
	其他	失败,详见错误代码

7.2.3 查询证书信任列表别名 SOF_GetCertTrustListAltNames

原型：BSTR SOF_GetCertTrustListAltNames()

描述：查询证书信任列表别名。

参数：无

返回值：		
	非空	信任列表别名的字符串组合,如“CA001@CA002@CA003”
	空	失败

7.2.4 查询证书信任列表 SOF_GetCertTrustList

原型：BSTR SOF_GetCertTrustList(BSTR CTLAltName)

描述：根据别名查询证书信任列表。

参数：		
	CTLAltName[in]	证书信任列表别名
返回值：	非空	base64 编码的证书信任列表
	空	失败

7.2.5 删除证书信任列表 SOF_DelCertTrustList

原型：long SOF_DelCertTrustList(BSTR CTLAltName)

描述：根据别名删除证书信任列表。

参数：		
	CTLAltName[in]	证书信任列表别名
返回值：	0	成功
	其他	失败,详见错误代码

7.2.6 初始化应用策略 SOF_InitCertAppPolicy

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，包括使用的密钥和证书、信任的根证书、证书验证的策略、验证方式等。配置内容自行定义。

参数：　PolicyName [in]　应用策略名称

返回值：　0　成功

其他　失败，详见错误代码

7.2.7　设置签名算法 SOF_SetSignMethod

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[in]　签名算法标识，详细定义见 GM/T 0006

返回值：　0　成功

其他　失败，详见错误代码

7.2.8　获得当前签名算法 SOF_GetSignMethod

原型：　long SOF_GetsignMethod()

描述：　获得组件签名运算使用的签名算法。

参数：　无

返回值：　非 0　当前的签名算法的预定义值

0　失败，没有设置算法

7.2.9　设置加密算法 SOF_SetEncryptMethod

原型：　long SOF_SetEncryptMethod(long EncryptMethod)

描述：　设置组件对数据加解密使用的对称算法。

参数：　EncryptMethod[in]　对称密码算法标识，详细定义见 GM/T 0006

返回值：　0　成功

其他　失败，详见错误代码

7.2.10　获得加密算法 SOF_GetEncryptMethod

原型：　long SOF_GetEncryptMethod()

描述：　获得组件使用的对称加解密算法。

参数：　无

返回值：　非 0　预先设置的加密算法的预定义值

0　失败，没有设置算法

7.2.11　获得服务器证书 SOF_GetServerCertificate

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[in]　证书用途，1：加密证书；2：签名证书

返回值：　非空　base64 编码的服务器证书

空　失败

7.2.12 产生随机数 SOF_GenRandom

原型：BSTR SOF_GenRandom(short RandomLen)

描述：产生指定长度的随机数。

参数：RandomLen[in] 待产生的随机数长度(字节)

返回值：非空 base64 编码的随机数值

空 失败

7.2.13 获得证书信息 SOF_GetCertInfo

原型：BSTR SOF_GetCertInfo(BSTR Base64EncodeCert,long type)

描述：根据指定类型,获取证书内的相关信息。

参数：Base64EncodeCert[in] base64 编码的数字证书

type[in] 证书信息的类型,详细定义见 GM/T 0006

返回值：非空 Type 对应的信息

空 失败

7.2.14 获得证书扩展信息 SOF_GetCertInfoByOid

原型：BSTR SOF_GetCertInfoByOid(BSTR Base64EncodeCert,BSTR oid)

描述：根据 OID 获取证书扩展项信息。

参数：Base64EncodeCert[in] base64 编码的证书

oid[in] 私有扩展对象 ID,如“1.2.156.xxx”

返回值：非空 证书私有扩展项信息

空 失败

7.2.15 验证证书有效性 SOF_ValidateCert

原型：long SOF_ValidateCert(BSTR Base64EncodeCert)

描述：根据应用的策略根据验证证书有效性。

最基本的证书验证策略应包括

a) 验证 CA 信任列表,各层都要进行签名和有效期验证;

b) 各层证书的有效期;

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。

参数：Base64EncodeCert[in] 待验证的 base64 编码证书

返回值：0 验证成功

其他

-1 证书不被信任

-2 超过有效期范围

-3 证书已作废

-4 证书已冻结

-5 证书未生效

-6 其他错误

7.2.16 数字签名 SOF_SignData

原型：BSTR SOF_SignData(BSTR InData)

描述：对字符串数据进行数字签名，返回的签名结果为数据类型 A。

参数：	InData[in]	待签名的数据原文
返回值：	非空	base64 编码的签名值
	空	失败

7.2.17 验证签名 SOF_VerifySignedData

原型：long SOF_VerifySignedData(BSTR Base64EncodeCert,BSTR InData,BSTR SignValue)

描述：验证数字签名。

参数：	Base64EncodeCert[in]	base64 编码的签名证书
	InData[in]	待验证的原文
	SignValue[in]	签名值，为 base64 编码的数据类型 A
返回值：	0	验证成功
	其他	验证失败，详见错误代码

7.2.18 文件签名 SOF_SignFile

原型：BSTR SOF_SignFile(BSTR InFile)

描述：对文件数字签名，得到 base64 编码的数据类型 A 的签名数据。

参数：	InFile[in]	待签名的全路径文件名称
返回值：	非空	base64 编码的签名数据
	空	失败

7.2.19 验证文件签名 SOF_VerifySignedFile

原型：long SOF_VerifySignedFile(BSTR Base64EncodeCert,BSTR InFile,BSTR SignValue)

描述：验证文件数字签名。

参数：	Base64EncodeCert[in]	base64 编码的签名证书
	InFile[in]	待验证的原文路径
	SignValue[in]	base64 编码的数据类型 A 的签名值
返回值：	0	验证成功
	其他	验证失败，详见错误代码

7.2.20 加密数据 SOF_EncryptData

原型：BSTR SOF_EncryptData(BSTR Cert,BSTR InData)

描述：使用证书对数据进行加密，得到数据类型 B 的密文数据。

参数：	BSTR Cert[in]	数据接收者的加密证书
	BSTR InData[in]	待加密的明文
返回值：	非空	base64 编码格式的密文
	空	失败

7.2.21 解密数据 SOF_DecryptData

原型：BSTR SOF_DecryptData(BSTR ContainerName,BSTR InData)

描述：　解密数据类型 B 的数字信封。

参数：

	BSTR ContainerName [in]	证书容器名
	BSTR InData[in]	base64 编码的待解密密文数据

返回值：

非空	解密后的明文
空	失败

7.2.22 文件加密 SOF_EncryptFile

原型：　long SOF_EncryptFile(BSTR Cert,BSTR InFile,BSTR OutFile)

描述：　使用对称算法加密文件。

参数：

BSTR Cert[in]	加密证书
BSTR InFile[in]	待加密的明文文件路径
BSTR OutFile[in]	密文文件保存路径，密文为数据类型 B

返回值：

0	成功
其他	失败，详见错误代码

7.2.23 文件解密 SOF_DecryptFile

原型：　long SOF_DecryptFile(BSTR ContainerName,BSTR InFile,BSTR OutFile)

描述：　使用对称算法解密文件。

参数：

BSTR ContainerName [in]	解密密钥对应的证书唯一标识
BSTR InFile[in]	待解密的密文文件路径，密文为数据类型 B
BSTR OutFile[in]	明文文件保存路径

返回值：

0	成功
其他	失败，详见错误代码

7.2.24 消息签名 SOF_SignMessage

原型：　BSTR SOF_SignMessage(BSTR InData)

描述：　对字符串数据进行数字签名，签名格式为数据类型 B。

参数：　InData[in]　待签名的数据原文

返回值：

非空	返回带原文消息结构的 base64 编码的签名数据
空	失败

7.2.25 验证消息签名 SOF_VerifySignedMessage

原型：　long SOF_VerifySignedMessage(BSTR SignedMessage)

描述：　验证消息签名包，签名格式为数据类型 B。

参数：　SignedMessage[in]　带原文消息结构的 base64 编码的签名数据

返回值：

0	成功
其他	失败，详见错误代码

7.2.26 不带原文的消息签名 SOF_SignMessageDetach

原型：　BSTR SOF_SignMessageDetach (BSTR InData)

描述：　对字符串数据进行数字签名，签名格式为不带原文的数据类型 B。

参数： InData[in] 待签名的数据原文
返回值： 非空 返回不带原文消息结构的 base64 编码的签名数据
空 失败

7.2.27 验证不带原文的消息签名 SOF_VerifySignedMessageDetach

原型： long SOF_VerifySignedMessageDetach (BSTR InData,BSTR SignedMessage)
描述： 验证消息签名包，签名格式为不带原文的数据类型 B。
参数： InData[in] 原文数据
SignedMessage[in] 不带原文消息结构的 base64 编码的签名数据
返回值： 0 成功
其他 失败，详见错误代码

7.2.28 解析消息签名 SOF_GetInfoFromSignedMessage

原型： BSTR SOF_GetInfoFromSignedMessage(BSTR SignedMessage,short type)
描述： 解析签名包内的信息，包括原文、签名值、签名证书等信息。签名包为数据类型 B。
参数： MessageData[in] base64 编码的签名数据
type[in] 类型 Type 为 1 时解析出原文；Type 为 2 时解析出 base64 编码的签名者证书；Type 为 3 时解析出 base64 编码的签名值。
返回值： 非空 解析出的信息
空 失败

7.2.29 XML 数字签名 SOF_SignDataXML

原型： BSTR SOF_SignDataXML(BSTR InData)
描述： 对 XML 数据进行数字签名，输出符合 RFC3275 的 XML 签名结果。
参数： InData[in] XML 格式的签名原文
返回值： 非空 签名结果
空 失败
备注： XML 签名标准为 RFC3275。

7.2.30 验证 XML 数字签名 SOF_VerifySignedDataXML

原型： long SOF_VerifySignedDataXML(BSTR InData)
描述： 验证 xml 格式的数字签名。
参数： InData[in] 带签名值的 XML 数据
返回值： 0 成功
其他 失败，详见错误代码
备注： XML 签名标准为 RFC3275。

7.2.31 解析 XML 签名数据 SOF_GetXMLSignatureInfo

原型： BSTR SOF_GetXMLSignatureInfo(BSTR XMLSignedData,short type)
描述： 解析 XML 签名数据，获取签名值、XML 原文、证书等信息。
参数： XMLSignedData[in] XML 格式的签名数据
type[in] 待解析的参数类型，Type 意义：1：xml 原文；2：摘要；3：签名值；4：签名证书；5：摘要算法；6：签名算法

返回值：　非空　　获得的信息
　　空　　失败
备注：　XML 签名标准为 RFC3275。

7.2.32　创建时间戳请求 SOF_CreateTimeStampRequest

原型：　BSTR SOF_CreateTimeStampRequest(BSTR InData)
描述：　创建时间戳请求。
参数：　InData[in]　　待创建时间戳请求的原文
返回值：　非空　　base64 编码格式的时间戳请求
　　空　　失败

7.2.33　创建时间戳响应 SOF_CreateTimeStampResponse

原型：　BSTR SOF_CreateTimeStampResponse(BSTR TimeStampRequest)
描述：　创建时间戳响应，即签发时间戳。
参数：　TimeStampRequest[in]　　时间戳请求
返回值：　非空　　base64 编码格式的时间戳响应
　　空　　失败

7.2.34　验证时间戳 SOF_VerifyTimeStamp

原型：　long SOF_VerifyTimeStamp(BSTR InData,BSTR tsResponseData)
描述：　验证时间戳。
参数：　InData[in]　　待验证的原文
　　tsResponseData[in]　　时间戳
返回值：　0　　成功
　　其他　　失败，详见错误代码

7.2.35　解析时间戳 SOF_GetTimeStampInfo

原型：　BSTR SOF_GetTimeStampInfo(BSTR tsResponseData,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、签名值等。
参数：　tsResponseData[in]　　时间戳
　　type[in]　　信息类型，type 意义：type =1，返回时间；type =2，返回签名值；type =3，返回签名证书
返回值：　非空　　解析出对应 type 的信息
　　空　　失败

7.2.36　获取最新的错误代码 SOF_GetLastError

原型：　long SOF_GetLastError()
描述：　获取接口最新的错误代码。
参数：　无
返回值：　详见错误代码

7.3 Java 组件接口函数

7.3.1 Java 组件接口函数定义

Java 组件接口函数如下：

a) 设置证书信任列表 SOF_setCertTrustList
b) 查询证书信任列表别名 SOF_getCertTrustListAltNames
c) 根据别名查询证书信任列表 SOF_getCertTrustList
d) 删除证书信任列表 SOF_delCertTrustList
e) 获取指定应用的实例 SOF_getInstance(java. lang. String PolicyName)
f) 设置签名算法 SOF_setSignMethod
g) 获得当前签名算法 SOF_getSignMethod
h) 设置加密算法 SOF_setEncryptMethod
i) 获得加密算法 SOF_getEncryptMethod
j) 获得服务器证书 SOF_getServerCertificate
k) 获得指定密钥用途的服务器证书 SOF_getServerCertificateByUsage
l) 产生随机数 SOF_genRandom
m) 获得证书信息 SOF_getCertInfo
n) 获得证书扩展信息 SOF_getCertInfoByOid
o) 验证证书有效性 SOF_validateCert
p) 数字签名 SOF_signData
q) 验证签名 SOF_verifySignedData
r) 文件签名 SOF_signFile
s) 验证文件签名 SOF_verifySignedFile
t) 加密数据 SOF_encryptData
u) 解密数据 SOF_decryptData
v) 文件加密 SOF_encryptFile
w) 文件解密 SOF_decryptFile
x) 消息签名 SOF_signMessage
y) 验证消息签名 SOF_verifySignedMessage
z) 解析消息签名 SOF_getInfoFromSignedMessage
aa) 不带原文的消息签名 SOF_signMessageDetach
bb) 验证不带原文的消息签名 SOF_verifySignedMessageDetach
cc) XML 数字签名 SOF_signDataXML
dd) 验证 XML 数字签名 SOF_verifySignedDataXML
ee) 解析 XML 签名数据 SOF_getXMLSignatureInfo
ff) 创建时间戳请求 SOF_createTimeStampRequest
gg) 创建时间戳响应 SOF_createTimeStampResponse
hh) 验证时间戳 SOF_verifyTimeStamp
ii) 解析时间戳 SOF_getTimeStampInfo
jj) 获取最新的错误代码 SOF_getLastError

本条的函数可以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。本规范仅对使用 SOF_getLastError 方式进行说明。

7.3.2 设置证书信任列表 SOF_setCertTrustList

原型：public 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

7.3.3 查询证书信任列表别名 SOF_getCertTrustListAltNames

原型：public java. lang. String SOF_getCertTrustListAltNames()

描述：查询证书信任列表别名。

参数：无

返回值：	非空	返回信任列表别名的组合，如："CA001@CA002@CA003"
	空	失败

7.3.4 根据别名查询证书信任列表 SOF_getCertTrustList

原型：public 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：	非空	返回 base64 编码格式的证书信任列表
	空	失败

7.3.5 删除证书信任列表 SOF_delCertTrustList

原型：public 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

7.3.6 获取指定应用的实例 SOF_getInstance(java. lang. String PolicyName)

原型：public 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。用户如果有多应用需求，可同时获取多个实例对象满足不同的使用需求，不同的实例在调用方法时会有不同效果(如：不同密钥的签名，不同算法的加密，不同的证书验证策略)。

参数：	PolicyName	应用策略名称
返回值：	非空	返回此应用策略名称所对应的实例
	空	失败

7.3.7 设置签名算法 SOF_setSignMethod

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(详见 GM/T 0006)
返回值： 无

7.3.8 获得当前签名算法 SOF_getSignMethod

原型： java.lang.long SOF_getSignMethod()
描述： 获得组件签名运算使用的签名算法标识。
参数： 无
返回值： 非 0 当前的签名算法标识(详见 GM/T 0006)
0 没有设置签名算法

7.3.9 设置加密算法 SOF_setEncryptMethod

原型： void SOF_setEncryptMethod(long encryptMethod)
描述： 设置组件对数据加解密使用的对称算法标识。
参数： encryptMethod 对称算法标识(详见 GM/T 0006)
返回值： 无

7.3.10 获得加密算法 SOF_getEncryptMethod

原型： java.lang.long SOF_getEncryptMethod()
描述： 获得组件使用的对称算法标识。
参数： 无
返回值： 非 0 当前控件使用的加密算法
0 没有设置加密算法

7.3.11 获得服务器证书 SOF_getServerCertificate

原型： java.lang.String SOF_getServerCertificate()
描述： 读取当前应用的服务器的签名证书。如果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： 非空 base64 编码的服务器证书
空 失败

7.3.12 获得指定密钥用途的服务器证书 SOF_getServerCertificateByUsage

原型： java.lang.String SOF_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，1：加密证书；2：签名证书
返回值： 非空 base64 编码的服务器证书
空 失败

7.3.13 产生随机数 SOF_genRandom

原型：java.lang.String SOF_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：非空　base64 编码格式的随机数值
空　失败

7.3.14 获得证书信息 SOF_getCertInfo

原型：java.lang.String SOF_getCertInfo(java.lang.String base64EncodeCert,int type)
描述：根据 type 解析证书内的相关信息。
参数：base64EncodeCert　base64 编码的数字证书
Type　获取证书信息的类型，见 GM/T 0006
返回值：非空　type 对应的信息
空　失败

7.3.15 获得证书扩展信息 SOF_getCertInfoByOid

原型：java.lang.String SOF_getCertInfoByOid(
java.lang.String base64EncodeCert,java.lang.String oid)
描述：根据 OID 获取证书私有扩展项信息。
参数：base64EncodeCert　base64 编码的证书
oid　私有扩展对象 ID，如“1.2.156.xxx”
返回值：非空　证书 OID 对应的值
空　失败

7.3.16 验证证书有效性 SOF_validateCert

原型：int SOF_validateCert(java.lang.String base64EncodeCert)
描述：根据应用的策略根据验证证书有效性。
参数：base64EncodeCert　待验证的 base64 编码证书
返回值：1　验证成功，证书有效
其他值为错误　-1 证书不被信任
-2 超过有效期范围
-3 证书已作废
-4 证书已冻结
-5 证书未生效
-6 其他错误

7.3.17 数字签名 SOF_signData

原型：java.lang.String SOF_SignData(byte[] inData)
描述：对字符串数据进行数字签名，签名格式为数据类型 A。
参数：inData　待签名的数据原文
返回值：非空　base64 编码签名值
空　失败

7.3.18 验证签名 SOF_verifySignedData

原型：boolean SOF_verifySignedData(
java.lang.String base64EncodeCert,
java.lang.String inData,java.lang.String signValue)

描述：验证数字签名。

参数：		
	base64EncodeCert	base64 编码的签名证书
	inData	待验证的原文
	signValue	签名值，签名格式为数据类型 A
返回值：	true	验证成功
	false	验证失败

7.3.19 文件签名 SOF_signFile

原型：java.lang.String SOF_signFile(java.lang.String inFile)

描述：对文件数字签名，得到 base64 编码格式数据类型 A 的签名数据。

参数：		
	inFile	待签名的文件路径
返回值：	非空	base64 编码签名数据
	空	失败

7.3.20 验证文件签名 SOF_verifySignedFile

原型：boolean SOF_VerifySignedFile(
java.lang.String base64EncodeCert,
java.lang.String inFile,java.lang.String signValue)

描述：验证文件数字签名。

参数：		
	base64EncodeCert	base64 编码的签名证书
	inFile	待验证的文件路径
	signValue	签名值，签名值为数据类型 A
返回值：	true	验证成功
	false	验证失败

7.3.21 加密数据 SOF_encryptData

原型：java.lang.String SOF_encryptData(
java.lang.String Cert,byte[] inData)

描述：使用数字证书加密数据，密文为数据类型 B 的数字信封格式。

参数：		
	Cert	加密证书
	inData	待加密的明文
返回值：	非空	base64 编码的数据类型 B 的密文
	空	失败

7.3.22 解密数据 SOF_decryptData

原型：byte[] SOF_decryptData(
java.lang.String ContainerName,java.lang.String inData)

描述：解密数据类型 B 的数字信封数据。

参数：　ContainerName　　解密密钥对应的证书唯一标识
　　InData　　base64 编码的待解密的数据类型 B 的密文
返回值：　非空　　解密后的明文
　　空　　失败

7.3.23　文件加密 SOF_encryptFile

原型：　boolean SOF_encryptFile(java. lang. String Cert,
　　java. lang. String inFile,java. lang. String outFile)
描述：　加密文件,得到数据类型 B 的密文文件。
参数：　Cert　　base64 编码的加密证书
　　inFile　　待加密的明文文件路径
　　outFile　　密文文件保存路径
返回值：　true　　成功
　　false　　失败

7.3.24　文件解密 SOF_decryptFile

原型：　boolean SOF_decryptFile(
　　java. lang. String ContainerName,
　　java. lang. String inFile,java. lang. String outFile)
描述：　解密密文文件。
参数：　ContainerName　　解密密钥对应的证书唯一标识
　　inFile　　待解密的密文文件路径,密文为数据类型 B
　　outFile　　明文文件保存路径
返回值：　true　　成功
　　false　　失败

7.3.25　消息签名 SOF_signMessage

原型：　java. lang. String SOF_SignMessage(byte[] inData)
描述：　对字符串数据进行数字签名,签名格式为带原文数据类型 B。
参数：　inData　　待签名的数据原文
返回值：　非空　　base64 编码的签名值
　　空　　失败

7.3.26　验证消息签名 SOF_verifySignedMessage

原型：　boolean SOF_verifySignedMessage(java. lang. String SignedMessage)
描述：　验证数字签名,签名格式为带原文数据类型 B。
参数：　SignedMessage　　base64 编码的消息签名包
返回值：　true　　成功
　　false　　失败

7.3.27　解析消息签名 SOF_getInfoFromSignedMessage

原型：　byte[] SOF_getInfoFromSignedMessage (
　　java. lang. String SignedMessage,short type)

描述：　解析数据类型 B 的签名包的信息，可获得原文、签名值、签名证书等信息。

参数：

SignedMessage	签名包
Type	type 定义：1：原文；2：签名者证书；3：签名值

返回值：

非空	返回 type 对应的值
空	失败

7.3.28　不带原文的消息签名 SOF_signMessageDetach

原型：　java.lang.String SOF_ signMessageDetach(byte[] inData)

描述：　对字符串数据进行数字签名，签名格式为不带原文数据类型 B。

参数：

inData	待签名的数据原文

返回值：

非空	base64 编码的签名值
空	失败

7.3.29　验证不带原文的消息签名 SOF_ verifySignedMessageDetach

原型：　boolean SOF_verifySignedMessageDetach (byte[] inData,
java.lang.String SignedMessage)

描述：　验证签名格式为不带原文数据类型 B 的数字签名。

参数：

inData	原文
SignedMessage	base64 编码的消息签名包

返回值：

true	成功
false	失败

7.3.30　XML 数字签名 SOF_signDataXML

原型：　java.lang.String SOF_signDataXML(java.lang.String inData)

描述：　对 XML 数据进行数字签名，输出符合 RFC3275 的 XML 签名结果。

参数：

InData	签名原文，XML 格式

返回值：

非空	输出 XML 格式的签名结果
空	失败

7.3.31　验证 XML 数字签名 SOF_verifySignedDataXML

原型：　boolean SOF_verifySignedDataXML(java.lang.String XMLSignedData)

描述：　验证 xml 签名。

参数：

XMLSignedData	带签名值的 XML 数据

返回值：

true	成功
false	失败

备注：　XML 签名标准为 RFC3275

7.3.32　解析 XML 签名数据 SOF_getXMLSignatureInfo

原型：　java.lang.String SOF_getXMLSignatureInfo(
java.lang.String XMLSignedData,short type)

描述：　解析 XML 签名数据，获取签名值、XML 原文、证书等信息。

参数：

XMLSignedData	XML 格式的签名数据
type	type 定义：1：XML 原文；2：摘要；3：签名值；4：签名证书；5：摘要算法；6：签名算法

返回值： 非空 type 对应的信息
空 失败

7.3.33 创建时间戳请求 SOF_createTimeStampRequest

原型： java.lang.String SOF_createTimeStampRequest(byte[]inData)
描述： 创建时间戳请求。
参数： inData 待创建时间戳请求的原文
返回值： 非空 base64 编码格式的时间戳请求
空 失败

7.3.34 创建时间戳响应 SOF_createTimeStampResponse

原型： java.lang.String SOF_createTimeStampResponse
(java.lang.String TimeStampRequest)
描述： 创建时间戳响应，即签发时间戳。
参数： TimeStampRequest 时间戳请求
返回值： 非空 base64 编码的时间戳响应
空 失败

7.3.35 验证时间戳 SOF_verifyTimeStamp

原型： boolean SOF_verifyTimeStamp(
java.lang.String inData,java.lang.String tsResponseData)
描述： 验证时间戳。
参数： inData 待验证的原文
tsResponseData 时间戳
返回值： true 成功
false 失败

7.3.36 解析时间戳 SOF_getTimeStampInfo

原型： java.lang.String SOF_getTimeStampInfo(
java.lang.String tsResponseData,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、签名值等。
参数： tsResponseData base64 编码的时间戳
type type 定义：1，返回时间；2，返回签名值；3，返回签名证书
返回值： 非空 type 对应的值
空 失败

7.3.37 获取最新的错误代码 SOF_GetLastError

原型： java.lang.long SOF_GetLastError()
描述： 获取接口最新的错误代码。
参数： 无
返回值： 详见错误代码 （详见错误代码表 A.1）。

附 录 A
（规范性附录）
证书应用综合服务接口错误代码定义

表 A.1 客户端控件和 COM 组件的错误代码表

宏描述	预定义值	说明
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_CspImprtPubKeyErr	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
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

表 A.1（续）

宏描述	预定义值	说明
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_CERTENCODEErr	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_PKCS7ENCODEErr	0X0B000417	PKCS7 编码格式错误
SOR_XMLENCODEErr	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

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

表 A.2（续）

异常描述	说　明
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_Pkcs7EncodeException	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
SOR_ParameterNotSupportException	不支持的参数

附 录 B
（资料性附录）
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。：

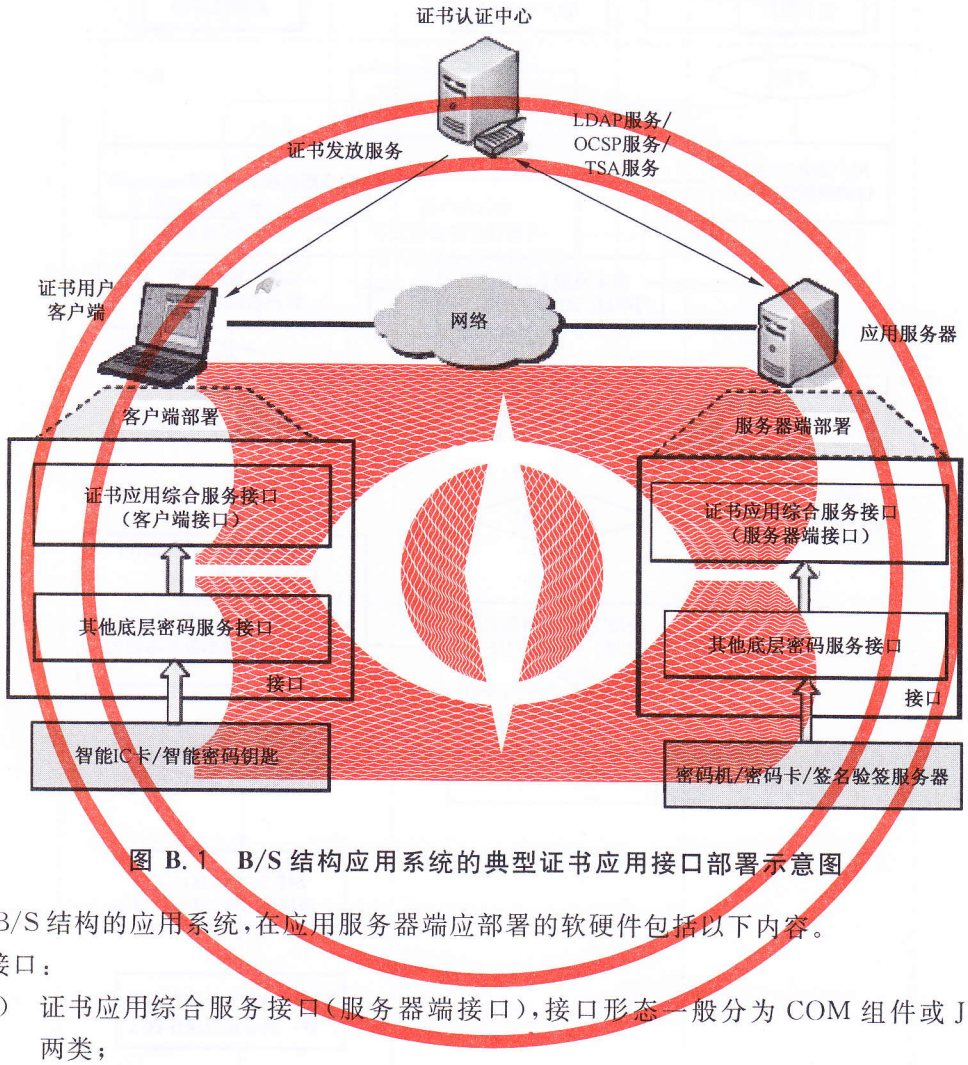

图 B.1 B/S 结构应用系统的典型证书应用接口部署示意图

对于 B/S 结构的应用系统，在应用服务器端应部署的软硬件包括以下内容。

a） 接口：

1） 证书应用综合服务接口（服务器端接口），接口形态一般分为 COM 组件或 JAVA 组件两类；

2） 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；

b） 密码设备：密码机、密码卡或签名验签服务器，用于服务器端的签名、验证、加密、解密等密码运算。

在证书用户客户端上应部署以下软硬件。

a） 接口：

1） 证书应用综合服务接口（客户端接口），接口形态一般分为 ActiveX 控件、DLL 动态库或 JAVA 类等三种形态，随着技术的发展接口形态可以进行扩展。

2） 其他底层密码服务接口，主要包括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。

b） 密码设备：智能密码钥匙（USBKey）等。

附 录 C
（资料性附录）
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：

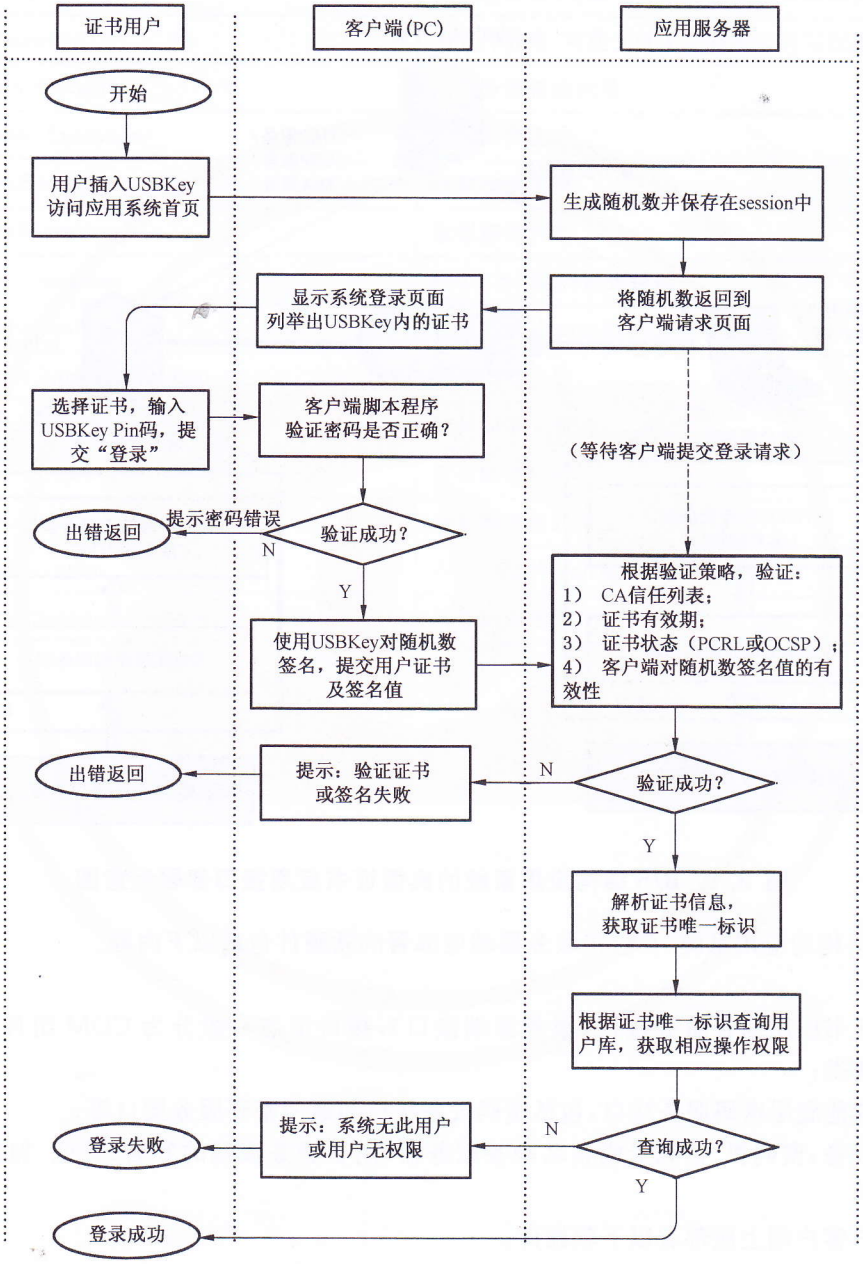

图 C.1 证书登录认证流程示意图

基于表单业务数据的签名和验签流程如图 C.2 所示：

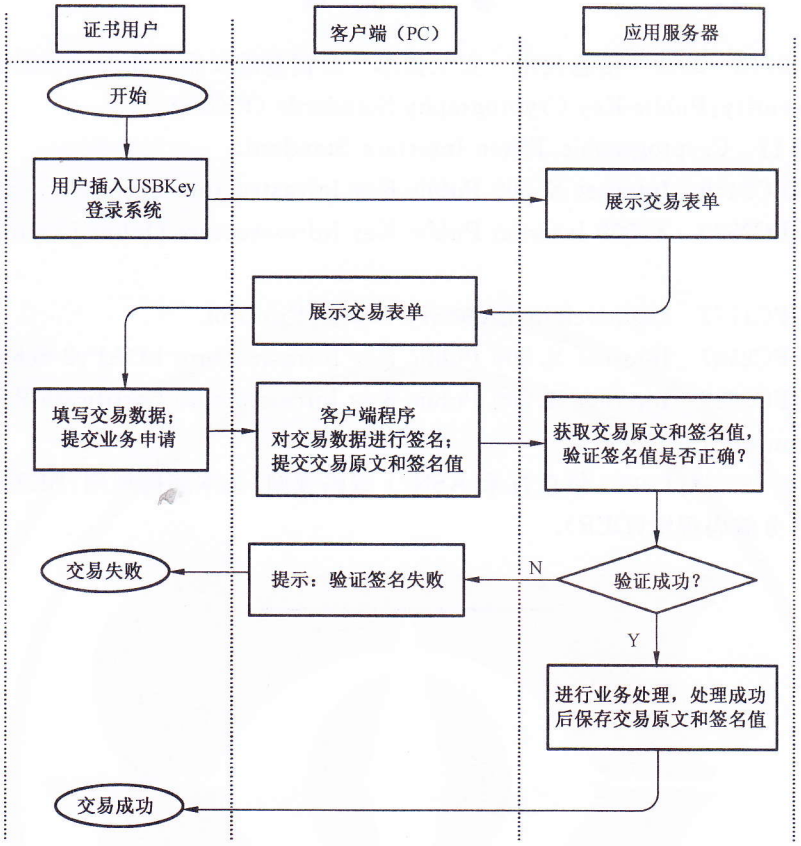

图 C.2　业务数字签名验证流程图

参 考 文 献

[1] GB/T 19713—2005 信息技术 安全技术 公钥基础设施 在线证书状态协议.

[2] RSA Security:Public-Key Cryptography Standards (PKCS).

[3] PKCS#11 Cryptographic Token Interface Standard.

[4] IETF RFC2459 Internet X.509 Public Key Infrastructure Certificate and CRL Profile.

[5] IETF RFC2560 X.509 Internet Public Key Infrastructure Online Certificate Status Protocol.

[6] IETF RFC1777 Lightweight Directory Access Protocol.

[7] IETF RFC2587 Internet X.509 Public Key Infrastructure LDAPv2 Schema.

[8] IETF RFC3647 Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework.

[9] ISO/IEC 8825-1:1998 信息技术-ASN.1 编码规则:基本编码规则(BER)的规范,正规编码规则(CER)和可区分编码规则(DER).

中华人民共和国密码
行业标准
证书应用综合服务接口规范
GM/T 0020—2012
*
中国标准出版社出版发行
北京市朝阳区和平里西街甲2号(100013)
北京市西城区三里河北街16号(100045)
网址 www.spc.net.cn
总编室:(010)64275323 发行中心:(010)51780235
读者服务部:(010)68523946
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 2.5 字数 63 千字
2013年1月第一版 2013年1月第一次印刷
*
书号:155066·2-24385 定价 36.00 元

GM/T 0020-2012