公钥密码基础设施应用技术体系

证书应用综合服务接口规范

Public Key Infrastructure Application Technology

Interface Specifications of Certificate Application Integrated Service

国 家 密 码 管 理 局

2010 年 8 月

目　　次

前　言

本规范是《公钥密码基础设施应用技术体系框架规范》系列规范之一，本规范为应用技术体系框架的典型密码服务层和应用层定义了统一的密码服务接口。

本规范的附录A为规范性附录，附录B为资料性附录。

本规范由国家密码管理局提出并归口。

本规范主要起草单位：北京数字证书认证中心、上海数字证书认证中心、卫士通信息产业股份有限公司、北京海泰方圆科技有限公司、兴唐通信科技股份有限公司。

本规范主要起草人：李述胜、崔久强、李元正、柳增寿等。

本规范责任专家：刘平。

本规范凡涉及密码算法相关内容，按国家有关法规实施。

引　　言

本规范依托于密码设备层的《公钥密码基础设施应用技术体系 通用密码服务接口规范》，向上为应用层规定了统一的、与密码协议无关、与密钥管理无关、与密码设备管理无关的高级密码服务接口。

证书应用综合服务接口在公钥密码基础设施支撑的前提下，向应用系统提供各类通用的密码服务，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。

本规范不涉及任何具体的密码运算，所有密码运算均在符合国家有关法规的密码设备中进行。

本规范编制过程中得到了国家商用密码应用技术体系总体工作组的指导。

公钥密码基础设施应用技术体系

证书应用综合服务接口规范

1 范围

本规范规定了与密码协议无关，与密钥管理无关，与密码设备管理无关的面向应用的统一应用服务接口。

本规范适用于公开密钥应用技术体系下密码应用服务的开发，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

下列标准所包含的条文，通过本规范中的引用而构成本规范的条文。考虑到标准的修订，使用本规范时，应研究使用下列标准最新版本的可能性。

GB/T 20518-2006 信息安全技术　公钥基础设施　数字证书格式

GB/T xxxxx 证书认证系统密码及其相关安全技术规范

GB/T AAAAA 信息技术 安全技术 密码术语

GB/T BBBBB 公钥密码基础设施应用技术体系 框架规范

GB/T CCCCC 公钥密码基础设施应用技术体系 密码设备应用接口规范

GB/T DDDDD 公钥密码基础设施应用技术体系 通用密码服务接口规范

GB/T EEEEE 公钥密码基础设施应用技术体系 密码设备管理规范；

GB/T FFFFF 公钥密码基础设施应用技术体系 责任认定密码技术规范；

GB/T GGGGG 公钥密码基础设施应用技术体系 身份鉴别接口规范；

GB/T HHHHH 公钥密码基础设施应用技术体系 单点登录接口规范；

GB/T IIIII 公钥密码基础设施应用技术体系 访问控制密码技术规范；

GB/T GGGGG 公钥密码基础设施应用技术体系 时间戳服务接口规范；

GB/T KKKKK 公钥密码基础设施应用技术体系 标识规范

GB/T LLLLL 智能IC卡及智能密码钥匙密码应用接口规范

GB/T MMMMM 信息技术 安全技术 密码术语

3 术语和定义

以下术语和定义适用于本规范。

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 pair

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存储空间的唯一性编号。

4 符号和缩略语

下列缩略语适用于本部分：

API　应用程序接口（Application Program Interface），简称应用接口
CA　证书认证机构（Certification Authority）
CN　通用名（Common Name）
CRL　证书撤销列表（Certificate Revocation List）
CSP　加密服务提供者（Cryptographic Service Provider）
DER　可区分编码规则（Distinguished Encoding Rules）
DN　可识别名（Distinguished Name）
LDAP　轻量级目录访问协议（Lightweight Directory Access Protocol）
OID　对象标识符（Object Identifier）
PKCS　公钥密码标准(the Public-Key Cryptography Standard)
SDS　国标密码设备应用接口

5　算法标识和数据结构

5.1　密码服务接口定义

5.1.1　常量定义

常量定义		
宏描述	预定义值	说明
#define SGD_TRUE	0x00000001	布尔值为真
#define SGD_FALSE	0x00000000	布尔值为假

5.1.2　全局参数定义

对称算法标识		
宏描述	预定义值	说明
#define SGD_SM1_ECB	0x00000101	SM1 算法 ECB 加密模式
#define SGD_SM1_CBC	0x00000102	SM1 算法 CBC 加密模式
#define SGD_SM1_CFB	0x00000104	SM1 算法 CFB 加密模式
#define SGD_SM1_OFB	0x00000108	SM1 算法 OFB 加密模式
#define SGD_SM1_MAC	0x00000110	SM1 算法 MAC 加密模式
#define SGD_SSF33_ECB	0x00000201	SSF33 算法 ECB 加密模式
#define SGD_SSF33_CBC	0x00000202	SSF33 算法 CBC 加密模式
#define SGD_SSF33_CFB	0x00000204	SSF33 算法 CFB 加密模式
#define SGD_SSF33_OFB	0x00000208	SSF33 算法 OFB 加密模式
#define SGD_SSF33_MAC	0x00000210	SSF33 算法 MAC 加密模式
非对称算法标识		
宏描述	预定义值	说明
#define SGD_RSA	0x00010000	RSA 算法
#define SGD_SM2_1	0x00020100	椭圆曲线签名算法
#define SGD_SM2_2	0x00020200	椭圆曲线密钥交换协议
#define SGD_SM2_3	0x00020400	椭圆曲线加密算法
杂凑算法标识		
宏描述	预定义值	说明
#define SGD_SM3	0x00000001	SM3 杂凑算法

#define SGD_SHA1	0x00000002	SHA1 杂凑算法
#define SGD_SHA256	0x00000004	SHA256 杂凑算法
接口描述标识		
宏描述	预定义值	说明
#define SGD_PROVIDER_CSP	0x00000001	CSP 接口
#define SGD_PROVIDER_PKCS11	0x00000002	PKCS#11 接口
#define SGD_PROVIDER_SDS	0x00000003	国标密码设备应用接口
#define SGD_KEYUSAGE_SIGN	0x00000001	签名/验证的密钥用途
#define SGD_KEYUSAGE_KEYEXCHANGE	0x00000002	加/解密的密钥用途
#define SGD_MODE_ECB	0x00000001	ECB 模式
#define SGD_MODE_CBC	0x00000002	CBC 模式
#define SGD_MODE_CFB	0x00000003	CFB 模式
#define SGD_MODE_OFB	0x00000004	OFB 模式
#define SGD_KEYINFO_DEV_GENERATE	0x00000001	设备产生
#define SGD_KEYINFO_KEY	0x00000002	外部输入 KEY
#define SGD_KEYINFO_IV	0x00000003	外部输入 IV
#define SGD_KEYINFO_PASSWORD_DERIVE_KEY	0x00000004	通过口令生成 KEY
证书解析标识		
宏描述	预定义值	说明
#define SGD_GET_CERT_VERSION	0x00000001	证书版本
#define SGD_GET_CERT_SERIAL	0x00000002	证书序列号
#define SGD_GET_CERT_ISSUER	0x00000005	证书颁发者 DN 信息
#define SGD_GET_CERT_VALID_TIME	0x00000006	证书有效期
#define SGD_GET_CERT_SUBJECT	0x00000007	证书拥有者 DN 信息
#define SGD_GET_CERT_DER_PUBLIC_KEY	0x00000008	证书公钥信息
#define SGD_GET_CERT_DER_EXTENSIONS	0x00000009	证书扩展项信息
#define SGD_EXT_AUTHORITYKEYIDENTIFIER	0x00000011	颁发者密钥标示符
#define SGD_EXT_SUBJECTKEYIDENTIFIER	0x00000012	证书持有者密钥标示符
#define SGD_EXT_KEYUSAGE	0x00000013	密钥用途
#define SGD_EXT_PRIVATEKEYUSAGEPERIOD	0x00000014	私钥有效期
#define SGD_EXT_CERTIFICATEPOLICIES	0x00000015	证书策略
#define SGD_EXT_POLICYMAPPINGS	0x00000016	策略映射
#define SGD_EXT_BASICCONSTRAINTS	0x00000017	基本限制
#define SGD_EXT_POLICYCONSTRAINTS	0x00000018	策略限制
#define SGD_EXT_EXTKEYUSAGE	0x00000019	扩展密钥用途
#define SGD_EXT_CRLDISTRIBUTIONPO	0x00000020	CRL 发布点
#define SGD_EXT_NETSCAPE_CERT_TYPE	0x00000021	netscape 属性
#define SGD_EXT_SELFDEFINED_EXTENSION	0x00000022	私有的自定义扩展项
#define SGD_EXT_IDENTIFYCARDNUMBER	0x00000023	个人身份证号码
#define SGD_EXT_INURANCENUMBER	0x00000024	个人社会保险号
#define SGD_EXT_ICREGISTRATIONNUMBER	0x00000025	企业工商注册号
#define SGD_EXT_ORGANIZATIONCODE	0x00000026	企业组织机构代码
#define SGD_EXT_TAXATIONNUMBER	0x00000027	企业税号

#define SGD_MAX_NAME_SIZE	0x00000080	名称最大长度
#define SGD_MAX_COUNT	0x00000100	列表最大长度
#define SGD_MAX_CONTAINER	0x00000800	容器容量
#define SGD_CERT_ISSUER_C	0x00000801	证书颁发者国家名
#define SGD_CERT_ISSUER_O	0x00000802	证书颁发者组织名
#define SGD_CERT_ISSUER_OU	0x00000803	证书颁发者部门名
#define SGD_CERT_ISSUER_S	0x00000804	证书颁发者省州名
#define SGD_CERT_ISSUER_CN	0x00000805	证书颁发者通用名
#define SGD_CERT_ISSUER_L	0x00000806	证书颁发者城市名
#define SGD_CERT_ISSUER_E	0x00000807	证书颁发者 EMAIL 地址
#define SGD_CERT_NOTBEFORE	0x00000808	证书起始时间
#define SGD_CERT_NOTAFTER	0x00000809	证书终止时间
#define SGD_CERT_SUBJECT_C	0x00000810	证书拥有者国家名
#define SGD_CERT_SUBJECT_O	0x00000811	证书拥有者组织名
#define SGD_CERT_SUBJECT_OU	0x00000811	证书拥有者部门名
#define SGD_CERT_SUBJECT_S	0x00000812	证书拥有者省州名
#define SGD_CERT_SUBJECT_CN	0x00000813	证书拥有者通用名
#define SGD_CERT_SUBJECT_L	0x00000814	证书拥有者城市名
#define SGD_CERT_SUBJECT_E	0x00000815	证书拥有者 EMAIL 地址

5.2 密码服务接口数据结构定义和说明

5.2.1 用户证书列表

字段名称	含义
certCount	证书总数
certificate	DER 编码的数字证书
certificateLen	数字证书的长度
containerName	容器名称
containerNameLen	容器名称的长度
keyUsage	密钥用途

实际数据结构定义：

```
typedef struct SGD_USR_CERT_ENUMLIST_ {
    unsigned int certCount;
    unsigned char *certificate[SGD_MAX_COUNT];
    unsigned int certificateLen[SGD_MAX_COUNT];
    unsigned char *containerName[SGD_MAX_COUNT];
    unsigned int containerNameLen[SGD_MAX_COUNT];
    unsigned int keyUsage[SGD_MAX_COUNT];
} SGD_USR_CERT_ENUMLIST;
```

5.2.2 用户密钥列表

字段名称	含义
keyPairCount	密钥对总数
containerName	容器名称

containerNameLen	容器名称的长度
keyUsage	密钥用途

实际数据结构定义：

```
typedef struct SGD_USR_KEYPAIR_ENUMLIST_ {
    unsigned int keyPairCount;
    unsigned char *containerName[SGD_MAX_COUNT];
    unsigned int containerNameLen[SGD_MAX_COUNT];
    unsigned int keyUsage[SGD_MAX_COUNT];
} SGD_USR_KEYPAIR_ENUMLIST;
```

5.2.3 证书中 DN 的结构

字段名称	数据长度（字节）	含义
dn_c	256	国家名称数组
dn_c_len	8	国家数组的长度
dn_s	256	省份或直辖市名称数组
dn_s_len	8	省份或直辖市名称数组的长度
dn_l	256	城市或地区的名称数组
dn_l_len	8	城市或地区的名称数组的长度
dn_o	640	机构名称数组
dn_o_len	20	机构名称数组的长度
dn_ou	640	机构单位名称数组
dn_ou_len	20	机构单位名称数组的长度
dn_cn	256	证书拥有者名称数组
dn_cn_len	8	证书拥有者名称数组的长度
dn_email	256	电子邮件数组
dn_email_len	8	电子邮件数组的长度

实际数据结构定义：

```
typedef struct{
    unsigned char dn_c[2][SGD_MAX_NAME_SIZE];
    unsigned int dn_c_len[2];
    unsigned char dn_s[2][SGD_MAX_NAME_SIZE];
    unsigned int dn_s_len[2];
    unsigned char dn_l[2][SGD_MAX_NAME_SIZE];
    unsigned int dn_l_len[2];
    unsigned char dn_o[5][SGD_MAX_NAME_SIZE];
    unsigned int dn_o_len[5];
    unsigned char dn_ou[5][SGD_MAX_NAME_SIZE];
    unsigned int dn_ou_len[5];
    unsigned char dn_cn[2][SGD_MAX_NAME_SIZE];
    unsigned int dn_cn_len[2];
    unsigned char dn_email[2][SGD_MAX_NAME_SIZE];
    unsigned int dn_email_len[2];
```

}SGD_NAME_INFO;

6 证书应用综合服务接口

6.1 证书应用综合服务接口在公钥密码基础设施应用技术体系框架中的位置

证书应用综合服务接口位于应用系统和通用密码服务接口之间，是数字证书应用支撑体系的软件基础，向应用层直接提供证书解析、证书认证、信息的机密性、完整性、不可否认性等高级通用密码服务，该层接口直接供应用系统调用，并将应用层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。统一证书应用接口为上层的证书应用系统提供简洁、易用的调用接口，屏蔽了各类密码设备（加密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。

统一的面向应用服务接在公钥密码基础设施应用技术体系框架内的位置如图1所示：

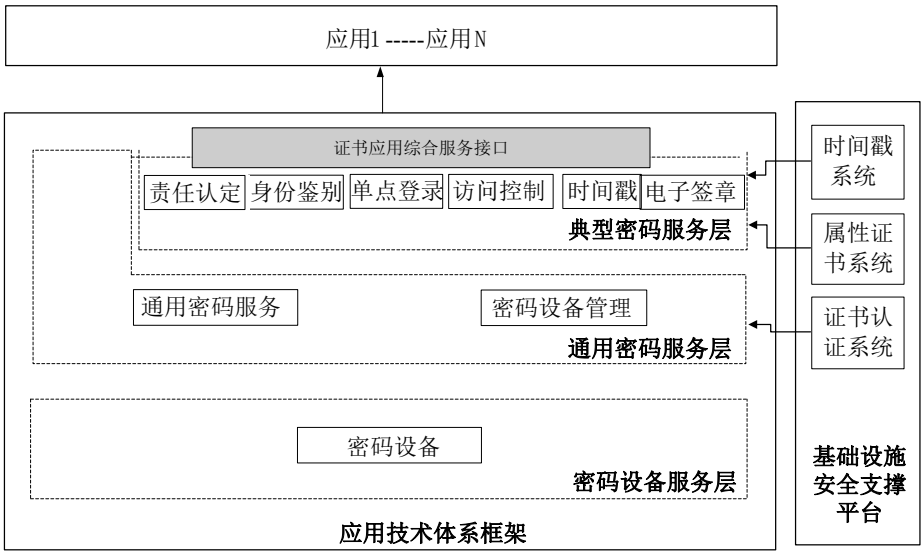


图1 证书应用综合服务接口在公钥密码基础设施应用技术体系框架内的位置

6.2 证书应用综合服务接口组成和功能

6.2.1 概述

证书应用综合服务接口分成客户端接口和服务器端接口两类，可满足B/S和C/S等多种架构的应用系统的调用需求。

6.2.2 客户端服务接口

客户端服务接口即客户端控件，客户端控件接口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，支持WindowsXP、Windows2000、Windows2003、Vista、Linux等操作终端类的主流操作系统。客户端控件接口的主要功能函数应包括：配置管理、数字证书解析、签名与验证、数据加密与解密、PKCS#7数据信封、XML数据的签名与验证、文件签名与加密等。

6.2.3 服务器端服务接口

服务器端服务接口（即服务器端组件）适用于服务器端程序调用，接口的形态包括COM组件、JAR格式的JAVA组件、WebService等形态，支持Windows、Linux、Unix、AIX、Solaris等所有服务器类的主流操作系统。服务器端组件接口的功能函数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、数据加密与解密、PKCS#7数据信封、XML数据的签名与验证、文件签名与加密等。

7 密码服务接口函数定义

7.1 客户端控件接口函数

7.1.1 客户端接口函数定义

本规范以ActiveX控件的形态接口为例，客户端控件包括以下函数：

A. 获取接口的版本号 SAF_GetVersion
B. 设置证书应用策略 SOF_SetCertAppPolicy
C. 设置签名算法 SOF_SetSignMethod
D. 获得当前签名算法 SOF_GetSignMethod
E. 设置加密算法 SOF_SetEncryptMethod
F. 获得加密算法 SOF_GetEncryptMethod
G. 获得证书列表 SOF_GetUserList
H. 导出用户证书 SOF_ExportUserCert
I. 导出用户证书 SOF_ExportExChangeUserCert
J. 获得证书信息 SOF_GetCertInfo
K. 获得证书扩展信息 vGetCertInfoByOid
L. 获得用户信息 SOF_GetUserInfo
M. 验证证书有效性 SOF_validateCert
N. 数据签名 SOF_SignData
O. 验证签名 SOF_VerifySignedData
P. 文件签名 SOF_SignFile
Q. 验证文件签名 SOF_VerifySignedFile
R. 加密数据 SOF_EncryptData
S. 解密数据 SOF_DecryptData
T. 文件加密 SOF_EncryptFile
U. 文件解密 SOF_DecryptFile
V. 公钥加密 SOF_PubKeyEncrypt
W. 私钥解密 SOF_PriKeyDecrypt
X. P7 数据签名 SOF_SignDataByP7
Y. 验证签名 SOF_VerifySignedDataByP7
Z. 解析 PKCS#7 签名包信息 SOF_GetP7SignDataInfo
AA. XML 数据签名 SOF_SignDataXML
BB. 验证 XML 数据签名 SOF_verifySignedDataXML
CC. 解析 XML 签名数据 SOF_getXMLSignatureInfo
DD. 拆分秘密 SOF_SecretSegment
EE. 秘密恢复 SOF_SecretRecovery
FF. 添加待处理的文件 SOF_AddFile
GG. 添加待处理的数据 SOF_AddString
HH. 清除文件和字符串 SOF_Clear
II. 开始对多个文件或数据签名 SOF_SignUpdate
JJ. 开始对多个文件或数据加密 SOF_EncUpdate
KK. 开始对多个文件或数据解密 SOF_DecUpdate
LL. 获得当前处理进度 SOF_getProgress
MM. 获得多文件或多字符串的总签名 SOF_GetTotalSignValue
NN. 获得多个文件和数据的时间戳请求 SOF_GetTotalTsReq
OO. 获得加密文件的个数 SOF_GetFileEncCount
PP. 获得加密字符串的个数 SOF_GetEncStringCount
QQ. 获得加密文件的路径 SOF_GetFileEncPath
RR. 获得加密文件的大小 SOF_GetFileEncSize
SS. 获得加密字符串 SOF_GetEncString

TT.检查控件支持 SOF_CheckSupport

UU.产生随机数 SOF_GenRandom

7.1.2 获取接口版本信息

原型： int SAF_GetVersion(unsigned int *puiVersion)

描述： 获取接口的版本号

参数： puiVersion [out] 版本号

返回值： 0 成功

非 0 失败，返回错误代码

备注： 版本号的格式为：0xAAAABBBB,其中 AAAA 为主版本号，BBBB 为次版本号。

7.1.3 设置证书应用策略 SOF_SetCertAppPolicy

原型： Void SOF_SetCertAppPolicy (BSTR AppPolicy);

描述： 设置证书应用策略。支持 “MS”。

“MS”：使用微软的证书存储规范，即 csp 规范，读取、使用证书。

参数： BSTR AppPolicy [in] 证书策略

返回值： 无 成功

7.1.4 设置签名算法 SOF_SetSignMethod

原型： Void SOF_SetSignMethod (BSTR SignMethod);

描述： 设置控件签名使用的签名算法。如果不调用此函数，控件缺省为“RSA-SHA1”。

参数： SignMethod [in] 签名算法

返回值： 无

备注： 签名算法支持的有：“RSA-SHA1”、“SHA256"等，参考表 5.1.2。

7.1.5 获得当前签名算法 SOF_GetSignMethod

原型： BSTR SOF_GetSignMethod ();

描述： 获得控件签名使用的签名算法

参数： 无

返回值： 当前的签名算法

7.1.6 设置加密算法 SOF_SetEncryptMethod

原型： Void SOF_SetEncryptMethod (BSTR EncryptMethod) ;

描述： 设置控件使用的对称加解密算法。

参数： SetEncryptMethod [IN] 对称加解密算法

返回值： 无

备注： 对称加解密算法参考表 5.1.2。

7.1.7 获得加密算法 SOF_GetEncryptMethod

原型： BSTR SOF_GetEncryptMethod ();

描述： 获得控件使用的对称加解密算法

参数： 无

返回值： 当前控件使用的加密算法

7.1.8 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： BSTR ret 用户列表字符串 数据格式：(用户 1||标识 1&&&用户 2||标识 2&&&…)

备注： 根据证书应用的策略不同会得到不同的证书列表

7.1.9 导出用户证书 SOF_ExportUserCert

原型： BSTR SOF_ExportUserCert(BSTR CertID);
描述： 根据证书惟一标识，获取 Base64 编码的证书字符串。
参数： BSTR CertID [in] 输入参数，证书惟一标识
返回值： BSTR rv 证书字符串
空 失败空值
备注： 如果是双证书，导出的是签名证书。
如果是单证书，导出的是签名加密证书。

7.1.10 导出用户证书 SOF_ExportExChangeUserCert

原型： BSTR SOF_ExportExChangeUserCert (BSTR CertID) ;
描述： 根据证书唯一标识，获取 Base64 编码的证书字符串。指定获取加密（交换）证书
参数： BSTR CertID[in] 证书唯一标识
返回值： BSTR rv 获取 Base64 编码的证书字符串
空值 失败
备注： CertID 证书唯一标识

7.1.11 获得证书信息 SOF_GetCertInfo

原型： BSTR SOF_GetCertInfo(BSTR Cert, short Type);
描述： 获取证书信息
参数： BSTR sCert[in] Base64 编码的证书
short Type[in] 获取信息的类型，TYPE 参数见 5.1.2“全局参数定义表”中的
返回值： BSTR ret 证书信息
空值 失败

7.1.12 获得证书扩展信息 SOF_GetCertInfoByOid

原型： BSTR SOF_GetCertInfoByOid(BSTR Cert, BSTR Oid) ;
描述： 根据 OID 获取证书私有扩展项信息
参数： BSTR sCert[in] Base64 编码的证书
BSTR oid [in] 私有扩展对象 ID，比如“1.2.18.21.88.2”
返回值： BSTR ret 证书 OID 对应的值
空值 失败

7.1.13 获得用户信息 SOF_GetUserInfo

原型： BSTR SOF_GetUserInfo (BSTR CertId, short type) ;
描述： 获得用户信息
参数： BSTR Certid[in] 证书标识
type[in] 类别，参数和意义如下表所示
返回值： BSTR ret type 对应的值
空值 失败

type 参数和意义如下表所示：

参数	值	意义
CERT_SUBJECT	1	用户名
CERT_UNIQUEID	2	唯一标示
CERT_DEPT	3	部门
CERT_ISSUE	4	颁发者

CERT_DEVICETYPE	8	介质类型
CERT_CATYPE	9	CA 类型
CERT_KEYTYPE	10	用户证书密钥类型，（双证或单证）
CERT_DEVICENAME	13	用户证书介质名称
CERT_DEVICEPROVIDER	14	用户证书介质提供者即 csp 名称
CERT_DEVICEAFFIX	15	用户证书介质附加库
CERT_SIGNPATH	16	用户签名证书路径
CERT_EXCHPATH	17	用户加密证书路径
CERT_SIGNPFXPATH	18	用户签名 P12 证书路径
CERT_EXCHPFXPATH	19	用户加密 P12 证书路径
CERT_UNIQUEIDOID	22	用户证书 UniqueID 的 OID

7.1.14 验证证书有效性 SOF_validateCert

原型： boolean SOF_ValidateCert(BSTR Cert) ;
描述： 验证证书有效性
参数： BSTR sCert[in] Base64 编码的证书
返回值： True 成功
False 失败
空值 失败

7.1.15 数据签名 SOF_SignData

原型： BSTR SOF_SignData(BSTR CertID, BSTR InData,short InDataLen) ;
描述： 对字符串数据进行数字签名，签名格式为 Pkcs#1
参数： BSTR sCertID[in] 证书标识
BSTR sInData[in] 签名原文
Short InDataLen[in] 签名原文长度
返回值： BSTR ret 签名结果
空值 失败

7.1.16 验证签名 SOF_VerifySignedData

原型： boolean SOF_VerifySignedData(BSTR Cert, BSTR InData, short InDataLen, BSTR SignValue) ;
描述： 验证数字签名
参数： BSTR sCert[in] 签名者证书，BASE64 编码
BSTR sInData[in] 签名原文
Short InDataLen[in] 签名原文长度
BSTR sSignValue[in] 签名值，BASE64 编码
返回值： True 成功
False 失败

7.1.17 文件签名 SOF_SignFile

原型： BSTR SOF_SignFile(BSTR CertID, BSTR InFile) ;
描述： 对文件数字签名
参数： BSTR sCertID[in] 证书标识

	BSTR sInFile[in]	签名原文文件路径
返回值：	BSTR ret	签名结果
	空值	失败

7.1.18 验证文件签名 SOF_VerifySignedFile

原型： boolean SOF_VerifySignedFile(BSTR Cert, BSTR InFile, BSTR SignValue) ;

描述： 验证文件数字签名

参数：	BSTR sCert[in]	签名者证书
	BSTR sInFile[in]	签名原文文件路径
	BSTR sSignValue[in]	签名值
返回值：	True	成功
	False	失败

7.1.19 加密数据 SOF_EncryptData

原型： BSTR SOF_EncryptData(BSTR SymmKey, BSTR Indata) ;

描述： 使用对称算法加密数据

参数：	BSTR sKey[in]	加密密钥
	BSTR sIndata[in]	待加密的明文
返回值：	BSTR rv	加密后的密文
	空值	失败

7.1.20 解密数据 SOF_DecryptData

原型： BSTR SOF_DecryptData(BSTR SymmKey, BSTR Indata) ;

描述： 使用对称算法解密数据

参数：	BSTR SymmKey[in]	解密密钥
	BSTR sIndata[in]	待解密的密文
返回值：	BSTR rv	解密后的明文
	空值	失败

7.1.21 文件加密 SOF_EncryptFile

原型： boolean SOF_EncryptFile(BSTR SymmKey, BSTR InFile, BSTR OutFile) ;

描述： 使用对称算法加密文件

参数：	BSTR SymmKey[in]	加密密钥
	BSTR InFile[in]	待加密的明文文件路径
	BSTR OutFile[in]	密文文件保存路径
返回值：	True	成功
	False	失败

7.1.22 文件解密 SOF_DecryptFile

原型： boolean SOF_DecryptFile(BSTR SymmKey, BSTR InFile, BSTR OutFile) ;

描述： 使用对称算法解密文件

参数：	BSTR SymmKey[in]	解密密钥
	BSTR InFile[in]	待解密的密文文件路径
	BSTR OutFile[in]	明文文件保存路径
返回值：	True	成功
	False	失败

7.1.23 公钥加密 SOF_PubKeyEncrypt

原型： BSTR SOF_PubKeyEncrypt(BSTR Cert, BSTR InData) ;

描述： 使用证书对数据加密。（Pkcs#1 格式）

参数：	BSTR Cert[in]	证书

BSTR InData[in] 待加密的数据

返回值： BSTR rv 成功加密后的密文

空值 失败

备注： 因为是 pkcs#1 格式，故加密的数据长度要小于证书的位数。比如 1024 位的证书，InData 长度必须小于 128

7.1.24 私钥解密 SOF_PriKeyDecrypt

原型： BSTR SOF_PriKeyDecrypt(BSTR CertID, BSTR InData) ;

描述： 私钥解密（Pkcs#1 格式）

参数： BSTR CertID[in] 证书 ID

BSTR InData[in] 待解密的数据

返回值： BSTR rv 成功解密后的明文

空值 失败

7.1.25 P7 数据签名 SOF_SignDataByP7

原型： BSTR SOF_SignDataByP7 (BSTR CertID, BSTR InData)

描述： 对字符串数据进行数字签名，签名格式为 Pkcs7

参数： BSTR sCertID[in] 证书标识

BSTR sInData[in] 签名原文

返回值： BSTR ret 签名结果

空值 失败

备注： PKCS#7 签名结果包含原文+签名者证书+签名值

7.1.26 验证签名 SOF_VerifySignedDataByP7

原型： boolean SOF_VerifySignedDatabyP7(BSTR P7Data) ;

描述： 验证数字签名

参数： BSTR P7Data[in] PKCS#7 签名包

返回值： True 成功

False 失败

7.1.27 解析 PKCS#7 签名包信息 SOF_GetP7SignDataInfo

原型： BSTR SOF_GetP7SignDataInfo (BSTR P7Data, short type) ;

描述： 解析 PKCS#7 签名包的信息，可获得原文、签名值、签名证书等信息

参数： BSTR P7Data[in] PKCS#7 签名包

short type[in] 类型

返回值： True 成功

False 失败

备注： 类型：1：原文；2：签名者证书；3：签名值

7.1.28 XML 数据签名 SOF_SignDataXML

原型： BSTR SOF_SignDataXML(BSTR CertID, BSTR InData) ;

描述： 对 XML 数据进行数字签名，输出符合国际标准的 XML 签名结果

参数： BSTR sCertID[in] 证书标识

BSTR InData[in] 签名原文，XML 格式

返回值： BSTR ret 签名结果

空值 失败

备注： XML 签名标准见 http://www.w3.org/TR/xmldsig-core/

7.1.29 验证 XML 数据签名 SOF_verifySignedDataXML

原型： boolean SOF_verifySignedDataXML (BSTR InData) ;

描述： 验证 xml 签名

参数： BSTR InData[in] XML 签名值

返回值： True 成功

False 失败

备注： XML 签名标准见 http://www.w3.org/TR/xmldsig-core/

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： BSTR XMLSignedData[in] XML 格式的签名数据

Type[in] 待解析的参数类型

返回值： 各项对应的信息

备注： type 可选的参数和意义：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。

7.1.31 拆分秘密 SOF_SecretSegment

原型： BSTR SOF_SecretSegment(BSTR Secert, short m, short n, short k) ;

描述： 门限算法，拆分秘密

参数： BSTR sSecert[in] 待拆分的秘密（Base64 编码后）

int m[in] 秘密分割份额

int n[in] 秘密恢复最小份额

int k[in] 设置为恢复秘密时的必选片段的个数。(此参数可选，不设置表示任意 n 份片段均可恢复秘密）

返回值： BSTR rv 拆分后的密码（Base64 编码)，共 m 份，以&&&相连。如“111&&&222&&&”。

空值 失败

备注： 若有必选片段，则前 k 份为必选

7.1.32 秘密恢复 SOF_SecretRecovery

原型： BSTR SOF_SecretRecovery(BSTR Seg) ;

描述： 门限算法，恢复秘密

参数： BSTR Seg[in] 密钥片段。以“&&&”相连的 n 段密钥片段

返回值： BSTR rv 恢复后的秘密

空值 失败

7.1.33 添加待处理的文件 SOF_AddFile

原型： boolean SOF_AddFile(BSTR FilePath) ;

描述： 针对多个文件或数据处理，添加待处理的文件

参数： BSTR FilePath[in] 文件路径

返回值： True 成功

False 失败

7.1.34 添加待处理的数据 SOF_AddString

原型： boolean SOF_AddString(BSTR Indata) ;

描述： 针对多个文件或数据处理，添加待处理的字符串

参数： BSTR sIndata[in] 字符串

返回值： True 成功

False 失败

7.1.35 清除文件和字符串 SOF_Clear

原型： void SOF_Clear();

描述： 清除添加的文件和字符串

参数： 无

返回值： 无

7.1.36 开始对多个文件或数据签名 SOF_SignUpdate

原型： boolean SOF_SignUpdate(BSTR CertID) ;

描述： 开始对多个文件或数据签名。先对文件签名后对字符串签名。对文件签名的顺序按照 AddFile 的顺序，对字符串签名的顺序对 AddString 的顺序。

参数：	BSTR CertID[in]	证书标识
返回值：	boolean True	成功
	Boolean False	失败

7.1.37 开始对多个文件或数据加密 SOF_EncUpdate

原型： boolean SOF_EncUpdate(BSTR Key) ;

描述： 开始对多个文件或数据加密。先对文件加密后对字符串加密。对文件加密的顺序按照 AddFile 的顺序，对字符串加密的顺序对 AddString 的顺序。

参数：	BSTR Key[in]	加密密钥
返回值：	无	

7.1.38 开始对多个文件或数据解密 SOF_DecUpdate

原型： boolean SOF_DecUpdate(BSTR Key) ;

描述： 开始对多个文件或数据解密。先对文件解密后对字符串解密。对文件解密的顺序按照 AddFile 的顺序，对字符串解密的顺序对 AddString 的顺序。

参数：	BSTR Key[in]	加密密钥
返回值：	无	

7.1.39 获得当前处理进度 SOF_getProgress

原型： short SOF_GetProgress();

描述： 获得当前操作处理的进度（多文件签名或多文件加密的进度）

参数： 无

返回值：	int rv	取值范围为 0-100

7.1.40 获得多文件或多字符串的总签名 SOF_GetTotalSignValue

原型： BSTR SOF_GetTotalSignValue();

描述： 获得多文件或多字符串的总签名。此函数须在调用 SignUpdate 函数成功后执行。

参数： 无

返回值：	BSTR rv	总签名值
	空值	失败

7.1.41 获得多个文件和数据的时间戳请求 SOF_GetTotalTsReq

原型： BSTR SOF_GetTotalTsReq ();

描述： 获得多文件或多字符串的总时间戳请求。此函数须是调用 SignUpdate 函数成功后才可调用。

参数： 无

返回值：	BSTR rv	总时间戳请求
	空值	失败

7.1.42 获得加密文件的个数 SOF_GetFileEncCount

原型： short SOF_GetFileEncCount();

描述： 获得多文件加密处理的文件个数。此函数必须是调用 EncUpdate 函数成功后才可调用。

参数： 无

返回值： 文件个数

7.1.43 获得加密字符串的个数 SOF_GetEncStringCount

原型： short SOF_GetEncStringCount ();

描述： 获得多字符串加密处理的字符串个数。此函数必须是调用 EncUpdate 函数成功件后才可调用。

参数： 无

返回值： 字符串个数

7.1.44 获得加密文件的路径 SOF_GetFileEncPath

原型： BSTR SOF_GetFileEncPath(short index);

描述： 获得多文件加密处理的文件路径。此函数必须是调用 EncUpdate 函数成功后才可调用。

参数： 文件索引 从 0 开始

返回值： 密文文件路径

7.1.45 获得加密文件的大小 SOF_GetFileEncSize

原型： Short SOF_GetFileEncSize (short index);

描述： 获得多文件加密处理的文件大小。此函数必须是调用 EncUpdate 函数成功后才可调用。

参数： 文件索引 从 0 开始

返回值： 密文文件大小

7.1.46 获得加密字符串 SOF_GetEncString

原型： BSTR SOF_GetEncString(short index);

描述： 获得多字符串加密处理的字符串。此函数必须是调用 EncUpdate 函数成功后才可调用。

参数： 字符串索引 从 0 开始

返回值： 密文文件大小

7.1.47 检查控件支持 SOF_CheckSupport

原型： short SOF_CheckSupport();

描述： 检查控件是否支持当前操作系统和 IE 版本

参数： 无

返回值： int rv 0 表示支持，1 表示不支持。

7.1.48 产生随机数 SOF_GenRandom

原型： BSTR SOF_GenRandom(short len);

描述： 产生随机数

参数： int RanddomLen[in] 待产生的随机数长度（bytes，字节长度）

返回值： BSTR rv 随机数值（Base64 编码后的）

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_QueryCertTrustListAltNames

C. 查询证书信任列表 SOF_QueryCertTrustList

D. 删除证书信任列表 SOF_DelCertTrustList

E. 设置 Web 应用名称 SOF_SetWebAppName

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. 数据签名 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. 拆分秘密 SOF_SecretSegment
T. 秘密恢复 SOF_SecertRecovery
U. 对称算法加密数据 SOF_EncryptData
V. 解密数据 SOF_DecryptData
W. 文件加密 SOF_EncryptFile
X. 文件解密 SOF_DecryptFile
Y. 公钥加密 SOF_PubKeyEncrypt
Z. 私钥解密 SOF_PriKeyDecrypt
AA. 添加待处理的文件 SOF_AddFile
BB. 添加待处理的数据 SOF_AddString
CC. 清除文件和字符串 SOF_Clear
DD. 对多个文件或数据签名 SOF_SignAll
EE. 对多个文件或数据验证签名 SOF_VerifySignAll
FF. 获得多个文件或数据的时间戳请求 SOF_GetTotalTsReq
GG. P7 数据签名 SOF_SignDataPkcs7
HH. P7 验证签名 SOF_VerifySignedDataPkcs7
II. 解析 PKCS#7 签名包信息 SOF_GetP7SignDataInfo
JJ. XML 数据签名 SOF_SignDataXML
KK. 验证 XML 数据签名 SOF_VerifySignedDataXML
LL. 解析 XML 签名数据 SOF_GetXMLSignatureInfo
MM. 创建时间戳请求 SOF_CreateTimeStampRequest
NN. 创建时间戳响应 SOF_CreateTimeStampResponse
OO. 验证时间戳 SOF_VerifyTimeStamp
PP. 解析时间戳 SOF_GetTimeStampInfo

7.2.2 设置证书信任列表 SOF_SetCertTrustList

原型： short SOF_SetCertTrustList(BSTR CTLAltName, BSTR CTLContent, int 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] 证书信任列表内容(Base64 编码格式)
CTLContentLen[in] 证书信任列表长度
返回值： 0 成功
其他 失败，详见错误码列表
备注： 错误代码：SOR_PARAMERR：参数错误

7.2.3 查询证书信任列表别名 SOF_QueryCertTrustListAltNames

原型： BSTR SOF_QueryCertTrustListAltNames();
描述： 查询证书信任列表别名
参数： 无
返回值： 非空 成功，信任列表别名的字符串组合，如“CA001@CA002@CA003”
其他 失败，详见错误码列表
备注： 错误代码：SOR_CTLNOTFOUND：没有信任列表

7.2.4 查询证书信任列表 SOF_QueryCertTrustList

原型： BSTR SOF_QueryCertTrustList (BSTR CTLAltName);
描述： 根据别名查询证书信任列表
参数： BSTR CTLAltName[in] 证书信任列表别名
返回值： 非空 成功，Base64 编码格式的证书信任列表；
其他 失败，详见错误码列表
备注： SOR_CTLNOTFOUND：没有信任列表
SOR_PARAMERR：参数错误。

7.2.5 删除证书信任列表 SOF_DelCertTrustList

原型： short 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，详见错误码列表
备注： SOR_NULLPOINTER：参数为空指针
SOR_CTLNOTFOUND：没有对应的信任列表

7.2.6 设置 Web 应用名称 SOF_SetWebAppName

原型： short SOF_SetWebAppName(BSTR WebAppName);
描述： 设置 web 应用的名称。该名称要和服务器 xml 文件对应。接口从 xml 文件中读取 webapp 的属性包括使用的证书、可信的根证书、证书验证的策略等
参数： WebAppName[in] 应用的名称
返回值： 0 成功
其他 失败，详见错误码列表
备注： SOR_NULLPOINTER：参数为空指针。
SOR_APPNOTFOUND：应用没发现

7.2.7 设置签名算法 SOF_SetSignMethod

原型： Short SOF_SetSignMethod (BSTR signMethod);
描述： 设置 COM 组件签名运算使用的签名算法。缺省为“RSA-SHA1”。
参数： signMethod[IN] 签名算法
返回值： 0 成功
其他失败 详见附录错误码定义
备注： SOR_NULLPOINTER：参数为空指针。
SOR_PARAMETERNOTSUPPORT ：不支持的参数

7.2.8 获得当前签名算法 SOF_getSignMethod

原型： BSTR SOF_signMethod getSignMethod ();
描述： 获得组件签名运算使用的签名算法
参数： 无
返回值： 当前的签名算法

7.2.9 设置加密算法 SOF_SetEncryptMethod

原型： short SOF_SetEncryptMethod (BSTR EncryptMethod);

描述： 设置组件对数据加解密使用的对称算法

参数： EncryptMethod [IN] 对称加解密算法参考表 5.1.2。

返回值： 0 成功

备注： 错误代码：

SOR_NULLPOINTER：参数为空指针。

SOR_PARAMETERNOTSUPPORT ：不支持的参数

7.2.10 获得加密算法 SOF_GetEncryptMethod

原型： BSTR SOF_GetEncryptMethod ();

描述： 获得组件使用的对称加解密算法

参数： 无

返回值： 当前控件使用的加密算法

7.2.11 获得服务器证书 SOF_GetServerCertificate

原型： BSTR SOF_GetServerCertificate (int CertUsage);

描述： 读取当前应用指定的服务器证书

参数： INT certUsage 证书用途 证书用途法 1：交换证书、2：签名证书

返回值： Base64 编码的服务器证书 成功

null 失败

备注： 错误代码 SOR_PARAMETERNOTSUPPORT：不支持的参数

7.2.12 产生随机数 SOF_GenRandom

原型： BSTR SOF_GenRandom(int len);

描述： 产生指定长度的随机数

参数： int len[in] 待产生的随机数长度（bytes，字节长度）

返回值： 随机数值 Base64 编码后的

备注： bytes，字节长度

7.2.13 获得证书信息 SOF_GetCertInfo

原型： BSTR SOF_GetCertInfo(BSTR Base64EncodeCert, int type);

描述： 获取证书信息

参数： BSTR Base64EncodeCert Base64 编码的 X.509 数字证书

int type 获取证书信息的类型，见 5.1.2 的参数表.

返回值： BSTR ret 证书信息

空值 失败

备注： 错误代码：

SOR_NULLPOINTER：某一个参数为空指针。

SOR_CERTENCODE：证书编码格式错误

SOR_PARAMETERNOTSUPPORT ：不支持的参数

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 获取证书私有扩展项信息

参数： BSTR Base64EncodeCert Base64 编码的证书

BSTR oid 私有扩展对象 ID，如“1.2.156.197.1.103”

返回值： BSTR ret 证书 OID 对应的值

空值 出错

备注： 错误代码：

SOR_NULLPOINTER：某一个参数为空指针。
SOR_CERTENCODE：证书编码格式错误。

7.2.15 验证证书有效性 SOF_ValidateCert

原型： short SOF_ValidateCert(BSTR Base64EncodeCert)
描述： 根据应用的策略根据验证证书有效性
参数： Base64EncodeCert[IN] 待验证的 base64 编码证书
返回值： True 验证成功
False 验证失败
备注： 错误代码：
SOR_NULLPOINTER: Base64EncodeCert 为 null。
SOR_CERTENCODE：证书编码格式错误。
SOR_CERTINVALID－证书无效，不是可信 ca 颁发的证书。
SOR_CERTNOTYETVALID－证书未生效。
SOR_CERTHASEXPIRED－证书已过期。
SOR_CERTREVOKED－证书已经被吊销

7.2.16 数据签名 SOF_SignData

原型： BSTR SOF_SignData(BSTR InData,int InDataLen);
描述： 对字符串数据进行数字签名，签名格式为 Pkcs#1
参数： InData[IN] 待签名的数据原文
InDataLen[IN] 待签名的数据原文长度
返回值： 返回 pkcs#1 格式的签名值的 base64 编码 成功
备注： 错误代码：
SOR_NULLPOINTER: InData 为 null。
SOR_SIGNDATA：签名失败。

7.2.17 验证签名 SOF_VerifySignedData

原型： short SOF_VerifySignedData(BSTR Base64EncodeCert, BSTR InData,int InDataLen ,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
InDataLen[IN] 待验证的原文长度
SignValue[IN] 签名值
返回值： 0 验证成功
其他 验证失败
备注： 错误代码：
SOR_NULLPOINTER: 其中一个输入参数为 null。
SOR_CERTENCODE：证书编码错误。
SOR_VERIFYSIGNDATA：验证签名失败

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 编码后的 Pkcs#1 格式的签名数据。
参数： InFile[IN] 待签名的文件路径
返回值： base64 编码后的 Pkcs#1 格式的签名数据

备注： 错误代码：
SOR_NULLPOINTER: InFile 为 null。
SOR_READFILE：读文件异常，可能文件不存在或没有读取权限等。

7.2.19 验证文件签名 SOF_VerifySignedFile

原型： short 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
SignValue[IN] 签名值
返回值： True 验证成功
False 验证失败
备注： 错误代码：
SOR_NULLPOINTER: 其中一个输入参数为 null。
SOR_READFILE：读文件异常，可能文件不存在或没有读取权限等。
SOR_CERTENCODE：证书编码错误。
SOR_VERIFYSIGNDATA：验证签名失败。

7.2.20 拆分秘密 SOF_SecretSegment

原型： BSTR SOF_SecretSegment(BSTR Secert, int m, int n, int k);
描述： 门限算法，拆分秘密
参数： BSTR Secert[IN] 待拆分的秘密（Base64 编码后的数据）
int m[IN] 秘密分割份额
int n[IN] 秘密恢复最小份额
int k[IN] 设置为恢复秘密时的必选片段的个数。（此参数可选，不设置标识任意 n 份片段均可恢复秘密）
返回值： 成功拆分后的密码(Base64 编码)。 共 m 份，以&&&相连。如:“111&&&222&&&”。
空值 失败
备注： 若有必选片段，则前 k 份为必选
错误代码：
SOR_NULLPOINTER: 其中一个输入参数为 null。
SOR_SECRETSEGMENT：门限分割算法失败。

7.2.21 秘密恢复 SOF_SecertRecovery

原型： BSTR SOF_SecertRecovery(BSTR seg);
描述： 门限算法，恢复秘密
参数： BSTR seg[IN] 密钥片段，“&&&”相连的 n 段密钥片段
返回值： 恢复后的秘密 成功
空值 失败
备注： SOR_NULLPOINTER: 其中一个输入参数为 null。
SOR_SECERTRECOVERY：门限恢复失败。

7.2.22 对称算法加密数据 SOF_EncryptData

原型： BSTR SOF_EncryptData(BSTR key, BSTR InData);
描述： 使用对称算法加密数据
参数： BSTR key[IN] 加密密钥
BSTR InData[IN] 待加密的明文

返回值： 成功加密后的密文 Base64 编码后的
空值 失败
备注： SOR_NULLPOINTER: 其中一个输入参数为 null。
SOR_ENCRYPTDATA：数据加密失败。

7.2.23 解密数据 SOF_DecryptData

原型： BSTR SOF_DecryptData(BSTR key, BSTR InData);
描述： 使用对称算法解密数据
参数： BSTR key[IN] 解密密钥
BSTR InData[IN] 待解密的密文
返回值： 解密后的明文 成功
空值 失败
备注： 错误代码：
SOR_NULLPOINTER: 其中一个输入参数为 null。
SOR_DECRYPTDATA： 数据解密失败。

7.2.24 文件加密 SOF_EncryptFile

原型： Short SOF_EncryptFile(BSTR key, BSTR InFile, BSTR OutFile);
描述： 使用对称算法加密文件
参数： BSTR key[IN] 加密密钥
BSTR InFile[IN] 待加密的明文文件路径
BSTR OutFile[IN] 密文文件保存路径
返回值： 0 成功
其他 见错误码
备注： SOR_NULLPOINTER: 其中一个输入参数为 null。
SOR_READFILE: 读文件异常，可能文件不存在或没有读取权限等。
SOR_WRITEFILE 写文件异常，可能文件不存在或没有写权限等
SOR_ENCRYPTDATA：数据加密失败

7.2.25 文件解密 SOF_DecryptFile

原型： short SOF_DecryptFile(BSTR key, BSTR InFile, BSTR OutFile);
描述： 使用对称算法解密文件
参数： BSTR key [IN] 解密密钥
BSTR InFile[IN] 待解密的密文文件路径
BSTR OutFile[IN] 明文文件保存路径
返回值： 0 成功
其他 见错误码
备注： 错误代码：
SOR_NULLPOINTER: 其中一个输入参数为 null。
SOR_READFILE: 读文件异常，可能文件不存在或没有读取权限等。
SOR_WRITEFILE 写文件异常，可能文件不存在或没有写权限等。
SOR_DECRYPTDATA：数据解密失败

7.2.26 公钥加密 SOF_PubKeyEncrypt

原型： BSTR SOF_PubKeyEncrypt(BSTR Base64EncodeCert, BSTR InData);
描述： 使用证书对数据加密。（Pkcs#1 格式）
参数： BSTR Base64EncodeCert [IN] 证书
BSTR InData[IN] 待加密的数据
返回值： 返回加密后的密文 base64 编码后的数据

	空	失败

备注：错误代码：

SOR_NULLPOINTER: 其中一个输入参数为 null。

SOR_CERTENCODE：证书编码格式错误。

SOR_ENCRYPTDATA：数据加密失败。

因为是 pkcs#1 格式，故加密的数据长度要小于证书的位数。比如 1024 位的证书，InData 长度必须小于 128。

7.2.27 私钥解密 SOF_PriKeyDecrypt

原型：BSTR SOF_PriKeyDecrypt(BSTR InData);

描述：私钥解密（Pkcs#1 格式）

参数：	BSTR InData[IN]	待解密的数据
返回值：	返回解密后的明文	成功
	空	失败

备注：错误代码：

SOR_NULLPOINTER: 其中一个输入参数为 null。

SOR_DECRYPTDATA：数据解密失败。

7.2.28 添加待处理的文件 SOF_AddFile

原型：Short SOF_AddFile(BSTR filePath);

描述：针对多个文件或数据处理，添加待处理的文件

参数：	BSTR filePath[IN]	文件路径
返回值：	0	成功
	其他	见错误码

备注：错误代码：

SOR_NULLPOINTER: 其中一个输入参数为 null。

SOR_READFILE：读文件异常，可能文件不存在或没有读取权限等。

7.2.29 添加待处理的数据 SOF_AddString

原型：short SOF_AddString(BSTR InData);

描述：针对多个文件或数据处理，添加待处理的字符串

参数：	BSTR InData[IN]	字符串
返回值：	0	成功
	其他	见错误码

备注：错误代码：

SOR_NULLPOINTER: 其中一个输入参数为 null

7.2.30 清除文件和字符串 SOF_Clear

原型：void SOF_Clear();

描述：清除添加的文件和字符串

参数：无

返回值：无

7.2.31 对多个文件或数据签名 SOF_SignAll

原型：BSTR SOF_SignAll();

描述：对多个文件或数据签名。先对文件签名后对字符串签名。对文件签名的顺序按照 AddFile 的顺序，对字符串签名的顺序对 AddString 的顺序。

参数：	无	
返回值：	返回全部数据的签名值	成功
	null	失败

备注： 错误代码：
SOR_SIGNDATA: 签名失败

7.2.32 对多个文件或数据验证签名 SOF_VerifySignAll

原型： short SOF_VerifySignAll (BSTR Base64EncodeCert, BSTR SignValue);
描述： 对多个文件或数据签名。先对文件签名后对字符串签名。对文件签名的顺序按照 AddFile 的顺序，对字符串签名的顺序对 AddString 的顺序
参数： Base64EncodeCert[IN] base64 编码的签名证书
SignValue[IN] 签名值
返回值： 0 成功
其他 见错误码
备注： 错误代码：
SOR_NULLPOINTER: 其中一个输入参数为 null。
SOR_CERTENCODE：证书编码错误。
SOR_VERIFYSIGNDATA：验证签名失败

7.2.33 获得多个文件或数据的时间戳请求 SOF_GetTotalTsReq

原型： BSTR SOF_GetTotalTsReq ();
描述： 获得多文件或多字符串的总时间戳请求
参数： 无
返回值： 返回全部数据的总时间戳请求 成功
空值 失败

7.2.34 P7 数据签名 SOF_SignDataPkcs7

原型： BSTR SOF_SignDataPkcs7 (BSTR InData);
描述： 对字符串数据进行数字签名，签名格式为 Pkcs7
参数： InData[IN] 待签名的数据原文
返回值： 返回 pkcs7 格式的签名值的 base64 编码 成功
备注： 错误代码：
SOR_NULLPOINTER: InData 为 null。
SOR_SIGNDATA：签名失败

7.2.35 P7 验证签名 SOF_VerifySignedDataPkcs7

原型： short SOF_VerifySignedDataPkcs7 (BSTR Pkcs7SignData);
描述： 验证数字签名
参数： BSTR Pkcs7SignData[IN] PKCS#7 签名包
返回值： 0 成功
其他 见错误码
备注： 错误代码：
SOR_NULLPOINTER: InData 为 null。
SOR_VERIFYSIGNDATA：签名失败。
SOR_PKCS7ENCODE：PKCS7 编码格式错误。

7.2.36 解析 PKCS#7 签名包信息 SOF_GetP7SignDataInfo

原型： BSTR SOF_GetP7SignDataInfo (BSTR Pkcs7SignData , int type);
描述： 解析 PKCS#7 签名包的信息，可获得原文、签名值、签名证书等信息
参数： BSTR Pkcs7SignData[IN] PKCS#7 签名包
int type[IN] 类型
返回值： 返回 type 对应的值 成功
null 失败

备注： 错误代码：
SOR_NULLPOINTER: InData 为 null。
SOR_VERIFYSIGNDATA：签名失败。
SOR_PKCS7ENCODE：PKCS#7 编码格式错误
Type 值：
1：原文;2：签名者证书;3：签名值

7.2.37 XML 数据签名 SOF_SignDataXML

原型： BSTR SOF_SignDataXML(BSTR InData);
描述： 对 XML 数据进行数字签名，输出符合国际标准的 XML 签名结果
参数： BSTR InData[IN] 签名原文，XML 格式
返回值： BSTR ret 签名结果
空值 失败
备注： 错误代码：
SOR_NULLPOINTER: 输入的某一个参数为 null。
SOR_XMLENCODE：不是合法的 xml 编码数据
SOR_SIGNDATA：签名失败。
XML 签名标准见： http://www.w3.org/TR/xmldsig-core/

7.2.38 验证 XML 数据签名 SOF_VerifySignedDataXML

原型： short SOF_VerifySignedDataXML (BSTR InData);
描述： 验证 xml 签名
参数： BSTR InData XML 签名值
返回值： 0 成功
其他 见错误码
备注： 错误代码：
SOR_NULLPOINTER: 输入的某一个参数为 null。
SOR_XMLENCODE：不是合法的 xml 编码数据
SOR_VERIFYSIGNDATA：验证签名失败
XML 签名标准见 http://www.w3.org/TR/xmldsig-core/

7.2.39 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： BSTR XMLSignedData XML 格式的签名数据
Type 待解析的参数类型
返回值： 各项对应的信息
备注： 错误代码：
SOR_NULLPOINTER: 输入的某一个参数为 null。
SOR_XMLENCODE：不是合法的 xml 编码数据
SOR_VERIFYSIGNDATA：验证签名失败。
SOR_PARAMETERNOTSUPPORT：不支持的参数
Type 值：
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

7.2.40 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： BSTR InData 待创建时间戳请求的原文

返回值： BSTR ret 时间戳请求（base64 编码格式）
空值 失败
备注： 错误代码：
SOR_NULLPOINTER: 输入的某一个参数为 null

7.2.41 创建时间戳响应 SOF_CreateTimeStampResponse

原型： BSTR SOF_CreateTimeStampRequest (BSTR InData);
描述： 创建时间戳响应，即签发时间戳
参数： BSTR InData 时间戳请求
返回值： BSTR ret 时间戳响应（base64 编码格式）
空值 失败
备注： 错误代码：
SOR_NULLPOINTER: 输入的某一个参数为 null。
SOR_SIGNDATA ：签发时间戳失败

7.2.42 验证时间戳 SOF_VerifyTimeStamp

原型： short SOF_VerifyTimeStamp (BSTR content, BSTR tsResponseData);
描述： 验证时间戳
参数： BSTR content 待验证的原文
BSTR tsResponseData 时间戳
返回值： 0 成功
其他 见错误码
备注： 错误代码：
SOR_NULLPOINTER: 输入的某一个参数为 null。
SOR_VERIFYSIGNDATA：验证时间戳失败

7.2.43 解析时间戳 SOF_GetTimeStampInfo

原型： BSTR SOF_GetTimeStampInfo (BSTR tsResponseData,int 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： BSTR tsResponseData 时间戳
int type 类型
返回值： type 对应的值 成功
null 出错
备注： 错误代码：
SOR_NULLPOINTER: 输入的某一个参数为 null。
SOR_VERIFYSIGNDATA：验证时间戳失败。
SOR_PARAMETERNOTSUPPORT：不支持的参数
type =1：返回时间；type ＝2：返回签名值；type ＝3：返回签名证书。

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. 初始化环境 SOF_getInstance
B. 释放环境 SOF_finalize
C. 设置证书信任列表 SOF_setCertTrustList
D. 查询证书信任列表别名 SOF_queryCertTrustListAltNames
E. 查询证书信任列表别名 SOF_queryCertTrustList
F. 删除证书信任列表 SOF_delCertTrustList
G. 设置 Web 应用名称 SOF_setWebAppName

H. 设置签名算法 SOF_setSignMethod
I. 获得当前签名算法 SOF_getSignMethod
J. 设置加密算法 SOF_setEncryptMethod
K. 获得加密算法 SOF_getEncryptMethod
L. 获得服务器证书 SOF_getServerCertificate
M. 读取当前应用指定的服务器证书 SOF_getServerCertificate
N. 产生随机数 SOF_genRandom
O. 产生指定长度随机数 SOF_genRandom
P. 获得证书信息 SOF_getCertInfo
Q. 获得证书扩展信息 SOF_getCertInfoByOid
R. 验证证书有效性 SOF_validateCert
S. 数据签名 SOF_signData
T. 验证签名 SOF_verifySignedData
U. 文件签名 SOF_signFile
V. 验证文件签名 SOF_verifySignedFile
W. 拆分秘密 SOF_secretSegment
X. 秘密恢复 SOF_secertRecovery
Y. 对称算法加密数据 SOF_encryptData
Z. 解密数据 SOF_decryptData
AA. 文件加密 SOF_encryptFile
BB. 文件解密 SOF_decryptFile
CC. 公钥加密 SOF_pubKeyEncrypt
DD. 私钥解密 SOF_priKeyDecrypt
EE. 添加待处理的文件 SOF_addFile
FF. 添加待处理的数据 SOF_addString
GG. 清除文件和字符串 SOF_clear
HH. 对多个文件或数据签名 SOF_signAll
II. 对多个文件或数据验证签名 SOF_verifySignAll
JJ. 获得多个文件和数据的时间戳请求 SOF_getTotalTsReq
KK. P7 数据签名 SOF_signDataPkcs7
LL. P7 验证签名 SOF_verifySignedDataPkcs7
MM. 解析 PKCS#7 签名包信息 SOF_getP7SignDataInfo
NN. XML 数据签名 SOF_signDataXML
OO. 验证 XML 数据签名 SOF_verifySignedDataXML
PP. 解析 XML 签名数据 SOF_getXMLSignatureInfo
QQ. 创建时间戳请求 SOF_createTimeStampRequest
RR. 创建时间戳响应 SOF_createTimeStampResponse
SS. 验证时间戳 SOF_verifyTimeStamp
TT. 解析时间戳 SOF_getTimeStampInfo
UU. 接口异常列表

7.3.2 初始化环境 SOF_getInstance

原型： public static SOF_SecurityEngineDeal getInstance();
描述： 获得一个对象实例，初始化对象
参数： 无
返回值： 对象实例

备注： 抛出：InitException ，初始化环境失败。

7.3.3 释放环境 SOF_finalize

原型： public boolean SOF_finalize ();

描述： 释放环境

参数： 无

返回值： True 成功

False 失败

7.3.4 设置证书信任列表 SOF_setCertTrustList

原型： public boolean SOF_setCertTrustList(java.lang.String ctlAltName, java.lang.String ctlContent, int ctlContentLen);

描述： 设置证书信任列表

参数： ctlAltName 证书信任列表别名

ctlContent 证书信任列表内容(Base64 编码格式)

ctlContentLen 证书信任列表长度

返回值： True 成功

False 失败

备注： 抛出：SOR_ParameterException ： 参数错误

7.3.5 查询证书信任列表别名 SOF_queryCertTrustListAltNames

原型： public java.lang.String SOF_queryCertTrustListAltNames();

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” 成功

null 失败

备注： 抛出：SOR_NoCTLsException ： 没有证书信任列表

7.3.6 查询证书信任列表别名 SOF_queryCertTrustList

原型： public java.lang.String SOF_queryCertTrustList(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，Base64 编码格式的证书信任列表 成功

null 失败

备注： 抛出：SOR_ParameterException ： 参数错误。

SOR_NoCTLsException ： 没有证书信任列表

7.3.7 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

备注： 抛出：SOR_ParameterException ： 参数错误。

SOR_NoCTLsException ： 没有证书信任列表

7.3.8 设置 Web 应用名称 SOF_setWebAppName

原型： public boolean SOF_setWebAppName(java.lang.String webAppName);

描述： 设置 web 应用的名称。该名称要和服务器 xml 文件对应。接口从 xml 文件中读取 webapp 的属性包括使用的证书、可信的根证书、证书验证的策略等

参数： WebAppName 应用的名称

返回值： True 成功

False 失败

备注： 抛出：SOR_AppNotfoundException：没有找到该应用

7.3.9 设置签名算法 SOF_setSignMethod

原型： void SOF_setSignMethod(java.lang.String signMethod);

描述： 设置 Java 组件签名运算使用的签名算法。缺省为“RSA-SHA1”

参数： signMethod[IN] 签名算法

返回值： 无

备注： 抛出：java.lang.NullPointerException：参数为空指针。

SOR_ParameterNotSupportException ：不支持的参数

7.3.10 获得当前签名算法 SOF_getSignMethod

原型： java.lang.String signMethod 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： 当前的签名算法

7.3.11 设置加密算法 SOF_setEncryptMethod

原型： void SOF_setEncryptMethod(java.lang.String encryptMethod);

描述： 设置组件对数据加解密使用的对称算法。

参数： encryptMethod [IN] 对称加密算法

返回值： 无

备注： 抛出：java.lang.NullPointerException：参数为空指针。

SOR_ParameterNotSupportException ：不支持的参数

7.3.12 获得加密算法 SOF_getEncryptMethod

原型： java.lang.String SOF_getEncryptMethod ();

描述： 获得组件使用的对称加解密算法

参数： 无

返回值： 当前控件使用的加密算法

7.3.13 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，否则得到交换证书

参数： 无

返回值： Base64 编码的服务器证书 成功

null 失败

7.3.14 读取当前应用指定的服务器证书 SOF_getServerCertificate (int certUsage)

原型： java.lang.String SOF_getServerCertificate (int certUsage);

描述： 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： Base64 编码的服务器证书

null 失败

备注： 抛出：SOR_ParameterNotSupportException：不支持的参数

7.3.15 产生随机数 SOF_genRandom

原型： java.lang.String SOF_genRandom();

描述： 产生随机数。默认为 10 个字节的随机数，然后 base64 编码输出

参数： 无

返回值： Base64 编码的随机数

7.3.16 产生指定长度随机数 SOF_genRandom(int len)

原型： java.lang.String SOF_genRandom(int len);

描述： 产生指定长度的随机数

参数： int len 待产生的随机数长度（bytes，字节长度）

返回值： 随机数值（Base64 编码后的）

7.3.17 获得证书信息 SOF_getCertInfo

原型： java.lang.String SOF_getCertInfo(java.lang.String base64EncodeCert, int type);

描述： 获取证书信息

参数： java.lang.String base64EncodeCert Base64 编码的 X.509 数字证书

int type 获取证书信息的类型，见 5.1.2 参数表

返回值： java.lang.String ret 证书信息

空值 失败

备注： 抛出：java.lang.NullPointerException：某一个参数为空指针。

SOR_CertEncodeException：证书编码格式错误。

SOR_ParameterNotSupportException ：不支持的参数。

7.3.18 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： java.lang.String base64EncodeCert Base64 编码的证书

java.lang.String oid 私有扩展对象 ID，如“1.2.156.197.1.102”

返回值： java.lang.String ret 证书 OID 对应的值

空值 失败

备注： 抛出：java.lang.NullPointerException：某一个参数为空指针。

SOR_CertEncodeException：证书编码格式错误

7.3.19 验证证书有效性 SOF_validateCert

原型： boolean SOF_validateCert(java.lang.String base64EncodeCert) throws java.lang.NullPointerException

描述： 根据应用的策略根据验证证书有效性

参数： base64EncodeCert[IN] 待验证的 base64 编码证书

返回值： True 验证成功

False 验证失败

备注： 抛出：java.lang.NullPointerException: base64EncodeCert 为 null。

SOR_CertEncodeException：证书编码格式错误。

SOR_CertEncodeException －证书无效，可能不是可信 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.

7.3.20 数据签名 SOF_signData

原型： java.lang.String SOF_SignData(java.lang.String inData);

描述： 对字符串数据进行数字签名，签名格式为 Pkcs#1

参数： inData[IN] 待签名的数据原文

返回值： 返回 pkcs#1 格式的签名值的 base64 编码 成功

备注： 抛出：java.lang.NullPointerException: inData 为 null。
SOR_SignDataException：签名失败

7.3.21 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[IN] base64 编码的签名证书
inData[IN] 待验证的原文
signValue[IN] 签名值

返回值： True 验证成功
False 验证失败

备注： 抛出：java.lang.NullPointerException: 其中一个输入参数为 null。
SOR_CertEncodeException：证书编码错误。
SOR_VerifySignDataException：验证签名失败

7.3.22 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 编码后的 Pkcs#1 格式的签名数据。

参数： inFile[IN] 待签名的文件路径

返回值： base64 编码后的 Pkcs#1 格式的签名数据

备注： 抛出：java.lang.NullPointerException: inFile 为 null。
ReadFileException：读文件异常，可能文件不存在或没有读取权限等。

7.3.23 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[IN] base64 编码的签名证书
inFile[IN] 待验证的原文
signValue[IN] 签名值

返回值： True 验证成功
False 验证失败

备注： 抛出：java.lang.NullPointerException: 其中一个输入参数为 null。
SOR_ReadFileException：读文件异常，可能文件不存在或没有读取权限等。
SOR_CertEncodeException：证书编码错误。
SOR_VerifySignDataException：验证签名失败

7.3.24 拆分秘密 SOF_secretSegment

原型： java.lang.String SOF_secretSegment(java.lang.String secert, int m, int n, int k);

描述： 门限算法，拆分秘密

参数： java.lang.String secert 待拆分的秘密（Base64 编码）
int m 秘密分割份额
int n 秘密恢复最小份额
int k 设置为恢复秘密时的必选片段的个数。（此参数可选，不设置表示任意 n 份片段均可恢复秘密）
返回值： 返回拆分后的密码（Base64 编码）。共 m 份，以&&&相连。例如“111&&&222&&&333”。 成功,若有必选片段，则前 k 份为必选
空值 失败
备注： 抛出：java.lang.NullPointerException: 其中一个输入参数为 null。
SOR_SecretSegmentException：门限分割算法失败

7.3.25 秘密恢复 SOF_secertRecovery

原型： java.lang.String SOF_secertRecovery(java.lang.String seg);
描述： 门限算法，恢复秘密
参数： java.lang.String seg 密钥片段,以“&&&”相连的 n 段密钥片段
返回值： 恢复后的秘密
空值 失败
备注： 抛出：java.lang.NullPointerException: 其中一个输入参数为 null。
SOR_SecertRecoveryException：门限恢复失败

7.3.26 对称算法加密数据 SOF_encryptData

原型： java.lang.String SOF_encryptData(java.lang.String key, java.lang.String inData);
描述： 使用对称算法加密数据
参数： java.lang.String key， 加密密钥
java.lang.String inData 待加密的明文
返回值： 返回加密后的密文（Base64 编码后的） 成功
空值 失败
备注： 抛出：java.lang.NullPointerException: 其中一个输入参数为 null。
SOR_EncryptDataException：数据加密失败

7.3.27 解密数据 SOF_decryptData

原型： java.lang.String SOF_decryptData(java.lang.String key, java.lang.String inData);
描述： 使用对称算法解密数据
参数： java.lang.String key 解密密钥
java.lang.String InData 待解密的密文
返回值： 解密后的明文
空值 出错
备注： 抛出：java.lang.NullPointerException: 其中一个输入参数为 null。
SOR_DecryptDataException： 数据解密失败

7.3.28 文件加密 SOF_encryptFile

原型： boolean SOF_encryptFile(java.lang.String key, java.lang.String inFile, java.lang.String outFile);
描述： 使用对称算法加密文件
参数： java.lang.String key 加密密钥

	java.lang.String inFile	待加密的明文文件路径
	java.lang.String outFile	密文文件保存路径
返回值：	True	成功
	False	失败
备注：	抛出：java.lang.NullPointerException: 其中一个输入参数为 null。 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_EncryptDataException：数据加密失败	

7.3.29 文件解密 SOF_decryptFile

原型：	boolean SOF_decryptFile(java.lang.String key, java.lang.String inFile, java.lang.String outFile);	
描述：	使用对称算法解密文件	
参数：	java.lang.String key	解密密钥
	java.lang.String inFile	待解密的密文文件路径
	java.lang.String outFile	明文文件保存路径
返回值：	True	成功
	False	失败
备注：	抛出：java.lang.NullPointerException: 其中一个输入参数为 null。 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_DecryptDataException：数据解密失败。	

7.3.30 公钥加密 SOF_pubKeyEncrypt

原型：	java.lang.String SOF_pubKeyEncrypt(java.lang.String base64EncodeCert, java.lang.String inData);	
描述：	使用证书对数据加密。（Pkcs#1 格式）	
参数：	java.lang.String base64EncodeCert	证书
	java.lang.String inData	待加密的数据
返回值：	返回加密后的密文(base64 编码后的)	成功
	空值	失败
备注：	抛出：java.lang.NullPointerException: 其中一个输入参数为 null。 SOR_CertEncodeException：证书编码格式错误。 SOR_EncryptDataException：数据加密失败 因为是 pkcs#1 格式，故加密的数据长度要小于证书的位数。比如 1024 位的证书，InData 长度必须小于 128。	

7.3.31 私钥解密 SOF_priKeyDecrypt

原型：	java.lang.String SOF_priKeyDecrypt(java.lang.String InData);	
描述：	私钥解密（Pkcs#1 格式）	
参数：	java.lang.String InData	待解密数据
返回值：	返回解密后的明文	成功
	空	失败
备注：	抛出：java.lang.NullPointerException: 其中一个输入参数为 null。 SOR_DecryptDataException：数据解密失败。	

7.3.32 添加待处理的文件 SOF_addFile

原型： boolean SOF_addFile(java.lang.String filePath);
描述： 针对多个文件或数据处理，添加待处理的文件
参数： java.lang.String filePath 文件路径
返回值： True 成功
False 失败
备注： 抛出：java.lang.NullPointerException: 其中一个输入参数为 null。
SOR_ReadFileException：读文件异常，可能文件不存在或没有读取权限等

7.3.33 添加待处理的数据 SOF_addString

原型： boolean SOF_addString(java.lang.String inData);
描述： 针对多个文件或数据处理，添加待处理的字符串
参数： java.lang.String inData 字符串
返回值： True 成功
False 失败
备注： 抛出：java.lang.NullPointerException: 其中一个输入参数为 null

7.3.34 清除文件和字符串 SOF_clear

原型： void SOF_clear();
描述： 清除添加的文件和字符串
参数： 无
返回值： 无

7.3.35 对多个文件或数据签名 SOF_signAll

原型： java.lang.String SOF_signAll();
描述： 对多个文件或数据签名。先对文件签名后对字符串签名。对文件签名的顺序按照 AddFile 的顺序，对字符串签名的顺序对 AddString 的顺序
参数： 无
返回值： 返回全部数据的签名值 成功
null 失败
备注： 抛出：SignDataException: 签名失败

7.3.36 对多个文件或数据验证签名 SOF_verifySignAll

原型： boolean SOF_verifySignAll(java.lang.String base64EncodeCert, java.lang.String signValue);
描述： 对多个文件或数据签名。先对文件签名后对字符串签名。对文件签名的顺序按照 AddFile 的顺序，对字符串签名的顺序对 AddString 的顺序
参数： base64EncodeCert[IN] base64 编码的签名证书
signValue[IN] 签名值
返回值： True 验证成功
False 验证失败
备注： 抛出：java.lang.NullPointerException: 其中一个输入参数为 null。
SOR_CertEncodeException：证书编码错误。
SOR_VerifySignDataException：验证签名失败

7.3.37 获得多个文件和数据的时间戳请求 SOF_getTotalTsReq

原型： java.lang.String SOF_getTotalTsReq();
描述： 获得多文件或多字符串的总时间戳请求
参数： 无
返回值： 返回全部数据的总时间戳请求 成功
空值 出错

7.3.38 P7 数据签名 SOF_signDataPkcs7

原型： java.lang.String SOF_SignDataPkcs7 (java.lang.String inData);

描述： 对字符串数据进行数字签名，签名格式为 Pkcs7

参数： inData[IN] 待签名的数据原文

返回值： 返回 pkcs7 格式的签名值的 base64 编码 成功

备注： 抛出：java.lang.NullPointerException: inData 为 null。
SOR_SignDataException：签名失败

7.3.39 P7 验证签名 SOF_verifySignedDataPkcs7

原型： boolean SOF_verifySignedDataPkcs7(java.lang.String pkcs7SignData);

描述： 验证数字签名

参数： java.lang.String pkcs7SignData PKCS#7 签名包

返回值： True 成功
False 失败

备注： 抛出：java.lang.NullPointerException: inData 为 null。
SOR_VerifySignDataException：签名失败。
SOR_Pkcs7EncodeException：PKCS7 编码格式错误

7.3.40 解析 PKCS#7 签名包信息 SOF_getP7SignDataInfo

原型： java.lang.String SOF_getP7SignDataInfo (java.lang.String pkcs7SignData , int type);

描述： 解析 PKCS#7 签名包的信息，可获得原文、签名值、签名证书等信息

参数： java.lang.String pkcs7SignData PKCS#7 签名包
int type 类型:1：原文;2：签名者证书;
3：签名值

返回值： 返回 type 对应的值 成功
null 失败

备注： 抛出：java.lang.NullPointerException: inData 为 null。
SOR_VerifySignDataException：签名失败。
SOR_Pkcs7EncodeException：PKCS7 编码格式错误

7.3.41 XML 数据签名 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，输出符合国际标准的 XML 签名结果

参数： java.lang.String InData 签名原文，XML 格式

返回值： java.lang.String ret 签名结果
空值 失败

备注： 抛出：java.lang.NullPointerException: 输入的某一个参数为 null。
SOR_XmlEncodeException：不是合法的 xml 编码数据
SOR_SignDataException：签名失败
XML 签名标准见 http://www.w3.org/TR/xmldsig-core/

7.3.42 验证 XML 数据签名 SOF_verifySignedDataXML

原型： boolean SOF_verifySignedDataXML (java.lang.String inData);

描述： 验证 xml 签名

参数： java.lang.String InData XML 签名值

返回值： True 成功

False 失败

备注： 抛出：java.lang.NullPointerException: 输入的某一个参数为 null。
SOR_XmlEncodeException：不是合法的 XML 编码数据
SOR_VerifySignDataException：验证签名失败
XML 签名标准见 http://www.w3.org/TR/xmldsig-core/

7.3.43 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：

java.lang.String XMLSignedData		XML 格式的签名数据
Type		待解析的参数类型: 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： 各项对应的信息

备注： 抛出：
java.lang.NullPointerException: 输入的某一个参数为 null。
SOR_XmlEncodeException：不是合法的 xml 编码数据
SOR_VerifySignDataException：验证签名失败。
SOR_ParameterNotSupportException：不支持的参数

7.3.44 创建时间戳请求 SOF_createTimeStampRequest

原型： java.lang.String SOF_createTimeStampRequest (java.lang.String inData); throws java.lang.NullPointerException

描述： 创建时间戳请求

参数：

java.lang.String	InData	待创建时间戳请求的原文
返回值： java.lang.String	ret	时间戳请求（base64 编码后的）
空值		失败

备注： 抛出：java.lang.NullPointerException: 输入的某一个参数为 null

7.3.45 创建时间戳响应 SOF_createTimeStampResponse

原型： java.lang.String SOF_createTimeStampRequest (java.lang.String inData);

描述： 创建时间戳响应，即签发时间戳

参数：

java.lang.String	inData	时间戳请求
返回值： java.lang.String	ret	时间戳响应（base64 编码后的）
空值		失败

备注： 抛出：java.lang.NullPointerException: 输入的某一个参数为 null。
SOR_SignDataException ：签发时间戳失败

7.3.46 验证时间戳 SOF_verifyTimeStamp

原型： boolean verifyTimeStamp SOF_(java.lang.String content , java.lang.String tsResponseData);

描述： 验证时间戳

参数：

java.lang.String	content	待验证的原文
java.lang.String tsResponseData		时间戳

返回值：

True	成功
False	失败

备注： 抛出：java.lang.NullPointerException: 输入的某一个参数为 null。
SOR_VerifySignDataException：验证时间戳失败

7.3.47 解析时间戳 SOF_getTimeStampInfo

原型： java.lang.String SOF_getTimeStampInfo(java.lang.String tsResponseData,int 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：

java.lang.String tsResponseData		时间戳
int type		类型:1,返回时间;2，返回签名值;3，返回签名证书

返回值：

type 对应的值	成功
null	失败

备注： 抛出：java.lang.NullPointerException: 输入的某一个参数为 null。
SOR_VerifySignDataException：验证时间戳失败。
SOR_ParameterNotSupportException：不支持的参数

附录 A
（规范性附录）
证书应用综合服务接口错误代码定义

客户端控件和 COM 组件的错误代码表		
宏描述	预定义值	说明
#define SOR_OK	0	成功
#define SOR_UnknownErr	0X0B000001	异常错误
#define SOR_NotSupportYetErr	0X0B000002	不支持的服务
#define SOR_FileErr	0X0B000003	文件操作错误
#define SOR_ProviderTypeErr	0X0B000004	服务提供者参数类型错误
#define SOR_LoadProviderErr	0X0B000005	导入服务提供者接口错误
#define SOR_LoadDevMngApiErr	0X0B000006	导入设备管理接口错误
#define SOR_AlgoTypeErr	0X0B000007	算法类型错误
#define SOR_NameLenErr	0X0B000008	名称长度错误
#define SOR_KeyUsageErr	0X0B000009	密钥用途错误
#define SOR_ModulusLenErr	0X0B000010	模的长度错误
#define SOR_NotInitializeErr	0X0B000011	未初始化
#define SOR_ObjErr	0X0B000012	对象错误
#define SOR_MemoryErr	0X0B000100	内存错误
#define SOR_TimeoutErr	0X0B000101	服务超时
#define SOR_IndataLenErr	0X0B000200	输入数据长度错误
#define SOR_IndataErr	0X0B000201	输入数据错误
#define SOR_GenRandErr	0X0B000300	生成随机数错误
#define SOR_HashObjErr	0X0B000301	HASH 对象错
#define SOR_HashErr	0X0B000302	HASH 运算错误
#define SOR_GenRsaKeyErr	0X0B000303	产生 RSA 密钥错
#define SOR_RsaModulusLenErr	0X0B000304	RSA 密钥模长错误
#define SOR_CspImprtPubKeyErr	0X0B000305	CSP 服务导入公钥错误
#define SOR_RsaEncErr	0X0B000306	RSA 加密错误
#define SOR_RSGDecErr	0X0B000307	RSA 解密错误
#define SOR_HashNotEqualErr	0X0B000308	HASH 值不相等
#define SOR_KeyNotFountErr	0X0B000309	密钥未发现
#define SOR_CertNotFountErr	0X0B000310	证书未发现
#define SOR_NotExportErr	0X0B000311	对象未导出
#define SOR_DecryptPadErr	0X0B000400	解密时做补丁错误
#define SOR_MacLenErr	0X0B000401	MAC 长度错误
#define SOR_KeyInfoTypeErr	0X0B000402	密钥类型错误
#define SOR_NULLPointerErr	0X0B000403	某一个参数为空指针
#define SOR_APPNOTFOUNDErr	0X0B000404	没有找到该应用

#define SOR_CERTENCODEErr	0X0B000405	证书编码格式错误。
#define SOR_CERTINVALIDErr	0X0B000406	证书无效，不是可信 ca 颁发的证书。
#define SOR_CERTHASEXPIREDErr	0X0B000407	证书已过期。
#define SOR_CERTREVOKEDErr	0X0B000408	证书已经被吊销。
#define SOR_SIGNDATAErr	0X0B000409	签名失败。
#define SOR_VERIFYSIGNDATAErr	0X0B000410	验证签名失败
#define SOR_READFILEErr	0X0B000411	读文件异常，可能文件不存在或没有读取权限等。
#define SOR_WRITEFILEErr	0X0B000412	写文件异常，可能文件不存在或没有写权限等
#define SOR_SECRETSEGMENTErr	0X0B000413	门限算法密钥分割失败。
#define SOR_SECERTRECOVERYErr	0X0B000414	门限恢复失败。
#define SOR_ENCRYPTDATAErr	0X0B000415	对数据的对称加密失败
#define SOR_DECRYPTDATAErr	0X0B000416	对称算法的数据解密失败。
#define SOR_PKCS7ENCODEErr	0X0B000417	PKCS7 编码格式错误
#define SOR_XMLENCODEErr	0X0B000418	不是合法的 xml 编码数据
#define SOR_PARAMETERNOTSUPPORTErr	0X0B000419	不支持的参数

JAVA 组件错误代码表：	
异常描述	说明
java.lang.NullPointerException	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_Pkcs7EncodeException	PKCS7 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_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	PKCS7 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

典型的基于 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下：

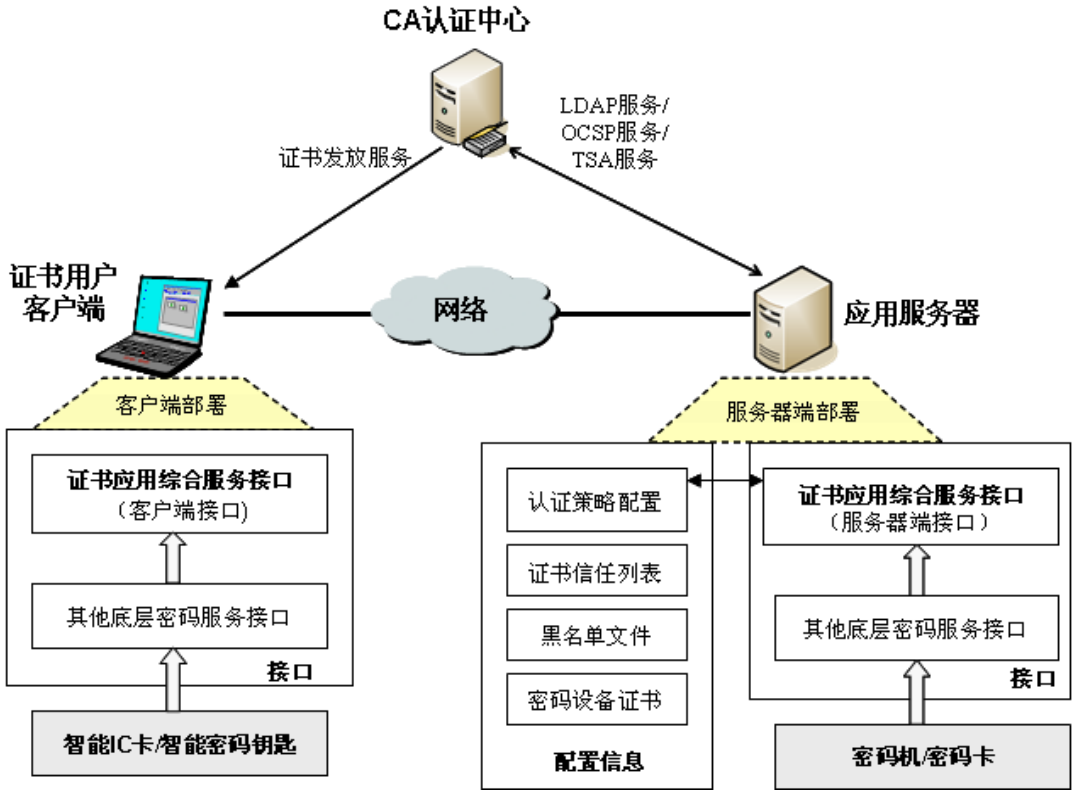


图 B1 B/S 结构应用系统的典型证书应用接口部署示意图

对于 B/S 结构的应用系统，建议在应用服务器端部署一下软硬件：

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，主要包括：密码设备接口和通用密码服务接口等；
b) 配置信息：
   1) 认证策略：配置 CRL 验证或 OCSP 验证的路径、策略等；
   2) 证书信任列表：应可配置多个信任列表，实现多 CA 证书的互信互认；
   3) 黑名单文件：对于 CRL 验证的应用模式，可将 CRL 文件定时下载到应用服务器，登录验证 CRL 时实现本地验证，黑名单文件为可选配置；
   4) 密码设备证书：服务器端配置的密码设备对应的数字证书。
c) 密码设备：
   1) 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。

在证书用户使用终端上部署以下软硬件：

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 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。

b) 密码设备：智能 IC 卡、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

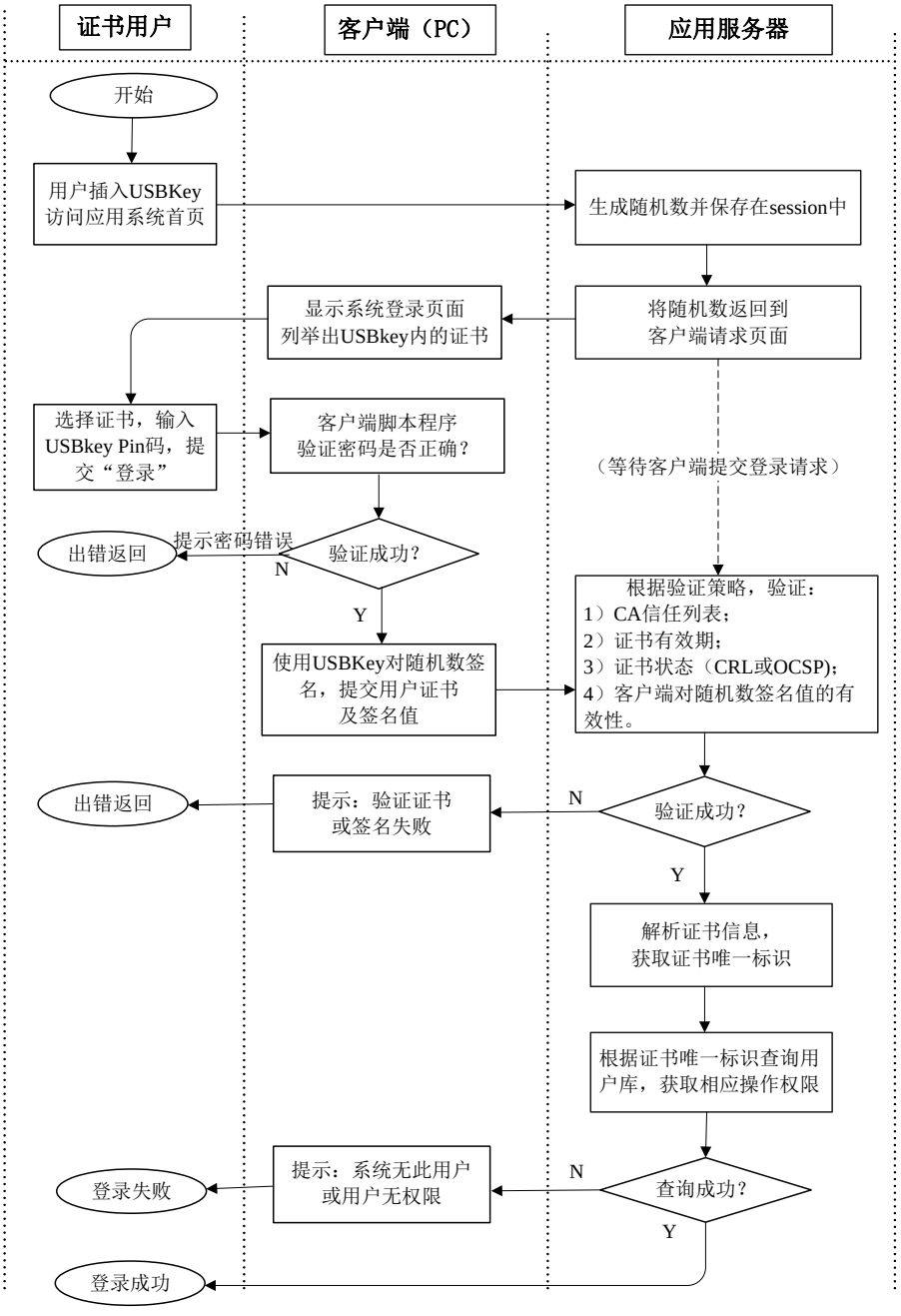
C.1. 证书登录认证流程示例

本标准规定密文数据封装包文件为扩展名为.bct、.ect及.cct的计算机文件。封装包文件的计算机文件名应和封装内容相关，用来查找、检索和利用电子文件封装包。它可以通过在电子文件管理系统中建立文件命名规则来确定。

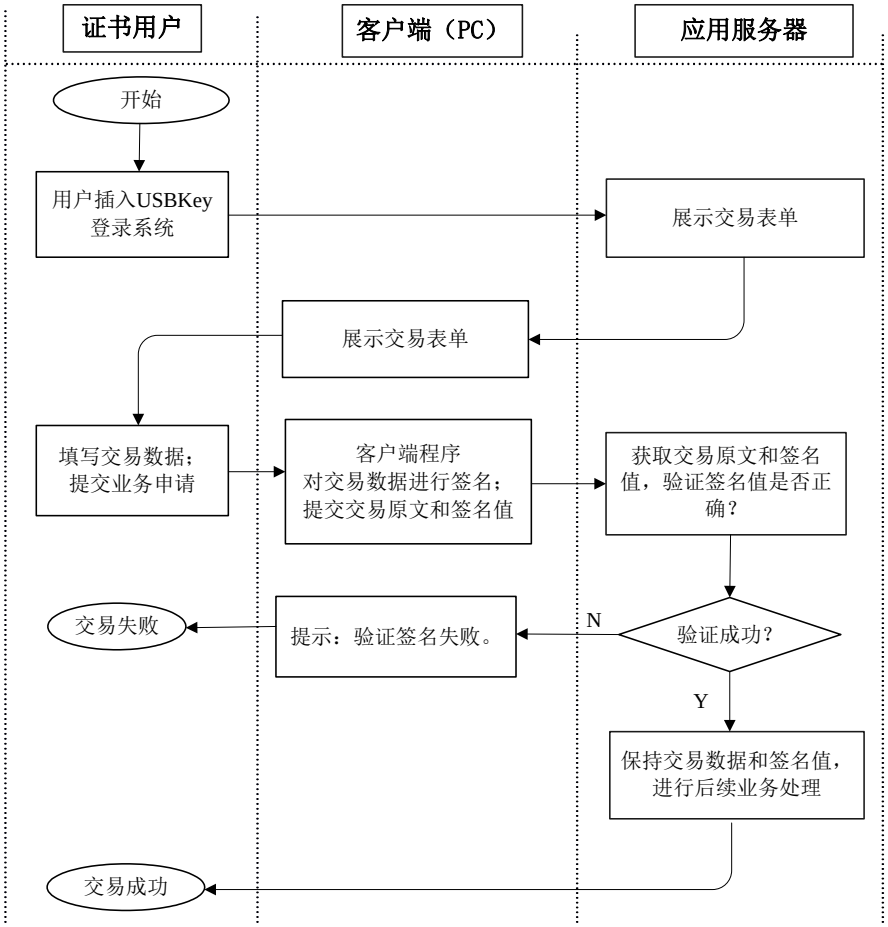
密文数据封装包文件是一个XML文件。

下面以 Java 代码为例，介绍 B/S 架构的应用系统实现数字证书登录认证的流程。

证书登录认证流程图如下图所示：



C.2. 交易数据的签名和验签示例



参考文献

GB/T 17964-2000 信息技术　安全技术　加密算法　第 1 部分：概述

GB/T 17964-2000 信息技术　安全技术　加密算法　第 2 部分：非对称加密

GB/T 17964-2000 信息技术　安全技术　加密算法　第 3 部分：对称加密

GB/T 17903.1-1999 信息技术 安全技术 抗抵赖 第 1 部分:概述

GB/T 17903.2-1999 信息技术 安全技术 抗抵赖 第 2 部分:使用对称技术的机制

GB/T 17903.3-1999 信息技术 安全技术 抗抵赖 第 3 部分:使用非对称技术的机制

GB/T 18238.1-2000 信息技术 安全技术 散列函数 第 1 部分:概述

GB/T 18238.2-2002 信息技术 安全技术 散列函数 第 2 部分:采用 n 位块密码的散列函数

GB/T 18238.3-2002 信息技术 安全技术 散列函数 第 3 部分:专用散列函数

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

GB/T 19771-2005 信息技术 安全技术 公钥基础设施 PKI 组件最小互操作规范

GB 15851 信息技术　安全技术　带消息恢复的数字签名方案

RFC 2560　　X.509 互联网公开密钥基础设施在线证书状态协议—OCSP

RFC 2459　　X.509 互联网公开密钥基础设施证书和 CRL 轮廓

RSA Security: Public-Key Cryptography Standards (PKCS)。

Pkcs#11 Cryptographic Token Interface Standard

IETF Rfc2459, Internet X.509 Public Key Infrastructure Certificate and CRL Profile

IETF Rfc2560, X.509 Internet Public Key Infrastructure Online Certificate Status Protocol

IETF Rfc1777, Lightweight Directory Access Protocol

IETF Rfc2587 Internet X.509 Public Key Infrastructure LDAPv2 Schema

IETF Rfc3647 , Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework

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）