ICS XX.XXX.XX
CCS R XX

DB2101

辽 宁 省 地 方 标 准

DB2101/T XXXX—2023

公共资源交易 数字证书（CA）互联互通技术规范 移动手机载体部分

Public resource transaction digital certificate (CA) interconnection technical specification Mobile phone supporter part

（征求意见稿）

2023-XX-XX 发布 2023-XX-XX 实施

辽宁省发展和改革委员会
联合发布
辽宁省市场监督管理局

前　　言

根据辽宁省公共资源交易协会针对《公共资源交易 数字证书（CA）互联互通技术规范 移动手机载体部分》标准编制大纲要求，经过编制组广泛调查研究，认真总结公共资源交易领域数字证书（CA）互联互通载体技术，参考有关国家标准，并在广泛征求意见的基础上，制订本标准。

本标准主要技术内容是：1.适用范围；2.规范性引用文件；3.术语和定义；4.移动证书二维码技术规范；5.数字证书技术规范；6.CA认证机构接入规范；7.数字证书应用集成规范。

本标准由辽宁省发展和改革委员会负责日常管理，由辽宁省公共资源交易协会负责技术内容的解释。请各单位在本标准执行的过程中，总结经验，积累资料，并将有关意见和建议寄送至辽宁省公共资源交易协会（地址：辽宁省沈阳市皇姑区北陵大街34 - 3 号 3 楼，电话：024-67781777），以便今后修订时参考。

本标准主编单位：辽宁省公共资源交易协会

本标准参编单位：辽宁省发展和改革委员会
大连市公共资源交易中心
丹东市公共资源交易中心
葫芦岛市公共资源交易中心
辽宁省网联数字科技产业有限公司
辽宁思创网联科技有限公司
广联达科技股份有限公司

主要起草人（按姓氏首字母排列）：

薄　红	陈洪岭	关　红	黄伊莹	金　萍	李宏达
李　智	陆　建	邱笑萌	沈　云	佟　伟	吴　楠
王　磊	张汪洋	邹　毅	郑文旭		

目　录

1. 适用范围 1
2. 规范性引用文件 1
3. 术语和定义 2
3.1 招标投标相关的术语及定义 2
3.1.1 电子招标投标 2
3.1.2 招标人 2
3.1.3 招标代理机构 2
3.1.4 投标人 2
3.2 平台及工具软件 2
3.2.1 公共服务平台 2
3.2.2 电子招标投标交易平台 2
3.2.3 工具软件 3
3.2.4 平台及工具软件标识码 3
3.3 主体的术语及定义 3
3.3.1 主体 3
3.3.2 主体代码类型 3
3.3.3 主体名称 3
3.4 CA 数字证书相关的术语定义 3
3.4.1 CA 3
3.4.2 数字证书 3
3.4.3 加密证书 4
3.4.4 签名证书 4
3.4.5 电子印章 4
3.4.6 电子印章系统 4
3.4.7 印章机构 4
3.4.8 用户 4
3.4.9 网络共享 CA 数字证书 4

辽宁省地方标准

3.4.10 CA 共享工具软件 4
3.4.11 数字证书使用入口 4
3.4.12 加密 4
3.4.13 解密 5
3.4.14 数字签名 5
3.4.15 电子签章 5
3.4.16 二维码 5
3.4.17 扫码 5
4. 移动证书二维码技术规范 5
4.1 接口规则 5
4.1.1 接口规则说明 5
4.1.2 数据结构体 5
4.1.3 请求报文结构体 5
4.1.4 响应报文结构体 6
4.2 接口服务 6
4.2.1 调用说明 6
4.2.2 接口说明 7
4.2.3 创建二维码接口 8
4.2.4 查询二维码信息接口 9
4.2.5 保存扫码操作结果接口 10
4.2.6 查询扫码操作结果接口 11
4.2.7 获取电子印章图片 12
4.3 SDK 包 13
4.3.1 验签方法、功能说明 14
4.3.2 方法参数说明 14
4.3.3 方法返回数据 14
4.3.4 加密方法、功能说明 14
4.3.5 方法参数说明 14
4.3.6 方法响应数据 15

4.4 数据类型定义 15
4.4.1 二维码类型 15
4.4.2 扫码状态 15
4.4.3 状态码 15
4.4.4 电子印章类型 15
4.5 业务信息对象 16
5. 数字证书技术规范 17
5.1 数字证书格式 17
5.1.1 数字证书基本结构 17
5.1.2 基本证书域 TBSCertificate 18
5.1.3 版本 Version 18
5.1.4 序列号 Serial Number 18
5.1.5 签名算法 Signature 19
5.1.6 颁发者 Issuer 19
5.1.7 有效期 Validity 20
5.1.8 主体 Subject 20
5.1.9 主体公钥信息 Subject Public Key Info 20
5.1.10 证书扩展项及数据结构 20
5.1.11 标准扩展 20
5.1.12 颁发机构密钥标识符 Authority Key Identifier 20
5.1.13 主体密钥标识符 Subject Key Identifier 21
5.1.14 密钥用法 KeyUsage 21
5.1.15 扩展密钥用法 Extension Key Usage 22
5.1.16 证书策略 CertificatePolicies 23
5.1.17 基本限制 23
5.1.18 证书撤销列表分发点 CRL Distribution Points 23
5.1.19 最新证书撤销列表 Freshest CRL 24
5.1.20 个人身份标识码 Identify Code 24
5.1.21 个人社会保险号 Insurance Number 24

辽宁省地方标准

5.1.22 企业工商注册号 IC Registration Number 25
5.1.23 企业组织机构代码 Organization Code 25
5.1.24 企业税号 Taxation Number 25
5.1.25 互联网证书扩展 25
5.1.26 主体信息访问 Subject Information Access 25
5.1.27 机构信息访问 Authority Info Access 25
5.2 证书撤销列表格式 26
5.2.1 TBSCertList 结构 27
5.2.2 版本号 version 27
5.2.3 签名算法 signature 27
5.2.4 颁发者 issuer 27
5.2.5 生效日期 thisUpdate 27
5.2.6 下次更新日期 nextUpdate 27
5.2.7 被撤销的证书列表 revokedCertificates 27
5.2.8 扩展项 crlExtensions 28
5.2.9 CRL 扩展项及数据结构 28
5.2.10 颁发机构密钥标识符 authorityKeyIdentifier 28
5.2.11 颁发者可替换名称 issuerAltName 28
5.2.12 证书撤销列表号 crlNumber 28
5.2.13 增量证书撤销列表指示 Delta CRL Indicator 28
5.2.14 颁发分布点 Issuing Distribution Point 28
5.2.15 最新证书撤销列表 freshest CRL(也可称为增量 CRL 分布点) 29
5.2.16 证书撤销列表条目扩展 CRL Entry Extensions 29
5.2.17 原因代码 Reason Code 29
5.2.18 撤销时间 Invalidity Date 29
5.2.19 证书颁发者 Certificate Issuer 30
6. CA 认证机构接入规范 30
6.1 系统接入总体要求 30
6.2 CA 系统功能要求 30

6.3 CA 系统接入接口要求 31
6.3.1 注册用户到平台接口 31
6.3.2 更新平台用户的数字证书接口 32
7. 数字证书应用集成规范 33
7.1 应用集成目标 33
7.2 应用集成要求 33
7.3 应用集成内容 33
7.4 统一证书应用接口规范 34
7.4.1 客户端接口 34
7.4.2 ICertificateCollectionX 接口 35
7.4.3 属性 37
7.4.4 ICertificateX 接口 38
7.4.5 ISKFTokenCollectionX 接口 51
7.4.6 ISKFTokenX 接口 53
7.4.7 服务端接口 54
7.4.8 SVS 签名验证服务接口 54
7.4.9 验证证书 54
7.4.10 PCS 密码运算服务接口 56
7.4.11 获取可用的密钥 ID 57
7.4.12 TSA 时间戳服务接口 61
7.4.13 签发时间戳 61
7.4.14 验证时间戳 62
7.5 应用接入规范 63
7.5.1 提交创建应用 63
7.5.2 应用集成 SDK 64

辽宁省地方标准

公共资源交易 数字证书（CA）互联互通技术规范
移动手机载体部分

1. 适用范围

本规范依据《公共资源交易电子认证服务管理办法（试行）》，参照国家密码管理局“公钥密码基础设施应用技术体系”系列技术规范，结合系统业务特点，提出系统数字证书应用集成目标、集成要求、集成内容，定义统一的证书应用接口，并提供证书应用接口的典型部署示例、登录认证流程示例和签名验证流程示例。

本规范用于指导并规范信息系统证书应用集成实施工作，指导电子认证服务机构发布标准统一的证书应用接口，规范信息系统实现基于数字证书的安全登录、数字签名和加密解密等安全功能。

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（包括所有修改单）适用于本文件。

《GB/T 35275-2017 信息安全技术 SM2 密码算法加密签名消息语法规范》

《GB/T 35276-2017 信息安全技术 SM2 密码算法使用规范》

《GB/T 35291-2017 信息安全技术 智能密码钥匙应用接口规范》

《GB/T 36322-2018 信息安全技术 密码设备应用接口规范》

《GB/T 38540-2020 信息安全技术 安全电子签章密码技术规范》

《GB/T 38629-2020 信息安全技术 签名验签服务器技术规范》

《GM/T 0020-2012 证书应用综合服务接口规范》

《GM_T 0025-2014 SSL VPN 网关产品规范》

《GM/T 0026-2014 安全认证网关产品规范》

《GM/T 0033-2014 时间戳接口规范》

《GB/T 7408-2005 数据和交换格式信息交换日期和时间表示法》

《GB 18030-2005 信息技术中文编码字符集》

《GB/T 7027-2002 信息分类和编码的基本原则与方法》

《GMT 0034-2014 基于 SM2 密码算法的证书认证 系统密码及其相关安全技术规范》

《GBT 25056-2018 信息安全技术 证书认证系统密码及其相关安全技术规范》

3. 术语和定义

3.1 招标投标相关的术语及定义

3.1.1 电子招标投标

根据招标投标相关法律法规规章，以数据电文为主要载体，应用信息技术完成招标投标活动的过程。

数据电文是指以电子、光学、磁或者类似手段生成、发送、接收或者储存的信息。本规范中的“电子文件”是指按照特定用途和规定的内容格式要求编辑生成的数据电文。

3.1.2 招标人

提出并组织实施招标项目的法人或其他组织。

3.1.3 招标代理机构

依法设立并提供招标代理服务的社会中介组织。

3.1.4 投标人

参加招投标活动，有意向为招标人提供工程、货物和服务的法人、其他组织或自然人。

3.2 平台及工具软件

3.2.1 公共服务平台

为满足各电子招标投标交易平台之间信息对接交换、资源共享的需要，并为市场主体、行政监督部门和社会公众提供信息交换、整合和发布的信息平台。公共服务平台具有招标投标相关信息对接交换、招标公告和公示发布、主体共享、CA 互联、智能交互、资格信用和业绩验证、行业统计分析、连接评标专家库、提供行政监督通道等服务功能。

公共服务平台是为对接互联电子招标投标交易平台与监督平台、入场交易中心，并为市场主体、监督部门和社会公众交互聚合、公开共享和智能分析招标投标交易基本信息以及提供相关公共技术支持的服务枢纽。

3.2.2 电子招标投标交易平台

通过网络技术以数据电文为主要载体，进行信息交换与传递，并为招标投标管理、交易、监督提供软件系统支撑的电子招标投标系统。

电子招标投标交易平台主要用于在线完成招标投标的全部交易过程，编辑、生成、对接、交

换和发布有关招标交易数据信息。

3.2.3 工具软件

与交易平台兼容对接，用于制作、生成项目投标报价文件的标书制作工具、项目开标工具及专家评标工具等电子招标投标过程中使用的工具软件。

3.2.4 平台及工具软件标识码

按照交易平台、公共服务平台、行政监督平台、发布工具的行业、区域、注册登记顺序，以特定的编码规则赋码，作为该平台或发布工具在电子招标投标网络体系中信息交互共享的唯一识别码。

3.3 主体的术语及定义

3.3.1 主体

招投标市场的参与者，使用电子招标投标系统或者工具软件进行电子招标投标活动的自然人、组织机构、国外企业等。

一个主体可以通过主体代码类型、主体代码、主体名称概念共同描述。

3.3.2 主体代码类型

主体包含招标交易主体、平台运营机构、监督机构等各类法人和其他组织，以及个人。

3.3.3 主体名称

主体的名称，可为组织机构名称、自然人姓名、国外企业名称等。

3.4 CA 数字证书相关的术语定义

3.4.1 CA

依据《电子签名法》和《电子认证服务管理办法》获得《电子认证服务许可证》向公众提供电子认证业务的机构，也称为电子认证服务机构。

3.4.2 数字证书

由 CA 基于 PKI 技术签发、认证和管理的一个可信数字化文件，具有数据电文交换中身份识别、电子签名、加密解密等功能。按主体不同，可分为个人证书、单位证书。

辽宁省地方标准

3.4.3 加密证书

用于证明加密公钥的数字证书，其中包含的公钥用来加密数据。

3.4.4 签名证书

用于证明签名公钥的数字证书，其中包含的公钥用来认证数字签名，而不能加密数据或执行其他的密码功能。

3.4.5 电子印章

一种由制作者签名的包括持有者信息和图形化内容的数据，可用于签署电子文件。

3.4.6 电子印章系统

支持电子印章制作、管理、使用和验证等过程的系统的统称。

3.4.7 印章机构

提供电子签章系统的机构。

3.4.8 用户

参与电子招标交易的市场主体。

3.4.9 网络共享 CA 数字证书

用户通过互联共享网络在线自主选择办理 CA 机构数字证书，并能在全国范围内不同平台上共享漫游使用，免插硬件介质完成全流程电子交易中的身份认证、电子签章签名和加密解密功能。

3.4.10 CA 共享工具软件

支撑实现网络共享 CA 数字证书的工具软件。

3.4.11 数字证书使用入口

交易平台/交易工具为用户提供的使用数字证书的入口。根据允许使用的数字证书不同，可以分为省内统一入口和省外互联入口，其中，省内统一入口是指可以使用省内数字证书的入口，省外互联入口是指可以支持省外数字证书漫游使用的入口。

3.4.12 加密

对数据进行密码变换以产生密文的过程。

3.4.13 解密

将密文转化为明文的过程。

3.4.14 数字签名

附加在数据单元上的一些数据，或是对数据单元所作的密码变换，这种数据或变换允许数据单元的接受者确认数据单元的来源和完整性，并防止数据单元被人伪造。

3.4.15 电子签章

利用图像处理技术、电子签名技术将数字签名操作转化为与纸质文件盖章操作相同的可视效果，并保障电子信息真实性、完整性以及签名人不可否认性的过程。

3.4.16 二维码

用某种特定的几何图形按一定规律在平面（二维方向上）分布的、黑白相间的、记录数据符号信息的图形。

3.4.17 扫码

通过 CA 共享工具软件扫描二维码进行身份识别、验证和授权确认的过程。

4. 移动证书二维码技术规范

4.1 接口规则

4.1.1 接口规则说明

1、接口协议：https。

2、数据格式：JSON 格式。

3、内容类型及编码格式：Content-type: application/json; charset=utf-8

4.1.2 数据结构体

4.1.3 请求报文结构体

```
{
    "head": {
        "platformcode": "",
        "timestamp": "",
        "token": ""
```

```
    },
    "params": {}
}
```

参数名称	参数说明	类型	数据长度	备注
platformcode	平台标识码	string	50	
timestamp	unix 时间戳	string	50	
token	认证令牌	string	200	
params	业务请求对象			具体见接口说明

4.1.4 响应报文结构体

```
{
    "msg": "操作成功",
    "code": 200,
    "data": {
        "token": ""
    }
}
```

参数名称	参数说明	类型	数据长度	备注
code	状态码；	int	1	
msg	描述；如果失败，错误信息放在这里	string	200	
data	业务请求对象			具体见接口说明

4.2 接口服务

4.2.1 调用说明

1、应用系统/移动证书 APP 请求其他接口前需先调用“获取认证令牌”接口，得到 token。

2、应用系统请求“创建二维码接口”，根据响应报文生成二维码，并提示用户扫码。

3、用户使用移动证书 APP 扫码，请求“查询二维码信息接口”获取业务信息，处理完成后，请求“保存扫码操作结果接口”。

4、应用系统异步轮询请求“查询扫码操作结果接口”获取扫码结果。若获取的二维码状态为已确认，系统根据响应报文中业务信息，进行业务逻辑操作。

4.2.2 接口说明

获取认证令牌

功能说明：请求其他接口前，需请求此接口获取认证令牌，此令牌用于标识身份并确保通信安全，认证令牌具有时效性，五分钟内有效。

调用方式：请求方式：Post

接口地址：/lytzs/cadog/applyToken

参数格式：

输入值	{
	"head":{
	"platformcode": "",
	"appkey": "",
	"appsecret": "",
	"timestamp": ""
	}
	"params": {
	}
	}

返回值	{
	"msg": "操作成功",
	"code": 200,
	"data": {
	"token": ""
	}
	}

请求参数：

参数名称	参数说明	类型	数据长度	备注
platformcode	平台标识码	string	50	
appkey	应用公钥	string	50	

appsecret	应用私钥	string	50	
timestamp	unix 时间戳	string	50	

响应参数：

参数名称	参数说明	类型	数据长度	备注
token	请求令牌	string	200	

4.2.3 创建二维码接口

功能说明：应用系统将业务信息上传。并获得二维码标识，可使用此二维码标识生成对应二维码。

调用方式：请求方式：Post

接口地址：/lytzs/cadog/createQR

参数格式：

输入值	{
	"head":{
	"platformcode": "",
	"timestamp": "",
	"token":""
	}
	"params": {
	"qrinfo": {
	……
	}
	}
	}

返回值	{
	"status": {
	"code": ,
	"text": "",
	"timestamp": ""
	},

	"custom": {
	"qrcodeid": ""
	}
	}

请求报文：

参数名称	参数说明	类型	数据长度	备注
qrinfo	业务信息对象	string		

响应报文：

参数名称	参数说明	类型	数据长度	备注
qrcodeid	二维码标识	String	200	用于生成二维码

4.2.4 查询二维码信息接口

功能说明：供移动证书 APP 调用，通过二维码标识获取业务信息，可对业务信息进行后续如签名、解密操作。

调用方式：请求方式：Post

接口地址：/lytzs/cadog/getQRInfo

参数格式：

输入值	{
	"head":{
	"platformcode": "",
	"timestamp": "",
	"token":""
	}
	"params": {
	"qrcodeid": ""
	}
	}

返回值	{ "msg": "操作成功", "code": 200,

	"data": {
	"qrinfo": {
	.....
	}
	}
	}

请求报文：

参数名称	参数说明	类型	数据长度	备注
qrcodeid	二维码标识	String	200	用于生成二维码

响应报文：

参数名称	参数说明	类型	数据长度	备注
qrinfo	业务信息对象	object		

4.2.5 保存扫码操作结果接口

功能说明：移动证书 APP 完成业务信息操作后，将操作结果上传，供应用系统获取。

调用方式：请求方式：Post

接口地址：/lytzs/cadog/saveQRResult

参数格式：

输入值	{
	"head":{
	"platformcode": "",
	"timestamp": "",
	"token":""
	}
	"params": {
	"qrcodeid": "",
	"qrinfo": {
	.....
	}
	}

	}

返回值	{
	"msg": "操作成功",
	"code": 200,
	"data": {}
	}

请求报文

参数名称	参数说明	类型	数据长度	备注
qrcodeid	二维码标识	String	200	用于生成二维码
qrinfo	业务信息对象	string		

响应报文：无

4.2.6 查询扫码操作结果接口

功能说明：应用系统根据二维码标识，获取扫码操作结果，根据响应数据进行业务操作。

调用方式：请求方式：Post

接口地址：/lytzs/cadog/getQRResult

参数格式：

输入值	{
	"head":{
	"platformcode": "",
	"timestamp": "",
	"token":""
	}
	"params": {
	"qrcodeid": "",
	}
	}

返回值	{ "msg": "操作成功", "code": 200,

	"data": {
	"qrcodeid": "",
	"qrstatus": "",
	"qrinfo": {.....}
	}
	}

请求报文：

参数名称	参数说明	类型	数据长度	备注
qrcodeid	二维码标识	String	200	用于生成二维码

响应报文：

参数名称	参数说明	类型	数据长度	备注
qrcodeid	二维码标识	String	200	用于生成二维码
qrstatus	二维码处理结果	String	50	详见附录 A 扫码状态
qrinfo	业务信息对象	object		按二维码类型定义不同，见附录 B 业务信息对象

4.2.7 获取电子印章图片

功能说明：

1. 应用系统签章时可通过此接口获取电子印章图片。
2. 签章公司需实现此接口，供移动证书互联互通服务调用。

调用方式：请求方式：Post

接口地址：/lytzs/cadog/getSeal

接口示例：

输入值	{
	"head":{
	"platformcode": "",
	"timestamp": "",
	"token":"",
	}
	"params": {
	"certsn": ""

	} }

返回值	{
	"msg": "操作成功",
	"code": 200,
	"data": {
	"seallist": {
	"sealtype": "",
	"sealalias": "",
	"image": ""
	}
	}
	}

接口请求报文：

参数名称	参数说明	类型	数据长度	备注
certsn	证书序列号	string	50	

接口响应报文：

参数名称	参数说明	类型	数据长度	备注
sealtype	电子印章类型	string	50	详见附录 A 电子印章类型
sealalias	印章别称	Json	50	最多 6 个字，用于展示在印章列表的时候；一般为个人章、法人章等
validate	印章有效期	yyyyMMdd HHmmss	50	
image	印章图片	string	2	

4.3 SDK 包

以 jar 或 dll 形式提供离线 SDK。应用系统可通过调用 SDK 方法，进行加密、验签操作。

4.3.1 验签方法、功能说明

1、根据传入的原文、签名证书、签名值，进行验签操作。

2、CA 机构需实现此方法，并提供离线 SDK，供移动证书互联互通服务整合发布。

3、方法名称：verifySign(org，signaturecert，signed)

4.3.2 方法参数说明

参数名称	参数说明	类型	数据长度	是否必填	备注
catype	CA 机构类型	string	2	是	规划为代码项
org	原文	string	500	是	
signaturecert	签名证书	string	500	是	可通过二维码的获取证书信息操作获取。
signed	签名数据	String	500	是	可通过二维码的签名操作获取。

4.3.3 方法返回数据

true：验签成功；false：验签失败

4.3.4 加密方法、功能说明

1、系统传入参数，调用 SDK 方法，进行加密操作。

2、CA 机构需实现此方法，并提供离线 SDK，供移动证书互联互通服务整合发布。

3、方法名称 encrypt(enccert,org)

4.3.5 方法参数说明

参数名称	参数说明	类型	数据长度	是否必填	备注
catype	CA 机构类型	string	2	是	规划为代码项
enccert	加密证书	string	500	是	
org	原文	string	500	是	

4.3.6 方法响应数据

返回加密密文

4.4 数据类型定义

4.4.1 二维码类型

代码项值	代码项文本	备注
01	签名	
02	验签	
03	解密	
04	加密	
05	获取证书信息	

4.4.2 扫码状态

代码项值	代码项文本	备注
00	未扫码	
01	已扫码未处理	
02	已处理	
03	无效	

4.4.3 状态码

代码项值	代码项文本	备注
0	失败	
1	成功	

4.4.4 电子印章类型

代码项值	代码项文本	备注
0	电子公章	电子公章
1	电子职务章	与机构关联的个人电子名章
2	电子私章	普通个人电子名章

辽宁省地方标准

代码项值	代码项文本	备注
3	电子部门章	业务章，部门章，在底部有横向一行字标识部门或者业务

4.5 业务信息对象

参数名称	参数说明	类型	数据长度	备注
qrcodetype	二维码类型	string	1	
platformcode	交易平台编号	string	50	
timestamp	unix 时间戳	string	50	

Certobject 节点：

参数名称	参数说明	类型	数据长度	备注
devicenum	硬件介质号	string	200	
dognum	证书序列号	string	200	
signcert	签名证书 base64	string	500	
decryptcert	加密证书 base64	string	500	

decryptobjects 节点：

参数名称	参数说明	类型	数据长度	备注
oid	对象标识	string	50	
org	原文	string	500	
encryed	密文	string	500	
result	解密结果	string	8	true/false
desc	描述	string	500	

signatureobjects 节点：

参数名称	参数说明	类型	数据长度	备注
oid	对象标识	string	50	
org	待签名原文	string	500	
desc	描述	string	500	
signed	签名值	string	200	

辽宁省地方标准

5. 数字证书技术规范

5.1 数字证书格式

5.1.1 数字证书基本结构

数字证书结构采用 ANS.1 DER 编码，基本结构由基本证书域 TBSCertificate、签名算法域 SignatureAlgorithm 和签名值 SignatureValue 三部分组成。

基本结构如下：

```
Certificate ::= SEQUENCE {
    tbsCertificate          TBSCertificate,
    signatureAlgorithm      AlgorithmIdentifier,
    signatureValue          BIT STRING }
TBSCertificate ::= SEQUENCE {
    version         [0]  EXPLICIT Version DEFAULT v1,
    serialNumber         CertificateSerialNumber,
    signature            AlgorithmIdentifier,
    issuer               Name,
    validity             Validity,
    subject              Name,
    subjectPublicKeyInfo SubjectPublicKeyInfo,
    issuerUniqueID  [1]  IMPLICIT UniqueIdentifier OPTIONAL,
                         -- If present, version MUST be v2 or v3
    subjectUniqueID [2]  IMPLICIT UniqueIdentifier OPTIONAL,
                         -- If present, version MUST be v2 or v3
    extensions      [3]  EXPLICIT Extensions OPTIONAL
                         -- If present, version MUST be v3
    }
Version ::= INTEGER { v1(0), v2(1), v3(2) }
CertificateSerialNumber ::= INTEGER
Validity ::= SEQUENCE {
    notBefore       Time,
```

```
        notAfter            Time }
    Time ::= CHOICE {
        utcTime             UTCTime,
        generalTime         GeneralizedTime }
    UniqueIdentifier   ::=   BIT STRING
    SubjectPublicKeyInfo   ::=   SEQUENCE   {
        algorithm                AlgorithmIdentifier,
        subjectPublicKey         BIT STRING   }
    Extensions   ::=   SEQUENCE SIZE (1.MAX) OF Extension
    Extension   ::=   SEQUENCE   {
        extnID          OBJECT IDENTIFIER,
        critical        BOOLEAN DEFAULT FALSE,
        extnValue       OCTET STRING   }
```

上述数据结构中，tbsCertificate 包含了主体名称和颁发者名称，主体的公钥，证书有效期及其他相关信息。signatureAlgorithm 包含了证书签发机构签发该证书时使用的密码算法标识符，一个算法的标识符的 ASN.1 结构如下：

```
AlgorithmIdentifier   ::=   SEQUENCE   {
        algorithm                OBJECT IDENTIFIER,
        parameters               ANY DEFINED BY algorithm OPTIONAL   }
```

该算法标识符必须与 tbsCertificate 中的 signature 的算法标识符相同。当算法为 SM2 时，算法标识定义见 GM / T 0006-2012《密码应用标识规范》。signatureValue 包含了对 tbsCertficate 进行数字签名的结果，签名结果按照 ASN.1 编码成 BIT STRING 类型并保存在证书签名值域内。

5.1.2 基本证书域 TBSCertificate

基本证书域包含了证书结构中主体名称及颁发者 CA 等信息，这些信息主要包括版本、序列号、签名算法、颁发者、有效期、主体、主体公钥信息等。

5.1.3 版本 Version

本项描述了数字证书的版本号。数字证书应使用版本 3（对应的数值是整数“2”）。

5.1.4 序列号 Serial Number

本项是 CA 分配给每张证书的一个正整数，一个 CA 签发的每张证书的序列号必须是唯一的。

序列号可以是长整数，最长可为 20 个 8 位字节的序列号值。证书更新时序列号应改变。

5.1.5 签名算法 Signature

本项包含 CA 签发该证书所使用的密码算法的标识符，算法标识符应与证书中 Signature Algorithm 项的算法标识符相同。签名算法必须符合国家密码主管部门对密码算法的规定，并根据国家密码主管部门批准的最新算法及时调整，以适应国家最新技术标准要求。当算法为 SM2 时，算法标识定义见 GM / T 0006-2012《密码应用标识规范》。

5.1.6 颁发者 Issuer

本项标识了签名和发布数字证书的实体。它必须包含一个非空的甄别名称。该项被定义为 X.501 的 Name 类型。

```
Name ::= CHOICE {
    RDNSequence }
  RDNSequence ::= SEQUENCE OF RelativeDistinguishedName
  RelativeDistinguishedName ::=
    SET OF AttributeTypeAndValue
  AttributeTypeAndValue ::= SEQUENCE {
    type      AttributeType,
    value     AttributeValue }
  AttributeType ::= OBJECT IDENTIFIER
  AttributeValue ::= ANY DEFINED BY AttributeType
  DirectoryString ::= CHOICE {
      teletexString        TeletexString (SIZE (1.MAX)),
      printableString      PrintableString (SIZE (1.MAX)),
      universalString      UniversalString (SIZE (1.MAX)),
      utf8String           UTF8String (SIZE (1. MAX)),
      bmpString            BMPString (SIZE (1.MAX)) }
```

Name 描述了由多个属性组成的层次结构的名称，如国家名、相应的值，例如“C=CN”。其中 AttributeValue 部分的类型由 AttributeType 确定，通常来说是一个 DirectoryString 类型。DirectoryString 类型可被定义为 PrintableString，TeletexString，BMPString，UTF8String 或 UniversalString，UTF8String 编码是首选编码。

5.1.7 有效期 Validity

本项是指一个时间段，在这个时间段内，CA 担保它将维护关于证书状态的信息。该项被表示成一个具有两个时间值的 SEQUENCE 类型数据：证书有效期的起始时间（notBefore）和证书有效期的终止时间（notAfter）。这两个时间都可作为 UTCTime 类型或者 GeneralizedTime 类型进行编码。GeneralizedTime 值必须用格林威治标准时间表示，且必须包含秒（即时间格式为 YYYYMMDDHHMMSSZ）。

5.1.8 主体 Subject

本项描述了与主体公钥项中的公钥相对应的实体。主体名称可以出现在主体项或主体替换名称扩展项中（SubjectAltName）。如果主体是一个 CA，那么主体项必须与其签发的所有证书的颁发者相同，一个 CA 认证的每个主体实体的甄别名称应是唯一的。一个 CA 可以为同一个主体实体以相同的甄别名称签发多个证书。该项不能为空。该项被定义为 X.501 的 Name 类型。

5.1.9 主体公钥信息 Subject Public Key Info

本项用来标识公钥和相应的公钥算法。公钥算法使用算法标识符 Algorithm Identifier 结构来表示。当算法为 SM2 时，算法标识定义见 GM / T 0006-2012《密码应用标识规范》。

5.1.10 证书扩展项及数据结构

本规范定义的证书扩展项提供了把一些附加属性同用户或公钥相关联的方法以及证书结构的管理方法。数字证书允许定义标准扩展项和专用扩展项。每个证书中的扩展可以定义成关键性的和非关键性的。一个扩展含有三部分，它们分别是扩展类型、扩展关键度和扩展项值。扩展关键度（extension criticality）告诉一个证书的使用者是否可以忽略某一扩展类型。证书的应用系统如果不能识别关键的扩展时，必须拒绝接受该证书，如果不能识别非关键的扩展，则可以忽略该扩展项的信息。

遵循本标准的 CA 必须支持密钥标识符、基本限制、密钥用法和证书策略等扩展。遵循本标准的应用必须至少能够识别密钥用法、证书策略、基本限制、颁发机构密钥标识符、主体密钥标识符和扩展的密钥用法。

5.1.11 标准扩展

5.1.12 颁发机构密钥标识符 Authority Key Identifier

本项提供了一种方式，以识别与证书签名私钥相应的公钥。当颁发者由于有多个密钥共存或由于发生变化而具有多个签名密钥时使用该扩展。识别可基于颁发者证书中的主体密钥标识符或

基于颁发者的名称和序列号。

CA 产生的所有证书应包括 AuthorityKeyIdentifier 扩展的 KeyIdentifier 项，以便于链的建立。CA 以“自签”（self-signed ）证书形式发放其公钥时，可以省略认证机构密钥标识符。

本项既可用作证书扩展亦可用作 CRL 扩展。本项标识用来验证在证书或 CRL 上签名的公开密钥。它能辨别同一 CA 使用的不同密钥。除了自签名证书外，所有证书必须包含本项，且必须为非关键扩展。

```
id-ce-authorityKeyIdentifier OBJECT IDENTIFIER ::=   { id-ce 35 }
    AuthorityKeyIdentifier ::= SEQUENCE {
        keyIdentifier              [0] KeyIdentifier              OPTIONAL,
        authorityCertIssuer        [1] GeneralNames               OPTIONAL,
        authorityCertSerialNumber [2] CertificateSerialNumber OPTIONAL   }
    KeyIdentifier ::= OCTET STRING
```

5.1.13 主体密钥标识符 Subject Key Identifier

本项提供一种识别包含有一个特定公钥的证书的方法。此扩展标识了被认证的公开密钥，它能够区分同一主体使用的不同密钥。

对于使用密钥标识符的主体的各个密钥标识符而言，每一个密钥标识符均应是唯一的。CA 签发证书时必须把 CA 证书中本扩展的值赋给终端实体证书 AuthorityKeyIdentifier 扩展中的 KeyIdentifier 项。 CA 证书的主体密钥标识符应从公钥中或者生成唯一值的方法中导出。终端实体证书的主体密钥标识符应从公钥中导出。

所有的 CA 证书必须包括本扩展，此扩展项为非关键项。

```
id-ce-subjectKeyIdentifier OBJECT IDENTIFIER ::=   { id-ce 14 }
    SubjectKeyIdentifier ::= KeyIdentifier
```

5.1.14 密钥用法 KeyUsage

密钥用法扩展定义了证书中密钥的用途（如加密、签名、签发证书等）。密钥用法的限制可用于限定密钥是否能够用于某种操作。密钥用法定义：

```
id-ce-keyUsage OBJECT IDENTIFIER ::= {id-ce 15}
KeyUsage::=BIT STRING{
digitalSignature          (0),
nonRepudiation            (1),
keyEncipherment           (2),
```

dataEncipherment (3),

keyAgreement (4),

keyCertSign (5),

cRLSign (6),

encipherOnly (7),

decipherOnly (8) }

所有的 CA 证书必须包括本扩展，而且必须包含 keyCertSign 这一密钥用途。用户证书则根据证书用途，分“签名”证书和“加密”证书，选择对应的密钥用途进行签发。此扩展可定义为关键的或非关键的，由证书颁发者选择。

5.1.15 扩展密钥用法 Extension Key Usage

本项指明已验证的数字证书公钥可以用于的一种或多种用途，可作为密钥用法扩展项中指明的基本用途的补充或替代。

id-ce-extKeyUsage OBJECT IDENTIFIER ::= {id-ce 37}

ExtKeyUsageSyntax ::= SEQUENCE SIZE (1.MAX) OF KeyPurposeId

KeyPurposeId ::= OBJECT IDENTIFIER

本项可标记为关键或非关键扩展，由证书颁发者决定。本标准定义下列密钥用途：

id-kp OBJECT IDENTIFIER ::= { id-pkix 3 }

id-kp-serverAuth OBJECT IDENTIFIER ::= {id-kp 1}

-- TLS Web server 鉴别

-- Key usage 可设置为 digitalSignature，keyEncipherment 或 keyAgreement

id-kp-clientAuth OBJECT IDENTIFIER ::= {id-kp 2}

-- TLS Web client 鉴别

-- Key usage 可设置为 digitalSignature 和/或 keyAgreement

id-kp-codeSigning OBJECT IDENTIFIER ::= {id-kp 3}

-- 可下载执行代码的签名

-- Key usage 可设置为 digitalSignature

id-kp-emailProtection OBJECT IDENTIFIER ::= {id-kp 4}

-- E-mail 保护

-- Key usage 可设置为 digitalSignature，nonRepudiation, 和/或 (keyEncipherment 或 keyAgreement)

id-kp-timeStamping OBJECT IDENTIFIER ::= { id-kp 8 }

-- 将对象的散列值与同一时间源提供的时间绑定

-- Key usage 可设置为 digitalSignature，nonrepudiation

5.1.16 证书策略 CertificatePolicies

本项包含了一系列策略信息条目，每个条目都有一个 OID 和一个可选的限定条件。

在用户证书中，这些策略信息条目描述了证书发放所依据的策略以及证书的应用目的；在 CA 证书中，这些策略条目指定了包含这个证书的验证路径的策略集合。

数字证书是否包括本扩展为可选的，是否为关键项也是可选的。

5.1.17 基本限制

本项用于识别证书主体是否是一个 CA，通过该 CA 可能存在的认证路径有多长。pathLenConstraint 字段仅在 CA 设置为 TRUE 时才有意义。它给出本证书之后认证路径中最多的 CA 证书数量。0 指明只可以向终端实体签发证书，本项出现时必须大于或等于 0。若未出现，则对人这功能路径的允许长度没有限制。CA 证书中必须包含本扩展且标记为关键扩展。

```
id-ce-basicConstraints OBJECT IDENTIFIER ::=    { id-ce 19 }
BasicConstraints ::= SEQUENCE {
        cA                          BOOLEAN DEFAULT FALSE,
        pathLenConstraint           INTEGER (0.MAX) OPTIONAL }
```

5.1.18 证书撤销列表分发点 CRL Distribution Points

本项标识了如何获得 CRL 信息的方法，本扩展应该被标记为非关键扩展。

```
id-ce-cRLDistributionPoints OBJECT IDENTIFIER ::=    { id-ce 31 }
cRLDistributionPoints ::= {
        CRLDistPointsSyntax }
CRLDistPointsSyntax ::= SEQUENCE SIZE (1.MAX) OF DistributionPoint
DistributionPoint ::= SEQUENCE {
        distributionPoint       [0]     DistributionPointName OPTIONAL,
        reasons                 [1]     ReasonFlags OPTIONAL,
        cRLIssuer               [2]     GeneralNames OPTIONAL }
DistributionPointName ::= CHOICE {
        fullName                [0]     GeneralNames,
```

nameRelativeToCRLIssuer [1] RelativeDistinguishedName }

ReasonFlags ::= BIT STRING {

unused (0),

keyCompromise (1),

cACompromise (2),

affiliationChanged (3),

superseded (4),

cessationOfOperation (5),

certificateHold (6) }

5.1.19 最新证书撤销列表 Freshest CRL

本项标识了如何获取增量 CRL 信息的方法，必须标记为非关键扩展项。

id-ce-freshestCRL OBJECT IDENTIFIER ::= { id-ce 46 }

FreshestCRL ::= CRLDistributionPoints

5.1.20 个人身份标识码 Identify Code

本项用于标识个人身份，标记为非关键扩展。

id-IdentifyCode OBJECT IDENTIFIER ::= { 1.2.156.10260.4.1.1}

IdentifyCode ::=CHOICE{

residenterCardNumber [0] PrintableString OPTIONAL,

--身份证号码

militaryOfficerCardNumber [1] UTF8String OPTIONAL,

--军官证号码

passportNumber [2] PrintableString OPTIONAL

--护照号码

}

5.1.21 个人社会保险号 Insurance Number

本项用于标识个人社会保险号，标记为非关键扩展。

Id-InsuranceNumber OBJECT IDENTIFIER :: ={1.2.156.10260.4.1.2}

InsuranceNumber::=PrintableString

5.1.22 企业工商注册号 IC Registration Number

本项用于标识企业工商注册号，标记为非关键扩展。

Id-RCRegistrationNumber OBJECT IDENTIFIER :: ={1.2.156.10260.4.1.3}

RCRegistrationNumber::=PrintableString

5.1.23 企业组织机构代码 Organization Code

本项用于标识企业组织机构代码，标记为非关键扩展。

Id-OrganizationCode OBJECT IDENTIFIER :: ={1.2.156.10260.4.1.4}

OrganizationCode::=PrintableString

5.1.24 企业税号 Taxation Number

本项用于标识企业税号，标记为非关键扩展。

Id-TaxationNumber OBJECT IDENTIFIER :: ={1.2.156.10260.4.1.5}

TaxationNumber::=PrintableString

5.1.25 互联网证书扩展

5.1.26 主体信息访问 Subject Information Access

本项指明了证书主体如何访问信息和服务，当证书主体是 CA 时，信息和服务中包含证书验证服务和 CA 策略数据，如果主体是终端用户，则描述提供的服务类型以及如何访问。本项可以包含在用户或 CA 证书中吗，且必须为非关键扩展项。

```
id-pe-subjectInfoAccess OBJECT IDENTIFIER ::= { id-pe 11 }
    SubjectInfoAccessSyntax    ::=
            SEQUENCE SIZE (1.MAX) OF AccessDescription
    AccessDescription    ::=    SEQUENCE {
            accessMethod              OBJECT IDENTIFIER,
            accessLocation            GeneralName    }
```

5.1.27 机构信息访问 Authority Info Access

本项指明了包含本项扩展的证书颁发者如何获得 CA 信息和服务。信息和服务包括在线验证服务和 CA 策略数据。本项可以包含在用户证书和 CA 证书中，且必须为非关键扩展。

```
id-pe-authorityInfoAccess OBJECT IDENTIFIER ::= { id-pe 1 }
    AuthorityInfoAccessSyntax    ::=
```

```
            SEQUENCE SIZE (1.MAX) OF AccessDescription
AccessDescription   ::=   SEQUENCE {
            accessMethod            OBJECT IDENTIFIER,
            accessLocation          GeneralName   }
id-ad OBJECT IDENTIFIER ::= { id-pkix 48 }
id-ad-caIssuers OBJECT IDENTIFIER ::= { id-ad 2 }
id-ad-ocsp OBJECT IDENTIFIER ::= { id-ad 1 }
```

5.2 证书撤销列表格式

证书撤销列表（CRL）是由 CA 签发的、加盖时间标记已撤销的数字证书列表，CRL 公开发布，可用于业务系统鉴别数字证书的有效性。其结构采用 ANS.1 DER 编码，其结构如下所示：

```
CertificateList   ::=   SEQUENCE   {
        tbsCertList             TBSCertList,
        signatureAlgorithm      AlgorithmIdentifier,
        signatureValue          BIT STRING   }
TBSCertList   ::=   SEQUENCE   {
        version                 Version OPTIONAL,
                                        -- if present, shall be v2
        signature               AlgorithmIdentifier,
        issuer                  Name,
        thisUpdate              Time,
        nextUpdate              Time OPTIONAL,
        revokedCertificates     SEQUENCE OF SEQUENCE   {
            userCertificate         CertificateSerialNumber,
            revocationDate          Time,
            crlEntryExtensions      Extensions OPTIONAL
                                        -- if present, shall be v2
                                    }   OPTIONAL,
        crlExtensions           [0]  EXPLICIT Extensions OPTIONAL
                                        -- if present, shall be v2
                                    }
```

CRL 基本结构由 tbsCertList、signatureAlgorithm 和 signatureValue 三部分组成。其中，tbsCertList 包含了主体名称和颁发者名称、颁发日期、下次更新日期、撤销的证书信息和 CRL 的扩展信息。signatureAlgorithm 包含了 CA 签发该 CRL 所使用的算法标识符，本项必须包含与 tbsCertList 中 signature 字段相同算法标识符。signatureValue 包含了对 tbsCertList 域进行签名的值，如果签名算法为 SM2，SM2 算法签名数据格式见《SM2 密码算法使用规范》GM/T 0009。

5.2.1 TBSCertList 结构

TBSCertList 也可称为待签发的数字证书列表，主要包含了版本号、签名算法、颁发者、生效日期、下次更新时间、被撤销的证书列表、扩展项。

5.2.2 版本号 version

CRL 版本号采用 v2 ，数值为“1”，表示版本 2 （V2）

5.2.3 签名算法 signature

CA 签发 CRL 所使用的密码算法的标识符，该算法必须与 CertificateList 中的 signatureAlgorithm 相同，当算法为 SM2 时，算法标识定义见 GM / T 0006-2012《密码应用标识规范》。

5.2.4 颁发者 issuer

本项标识了签名并发布 CRL 的实体，颁发者必须包含一个非空的 X.500 甄别名称，颁发者名称字段定义为 X.501 类型 Name。

5.2.5 生效日期 thisUpdate

本项标识了 CRL 的办法日期，可以使用 UTCTime 或 GeneralizedTime 编码。遵循本标准的 CA 在 2049 年以前必须使用 UTCTime 编码本项，在 2050 年以后，必须使用 GeneralizedTime 编码本项。

5.2.6 下次更新日期 nextUpdate

本项标识了 CRL 的下次颁发时间，下次 CRL 可以在此时间之前签发，但不能晚于此时间签发。遵循本标准的 CA 签发 CRL 时必须包含 nextUpdate。

5.2.7 被撤销的证书列表 revokedCertificates

当没有被撤销的证书时，证书撤销列表必须为空。否则被撤销的证书列表是一个集合，其中

每一条 CA 机构撤销的证书都包含证书序列号、撤销时间、撤销理由。被 CA 撤销的证书由证书序列号唯一标识。

5.2.8 扩展项 crlExtensions

本项仅在 CRL version 为 2 时出现，如果出现，本项必须是一个或多个 CRL 扩展的序列。

5.2.9 CRL 扩展项及数据结构

CRL 扩展项采用 ANS.1 编码，每个 CRL 中的扩展项可以被指定为关键或非关键扩展。

5.2.10 颁发机构密钥标识符 authorityKeyIdentifier

颁发机构密钥标识符识提供了一种识别与 CRL 签名私钥相对应的公钥的方式。在颁发者因有多个签名密钥对或签名密钥发生变化而同时具有多个签名私钥时，可通过本项有效识别出颁发者。

5.2.11 颁发者可替换名称 issuerAltName

本项是一个非关键扩展，可包含一个或多个可替换名称以供 CRL 颁发者使用。

5.2.12 证书撤销列表号 crlNumber

本项是一个非关键的 CRL 扩展，表示 CA 机构为其签发的每个 CRL 分配一个单调增长的序列号。用户可通过本项确定一个特定的 CRL 何时取代另一个 CRL，遵守本标准的 CA 必须使用不大于 20 字节的证书撤销列表号。

5.2.13 增量证书撤销列表指示 Delta CRL Indicator

本项是一个关键的 CRL 扩展，表明一个 CRL 是增量 CRL。在一些存储为非 CRL 结构格式的应用中，使用增量 CRL 能够显著的减少处理时间。

5.2.14 颁发分布点 Issuing Distribution Point

本项是一个关键 CRL 扩展，表明一个特定 CRL 的分发点和范围，还表明这个 CRL 是否只包含了用户证书的撤销、CA 证书的撤销或者一系列的原因代码。颁发机构分发点的数据结构如下：

```
id-ce-issuingDistributionPoint OBJECT IDENTIFIER ::= { id-ce 28 }
  issuingDistributionPoint ::= SEQUENCE {
        distributionPoint          [0] DistributionPointName OPTIONAL,
        onlyContainsUserCerts      [1] BOOLEAN DEFAULT FALSE,
        onlyContainsCACerts        [2] BOOLEAN DEFAULT FALSE,
```

```
        onlySomeReasons          [3] ReasonFlags OPTIONAL,
        indirectCRL              [4] BOOLEAN DEFAULT FALSE }
```

5.2.15 最新证书撤销列表 freshest CRL(也可称为增量 CRL 分布点)

本项扩展确定了如何获取完整 CRL 的增量 CRL 信息，本项必须为非关键扩展，并且不能出现在增量 CRL 中。

```
id-ce-freshestCRL OBJECT IDENTIFIER ::=  { id-ce 46 }
    FreshestCRL ::= CRLDistributionPoints
```

5.2.16 证书撤销列表条目扩展 CRL Entry Extensions

本标准中所有的证书撤销列表条目扩展都是非关键扩展。无论本项是否可用，CRL 颁发者都应该包括原因代码和撤销时间。

5.2.17 原因代码 Reason Code

本项为非关键扩展，表明证书撤销的原因。CA 应尽可能在证书撤销列表条目中使用有意义的原因代码。如果的确是没有原因，可以选择不设置原因代码， 而不是使用“未指定(0)”的原因代码。

```
id-ce-cRLReason OBJECT IDENTIFIER ::= { id-ce 21 }
    -- reasonCode ::= { CRLReason }
    CRLReason ::= ENUMERATED {
        unspecified             (0),
        keyCompromise           (1),
        cACompromise            (2),
        affiliationChanged      (3),
        superseded              (4),
        cessationOfOperation    (5),
        certificateHold         (6),
        removeFromCRL           (8) }
```

5.2.18 撤销时间 Invalidity Date

本项是一个非关键扩展，表明知道或者怀疑私钥泄漏或者证书失效时间。这个时间可以比证书撤销列表中的撤销时间更早。本项包含的 GeralizedTime 必须使用格林威治标准时间。

id-ce-invalidityDate OBJECT IDENTIFIER ::= { id-ce 24 }

invalidityDate ::= GeneralizedTime

5.2.19 证书颁发者 Certificate Issuer

证书撤销列表条目标识了在一个间接 CRL 中条目相关联的证书颁发者。即是一个在颁发者分发点扩展中具有间接 CRL 指示者集合的证书撤销列表。如果本项存在，证书颁发者扩展包含一个或多个和 CRL 条目对应的，从证书的颁发者域或颁发者替换名称域得到的名字。

id-ce-certificateIssuer OBJECT IDENTIFIER ::= { id-ce 29 }

certificateIssuer ::= GeneralNames

6. CA 认证机构接入规范

6.1 系统接入总体要求

接入到平台中的电子认证服务机构必须符合以下要求：

1、电子认证服务机构的 CA 系统建设必须遵循《基于 SM2 密码算法的证书认证系统密码及其相关安全技术规范》

2、电子认证服务机构的 CA 系统应遵循《电子政务电子认证体系建设总体规划》（国密局联字〔2007〕2 号）中关于电子认证体系建设的相关要求，符合《电子政务电子认证服务管理办法》(国密局发[2009]7）相关要求。电子认证服务机构的 CA 系统必须具有电子认证服务许可证和电子认证服务使用密码许可证

3、电子认证服务机构的 CA 系统签发的证书为双证书（签名证书和加密证书），证书格式应遵循《平台数字证书格式规范》，使用的证书介质应符合《平台数字证书介质接口及使用规范》，相关应用接入应符合《平台数字证书应用集成规范》。

电子认证服务机构接入平台，必须满足以下条件：

1、提供完整的 CA 业务证书链，CA 业务证书链可以是一个或多个；提供的方式可以由 CA 机构直接以文件方式提交，也可以由 CA 机构提交 CA 业务证书链的 LDAP 或 HTTP 链接地址。

2、提供与 CA 业务证书链相对应的 CRL 发布服务，且按照发布策略及时发布。CA 机构可将 CRL 发布到 LDAP 服务器或 HTTP 服务器上，必须保证 CRL 发布服务可持续访问。

3、提供稳定在线的在线证书状态查询服务，该服务必须按照 RFC 6960 中定义的 OCSP 协议，提供实时在线查询证书的状态。

6.2 CA 系统功能要求

接入到平台的电子认证服务机构的 CA 系统应该具备如下基本功能：

1、证书申请/签发。用户可以以在线或离线两种方式向 CA 机构申请数字证书，在线方式指用户通过互联网登录到 CA 机构的用户注册管理系统申请证书，离线方式指用户到 CA 机构指定的注册机构中申请证书。用户数字证书由 CA 系统的 CA 证书进行签发。CA 机构签发的数字证书格式必须符合《平台数字证书格式规范》。

2、证书更新。CA 系统应提供证书更新功能，更新可分为证书有效期更新和证书可甄别名称更新。CA 系统在对证书更新后，须撤销更新之前的数字证书。

3、证书撤销。CA 系统应提供证书撤销功能，证书撤销后，CA 系统应实时将该证书加入证书撤销列表中，并根据 CRL 发布策略及时将 CRL 发布到 CRL 发布服务中。

4、密钥恢复。CA 系统应提供加密证书的密钥恢复功能，以保证用户可使用恢复的加密证书和加密密钥完成对历史密文信息的解密操作。

5、CRL 签发。CA 系统应提供 CRL 签发功能，并根据 CRL 发布策略及时将 CRL 发布到 CRL 发布服务上。CA 系统签发的 CRL 必须符合《平台数字证书格式规范》。

6.3 CA 系统接入接口要求

电子认证服务机构的 CA 系统根据业务需要可调用平台提供的 HTTP/HTTPS 接口，实现将数字证书注册到平台以及更新平台中数字证书等功能，接口描述如下：

6.3.1 注册用户到平台接口

CA 系统调用本接口可根据业务需要将用户证书及其他信息（企业名称、企业证件号码、联系电话）注册到平台中。详细描述如下：

<table>
<tr><td>名称</td><td colspan="2">createAccount</td></tr>
<tr><td>url</td><td colspan="2">http(s)://xx.com/api/classified/createAccount</td></tr>
<tr><td>请求数据格式</td><td colspan="2">1）请求数据格式为 PKCS#7 SignedData 类型的签名数据格式
2）PKCS#7 中需包含签名证书，签名的原文为 UTF8 编码的 JSON 字符串，JSON 字符串承载请求参数描述如下：
<table>
<tr><td>certificate</td><td>Base64 编码的数字证书字符串，签名证书</td></tr>
<tr><td>name</td><td>企业名称</td></tr>
<tr><td>idno</td><td>企业证件号码</td></tr>
<tr><td>phone</td><td>联系电话</td></tr>
</table>
签名证书在 CA 接入平台时，由平台分配或赋予权限</td></tr>
<tr><td>http 请</td><td colspan="2">POST</td></tr>
</table>

求方式		
返回结果说明	1）XX 返回的数据格式为 PKCS#7 SignedData 类型的签名数据格式 2）PKCS#7 中签名的原文为 UTF8 编码的 JSON 字符串，JSON 字符串承载请求参数描述如下：	
	ret	Ba 返回码，0 表示成功
	msg	UTF 编码的错误提示信息
	签名证书为平台证书	

6.3.2 更新平台用户的数字证书接口

CA 系统调用本接口可根据业务需要更新已经注册到平台的用户数字证书。详细描述如下：

名称	updateCertificate	
url	http(s)://xx.com/api/classified/updateCertificate	
请求数据格式	1）请求数据格式为 PKCS#7 SignedData 类型的签名数据格式 2）PKCS#7 中签名的原文为 UTF8 编码的 JSON 字符串，JSON 字符串承载请求参数描述如下：	
	oldCertificate	Base64 编码的数字证书字符串，旧证书
	newCertificate	Base64 编码的数字证书字符串，新证书
	name	企业名称
	idno	企业证件号码
	phone	联系电话
	签名证书在 CA 接入平台时，由平台分配或赋予权限	
http 请求方式	POST	
返回结果说明	1）XX 返回的数据格式为 PKCS#7 SignedData 类型的签名数据格式 2）PKCS#7 中签名的原文为 UTF8 编码的 JSON 字符串，JSON 字符串承载请求参数描述如下：	
	ret	Ba 返回码，0 表示成功
	msg	UTF 编码的错误提示信息
	签名证书为平台证书	

7. 数字证书应用集成规范

7.1 应用集成目标

1、遵循本规范要求新建或者修改的应用系统，在基于统一认证的多 CA 数字证书的身份认证机制下，可使用不同 CA 机构颁发的数字证书，实现数字证书的互通互认，并确保系统访问控制的高安全性和高可靠性；

2、对信息系统的重要操作环节和重要数据实现基于多 CA 数字证书的数字签名功能，保护数据的完整性，并为后期纠纷处理及责任认定提供合法电子证据；

3、对信息系统的敏感信息实现支持多 CA 的数字证书的数据加密功能，确保敏感信息在传输和存储阶段的安全性及多 CA 支持。

7.2 应用集成要求

在证书应用集成时，应根据应用系统本身的业务特点和业务需求，确定需要使用数字证书认证的用户及范围、确定需要加密、签名的重要操作环节和数据，具体集成要求如下：

1、对于具有操作行为责任认定、证据保存需求的应用系统，应集成基于数字证书的数字签名的功能；

2、对于具有数据加密和解密需求的信息系统，应集成基于数字证书的信息加密、信息解密功能；

3、对于具有可信时间需求的应用系统，应集成时间戳功能，时间戳应遵循《GM/T 0033-2014 时间戳接口规范》；

4、对于具有信息共享需求的多个应用系统，可采用统一的身份认证模式，实现统一的身份认证管理、用户信息共享和单点登录等功能。

7.3 应用集成内容

应用系统数字证书应用集成内容如下：

1、基于多 CA 数字证书的互通互认的统一身份认证

2、数字签名和验证

信息系统中关键业务数据和操作的数字签名，应满足《电子签名法》以及其他相关政策法规规定的书面形式、原件形式及文件保存等要求，至少应包括数据原文、电子签名、可信时间等内容，并可在需要时查询、阅读、下载、验证，具备作为电子证据的真实性、可靠性和可验证性。

数字签名可选和图章结合起来应用，实现电子签章功能，从而实现电子签名的可视化管理，方便用户查看、审阅和验证。

3、数据加密和解密

数据加密时应采用对称算法和非对称算法相结合的方式，既保障加密机制的安全性、密钥分发的方便性，同时又提高了加解密操作的效率。根据业务需求，可使用单个数字证书加密，也可使用多个数字证书对数据共同加密，形成密文数据。

密文数据应安全存储在数据库或磁盘上，待解密时间达到后方可解密。数据解密时，须使用数字证书对应的密码设备或证书介质解密。如果证书介质损坏或丢失，电子认证服务机构应在安全可控的前提下，为用户及时提供密钥恢复服务。

4、时间戳应用

电子认证服务机构提供的时间戳服务应基于可靠的标准时间源，确保时间的准确和可信。

5、密码设备应用

信息系统在使用数字证书安全功能时，应基于密码设备提供的密码服务。密码设备包括客户端用户使用的证书介质和服务器端使用的密码设备。

客户端证书介质是指具有密码许可资质的 USBKey、智能 IC 卡等 PC 终端上的密码设备，以及符合国家密码政策管理规定的 SIM 卡、SD 卡等手机终端上的密码设备。证书介质应用接口应支持所有主流操作系统，并符合《平台数字证书介质接口及使用规范》。

服务器端密码设备是指具有密码许可资质的加密机、加密卡等，应支持所有主流操作系统，并符合《公钥密码基础设施应用技术体系 密码设备应用接口规范》。

通用密码服务接口调用证书介质应用接口或密码设备应用接口，实现对底层密码设备和证书介质的调用，应支持所有主流操作系统，并符合《公钥密码基础设施应用技术体系 通用密码服务接口规范》。

7.4 统一证书应用接口规范

统一证书应用接口是应用系统可直接调用的应用接口，分为客户端接口和服务端接口。

7.4.1 客户端接口

客户端接口是供应用在系统客户端应用程序直接调用的接口，应支持主流操作系统，支持 IE、Chrome、Firefox 等主流浏览器（Chrome 和 Firefox 仅包含支持 NPAPI 的版本）。客户端接口应支持符合《平台数字证书格式标准规范》的数字证书，支持使用符合《平台数字证书介质接口及使用规范》的证书介质。客户端接口可以以 COM 组件、DLL 动态链接库和 ActiveX 控件的产品形态方式提供，支持 B/S 和 C/S 架构的应用。客户端接口如下：

7.4.2 ICertificateCollectionX 接口

选择数字证书 SelectCertificateDialog

函数原型 HRESULT ICertificateCollectionX::SelectCertificateDialog ([out, retval] ICertificateX ** ppResult)

功能描述 根据当前证书集显示证书选择对话框，如果用户选择了证书，则返回该证书的 ICertificateX 对象

参数 ppResult [OUT] ICertificateX 对象

返回值 ICertificateX 对象： 成功

NULL： 失败

加载证书集 Load

函数原型 HRESULT ICertificateCollectionX::Load ([out, retval] LONG * pVal)

功能描述 根据指定的证书过滤标识加载证书集

参数 ppResult [OUT] ICertificateX 对象

返回值 证书集中的证书数量>＝0： 成功

证书集中的证书数量<0： 失败

获得证书对象 GetAt

函数原型 HRESULT ICertificateCollectionX::GetAt ([in] LONG pos, [out, retval] ICertificateX ** ppResult)

功能描述 根据索引号获取证书对象

参数 pos 证书在证书集中国年的索引号，从 0 开始

ppResult [OUT] ICertificateX 对象

返回值 ICertificateX 对象： 成功

NULL： 失败

获取证书对象 Find

函数原型 HRESULT ICertificateCollectionX::Find ([in] BSTR thumb, [out, retval] ICertificateX ** ppResult)

功能描述 根据证书指纹获取证书对象

参数 thumb 证书指纹字符串，可以是 SHA1 的结果，也可以是 MD5 的结果

ppResult [OUT] ICertificateX 对象

返回值 ICertificateX 对象： 成功

NULL： 失败

构造证书对象 CreateCertificateBase64

函数原型 HRESULT ICertificateCollectionX:: CreateCertificateBase64 ([in] BTSR data,

[out, retval] ICertificateX ** ppResult

)

功能描述 根据证书内容（Base64 编码）构造证书对象

参数 data Base64 编码的证书内容

ppResult [OUT] ICertificateX 对象

返回值 ICertificateX 对象： 成功

NULL： 失败

构造证书对象 CreateCertificateDER

函数原型 HRESULT ICertificateCollectionX:: CreateCertificateDER ([in] VARIANT data,

[out, retval] ICertificateX ** ppResult

)

功能描述 根据证书内容（DER 编码数据）构造证书对象

参数 data Base64 编码的证书内容

ppResult [OUT] ICertificateX 对象

返回值 ICertificateX 对象： 成功

NULL： 失败

构造证书对象 CreateCertificateFile

函数原型 HRESULT ICertificateCollectionX:: CreateCertificateFile ([in] BSTR filename,

[out, retval] ICertificateX ** ppResult

)

功能描述 根据证书内容构造证书对象

参数 filename 证书文件路径，如果 filename 为空，则一个“打开文件对话框”会出现，让用户自行选择证书文件

ppResult [OUT] ICertificateX 对象

返回值 ICertificateX 对象： 成功

NULL： 失败

7.4.3 属性

DEBUG 标志：DEBUG

原型 LONG ICertificateCollectionX::DEBUG

功能描述 DEBUG 标志位， 缺省是 0.

如果该属性被设置成 1, 则在执行过程中， 会通过 MessageBox 的形式打印调试信息， 可用于错误追踪

报告错误信息：Quiet

原型 LONG ICertificateCollectionX::Quiet

功能描述 报告错误信息， 缺省为 0

如果该值等于 0, 控件将通过 MessageBox 报告执行过程中的错误， 并返回错误代码， 如果该值不等于 0, 控件将仅仅返回错误结果， 而不通过 MessageBox 报告错误.

证书集中有效证书的数量：Size

原型 BSTR ICertificateCollectionX::Size

功能描述 证书集中有效证书的数量

最近一次执行失败的错误代码：LastError

原型 LONG ICertificateCollectionX::LastError

功能描述 最近一次执行失败的错误代码

最近一次失败的错误信息：ErrorString

原型 BSTR ICertificateCollectionX::ErrorString

功能描述 最近一次执行失败的错误信息

密钥用于过滤标识：CF_KeyUsage

原型 LONG ICertificateCollectionX::CF_KeyUsage

功能描述 密钥用途过滤标识，在 Load()的时候仅仅包含指定密钥用途的证书。0x01 表示证书必须包含私钥；0x10 表示证书必须是有加密用途的证书； 0x20 表示证书必须是有签名用途的证书.可以使用 OR 来组合使用上述标志。注意：CFxxxxx 属性组合使用时，相互之间是"与"的关系。。

证书主题项过滤标识：CF_Subject_Contains

原型　　BSTR ICertificateCollectionX::CF_Subject_Contains

功能描述　　证书主题项过滤标识，在 Load()的时候仅仅包含证书主题项中包含指定字符串的证书。注意：匹配时， 1)大小写敏感， 2)字符集必须一致，CFxxxxx 属性组合使用时， 相互之间是"与"的关系。

签发者主题项过滤标识：CF_Issuer_Contains

原型　　BSTR ICertificateCollectionX:: CF_Issuer_Contains

功能描述　　签发者主题项过滤标识，在 Load()的时候仅仅显示签发者主题项中包含指定字符串的证书。注意：匹配时， 1)大小写敏感， 2)字符集必须一致。CFxxxxx 属性组合使用时， 相互之间是"与"的关系。

密码运算接口标识：CryptoInterface

原型　　LONG ICertificateCollectionX:: CryptoInterface

功能描述　　用于兼容 SKF 接口(国密 SM2)的模式设置参数，可按照指定的参数采用不同的加密接口实现数字签名

1 表示使用 SKF 接口；

2 表示使用 CSP 接口(缺省);

3 表示同时使用上面两个接口

签发者主题项过滤标识： CF_Issuer_Regex

原型　　BSTR ICertificateCollectionX::ErrorString

功能描述　　签发者主题项过滤标识，在 Load()的时候仅仅显示签发者主题项符合正则表达式(PCRE)的证书. 可用于匹配签发自多个 CA 的证书(例如： (CA1|CA2))。

注意：匹配时， 1)大小写不敏感， 2)字符集必须一致。CFxxxxx 属性组合使用时， 相互之间是"与"的关系。

证书主题项过滤标识：CF_Subject_Regex

原型　　BSTR ICertificateCollectionX::CF_Subject_Regex

功能描述　　证书主题项过滤标识，在 Load()的时候仅仅显示证书主题项中符合正则表达式(PCRE)的证书. 可用于匹配多张证书(例如： (证书 1丨证书 2))。

注意：匹配时， 1)大小写不敏感， 2)字符集必须一致。CFxxxxx 属性组合使用时， 相互之间是"与"的关系。

7.4.4 ICertificateX 接口

PKCS1 签名 PKCS1String

函数原型　HRESULT ICertificateX:: PKCS1String　(　[in] VARIANT　data,

[out, retval] BSTR *　pRet

)

功能描述　使用当前证书对数据做 PKCS1 签名，成功返回 BASE64 编码的 PKCS1 签名结果，失败返回 NULL。与 PKCS1Bytes 相比，本方法仅仅是返回值类型不同

参数　data　待签名数据。数据可以是以下类型：

字符串(VT_BSTR 或者 VT_BYREF|VT_BSTR)

字节流(VT_I1 或者 VT_UI1 类型的 Array)

pRet　[OUT] PKCS1 签名结果

返回值　Base64 编码的 PKCS1 签名结果：　成功

NULL：　失败

PKCS1 签名 PKCS1Base64

函数原型　HRESULT ICertificateX:: PKCS1Base64　(　[in] VARIANT　data,

[out, retval] BSTR *　pRet

)

功能描述　使用当前证书对数据做 PKCS1 签名，成功返回 BASE64 编码的 PKCS1 签名结果，失败返回 NULL。

参数　data　待签名数据，必须 Base64 编码。

pRet　[OUT] PKCS1 签名结果

返回值　Base64 编码的 PKCS1 签名结果：　成功

NULL：　失败

PKCS1 签名 PKCS1Bytes

函数原型　HRESULT ICertificateX:: PKCS1Bytes　(　[in] VARIANT　data,

[out, retval] VARIANT*　pRet

)

功能描述　使用当前证书对数据做 PKCS1 签名，成功返回字节流格式的 PKCS1 签名结果，失败返回 NULL。与 PKCS1String 相比，本方法仅仅是返回值类型不同

参数　data　待签名数据。数据可以是以下类型：

字符串(VT_BSTR 或者 VT_BYREF|VT_BSTR)

字节流(VT_I1 或者 VT_UI1 类型的 Array)

pRet　[OUT] PKCS1 签名结果

返回值　　字节流格式的 PKCS1 签名结果：　　成功

NULL：　　失败

PKCS1 签名 PKCS1File

函数原型　HRESULT ICertificateX:: PKCS1File ([in] BSTR fileName,

[out, retval] BSTR* pRet

)

功能描述　使用当前证书对文件做 PKCS1 签名。成功返回 BASE64 编码的 PKCS1 签名结果，失败返回 NULL 参数

fileName　　待签名的文件路径

pRet　　[OUT] PKCS1 签名结果

返回值　　Base64 编码的 PKCS1 签名结果：　　成功

NULL：　　失败

PKCS1 签名 PKCS1Digest

函数原型　HRESULT ICertificateX:: PKCS1Digest ([in] VARIANT data,

[in] BSTR digestAlgo,

[out, retval] BSTR* pRet

)

功能描述　使用当前证书对摘要数据做 PKCS1 签名，成功返回 BASE64 编码的 PKCS1 签名结果，失败返回 NULL

data　　待签名数据。数据可以是以下类型：

字符串(VT_BSTR 或者 VT_BYREF|VT_BSTR) 该符串必须是经过 base64 编码的摘要数据

字节流(VT_I1 或者 VT_UI1 类型的 Array)

digestAlgo　　摘要数据的 Hash 算法，RSA 证书可以选择

"md5"、"sha1"，SM2 证书可选择"ecdsa-sm2-with-sm3"。

pRet　　[OUT] PKCS1 签名结果

返回值　　Base64 编码的 PKCS1 签名结果：　　成功

NULL：　　失败

验证 PKCS1 签名 PKCS1Verify

函数原型　HRESULT ICertificateX:: PKCS1Verify ([in] VARIANT sig,

[in] VARIANT data,

[out, retval] VARIANT* pRet

)

功能描述　使用当前证书验证 PKCS1 签名（原文模式）

参数　sig　签名结果。数据可以是以下类型：

字符串(VT_BSTR 或者 VT_BYREF|VT_BSTR)，必须是 Base64 编码的数据

字节流(VT_I1 或者 VT_UI1 类型的 Array)

data　签名原文数据。数据可以是以下类型：

字符串(VT_BSTR 或者 VT_BYREF|VT_BSTR)，必须是 Base64 编码的数据

字节流(VT_I1 或者 VT_UI1 类型的 Array)

pRet　[OUT] PKCS1 签名结果

返回值　0：　成功

其他：　失败

验证 PKCS1 签名　PKCS1VerifyFile

函数原型　HRESULT ICertificateX:: PKCS1VerifyFile (　[in] VARIANT　sig,

[in] BSTR　fileName,

[out, retval] VARIANT*　pRet

)

功能描述　使用当前证书验证 PKCS1 签名

参数　sig　签名结果。数据可以是以下类型：

字符串(VT_BSTR 或者 VT_BYREF|VT_BSTR)，必须是 Base64 编码的数据

字节流(VT_I1 或者 VT_UI1 类型的 Array)

fileName　待验证的文件路径。

pRet　[OUT] PKCS1 签名结果

返回值　0：　成功

其他：　失败

验证 PKCS1 签名　PKCS1VerifyDigest

函数原型　HRESULT ICertificateX:: PKCS1VerifyDigest　(　[in] VARIANT　sig,

[in] VARIANT　data,

[out, retval] VARIANT*　pRet

)

功能描述　使用当前证书验证 PKCS1 签名（摘要模式）

参数 sig 签名结果。数据可以是以下类型：

字符串(VT_BSTR 或者 VT_BYREF|VT_BSTR)，必须是 Base64 编码的数据

字节流(VT_I1 或者 VT_UI1 类型的 Array)

digest 原文数据摘要。数据可以是以下类型：

字符串(VT_BSTR 或者 VT_BYREF|VT_BSTR)，必须是 Base64 编码的数据

字节流(VT_I1 或者 VT_UI1 类型的 Array)

pRet [OUT] PKCS1 签名结果

返回值 0： 成功

其他： 失败

PKCS7 签名 PKCS7String

函数原型 HRESULT ICertificateX:: PKCS7String ([in] VARIANT data,

[in] LONG fDetached,

[out, retval] VARIANT* pRet

)

功能描述 使用当前证书对数据做 PKCS7 签名。成功返回 Base64 编码格式的 PKCS7 签名结果，失败返回 NULL

参数 data 待签名数据。数据可以是以下类型：

字符串(VT_BSTR 或者 VT_BYREF|VT_BSTR)

字节流(VT_I1 或者 VT_UI1 类型的 Array)

fDetached 0: 创建 Attached 格式的 PKCS7 签名，1: 创建 Detached 格式的 PKCS7 签名

pRet [OUT] PKCS7 签名结果

返回值 Base64 编码的 PKCS7 签名结果： 成功

NULL： 失败

PKCS7 签名 PKCS7Byte

函数原型 HRESULT ICertificateX:: PKCS7String ([in] VARIANT data,

[in] LONG fDetached,

[out, retval] VARIANT* pRet

)

功能描述 使用当前证书对数据做 PKCS7 签名。成功返回字节流编码格式的 PKCS7 签名

结果，失败返回 NULL

参数　　data　　待签名数据。数据可以是以下类型：

字符串(VT_BSTR 或者 VT_BYREF|VT_BSTR)

字节流(VT_I1 或者 VT_UI1 类型的 Array)

fDetached　　0: 创建 Attached 格式的 PKCS7 签名，　1: 创建 Detached 格式的 PKCS7 签名

pRet　　[OUT] PKCS7 签名结果

返回值　　字节流编码格式的 PKCS7 签名结果：　　成功

NULL：　　失败

PKCS7 签名 PKCS7Base64

函数原型　　HRESULT ICertificateX:: PKCS7Base64　(　[in] VARIANT b64data,

[in] LONG　fDetached,

[out, retval] VARIANT*　　pRet

)

功能描述　　使用当前证书对数据做 PKCS7 签名。成功返回 Base64 编码格式的 PKCS7 签名结果，失败返回 NULL

参数　　b64data　　待签名数据。必须是 Base64 编码后的字符串

fDetached　　0: 创建 Attached 格式的 PKCS7 签名，　1: 创建 Detached 格式的 PKCS7 签名

pRet　　[OUT] PKCS7 签名结果

返回值　　Base64 编码的 PKCS7 签名结果：　　成功

NULL：　　失败

证书公钥加密 PublicEncrypt

函数原型　　HRESULT ICertificateX:: PublicEncrypt　(　[in] VARIANT　data,

[out, retval] BSTR *　　pRet

)

功能描述　　使用当前证书公钥对数据进行公钥加密

参数　　data　　待加密数据。数据可以是以下类型：

字符串(VT_BSTR 或者 VT_BYREF|VT_BSTR)

字节流(VT_I1 或者 VT_UI1 类型的 Array)

pRet　　[OUT] 加密结果

返回值　　Base64 编码的加密结果：　　成功

NULL：　　失败

证书私钥解密 PrivateDecrypt

函数原型　HRESULT ICertificateX:: PrivateDecrypt　(　[in] VARIANT　data,

[out, retval] BSTR *　pRet

)

功能描述　使用当前证书私钥对数据进行私钥解密

参数　　data　　待加密数据。数据可以是以下类型：

字符串(VT_BSTR 或者 VT_BYREF|VT_BSTR)

字节流(VT_I1 或者 VT_UI1 类型的 Array)

pRet　　[OUT]解密结果

返回值　　Base64 编码的解密结果：　　成功

NULL：　　失败

数字信封封包 EnvSeal

函数原型　HRESULT ICertificateX:: EnvSeal　(　[in] VARIANT　data,

[out, retval] BSTR *　b64Sealed

)

功能描述　数字信封封包，当前证书为接收方证书，无需包含私钥，封包时只加密，不签名。

参数　　data　　待封包数据。数据可以是以下类型：

字符串(VT_BSTR 或者 VT_BYREF|VT_BSTR)

字节流(VT_I1 或者 VT_UI1 类型的 Array)

b64Sealed　　[OUT]封包结果

返回值　　Base64 编码的封包结果：　　成功

NULL：　　失败

数字信封封包 EnvSealFile

函数原型　HRESULT ICertificateX:: EnvSealFile (　[in] BSTR　srcFile,

[in] BSTR　dstFile,

[out, retval] LONG *　pRet

)

功能描述　对文件进行数字信封封包，当前证书为接收方证书，无需包含私钥，封包时只

加密，不签名。

参数　　srcFile　　待封包的文件路径

　　dstFile　　封包后的保存的文件路径

返回值　　0：　　成功

　　其他：　　失败

数字信封封包 ExtSealFile

函数原型　HRESULT ICertificateX:: ExtSealFile (　[in] BSTR　srcFile,

　[in] BSTR　dstFile,

[out, retval] LONG *　pRet

)

功能描述　对大文件数字信封封包(EnvSeal 需要把整个文件读入内存封包，在封包大文件时，有可能由于内存不足发生错误，如果要对大文件封包，请使用这个函数)，当前证书为接收方证书，无需包含私钥。

参数　　srcFile　　待封包的文件路径

　　dstFile　　封包后的保存的文件路径

返回值　　0：　　成功

　　其他：　　失败

数字信封解包 EnvOpen

函数原型　HRESULT ICertificateX:: EnvOpen　(　[in] VARIANT　data,

[out, retval] BSTR *　b64Opened

)

功能描述　数字信封解包，当前证书为接收方证书，必须包含私钥。

参数　　data　　待解包数据。数据可以是以下类型：

字符串(VT_BSTR 或者 VT_BYREF|VT_BSTR)

字节流(VT_I1 或者 VT_UI1 类型的 Array)

b64Sealed　　[OUT]封包结果

返回值　　Base64 编码的解包结果：　　成功

　　NULL：　　失败

数字信封解包 EnvOpenFile

函数原型　HRESULT ICertificateX:: EnvOpenFile　(　[in] BSTR　srcFile,

[in] BSTR dstFile,

[out, retval] LONG * pRet

)

功能描述 对文件进行数字信封解包，当前证书为接收方证书，且必须包含私钥。

参数 srcFile 待解包的文件路径

dstFile 解包后的保存的文件路径

返回值 0： 成功

其他： 失败

数字信封解包 ExtOpenFile

函数原型 HRESULT ICertificateX:: ExtOpenFile ([in] BSTR srcFile,

[in] BSTR dstFile,

[out, retval] LONG * pRet

)

功能描述 对大文件数字信封解包(EnvOpen 需要把整个文件读入内存解包， 在解包大文件时， 有可能由于内存不足发生错误， 如果要对大文件解包， 请使用这个函数), 当前证书为接收方证书， 且必须包含私钥。

参数 srcFile 待解包的文件路径

dstFile 解包后的保存的文件路径

返回值 0： 成功

其他： 失败

XML 签名 XMLSign

函数原型 HRESULT ICertificateX:: XMLSign ([in] BSTR xml,

[out, retval] BSTR * xmlSigned

)

功能描述 使用当前证书对数据做 XML 签名。成功返回 XML 签名结果，失败返回 NULL

参数 xml 待签名数据。数据必须是 XML 字符串

返回值 XML 签名结果： 成功

NULL： 失败

XML 签名 XMLSignEnveloping

函数原型 HRESULT ICertificateX:: XMLSignEnveloping ([in] BSTR xml,

[out, retval] BSTR * xmlSigned

)

功能描述　使用当前证书对数据做 Enveloping 方式的 XML 签名。成功返回 XML 签名结果，失败返回 NULLL

参数　xml　待签名数据。数据必须是 XML 字符串

返回值　Enveloping 方式的 XML 签名结果：　成功

NULL：　失败

获得证书扩展信息 GetExtensionString

函数原型　HRESULT ICertificateX:: GetExtensionString　(　[in] BSTR　oid,

LONG　rawHex,

[out, retval] BSTR *　pRet

)

功能描述　使用当前证书的扩展信息

参数　oid　证书扩展项 OID

rawHex　1：返回 16 进制数据；0：返回可打印字符串

返回值　证书扩展项信息：　成功

NULL：　失败

属性：

DEBUG 标志：DEBUG

原型　LONG ICertificateX::DEBUG

功能描述　DEBUG 标志位， 缺省是 0.

如果该属性被设置成 1, 则在执行过程中， 会通过 MessageBox 的形式打印调试信息， 可用于错误追踪

报告错误信息：Quiet

原型　LONG ICertificateX::Quiet

功能描述　报告错误信息， 缺省为 0

如果该值等于 0, 控件将通过 MessageBox 报告执行过程中的错误， 并返回错误代码， 如果该值不等于 0, 控件将仅仅返回错误结果， 而不通过 MessageBox 报告错误.

最近一次执行失败的错误代码：LastError

原型　LONG ICertificateX::LastError

功能描述　最近一次执行失败的错误代码

最近一次失败的错误信息：ErrorString

原型　　BSTR ICertificateX::ErrorString

功能描述　　最近一次执行失败的错误信息

证书主题项：Subject

原型　　BSTR ICertificateX::Subject

功能描述　　证书主题项

证书指纹：ThumbprintSHA1

原型　　BSTR ICertificateX::ThumbprintSHA1

功能描述　　证书指纹（SHA1）

证书指纹：ThumbprintMD5

原型　　BSTR ICertificateX::ThumbprintMD5

功能描述　　证书指纹（MD5）

证书友好名：FriendlyName

原型　　BSTR ICertificateX::FriendlyName

功能描述　　证书友好名，如果证书中包含 FriendlyName，则返回 FriendlyName，否则返回 CN(OU)

证书内容：Content

原型　　BSTR ICertificateX::Content

功能描述　　证书内容（Base64 编码）

证书序列号：SerialNumber

原型　　BSTR ICertificateX::SerialNumber

功能描述　　证书序列号（十六进制表示）

证书颁发者：Issuer

原型　　BSTR ICertificateX::Issuer

功能描述　　证书颁发者

证书版本号：　Version

原型　　ULONG ICertificateX::Version

功能描述　　证书版本号

签名算法的 OID 值：SignatureAlgorithm

原型　　BSTR ICertificateX::SignatureAlgorithm

功能描述　　签名算法的 OID 值

生效时间：NotBeforeTimestamp

原型　　ULONG ICertificateX::NotBeforeTimestamp

功能描述　　生效时间（Unix Timestamp 格式）

失效时间：NotAfterTimestamp

原型　　ULONG ICertificateX::NotAfterTimestamp

功能描述　　失效时间（Unix Timestamp 格式）

生效时间：NotBeforeSystemTime

原型　　BSTR ICertificateX:: NotBeforeSystemTime

功能描述　　生效时间(YYYY/MM/DD HH:MM:SS)

失效时间：NotAfterSystemTime

原型　　BSTR ICertificateX:: NotAfterSystemTime

功能描述　　失效时间(YYYY/MM/DD HH:MM:SS)

IssuerUniqueId：IssuerUniqueIdHexString

原型　　BSTR ICertificateX:: IssuerUniqueIdHexString

功能描述　　IssuerUniqueId 的 16 进制字符串

SubjectUniqueId：SubjectUniqueIdHexString

原型　　BSTR ICertificateX:: SubjectUniqueIdHexString

功能描述　　SubjectUniqueId 的 16 进制字符串

SubjectUniqueId：SubjectUniqueIdPrintable

原型　　BSTR ICertificateX:: SubjectUniqueIdPrintable

功能描述　　SubjectUniqueId 的 ASCII 字符串

IssuerUniqueId：IssuerUniqueIdPrintable

原型　　BSTR ICertificateX:: IssuerUniqueIdPrintable

功能描述　　IssuerUniqueId 的 ASCII 字符串

证书序列号：SerialNumberDec

原型　　BSTR ICertificateX:: SerialNumberDec

功能描述　　证书序列号(十进制表示)

证书序列号：SerialNumberHex

原型　　BSTR ICertificateX:: SerialNumberHex

功能描述　　证书序列号(十六进制表示)

证书容器名：ContainerName

原型　　BSTR ICertificateX:: ContainerName

功能描述　　证书容器名

证书提供者：ProviderName

原型　　BSTR ICertificateX:: ProviderName

功能描述　　证书提供者

包含私钥标识：IsPrivateKeyAccessible

原型　　VARIANT_BOOL ICertificateX:: IsPrivateKeyAccessible

功能描述　　证书是否包含可用私钥

密钥长度：Keybits

原型　　LONG ICertificateX:: Keybits

功能描述　　密钥长度

密码运算调用接口的名称：CryptoInterfaceName

原型　　BSTR ICertificateX:: CryptoInterfaceName

功能描述　　密码运算调用接口的名称， 可能是下列名称中的一个： "CSP", "SKF", "PCS"

私钥访问口令：UserPIN

原型　　BSTR ICertificateX:: UserPIN

功能描述　　私钥访问口令。如果该证书来自 CSP 或者 SKF 接口， 可以在这里设置私钥访问的用户口令， 如果该口令非空， 控件将直接使用该口令作为私钥访问口令， 不再弹出用户口令输入对话框

密钥用途：KeyUsage

原型　　ULONG ICertificateX:: KeyUsage

功能描述　　密钥用途：0x01 密钥可用于签名，0x02 密钥可用于加密

算法名称：Algorithm

原型　　BSTR ICertificateX:: Algorithm

功能描述　　算法名称， 可能是下列字符串之一： "RSA", "SM2"

对称加密算法的名称：DefaultCipher

原型　　BSTR ICertificateX:: DefaultCipher

功能描述　　对称加密算法的名称. 例如： aes256, sm4 等等. 目前仅仅在数字信封封包的时候使用. 这个属性可以为空， 如果为空， 则使用的缺省对称加密算法

SKF 类型的证书， 缺省对称加密算法是 aes256

CSP 类型的证书不支持这个属性， 不论这个属性设置成什么， 对称加密算法始终是 3des

PKCS1/Q1 数据摘要算法的名称：DefaultP1Digest

原型　　BSTR ICertificateX:: DefaultP1Digest

功能描述　　PKCS1/Q1 数据摘要算法的名称. 例如： sha1, ecdsa-sm2-with-sm3 等等. 这个属性可以为空， 如果为空， 则使用的缺省数据摘要算法

SKF 类型的证书， 缺省数据摘要算法是 sha1(RSA), ecdsa-sm2-with-sm3(SM2)

CSP 类型的证书， 缺省数据摘要算法是 sha1(RSA)

PKCS7/Q7 数据摘要算法的名称：DefaultP7Digest

原型　　BSTR ICertificateX:: DefaultP7Digest

功能描述　　PKCS7/Q7 数据摘要算法的名称. 例如： sha1, ecdsa-sm2 等等. 这个属性可以为空， 如果为空， 则使用的缺省数据摘要算法

SKF 类型的证书， 缺省数据摘要算法是 sha1(RSA), ecdsa-sm2(SM2)

CSP 类型的证书， 缺省数据摘要算法是 sha1(RSA)

7.4.5 ISKFTokenCollectionX 接口

选择数字证书 SelectTokenDialog

函数原型　HRESULT ISKFTokenCollectionX:: SelectTokenDialog ([out, retval] ISKFTokenX** ppResult)

功能描述　　显示选择 SKF 设备对话框

参数　　ppResult　　[OUT] ISKFTokenX 对象

返回值　　设备数量：　　成功

　　NULL：　　失败

加载 SKF 设备 Load

函数原型　HRESULT ISKFTokenCollectionX::Load ([out, retval] LONG * pRet)

功能描述　　加载 SKF 设备

返回值　　加载的设备数量>＝0：　　成功

　　加载的设备数量<0：　　失败

获得 SKF 设备对象 GetAt

函数原型　HRESULT ISKFTokenCollectionX::GetAt ([in] LONG pos, [out, retval] ISKFTokenX ** ppResult)

功能描述　根据索引号获取 SKFTokenX 对象

参数　　pos　　证书在证书集中国年的索引号，从 0 开始

ppResult　　[OUT] ISKFTokenX 对象

返回值　　ISKFTokenX 对象：　　成功

　　NULL：　　失败

获取 SKF 设备 Find

函数原型　HRESULT ISKFTokenCollectionX::Find　(　[in] BSTR　name,

[out, retval] ISKFTokenX **　ppResult

)

功能描述　根据名称获取 SKF 设备对象

参数　　name　　设备名称

ppResult　　[OUT] ISKFTokenX 对象

返回值　　ISKFTokenX 对象：　　成功

　　NULL：　　失败

属性：

DEBUG 标志：DEBUG

原型　　LONG ISKFTokenCollectionX::DEBUG

功能描述　　DEBUG 标志位， 缺省是 0.

如果该属性被设置成 1, 则在执行过程中， 会通过 MessageBox 的形式打印调试信息， 可用于错误追踪

报告错误信息：Quiet

原型　　LONG ISKFTokenCollectionX::Quiet

功能描述　　报告错误信息， 缺省为 0

如果该值等于 0, 控件将通过 MessageBox 报告执行过程中的错误， 并返回错误代码， 如果该值不等于 0, 控件将仅仅返回错误结果， 而不通过 MessageBox 报告错误.

当前 SKF 设备数量：Size

原型　　BSTR ISKFTokenCollectionX::Size

功能描述　　当前 SKF 设备数量

版权信息：Copyright

原型　　BSTR ISKFTokenCollectionX::Copyright

功能描述　　版权信息

最近一次执行失败的错误代码：LastError

原型　　LONG ISKFTokenCollectionX::LastError

功能描述　　最近一次执行失败的错误代码

最近一次失败的错误信息：ErrorString

原型　　BSTR ISKFTokenCollectionX::ErrorString

功能描述　　最近一次执行失败的错误信息

7.4.6　ISKFTokenX 接口

ToString

函数原型　HRESULT ISKFTokenX:: ToString　(　[out, retval] BSTR *　pVal)

功能描述　　输出 SKF 设备信息

返回值　　SKF 设备信息：　　成功

　　NULL：　　失败

属性：

设备名称：Name

原型　　BSTR ISKFTokenX::Name

功能描述　　SKF 设备名称

设备制造商：Manufacturer

原型　　BSTR ISKFTokenX:: Manufacturer

功能描述　　SKF 设备制造商

设备序列号：SerialNumber

原型　　BSTR ISKFTokenX:: SerialNumber

功能描述　　SKF 设备序列号

设备硬件版本：HWVersion

原型　　BSTR ISKFTokenX:: HWVersion

功能描述　　SKF 设备硬件版本

设备软件版本：FirmwareVersion

原型　　BSTR ISKFTokenX:: FirmwareVersion

功能描述　　SKF 设备软件版本

密码运算接口标识：CryptoInterface

原型　　LONG ISKFTokenX::CryptoInterface

功能描述　　密码运算接口

SKF 驱动名称和路径：ProviderModuleName

辽宁省地方标准

原型　　BSTR ISKFTokenX:: ProviderModuleName

功能描述　　SKF 设备驱动名称和路径

7.4.7　服务端接口

服务器端接口是供应用系统服务器端程序直接调用的高级接口，应支持所有主流操作系统，支持 B/S 和 C/S 等系统架构，支持符合《平台数字证书格式规范》的数字证书，可通过添加证书信任列表的方式实现不同电子认证服务机构证书之间的交叉认证和互信互认。服务端接口使用 HTTP/HTTPS 作为通讯接口协议，所有的请求都使用 POST 方法，请求参数保存在 POST 表单域内。

7.4.8　SVS 签名验证服务接口

签名验证服务(Signature Verification Service，以下简称 SVS)提供了对数字证书及数字证书签名信息进行可信的验证服务。主要包括以下功能：

1、证书验证，包括验证证书及颁发者的有效性，证书状态的验证（CRL/OCSP）。

2、业务数据签名验证。验证数据电文的真实性和有效性。

7.4.9　验证证书

URL	http(s)://[主机地址][：端口号]/vc.svr
参数	cert　所需要验证的证书(B64 编码)。 greenpass　证书验证绿色通道。其包括以下取值范围。 ➢ 0：完整验证 ➢ 0x01：允许过期证书通过 ➢ 0x02：允许未生效证书通过 ➢ 0x04：不验证 CRL 例如：如果既允许过期证书通过，又允许未生效证书通过，则 greenpass 值为：(0x01|0x02)=0x03 greenpass 为可选参数，如果在 request 里面没有包含这些数据项，SVS 将使用缺省设置。
返回	以 HTTP 响应码的方式返回验证的结果，如：200 代表验证成功，4xx：代表验证错误，5xx：服务端异常错误。

验证数据签名(PKCS1)

URL	http(s)://[主机地址][：端口号]/vp1.svr
参数	cert　　签名证书(B64 编码) signature　　签名数据(B64 编码) algo　　签名算法。格式：[签名算法]-[摘要算法] 算法标识说明： ➢ RSA-MD5 ➢ RSA-SHA1 data　　原文数据((B64 编码) datt　　原文数据类型，包括两种 ➢ 0 原文数据类型 ➢ 1 摘要数据类型 greenpass　　证书验证绿色通道。其包括以下取值范围。 ➢ 0：完整验证 ➢ 0x01：允许过期证书通过 ➢ 0x02：允许未生效证书通过 ➢ 0x04：不验证 CRL 例如：如果既允许过期证书通过，又允许未生效证书通过，则 greenpass 值为：(0x01|0x02)=0x03 greenpass 为可选参数，如果在 request 里面没有包含这些数据项，SVS 将使用缺省设置。
返回	以 HTTP 响应码的方式返回验证的结果，如：200 代表验证成功，4xx：代表验证错误 ，5xx：服务端异常错误。

验证数据签名(PKCS7)

URL	http(s)://[主机地址][：端口号]/vp7.svr
参数	p7data　　PKCS7 格式数据(B64 编码)。注意：该 pkcs7 数据中应包含签名证书(链)、签名数据以及原文数据 。 p7odat　　原文数据(B64 编码)。注意：该参数只在 Detached 模式数字签名验证时需要提交，非必要参数。 greenpass　　证书验证绿色通道。其包括以下取值范围。

	➢ 0：完整验证 ➢ 0x01：允许过期证书通过 ➢ 0x02：允许未生效证书通过 ➢ 0x04：不验证 CRL 例如：如果既允许过期证书通过，又允许未生效证书通过，则 greenpass 值为：(0x01|0x02)=0x03 greenpass 为可选参数，如果在 request 里面没有包含这些数据项，SVS 将使用缺省设置。
返回	以 HTTP 响应码的方式返回验证的结果，如：200 代表验证成功，4xx：代表验证错误 ，5xx：服务端异常错误。

验证 XML 签名

URL	http(s)://[主机地址][：端口号]/vx.svr	
参数	data	待验证的 XML 数据(B64 编码)。
	greenpass	证书验证绿色通道。其包括以下取值范围。 ➢ 0：完整验证 ➢ 0x01：允许过期证书通过 ➢ 0x02：允许未生效证书通过 ➢ 0x04：不验证 CRL 例如：如果既允许过期证书通过，又允许未生效证书通过，则 greenpass 值为：(0x01|0x02)=0x03 greenpass 为可选参数，如果在 request 里面没有包含这些数据项，SVS 将使用缺省设置。
返回	以 HTTP 响应码的方式返回验证的结果，如：200 代表验证成功，4xx：代表验证错误 ，5xx：服务端异常错误。	

7.4.10 PCS 密码运算服务接口

PKI 密码运算服务(PKI Crypto Service)可用户服务端签名、加密、解密、数字信封等业务，主要功能如下：

1. 私钥的管理.
2. 私钥的加解密操作.

3. 生成 PKCS#1 数字签名.
4. 生成 PKCS#7 数字签名
5. XML 签名
6. 数字信封封包
7. 数字信封解包

7.4.11 获取可用的密钥 ID

URL	http(s)://[主机地址][：端口号]/sl.svr
参数	无
返回	以 HTTP 响应码的方式返回执行的结果， 如：200 代表执行成功， 404 表示密钥没有找到， 5xx：服务端异常错误. Http Response Content 中保存返回可用的密钥 ID 列表。如果有多个可用的密钥 ID，这些 ID 之间通过换行符分割 Http header 中包含 Content-Length 项指明数据长度

获取密钥 ID 所对应的数字证书

URL	http(s)://[主机地址][：端口号]/sg.svr
参数	id 密钥 ID
返回	以 HTTP 响应码的方式返回执行的结果， 如：200 代表执行成功， 404 表示密钥没有找到， 5xx：服务端异常错误. Http Response Content 中保存返回 BASE64 编码的数字证书 Http header 中包含 Content-Length 项指明数据长度

私钥加密

URL	http(s)://[主机地址][：端口号]/spe.svr
参数	id 密钥 ID passwd 口令(私钥保护的口令) data BASE64 编码的待加密数据（注意，RSA 私钥加密的明文最大长度不能超过私钥长度的 2 倍，SM2 不支持私钥加密）
返回	以 HTTP 响应码的方式返回执行的结果， 如：200 代表执行成功. 400：代表参数错误，403 代表无权访问或者密钥， 404 表示密钥没有找到， 5xx：服

	务端异常错误. Http Response Content 中保存返回 BASE64 编码的加密结果. Http header 中包含 Content-Length 项指明数据长度

私钥解密

URL	http(s)://[主机地址][：端口号]/spd.svr
参数	id 密钥 ID passwd 口令(私钥保护的口令) data BASE64 编码的待解密数据（注意，RSA 私钥解密的明文最大长度不能超过私钥长度的 2 倍，SM2 私钥无此限制）
返回	以 HTTP 响应码的方式返回执行的结果， 如：200 代表执行成功. 400：代表参数错误，403 代表无权访问或者密钥， 404 表示密钥没有找到， 5xx：服务端异常错误. Http Response Content 中保存返回 BASE64 编码的解密结果. Http header 中包含 Content-Length 项指明数据长度

公钥加密

URL	http(s)://[主机地址][：端口号]/ppe.svr
参数	id 密钥 ID data BASE64 编码的待加密数据（注意，RSA 公钥加密的明文最大长度不能超过私钥长度的 2 倍，SM2 公钥无此限制）
返回	以 HTTP 响应码的方式返回执行的结果， 如：200 代表执行成功. 400：代表参数错误，403 代表无权访问或者密钥， 404 表示密钥没有找到， 5xx：服务端异常错误. Http Response Content 中保存返回 BASE64 编码的加密结果. Http header 中包含 Content-Length 项指明数据长度

公钥解密

URL	http(s)://[主机地址][：端口号]/ppd.svr
参数	id 密钥 ID

	data　　BASE64 编码的待解密数据（注意，RSA 公钥解密的明文最大长度不能超过私钥长度的 2 倍，SM2 不支持公钥解密）
返回	以 HTTP 响应码的方式返回执行的结果， 如：200 代表执行成功. 400：代表参数错误，403 代表无权访问或者密钥， 404 表示密钥没有找到， 5xx：服务端异常错误. Http Response Content 中保存返回 BASE64 编码的解密结果. Http header 中包含 Content-Length 项指明数据长度

创建 PKCS#1 签名

URL	http(s)://[主机地址][：端口号]/smp1.svr
参数	id　　密钥 ID passwd　　口令(私钥保护的口令) data　　BASE64 编码的待签名数据，数据长度不限 algo　　签名时的数据摘要算法。可以是（RSA-MD5，RSA-SHA1）。algo 为可选参数，如果在 request 里面没有包含这些数据项，PCS 将使用缺省设置 RSA-SHA1 datt　　datt 为可选参数，如果在 request 里面没有包含这些数据项，PCS 将使用缺省设置 0 待签名数据类型： 0: 原文 1: 摘要 (二进制格式) 2: 摘要 (HEX string 格式)
返回	以 HTTP 响应码的方式返回执行的结果， 如：200 代表执行成功. 400：代表参数错误，403 代表无权访问或者密钥， 404 表示密钥没有找到， 5xx：服务端异常错误. Http Response Content 中保存返回 BASE64 编码的签名结果. Http header 中包含 Content-Length 项指明数据长度

创建 PKCS#7 签名

URL	http(s)://[主机地址][：端口号]/smp7.svr

参数	id 密钥 ID passwd 口令(私钥保护的口令) data BASE64 编码的待签名数据，数据长度不限 fullchain 是否嵌入整个证书链，可以为(0,1).fullchain 为可选参数，如果在 request 里面没有包含这些数据项，PCS 将使用缺省设置为 0。 attach 是否启用 Detached 签名模式，0=包含原文的 P7 签名（缺省设置），1=不包含原文的 P7 签名。 注意：Detached 模式下产生的签名数据将不包含原文数据。
返回	以 HTTP 响应码的方式返回执行的结果， 如：200 代表执行成功. 400：代表参数错误，403 代表无权访问或者密钥， 404 表示密钥没有找到， 5xx：服务端异常错误. Http Response Content 中保存返回 BASE64 编码的签名结果. Http header 中包含 Content-Length 项指明数据长度

创建 XML 签名

URL	http(s)://[主机地址][：端口号]/sxs.svr
参数	id 密钥 ID passwd 口令(私钥保护的口令) data BASE64 编码的待签名数据，数据长度不限 sigmode 签名模式。可以是（0,1），0 表示 enveloped, 1 表示 enveloping。Sigmode 为可选参数，如果在 request 里面没有包含这些数据项，PCS 将使用缺省设置 0
返回	以 HTTP 响应码的方式返回执行的结果， 如：200 代表执行成功. 400：代表参数错误，403 代表无权访问或者密钥， 404 表示密钥没有找到， 5xx：服务端异常错误. Http Response Content 中保存返回 BASE64 编码的签名结果. Http header 中包含 Content-Length 项指明数据长度

数字信封封包

URL	http(s)://[主机地址][：端口号]/envseal.svr

参数	id　　密钥 ID passwd　　口令(私钥保护的口令) data　　BASE64 编码的待封包数据，数据长度不限 peer　　BASE64 编码的接收方证书
返回	以 HTTP 响应码的方式返回执行的结果， 如：200 代表执行成功. 400：代表参数错误，403 代表无权访问或者密钥， 404 表示密钥没有找到， 5xx：服务端异常错误. Http Response Content 中保存返回 BASE64 编码的数字信封. Http header 中包含 Content-Length 项指明数据长度

数字信封解包

URL	http(s)://[主机地址][：端口号]/envopen.svr
参数	id　　密钥 ID passwd　　口令(私钥保护的口令) data　　BASE64 编码的待解包数据，数据长度不限
返回	以 HTTP 响应码的方式返回执行的结果， 如：200 代表执行成功. 400：代表参数错误，403 代表无权访问或者密钥， 404 表示密钥没有找到， 5xx：服务端异常错误. Http Response Content 中保存返回 BASE64 编码的数据原文. Http header 中包含 Content-Length 项指明数据长度

7.4.12　TSA 时间戳服务接口

时间戳服务(Time Stamp Authority Service)提供了对数据时间的可信服务。时间戳应遵循 GM/T 0033-2014《时间戳接口规范》，TSA 时间戳服务主要功能有：

1) 签发时间戳。
2) 验证时间戳。

7.4.13　签发时间戳

URL	http(s)://[主机地址][：端口号]/tsac.svr

参数	digest	（必选）：所需要签名数据的数据摘要信息(hex string 模式，例如摘要是 0x010203...则 digest 则是 string 形式的 010203...)，digest 的长度必须和摘要算法所需的长度一致，例如 md5 应该是 32 个 char，sha1 应该是 40 个 char
	algo	(可选）：数据摘要算法(md5 或者 sha1)。algo 为可选参数，如果在 request 里面没有包含这些数据项，TSA 将根据 digest 的长度自动选择相应的算法。
返回	以 HTTP 响应码的方式返回签发的结果，如：200 代表成功. 4xx：代表失败 5xx：服务端异常错误。 如果成功，则将返回 DER 编码格式的时间戳。在 http response header 里面将指定'Content-Type: application/timestamp-reply'。 如果失败，将记录日志，并返回 400 错。	

7.4.14 验证时间戳

URL	http(s)://[主机地址][：端口号]/tsav.svr	
参数	tsr	必选）: base64 编码格式的时间戳。
	digest	(可选）：数据摘要信息(hex string 模式，例如摘要是 0x010203... 则 digest 则是 string 形式的 010203...)。如果 digest 数据为空，则仅仅验证时间戳签名，而不验证时间戳内容。
	algo	(可选）：数据摘要算法(md5 或者 sha1)。algo 为可选参数， 如果在 request 里面没有包含这些数据项，TSA 将根据 digest 的长度自动选择相应的算法。如果指定了 digest，digest 的长度必须和摘要算法所需的长度一致，例如 md5 应该是 32 个 char；sha1 应该是 40 个 char。
返回	以 HTTP 响应码的方式返回验证的结果，如：200 代表成功。4xx：代表失败 5xx：服务端异常错误。 如果失败，将记录日志，并返回 400 错。	

辽宁省地方标准

7.5 应用接入规范

平台提供了基于多 CA 数字证书互通互认的统一认证服务，为接入到平台的应用系统提供统一身份认证功能。平台基于多 CA 数字证书互通互认的统一认证服务遵循 OAuth2.0 协议开发，OAuth2.0 的授权验证示意图如下所示：

```
+----------+
| resource |
|   owner  |
|          |
+----------+
     ^
     |
    (B)
+----|-----+          Client Identifier      +---------------+
|         -+----(A)-- & Redirection URI ---->|               |
|  User-   |                                 | Authorization |
|  Agent  -+----(B)-- User authenticates --->|     Server    |
|          |                                 |               |
|         -+----(C)-- Authorization Code ---<|               |
+-|----|---+                                 +---------------+
  |    |                                         ^      v
 (A)  (C)                                        |      |
  |    |                                         |      |
  ^    v                                         |      |
+----------+                                     |      |
|         |>---(D)-- Authorization Code ---------'      |
|  Client |          & Redirection URI                 |
|         |                                            |
|         |<---(E)----- Access Token -------------------'
+----------+       (w/ Optional Refresh Token)
```

平台提供 Java/C#/PHP 等多个版本的 SDK 供应用接入使用。

7.5.1 提交创建应用

应用管理员可向平台提交创建应用。应用管理员可以由 XX 平台超级管理员分配。应用系统开发者或管理者可以向平台超级管理员申请应用管理员权限。申请应用管理员权限时需要提供有效的身份证明材料。

提交应用时所需要提交的信息包括：应用名称、应用 LOGO、应用回调地址、应用 URL、应用类型、允许 CA 机构证书、应用图片介绍等，所提交的信息必须真实有效。

应用创建后，可获得由 XX 平台分配的 APP ID 和 APP KEY。APP ID 和 APP KEY 是应用系统的身份标识，特别是 APP KEY 不能泄漏或者扩散。

7.5.2 应用集成 SDK

应用开发者下载平台提供的平台接入 SDK，使用在上一步中的 APP ID 和 APP KEY 作为身份标识，修改应用以集成到平台。修改的内容包括：

1、在应用首页放置“平台统一认证登录”链接，引导用户到 XX 平台统一认证服务进行身份认证。

2、XX 平台提供了多个 API，详见 XX 平台 OpenAPI 列表。应用开发者应遵循 OAuth2.0 规范，根据需要使用 XX 平台提供的多语言的 SDK 以更快地实现 OpenAPI 的调用，完成用户授权确认、身份认证的流程。

3、应用接入完成后，需要进行开发调试，开发调试完成后，即可由应用管理员为该应用申请上线。

+ 应用授权

URL	http(s)://[XX 统一认证服务 URL]/oauth/authorize	
支持验证方式	OAuth2.0	
格式	html/text	
Http 请求方式	GET	
请求参数	response_type 返回类型，固定值：code client_id 应用的 APP ID scope 请求授权的权限 如 user app redirect_uri 授权回调地址 state 随机数	
是否需要鉴权	不需要	
返回值	code	授权码
	state	应用请求的随机数

获取 token

URL	http(s)://[XX 统一认证服务 URL]/oauth/token
支持验证方式	OAuth2.0
格式	JSON
Http 请求方式	POST
请求参数	grant_type　返回类型，固定值：authorization_code client_id 应用的 APP ID scope　请求授权的权限 如 user app redirect_uri　授权回调地址 state 随机数 client_secret　应用的 APP KEY code 获得的授权码
是否需要鉴权	不需要
返回值	ret　返回码 msg 如果 ret<0，会有相应的错误提示信息，返回数据全部用 UTF-8 编码 access_token　token token_type　token 类型 refresh_token 刷新 token expire_in token 过期时间 scope　请求的权限

辽宁省地方标准

获取 OPEN ID

URL	http(s)://[XX 统一认证服务 URL]/oauth/me	
支持验证方式	OAuth2.0	
格式	JSON	
Http 请求方式	GET	
请求参数	access_token 用户授权的 token client_id 应用的 APP ID	
是否需要鉴权	需要	
返回值	ret 返回码 msg 如果 ret<0，会有相应的错误提示信息，返回数据全部用 UTF-8 编码 openid OPEN ID	

获取用户信息

URL	http(s)://[XX 统一认证服务 URL] /user/get_usr_info
支持验证方式	OAuth2.0
格式	JSON
Http 请求方式	POST
请求参数	access_token 用户授权的 token client_id 应用的 APP ID openid 用户与应用唯一对应的 ID
是否需要鉴权	需要
返回值	ret 返回码 msg 如果 ret<0，会有相应的错误提示信息，返回数据全部用 UTF-8 编码 certvalue 身份认证时用的证书（Base64 编码之） login_account 账号口令认证方式的账号 nick_name 昵称

获取用户应用扩展信息

参数	描述
oid	应用扩展 mask，如 2.5.4.13
value	OID 对应值
URL	http(s)://[XX 统一认证服务 URL]/usr/get_usr_extension
返回值	ret　返回码 msg 如果 ret<0，会有相应的错误提示信息，返回数据全部用 UTF-8 编码 app_not_before　用户应用启用时间 app_not_after 用户应用失效时间 action　扩展动作 serial　序列号 oids 扩展 OID 信息集合， 参数　描述 oid　应用扩展 mask，如 2.5.4.13 value　OID 对应值

辽宁省地方标准

解析证书

参数	描述	
oid	应用扩展 mask，如 2.5.4.13	
value	OID 对应值	
URL	http(s)://[XX 统一认证服务 URL]/usr/parse_user_cert	
返回值	ret	返回码
	msg	如果 ret<0，会有相应的错误提示信息，返回数据全部用 UTF-8 编码
	subject_cn	通用名
	subject_ou	组织机构
	subject_o	组织
	subject_l	城市
	subject_s	省
	subject_c	国家
	subject_oids	扩展 OID 信息集合， 参数　　描述 oid　　应用扩展 mask，如 2.5.4.13 value　　OID 对应值
	issuer_cn	证书颁发者通用名
	issuer_ou	证书颁发者组织机构
	issuer_o	证书颁发者组织
	issuer_l	证书颁发者城市
	issuer_s	证书颁发者省
	issuer_c	证书颁发者国家
	subject	证书主题项
	issuer	证书颁发者
	serial_hex	十六进制序列号
	serial_dec	十进制序列号
	not_before	有效起始日期
	not_after	有效截止日期

辽宁省地方标准

	algo	证书算法
	keysize	密钥长度
	fingerprint	证书指纹

验证证书

URL	http(s)://[XX 统一认证服务 URL]/user/verify/verify_user_cert
支持验证方式	OAuth2.0
格式	JSON
Http 请求方式	POST
请求参数	access_token 用户授权的 token client_id 应用的 APP ID openid 用户与应用唯一对应的 ID certvalue 可选项。证书的 Base64 编码字符串 fopenid 可选项。证书对应的 openid。Fopenid 和 certvalue 至少选一个，若同时存在以 certvalue 为主。 format API 返回数据的格式，XML/JSON
是否需要鉴权	不需要
返回值	ret 返回码 msg 如果 ret<0，会有相应的错误提示信息，返回数据全部用 UTF-8 编码 result 验证结果。 验证结果为三位由 0 或者 1 组成的数字，由左至右每一位分别代表验证 CRL 的结果、验证证书链的结果、验证证书有效期的结果。 每一位值为 1 代表相应验证项验证失败，0 代表相应验证项验证通过。 即 0x01-证书有效期验证失败，0x02-证书链验证失败，0x04-CRL 验证失败。 返回的验证结果=（验证 CRL丨验证证书链丨验证有效期）

验证 PKCS1 签名

URL	http(s)://[XX 统一认证服务 URL] /verify/verify_pkcs1
支持验证方式	OAuth2.0
格式	JSON
Http 请求方式	POST
请求参数	access_token 用户授权的 token client_id 应用的 APP ID openid 用户与应用唯一对应的 ID certvalue 可选项。证书的 Base64 编码字符串 fopenid 可选项。证书对应的 openid。Fopenid 和 certvalue 至少选一个，若同时存在以 certvalue 为主。 Pkcs1 必选。Base64 编码的 pkcs1 签名数据 data 必选。Base64 编码的原文数据 datatype 可选项。签名数据的类型，1 代表 16 进制摘要，0 代表原文，默认为原文 algo 签名算法
是否需要鉴权	不需要
返回值	ret 返回码 msg 如果 ret<0，会有相应的错误提示信息，返回数据全部用 UTF-8 编码 result 验证结果

验证 PKCS7 签名

URL	http(s)://[XX 统一认证服务 URL] /verify/verify_pkcs7
支持验证方式	OAuth2.0
格式	JSON
Http 请求方式	POST
请求参数	access_token 用户授权的 token client_id 应用的 APP ID

	openid　用户与应用唯一对应的 ID pkcs7　必选。Base64 编码的 pkcs7 签名数据 verify　可选。是否验证证书有效性
是否需要鉴权	不需要
返回值	ret　返回码 msg 如果 ret<0，会有相应的错误提示信息，返回数据全部用 UTF-8 编码 result　验证结果

返回代码说明见下表：

返回码	说明
0	正确
9000	无效的请求
9001	无效的应用标识
9002	无效的授权
9003	未授权应用
9004	不支持授权类型
9005	无效的 SCOPE
9006	无效的 token
9007	无法匹配回调地址
9008	不支持返回类型
9009	访问拒绝
10000	未知异常
10001	缺少参数 client_id
10002	没有提供 openid 信息
10003	请求应用与授权应用不是同一应用
10004	请求 openid 不在请求应用中
10005	没有查到用户信息
10006	没有查到应用信息
10007	没有提供 token
10009	缺少参数 client_secret

辽宁省地方标准

10010	缺少参数 client_cert
10011	没有绑定应用，没有应用扩展信息
10012	证书解析失败，请确认证书格式是否正确
10016	缺少参数 user_extension
10017	无效的 openId
10018	无效的设备 ID

辽宁省地方标准