ICS 35.040
CCS L 80

LD

中华人民共和国劳动和劳动安全行业标准

LD/T 02.4—2022
代替 LD/T 30.4—2009

人力资源社会保障电子认证体系规范 第4部分：数字证书应用接口规范

Specifications for human resources and social security electronic authentication system— Part 4: Specification for certificate application interface

2022-06-22 发布　　2022-07-01 实施

中华人民共和国人力资源和社会保障部　发布

目　　次

前　言

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

LD/T 02人力资源社会保障电子认证体系系列规范，已经发布了以下五个部分：

----第1部分：框架规范

----第2部分：电子认证系统技术规范

----第3部分：数字证书格式规范

----第4部分：数字证书应用接口规范

----第5部分：数字证书载体规范

本文件为LD/T 02的第4部分。

本文件代替LD/T 30.4—2009《人力资源社会保障电子认证体系 第4部分： 证书应用管理规范》，与LD/T 30.4—2009相比，除结构调整和编辑性改动外，主要技术变化如下：

a) 更改本部分规范名称为《人力资源社会保障电子认证体系 第4部分：数字证书应用接口规范》；
b) 增加了部分规范性引用文件（见第2章，2009版第2章）；
c) 删除了部分不必要的术语定义，同时增加了的术语和定义的来源（见第3章，2009版第3章）；
d) 更改了人力资源社会保障数字证书的应用领域（见第5章，2009版第5章）；
e) 更改了人力资源社会保障数字证书总体技术框架，增加应用系统调用密码应用服务平台服务接口的方式，并根据人力资源社会保障数字证书业务发展实际情况，将第三代社会保障卡及其相应接口纳入到人力资源社会保障数字证书应用技术框架中（见6.1，2009版6.1）；
f) 更改了密码设备组成以及对密码设备的要求（见6.2，2009版6.2）；
g) 更改了对基础应用接口的描述，并根据人力资源社会保障数字证书应用总体技术框架对基础应用接口的组成及各部分进行定义，增加基础应用接口各组成部分的具体技术要求（见6.3，2009版6.3）；
h) 更改了对高级应用接口的描述，根据人力资源社会保障数字证书应用总体技术框架，对高级应用接口的组成及各部分进行定义，增加对各接口形态的技术要求（见6.4，2009版6.4）；
i) 更改了证书应用接口技术要求（见6.5，2009版6.5）；
j) 更改了典型证书应用流程，按照新规范定义的客户端应用接口和服务端应用接口对应用流程中应用系统需调用的接口名称进行修订（见第7章，2009版第7章）；
k) 更改和增加了证书应用场景，根据服务器端应用接口分类，按照COM组件接口和Java组件接口调用、密码应用服务接口调用，分别描述应用集成部署模型和数字证书应用示例（见附录A，2009版附录A）；
l) 增加了数字证书客户端与服务端应用接口技术要求（见附录B，2009版附录B）。

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

本文件由中华人民共和国人力资源社会保障部信息中心提出并归口。

本文件起草单位：中华人民共和国人力资源社会保障部信息中心、普华诚信信息技术有限公司、北京数字认证股份有限公司、北京信安世纪科技股份有限公司。

本文件主要起草人：马丹蕾、张嵩、王岩、耿建军、唐淑静、韩晓颖、成勇、王祥宇、李娜、王智飞、郭丽芳、高五星、李述胜、叶鹏。

本文件所代替的历次版本发布情况为：

----LD/T 30.4—2009《人力资源社会保障电子认证体系 第4部分： 证书应用管理规范》；

----本次为第一次修订。

引　言

为适应人力资源社会保障信息化发展要求，满足人力资源社会保障网络信任体系建设和管理的需要，人力资源社会保障部组织并制定了人力资源社会保障电子认证体系系列规范。随着我国商用密码技术的发展、国产密码算法的标准发布，以及人力资源社会保障行业的业务发展，需要对行业标准LD/T 30—2009《人力资源社会保障电子认证体系规范》进行修改和完善。

本次修订，是在充分借鉴原标准的框架和结构的基础上，根据人力资源社会保障行业特点和电子认证业务发展需求，对电子认证体系总体结构和电子认证系统整体建设规划进行扩充完善，以符合国家及国家密码管理部门相关标准规范要求，满足人力资源社会保障业务和管理需求，推进SM2算法在人社信息系统中的应用，另一方面，也可有效配合《中华人民共和国密码法》、《中华人民共和国网络安全法》、密码管理及密码应用安全测评工作、等级保护工作的落实与实施。

LD/T 02描述了人力资源社会保障电子认证体系总体结构和电子认证系统整体建设规划，规定了各级人力资源社会保障部门电子认证系统建设和应用要求，由以下五个部分构成。

----第1部分：框架规范

----第2部分：电子认证系统技术规范

----第3部分：数字证书格式规范

----第4部分：数字证书应用接口规范

----第5部分：数字证书载体规范

LD/T 02的第1部分，是人力资源社会保障电子认证体系系列规范的总纲，规定了电子认证体系规范的总体框架。LD/T 02的第2部分~第5部分分别从电子认证系统技术、数字证书格式、数字证书应用接口、数字证书载体四个方面提出具体规范要求。

本部分重点引用了GM/T 0020-2012相关规范，并在此基础上，扩展了基于人力资源社会保障密码应用服务平台的密码应用服务接口、典型应用流程等相关内容，给出了几类常见的人力资源社会保障业务系统的数字证书应用场景，从满足人力资源社会保障业务需求的角度，对本行业应用系统使用数字证书的接口和流程提出规范和要求。

人力资源社会保障电子认证体系规范
第4部分：数字证书应用接口规范

1 范围

本文件给出了数字证书的应用领域，规定了数字证书应用技术框架和典型应用流程。

本文件适用于指导人力资源社会保障应用系统实现基于电子认证体系的安全功能，有助于各级人力资源社会保障部门采用统一的数字证书应用开发接口。

2 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中，注日期的引用文件仅该日期对应的版本适用于本文件；不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

GB/T 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

LD/T 01-2022（所有部分） 人力资源社会保障电子印章体系规范

LD/T 32 社会保障卡规范

LD/T 33 社会保障卡读写终端规范

PKCS#7 Cryptographic Message Syntax 加密消息的语法标准

PKCS#10（V1.7） Certification Request Syntax Standard 认证请求语法标准

3 术语和定义

GB/T 35291、GB/T 0020界定的以及下列术语和定义适用于本文件。

3.1

容器 container

密码设备中用于保存密钥所划分的唯一性存储空间。

［来源：GB/T 35291-2017，3.1］

3.2

BSTR batch stirred tank reactor

在定义客户端服务接口时，本规范以 ActiveX 控件为例进行描述，其中 BSTR 代表函数返回值或参数类型为 OLECHAR 字符串类型，不同的开发语言应采取对应的类型定义，如：char * 、CString、java.lang.String 等。

[来源：GM/T 0020-2012，6.2]

4 缩略语

下列缩略语适用于本文件：

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

CA：证书认证机构（Certification Authority）

CRL：证书撤销列表（Certificate Revocation List）

CSP：加密服务提供者（Cryptographic Service Provider）

KMC：密钥管理中心（Key Management Center）

LDAP：轻量级目录访问协议（Lightweight Directory Access Protocol）

OID：对象标识符（Object Identifier）

HTTP：超文本传输协议（Hypertext Transfer Protocol）

PKCS：公钥密码标准（the Public-Key Cryptography Standard)

RA：证书注册机构（Registration Authority）

5 应用领域

人力资源社会保障数字证书的应用领域包括全国性、区域性的各类应用系统，按照业务类别划分为以下四类：

a) 就业创业类：就业创业、失业监测等业务系统；
b) 社会保障类：养老、工伤等社会保险业务系统、社会保障卡持卡库等业务系统、社会保险公共服务平台；
c) 人才人事类：事业单位人事管理、专业技术人员管理、人力资源市场、职业资格管理、职业技能培训等业务系统；
d) 劳动关系类：劳动用工备案、调解仲裁、劳动监察等业务系统。

6 数字证书应用技术体系

6.1 总体技术框架

人力资源社会保障数字证书应用技术框架由密码设备、基础应用接口、高级应用接口组成，如图 1 所示。

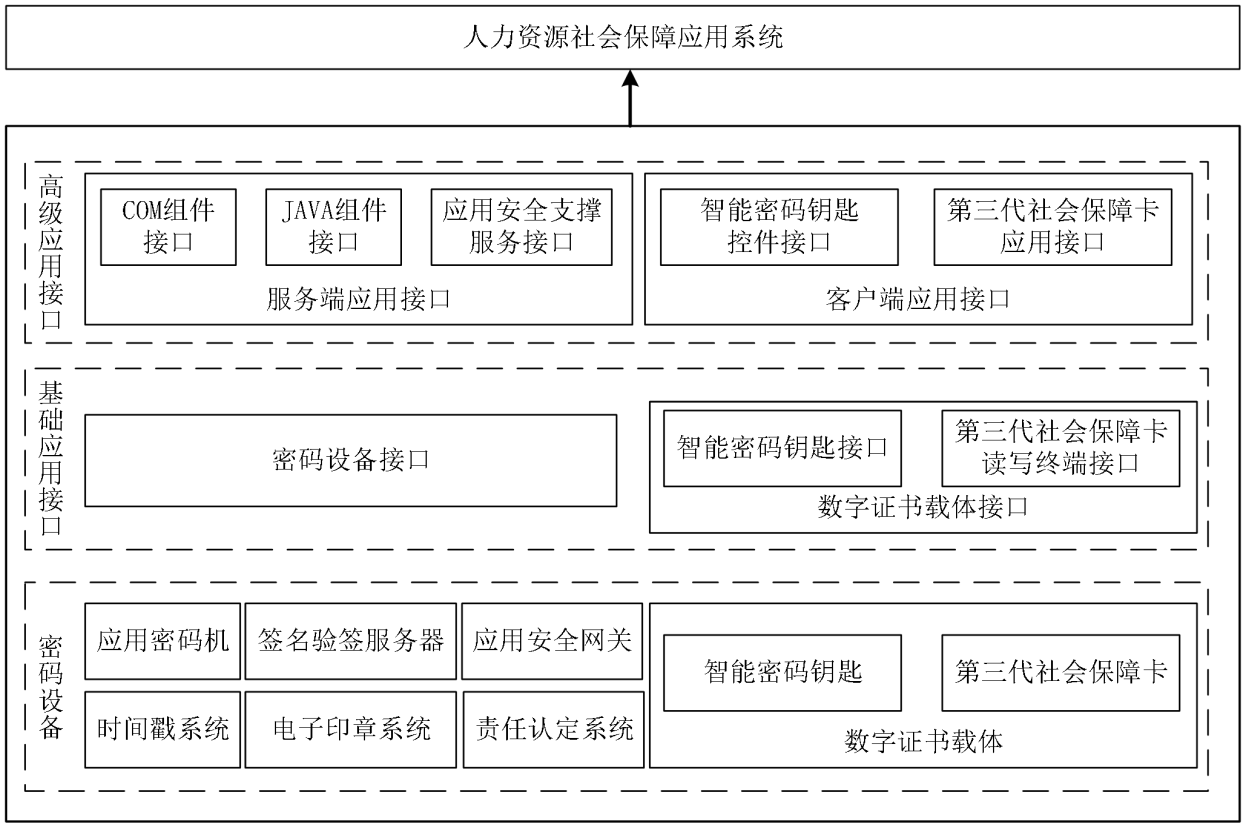

图 1 人力资源社会保障数字证书应用总体技术框架

6.2 密码设备

密码设备由应用密码机、签名验签服务器、应用安全网关、电子印章系统、时间戳系统、责任认定系统、证书载体等密码产品组成，密码产品通过基础应用接口向上层应用提供基础的、全面的密码服务。

应用密码机、签名验签服务器、应用安全网关、时间戳服务器等必须采用支持 SM1、SM2、SM3、SM4 密码算法的、经国家密码管理部门批准使用的密码设备，具有国家密码管理部门认证的产品认证证书。

数字证书载体主要指用于存储密钥和数字证书并具有密码运算功能的载体，目前人力资源社会保障电子认证系统所采用的数字证书载体包括智能密码钥匙和第三代社会保障卡两类，为应用系统的客户端提供密码服务。必须采用支持 SM1、SM2、SM3、SM4 密码算法的、经国家密码管理部门批准使用的密码设备，具有国家密码管理部门认证的产品认证证书。

数字证书载体的各项技术参数应符合 LD/T 02.5 规定的要求。

6.3 基础应用接口

基础应用接口位于密码设备之上，密码设备通过基础应用接口向上层的高级应用接口层提供密钥生成、密码运算等基础密码服务。基础应用接口分为服务器端密码设备接口和证书载体接口两类，其中，数字证书载体接口包括智能密码钥匙接口和第三代社会保障卡读写终端接口。

6.3.1 密码设备应用接口

密码设备应用接口应支持包括国产操作系统在内的主流操作系统，例如 Windows、Linux、Unix 等主流操作系统。

应用密码机接口应符合 GB/T 36322-2018 规定的要求。

签名验签服务器接口应符合 GB/T 38629-2020 规定的要求。

应用安全网关接口应符合 GM/T 0026-2014、GM/T 0025-2014 规定的要求。

电子印章系统接口应符合 GB/T 38540-2020 以及 LD/T 01-2022 规定的要求。

时间戳系统接口应符合 GM/T 0033-2014 规定的要求。

6.3.2 证书载体接口

智能密码钥匙接口应符合 GB/T 35291-2017 和 LD/T 02.5 规定的要求，支持国产操作系统及 Windows、Linux 等主流操作系统，支持国产浏览器以及 IE、火狐、Chrome 等主流浏览器及相应版本。

第三代社会保障卡接口应符合 LD/T 32 规定的要求。

6.4 高级应用接口

高级应用接口是位于基础应用接口之上，为应用系统提供身份认证、数字签名、数据加解密、时间戳、电子印章、责任认定等安全保障服务，该接口可直接被应用系统调用，将应用系统的密码服务请求转向基础应用接口，通过基础应用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。高级应用接口包括客户端应用接口和服务器端应用接口两大部分。

客户端应用接口根据数字证书载体类型形态分为两类：数字证书载体为智能密码钥匙时，客户端应提供跨浏览器支持组件、ActiveX 控件、动态链接库、JAR 等开发包，客户端应用接口符合 GM/T 0020-2012 中 7.1 规定的要求，见附录 B.1；证书载体为第三代社会保障卡时，客户端应提供 ActiveX 控件或 DLL 动态连接库，客户端应用接口符合 LD/T 33 规定的要求。本文件主要规范智能密码钥匙形态的客户端应用接口（客户端控件接口）。

服务器端应用接口根据功能及接口形态的不同，分为 COM 组件接口、Java 组件接口和密码应用服务接口。应用系统服务器端直接调用密码设备时，通过 COM 组件接口和 Java 组件接口实现，其主要功能包括：身份认证、数字签名、数据加解密等，COM 组件接口符合 GM/T 0020-2012 中 7.2 规定的要求，见附录 B.2.1，Java 组件接口符合 GM/T 0020-2012 中 7.3 规定的要求，见附录 B.2.2。应用系统服务器端调用密码应用服务平台时，通过密码应用服务接口实现，该接口的主要功能包括：身份认证、数字签名、数据加解密、时间戳、电子印章、责任认定等，密码应用服务接口形态为 HTTP restful，见附录 B.2.3。

6.5 应用接口技术要求

6.5.1 密码算法

密码设备和密码服务接口应提供符合国家密码管理部门规定的密码算法和接口规范，并根据国家密码管理部门批准的算法及时调整，以适应国家最新技术标准要求。密码算法及算法类型如表 1 所示。

a) 对称密码算法：SM1、SM4 等；
b) 公钥密码算法：SM2 算法，兼容 RSA 算法但不推荐使用；
c) HASH 算法：SM3、SHA256 等。

表 1 密码算法

算法名称	算法类型	Oid
SM1 算法	我国分组密码算法	1.2.156.10197.1.102
SM4 算法	我国分组密码算法	1.2.156.10197.1.104
SM2 椭圆曲线密码算法	我国非对称密码算法	1.2.156.10197.1.301

算法名称	算法类型	Oid
SM2 椭圆曲线签名算法	我国非对称密码算法	1.2.156.10197.1.301.1
SM2 椭圆曲线密钥交换协议	我国非对称密码算法	1.2.156.10197.1.301.2
SM2 椭圆曲线加密算法	我国非对称密码算法	1.2.156.10197.1.301.3
RSA 算法	国外非对密码密算法	1.2.840.113549.1.1.1
SM3 杂凑算法	我国杂凑密码算法	1.2.156.10197.1.401
SHA-256 杂凑算法	国外杂凑密码算法	2.16.840.1.101.3.4.2.1
SM3withSM2	我国数字签名算法	1.2.156.10197.1.501
SM3withRSA	国外数字签名算法	1.2.156.10197.1.504
SHA1withRSA	国外数字签名算法	1.2.840.113549.1.1.5
SHA256withRSA	国外数字签名算法	1.2.840.113549.1.1.11

6.5.2 应用接口运行环境

密码服务接口应支持在以下环境中可靠运行：

a) 客户端支持国产 Linux 桌面操作系统及 WindowsXP、Windows7、Windows8、Windows10 等主流操作系统；
b) 服务器端支持国产 Linux 操作系统及 Windows2012（或更高版本）、Linux、Unix 等所有主流操作系统；
c) 服务器端支持硬件密码机或加密卡，客户端证书载体支持智能密码钥匙、智能 IC 卡（第三代社会保障卡）等设备；
d) 应用系统支持 C/S 和 B/S 的系统结构。

6.5.3 数据结构

本文件对应的接口处理的数据分为两种类型：

a) 数据类型A：当公钥算法为SM2时，数据的结构符合GB/T 35276-2017规定的要求；当公钥算法为RSA时，数据的结构应符合PKCS#1；
b) 数据类型 B：当公钥算法为 SM2 时，消息的结构符合 GB/T 35275-2017 规定的要求；当公钥算法为 RSA 时，消息的结构应符合 PKCS#7。

7 数字证书典型应用流程

7.1 概述

数字证书典型应用流程主要包括以下几类：

a) 数字证书登录认证；
b) 单向数字签名与验签；
c) 双向数字签名与验签；
d) 加密与解密；
e) 加密签名与解密验签。

另外，数字证书应用场景相关示例见附录 A。

7.2 数字证书登录认证

基于数字证书的登录认证是指用户通过数字证书方式安全登录人力资源社会保障业务系统的过程。

在应用过程中，用户应在业务系统中进行注册，同时需要将业务系统中的注册用户与用户数字证

书进行关联。用户使用数字证书登录前，应安装数字证书客户端软件，准备好证书运行环境。

数字证书登录认证流程如图 2 所示。

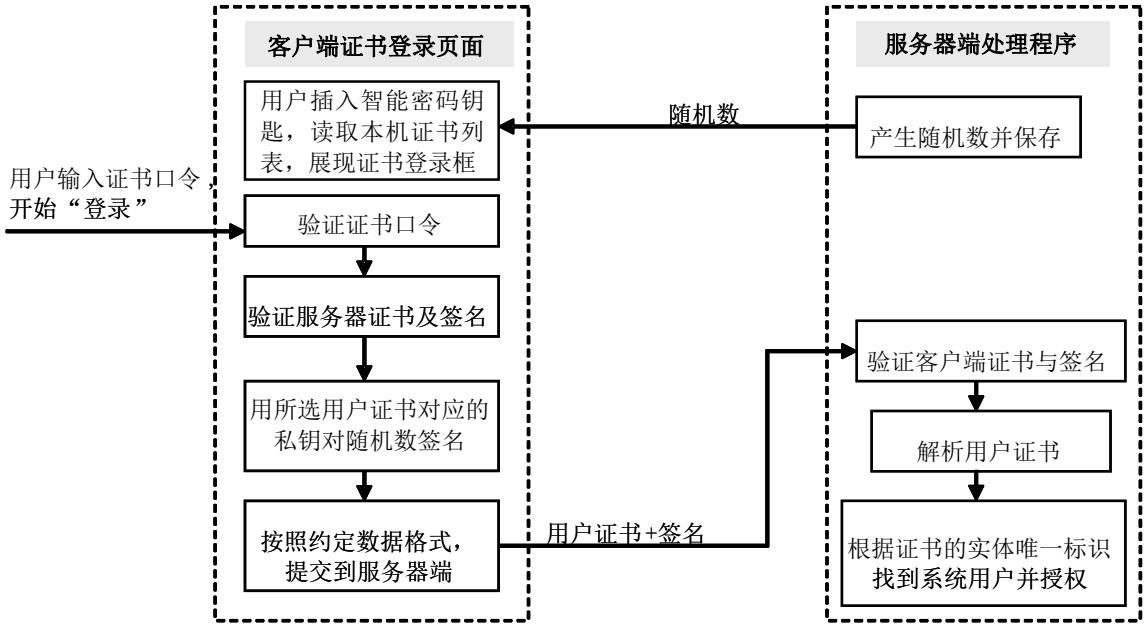

图 2 数字证书登录认证流程图

数字证书登录认证流程：

a) 用户将证书载体（智能密码钥匙）插入计算机，浏览证书登录首页。

b) 服务器端处理程序产生一个随机数，通过密码设备签名，并将设备证书、随机数及其电子签名值下载到客户端。登录页面获取本机注册的数字证书，展现给用户一个证书登录框。

c) 用户选择数字证书，输入证书保护密码（即智能密码钥匙的用户口令），点击“登录”按钮。此时，客户端程序（如 JavaScript）应调用证书高级应用接口（ActiveX 控件）对证书密码进行校验。以 ActiveX 控件为例，分别调用 SOF_GetDeviceInstance 和 SOF_Login ()函数。

d) 登录成功后，使用证书载体（智能密码钥匙）对随机数进行签名。以 ActiveX 控件为例，调用 SOF_SignData ()函数。

e) 签名成功后，按照约定的数据格式将数据提交给服务器。数据格式中至少应包括用户证书和证书载体对随机数的签名。

f) 服务器接收到客户端提交的登录请求后，应调用证书应用接口或密码应用服务接口验证客户端证书及其签名的有效性，包括以下几个方面：

   1) 用户证书是否在有效期内；
   2) 用户证书是否为服务器端配置的 CA 证书所签发；
   3) 用户证书是否已被注销；
   4) 验证用户证书对随机数签名的有效性。

   以 Java 组件为例，分别调用 SOF_validateCert ()和 SOF_VerifySignedData ()函数。

   以密码应用服务接口为例，分别调用 B.2.3.2.6 证书验证接口和 B.2.3.3.2 RAW 方式验签接口。

g) 服务器端调用证书应用高级接口，解析用户证书信息，获取证书的实体唯一标识。

h) 服务器程序根据证书的实体唯一标识，在系统中找到对应的用户进行相应授权。

7.3 单向数字签名与验签

单向数字签名与验签是指客户端对业务数据进行数字签名，服务器验证后进行业务处理的过程，

从而实现客户端向服务器端提交数据的完整性和用户业务操作的不可抵赖性。

单向数字签名和验签流程如图 3 所示。

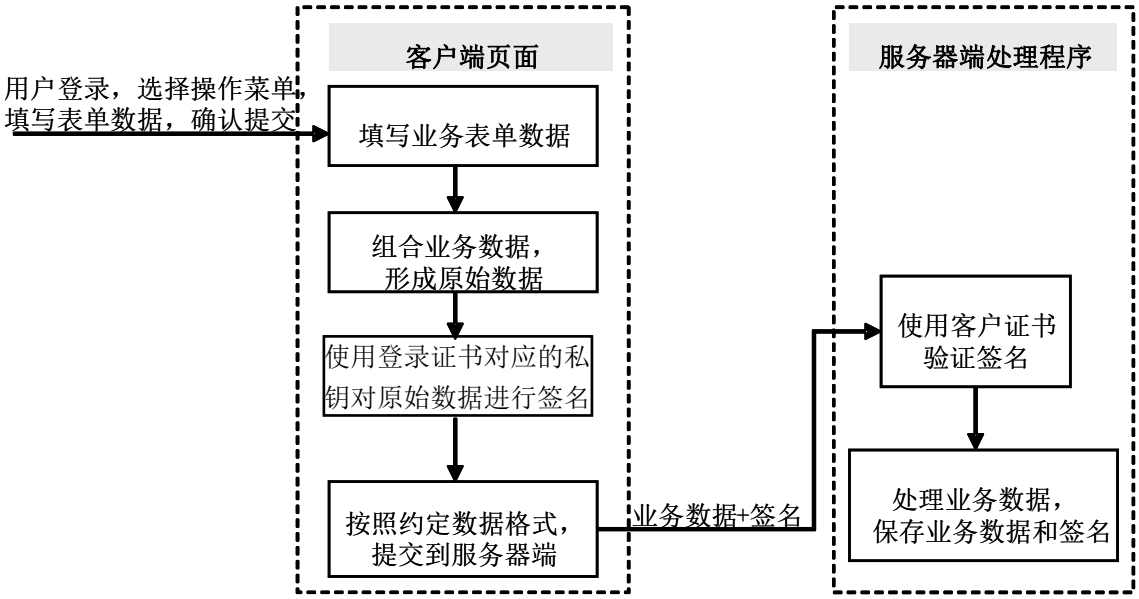

图 3 单向数字签名与验签流程图

单向数字证书签名与验签流程：

a) 用户使用数字证书登录系统，选择操作菜单，根据提示填写表单数据，提交业务数据。

b) 客户端组合业务数据，形成原始数据。

c) 客户端调用证书高级应用接口（ActiveX 控件），使用用户数字证书对应的私钥对原始数据进行数字签名，数字签名时应采用签名密钥，验证时采用签名证书。以 ActiveX 控件为例，分别调用 SOF_GetDeviceInstance ()、SOF_Login()、SOF_SignData()和 SOF_GetCertInfo()函数。

d) 按照约定格式，将原始数据和数字签名提交到服务器端。

e) 服务器端接收到客户端的数据后，应调用证书应用接口或密码应用服务接口验证客户端对业务数据的签名。

以 Java 组件为例，调用 SOF_verifySignedData()函数。

以密码应用服务接口为例，调用 B.2.3.3.2 RAW 方式验签接口。

f) 验证签名通过后，服务端程序处理业务数据，并保存业务数据和签名。

7.4 双向数字签名与验签

双向数字签名与验签是指客户端和服务器端双方分别对业务数据进行数字签名和验证，从而实现客户端和服务器端双向数据传输的数据完整性和双方业务操作的不可抵赖性。

双向数字签名和验签流程如图 4 所示。

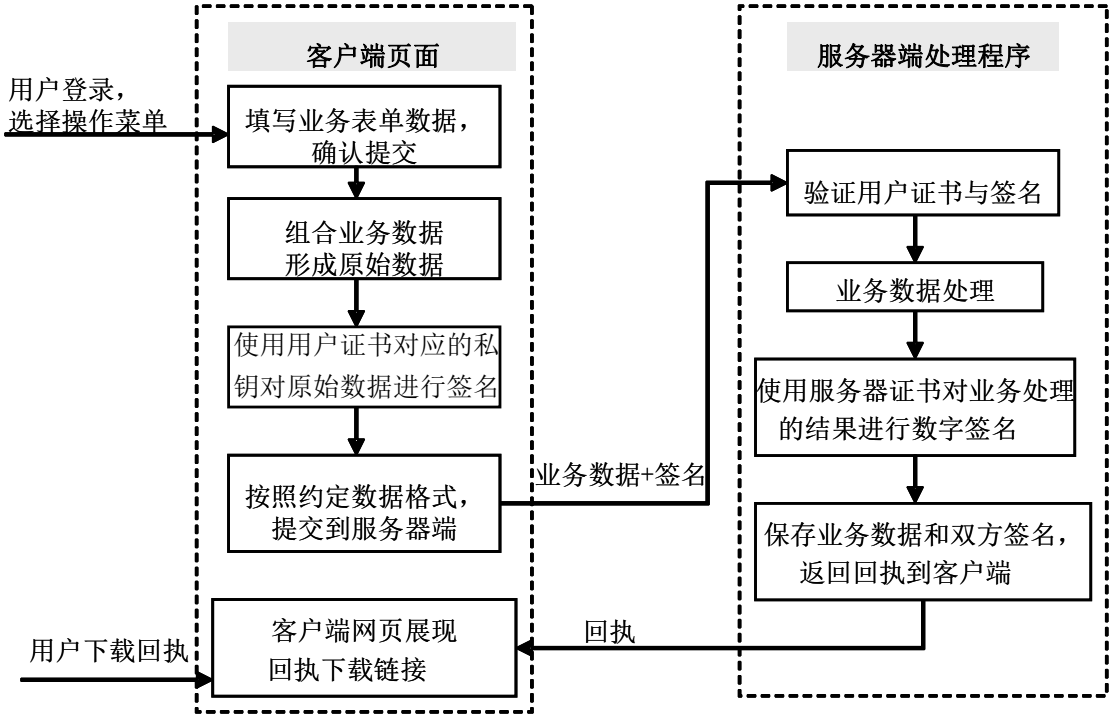

图 4 双向数字签名与验签流程图

双向的数字签名和验签的流程：

a) 用户使用数字证书登录系统，选择操作菜单，根据提示填写表单数据，提交业务数据。
b) 客户端组合业务数据，形成原始数据。
c) 客户端调用证书高级应用接口（ActiveX 控件），使用用户数字证书对应的私钥对原始数据进行数字签名，数字签名时应采用签名密钥，验证时采用签名证书。以 ActiveX 控件为例，分别调用 SOF_GetDeviceInstance ()、SOF_Login()、SOF_SignData()和 SOF_GetCertInfo()函数。
d) 按照约定格式，将原始数据和数字签名提交到服务器端。
e) 验证签名通过后，服务端程序处理业务数据，并保存业务数据和签名。
f) 服务端程序处理成功后，使用设备证书对应的私钥对业务操作的结果进行数字签名（设备证书对应的私钥一定从密码设备中生成）。
以 Java 组件为例，调用 SOF_signData()函数。
以密码应用服务接口为例，调用 B.2.3.3.4 RAW 签名接口。
g) 服务端程序保存原始数据、客户端签名、服务器端的数字签名，作为操作证据安全存储在服务器端，返回回执到客户端供用户下载。
h) 用户下载回执文件并保存。

7.5 加密与解密

加密与解密是指业务数据的加密与解密，实现业务数据的保密性。人力资源社会保障业务系统中数据加密与解密的方式包括：

a) 使用标准 SSL 协议对业务数据加密与解密；
b) 使用数字信封对业务数据进行加密与解密。

业务系统服务器端不需要保存密文，可选择标准 SSL 协议进行加密和解密。服务器端需要保存密文，应通过高级证书应用接口实现基于数字信封方式的加密与解密，如图 5 所示。

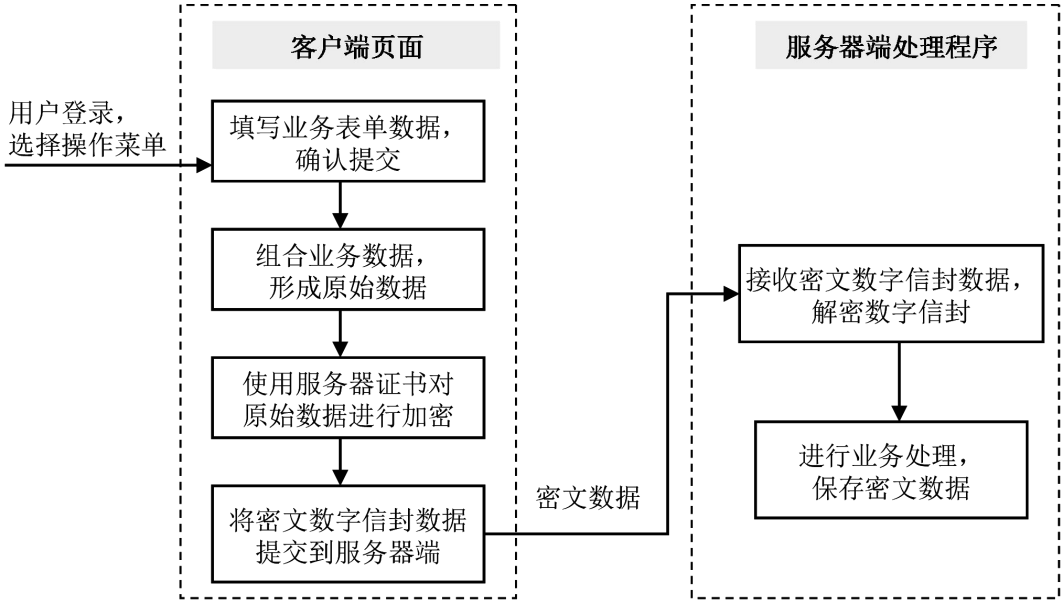

图 5 加密与解密流程图

加密与解密的流程：

a) 用户使用数字证书登录系统后，选择操作菜单，根据提示填写表单数据，提交业务数据。

b) 客户端组合业务数据，形成原始数据。

c) 客户端调用高级证书应用接口（控件），使用服务器端设备证书对原始数据进行加密，形成密文数字信封数据。以 ActiveX 控件为例，调用 SOF_EncryptData()函数。

d) 按照约定格式，将密文数字信封数据提交到服务器端。

e) 服务端程序接收到密文数字信封的业务数据后，使用服务器设备证书对应的密码设备解密，形成明文格式的业务数据。

以 Java 组件为例，调用 SOF_decryptData ()函数。

以密码应用服务接口为例，调用 B.2.3.4.11 数据解密接口。

f) 服务端程序将密文数字信封数据解密后，进行相应的业务处理，是否保存密文数字信封数据根据应用需求而定。

7.6 加密签名与解密验签

加密签名与解密验签是指在客户端对业务数据进行加密和签名，服务器端在业务处理前对密文进行解密和验证签名的过程，从而实现数据保密性、完整性和操作的不可否认性。

加密签名与解密验签流程如图 6 所示。

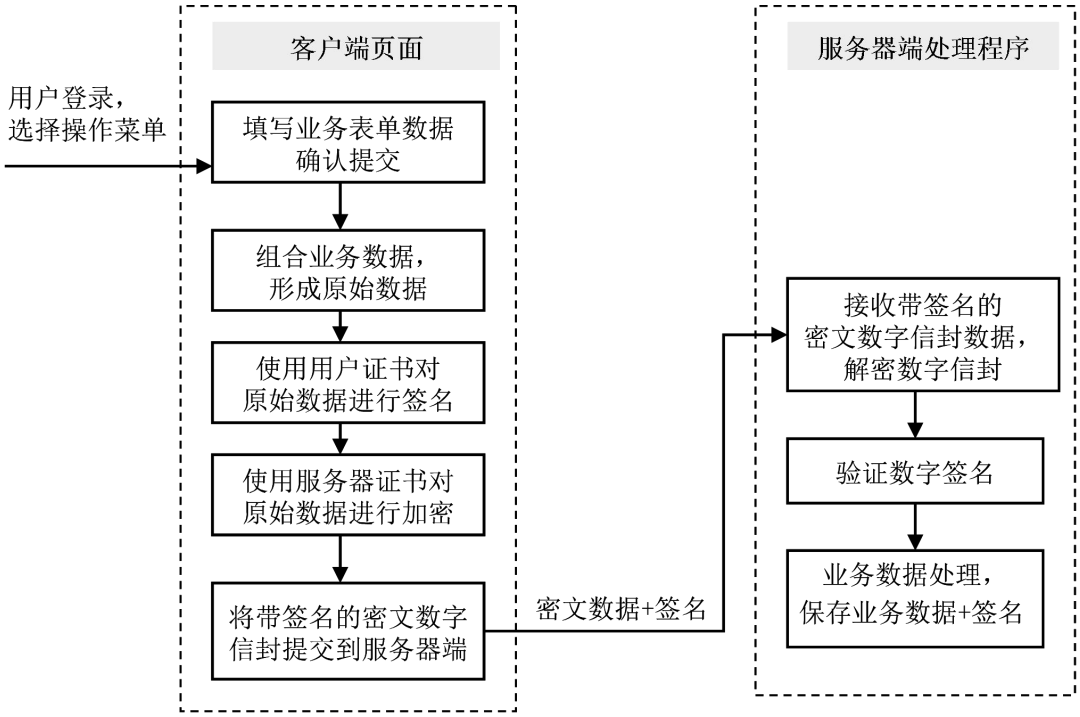

图 6 加密签名与解密验签流程图

加密签名与解密验签的流程：

a) 用户使用数字证书登录系统后，选择操作，根据提示填写表单数据，提交业务数据。

b) 客户端组合业务数据，形成原始数据。

c) 客户端调用高级证书应用接口（控件），使用用户证书对原始数据进行数字签名，数字签名时应采用发送方的签名密钥，验证时采用发送方签名证书。以 ActiveX 控件为例，分别调用 SOF_GetDeviceInstance ()、SOF_Login()、SOF_SignData()和 SOF_GetCertInfo()函数。

d) 客户端调用高级证书应用接口（控件），使用设备证书对原始数据进行加密，形成密文数据。以 ActiveX 控件为例，调用 SOF_EncryptData()函数。

e) 按照约定格式，将带签名的密文数字信封数据提交到服务器端。

f) 服务端程序接收到带签名的密文数字信封数据后，使用服务器端设备证书对应的密码设备解密。以 Java 组件为例，调用 SOF_decryptData ()函数。
以密码应用服务接口为例，调用 B.2.3.4.11 数据解密接口。

g) 服务端程序将密文数字信封数据解密后，使用用户证书验证数字签名，进行相应的业务处理。以 Java 组件为例，调用 SOF_verifySignedData()函数。
以密码应用服务接口为例，调用 B2.3.3.2 RAW 方式验签接口。

h) 保存原始业务数据和客户端签名，作为操作证据安全存储在服务器端。

附 录 A
（资料性）
数字证书应用场景

随着政策环境和业务需求的变化，人力资源社会保障业务系统会随之变化，因此本文件并不针对具体的业务系统进行详细分类和需求分析，而是从证书应用功能角度分析证书应用的安全需求。

A.1 COM组件接口和Java组件接口调用

A.1.1应用集成部署模型

数字证书应用软件分为客户端和服务器端两个模块，应用系统服务器端通过调用COM组件接口和Java组件接口实现身份认证、数字签名、数据加解密等基本功能。典型的数字证书应用集成部署模型如图A.1所示。

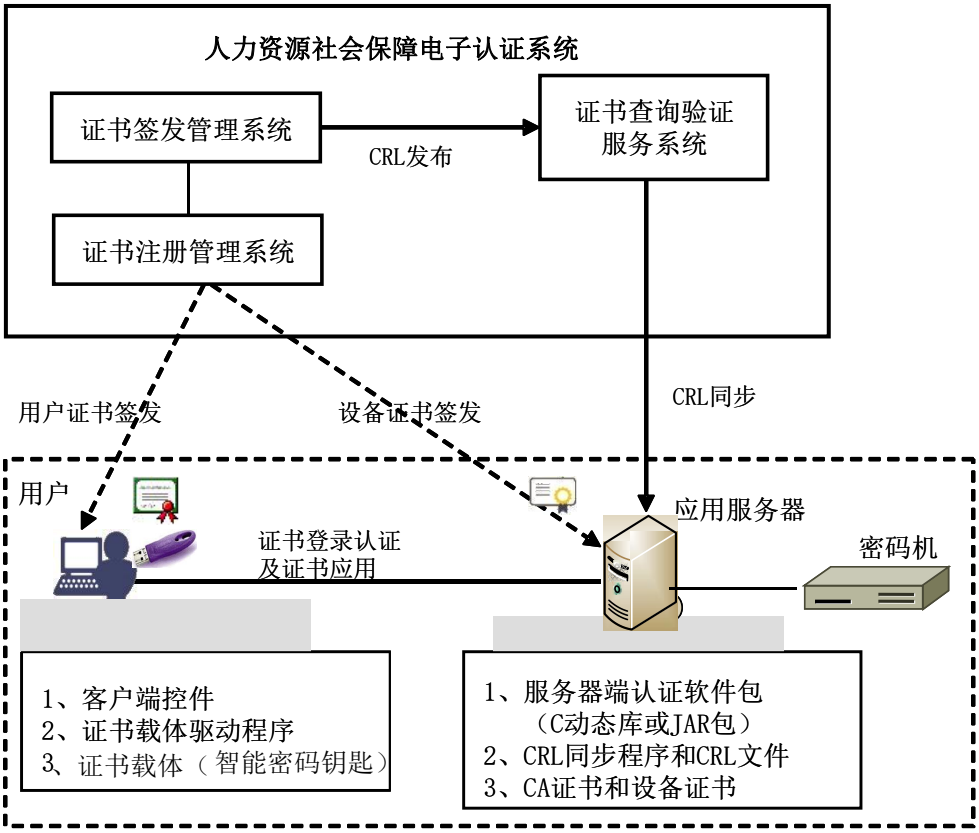

图 A.1 证书应用集成部署模型

根据图A.1所示，基于数字证书的应用系统涉及三个方面：人力资源保障电子认证系统、应用服务器和客户端控件及证书载体等。

人力资源社会保障电子认证系统负责为用户签发用户数字证书，为应用服务器签发设备证书，并通过证书查验服务系统向应用服务器同步CRL文件。

应用服务器端需要部署的内容包括：

a) 服务器认证软件包：进行加密解密、数字签名验签的软件接口。软件包为C语言的动态库文件或Java语言的Jar包；

b) CRL同步程序：定时从证书查验服务系统下载CRL文件的服务程序；

c) CRL文件：CA系统发布的黑名单文件；

d) CA证书：行业CA证书和业务CA证书；

e) 设备证书：代表服务器身份的数字证书，私钥由密码机产生；

f) 密码机：存储服务器设备证书对应密钥的密码设备。

客户端用户需要安装证书应用软件客户端控件及证书载体的驱动程序。在日常登录和使用时应将证书载体（智能密码钥匙）插入客户端的电脑中。

客户端和服务器端通过双方的数字证书进行双向身份认证，在数据传输过程中，双方可通过数字证书进行签名、验证、加密、解密等安全操作，实现数据完整性、操作的不可抵赖性和数据保密性。

A.1.2 数字证书应用示例

A.1.2.1 数字证书初始化绑定

用户首先应按照“证书申请流程”完成能够代表自己身份的数字证书申请，以及业务系统的用户注册。在登录业务系统前，首先应完成数字证书的绑定，即将数字证书的唯一标识与该用户在系统中的账户（或用户名）进行绑定，实现数字证书与系统用户及其权限的一一对应。证书绑定成功后，应及时修改数字证书的pin码。

数字证书绑定的操作流程如图A.2所示。

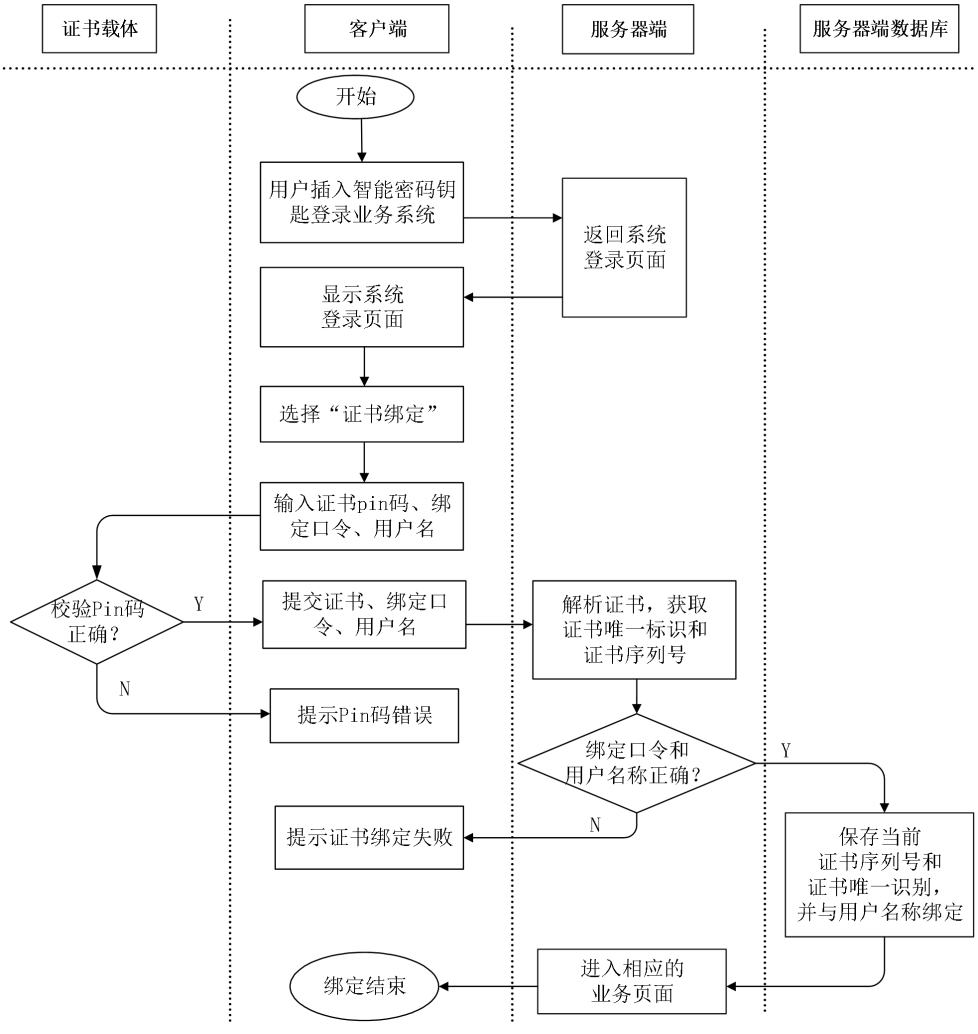

图 A.2 数字证书绑定流程图

A.1.2.2 数字证书身份认证

证书用户完成数字证书绑定后，即可使用数字证书登录业务系统，数字证书登录认证处理流程如下：

a) 用户将数字证书载体（智能密码钥匙）插入计算机后，输入业务系统URL地址，访问系统；
b) 服务器端程序调用认证软件包产生一个随机数，并保存在会话（Session）中；
c) 服务器端将随机数传递给客户端；
d) 用户输入证书pin码，由数字证书载体（智能密码钥匙）验证证书pin码的正确性；
e) 验证证书pin成功后，使用智能密码钥匙对随机数签名；
f) 客户端将用户证书和签名值提交到服务器端；
g) 服务器端程序接收到上述数据后，验证用户证书和签名值的有效性。其中，验证证书的有效性包括：证书有效期、信任源以及是否被吊销；
h) 解析证书中的唯一标识，根据唯一标识获取用户权限，显示用户权限对应的操作页面。

数字证书身份认证处理流程如图A.3所示。

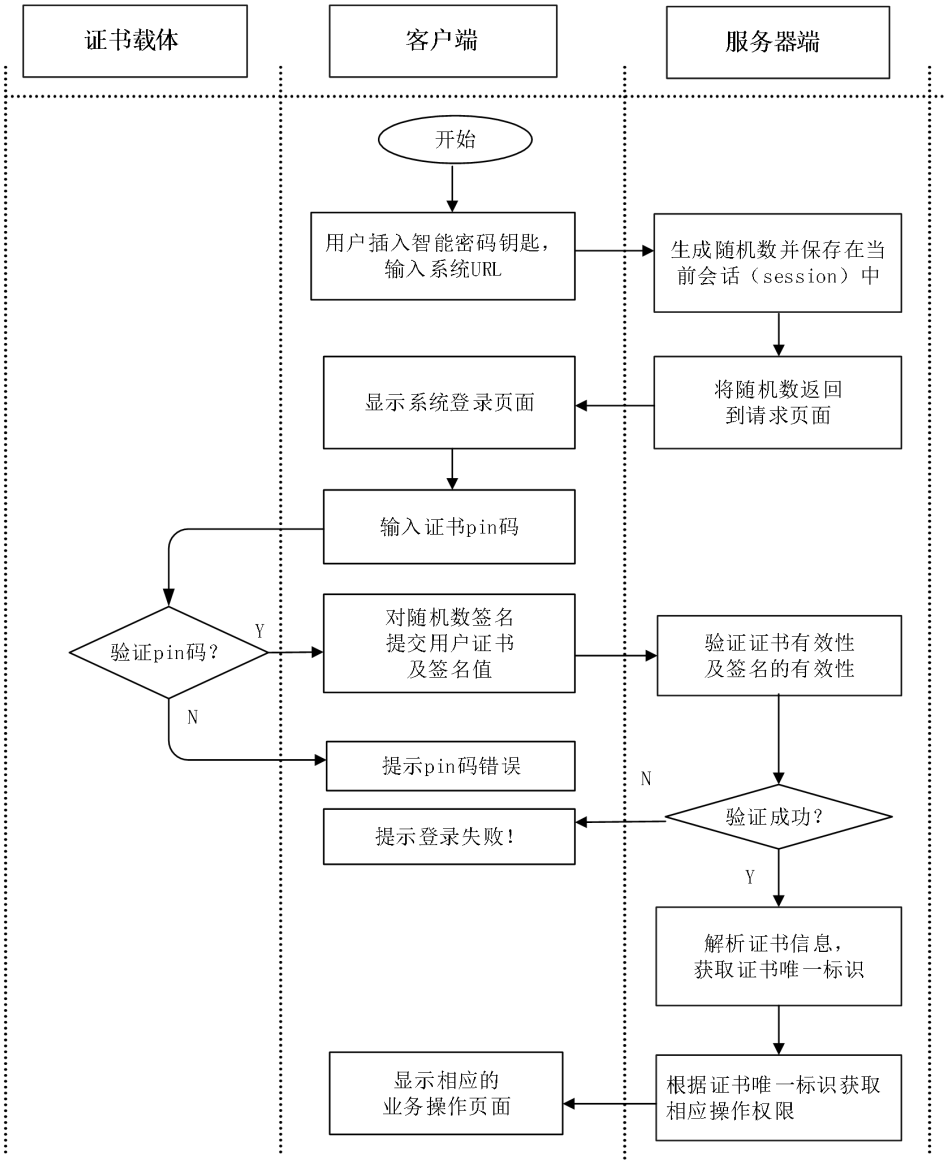

图 A.3 数字证书身份认证流程图

A.1.2.3 签名验签示例1：金保工程异地业务系统

以金保工程异地业务系统为例，介绍在业务系统中如何利用数字证书实现身份认证和对传输的重要业务数据的签名保护，从而实现数据完整性和不可抵赖性。

异地业务系统主要是实现异地社会保险关系转移信息交换的业务系统，异地社会保险关系转移的主要业务操作流程如下：

a) A地的参保人员首先登录到本地的异地转移系统，申请保险关系转移业务，并向异地系统上传申请接收函信息；

b) B地异地系统查询下载接收函信息，进行办理转出操作，并向异地系统上传分险种的转移单；

c) A地异地系统查询转移单，并下载办理转入，完成后上传转移单，操作完毕。

以A地办理转入业务申请为例，签名验签处理流程如下：

a) 用户将数字证书载体（智能密码钥匙）插入计算机，登录到异地业务系统；

b) 编辑、输入、上传接收函文件；

c) 使用智能密码钥匙对接收函文件进行签名；

d) 提交接收函文件、签名及签名证书；

e) 服务器端程序验证签名有效性，对接收函验证签名；

f) 验证成功后，保存当前的接收函，业务处理结束。

异地业务系统证书签名验签业务处理流程如图A.4所示。

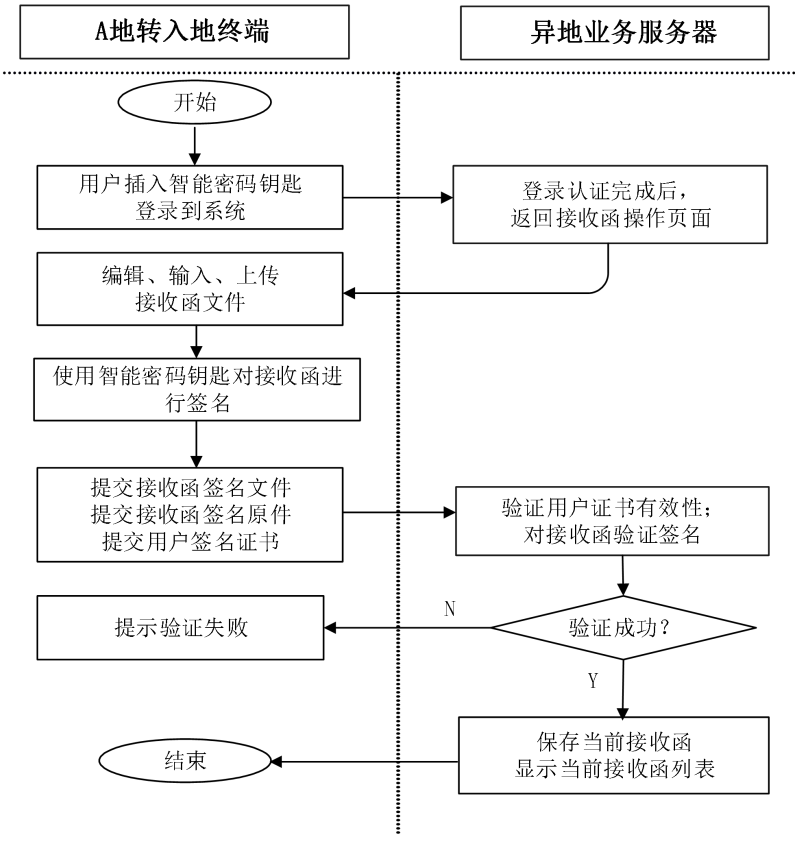

图 A.4 金保工程异地业务系统证书签名验签流程图

A.1.2.4 签名验签示例2：金保工程传输总线系统

金保工程传输总线系统通过建立数据总线通道，利用数字签名技术实现部门间数据的安全上传和下发，主要应用在金保工程联网软件系统数据报表的上传和下发。

金保工程传输总线系统中数字证书签名验签处理流程如下：

a) 下级应用系统（如：地市联网软件系统）需要上传数据报表到上级部门时，操作联网软件的业务人员将数据上传到本地总线服务器上；
b) 本地总线服务器定时轮询目录中需要上传的报表文件。当检测到上报文件后，产生随机数A，然后向目的地总线服务器发出服务请求；
c) 目的地总线服务器接收到请求后，产生随机数B，使用目的地总线服务器的设备证书对随机数A和随机数B签名，并将随机数B和签名值返回给本地总线服务器；
d) 本地总线服务器验证目的地总线服务器的证书和签名后，将随机数B和需传输的业务数据进行组合，使用本地总线服务器的设备证书对组合数据进行签名，将业务数据和签名值发送到目的地总线服务器；
e) 目的地总线服务器验证本地总线服务器的设备证书及签名后，保存本地总线服务器传输的业务数据，并返回成功信息；
f) 至此，从本地总线服务器到目的地总线服务器数据文件传输结束。

传输总线系统数字证书签名验签业务处理流程如图A.5所示。

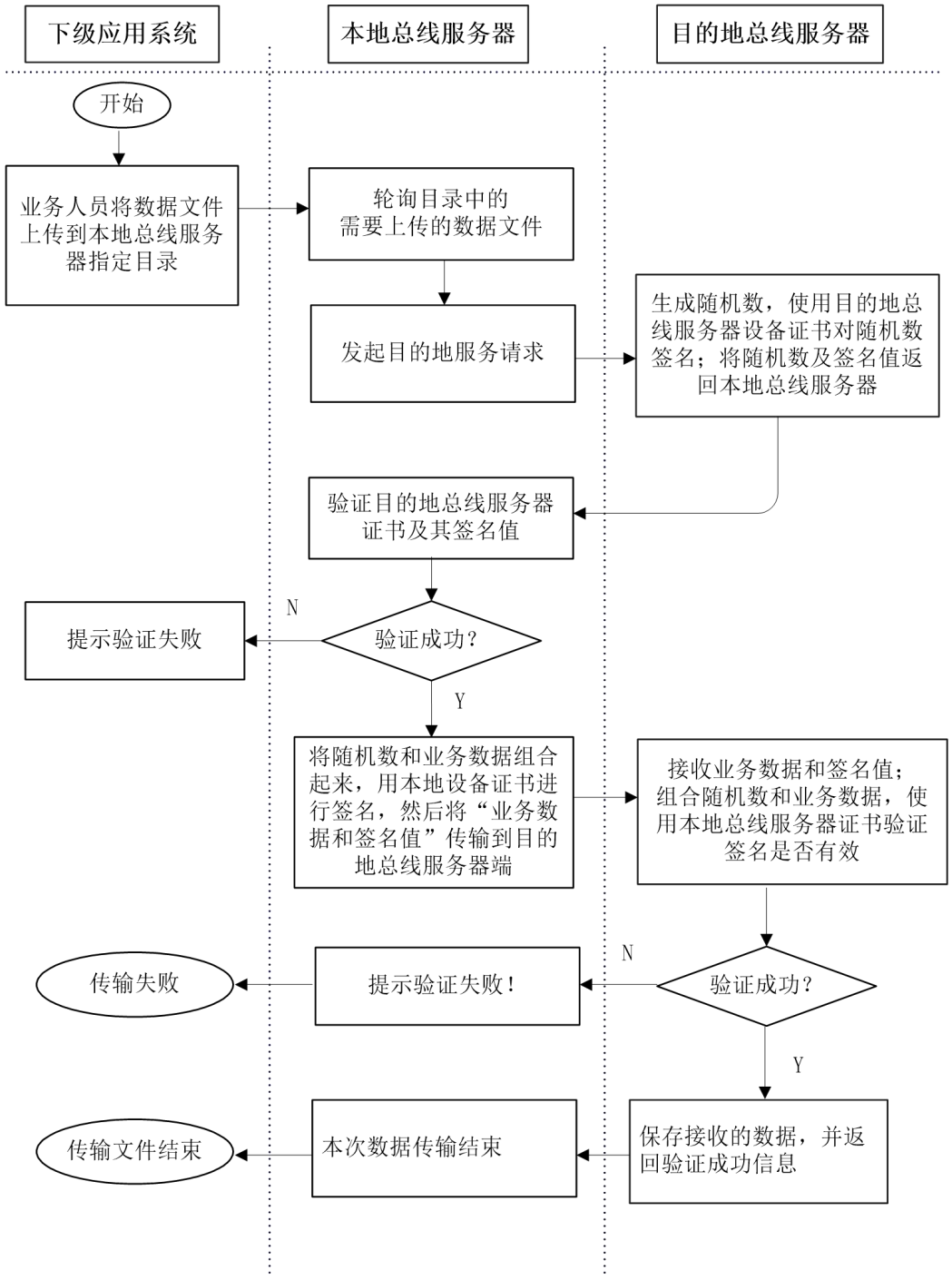

图 A.5 金保工程传输总线系统证书签名验签流程图

A.2密码应用服务接口调用

A.2.1 应用集成部署模型

应用系统通过调用密码应用服务接口实现身份认证、数据加解密、数字签名、时间戳、电子印章、责任认定等服务功能。典型的应用集成部署模型如图 A.6 所示。

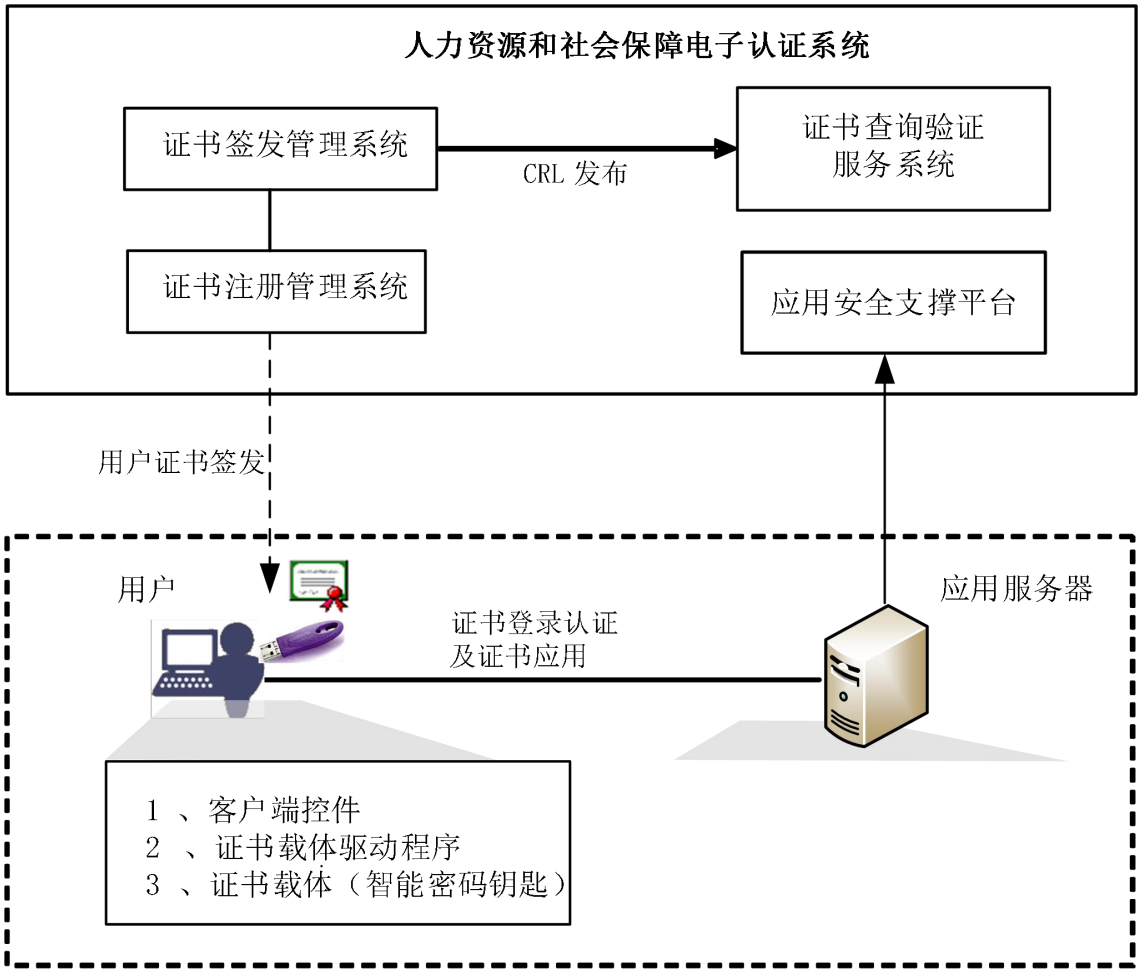

图 A.6 证书应用集成部署模型

根据图A.6所示，基于证书的应用系统涉及三个方面：人力资源社会保障电子认证系统、应用服务器和客户端认证类软件及证书载体等。

人力资源社会保障电子认证系统负责为用户签发用户数字证书。

应用服务器端通过密码应用服务接口调用密码应用服务平台实现身份认证、签名、验签等操作。客户端用户需要安装证书应用软件客户端控件及证书载体的驱动程序。在日常登录和使用时应将证书载体（智能密码钥匙）插入客户端的电脑中。

客户端和服务器端通过双方的数字证书进行双向身份认证，在数据传输过程中，双方可通过数字证书进行签名、验证、加密、解密等安全操作，实现数据完整性、操作的不可抵赖性和数据保密性。

密码应用服务接口调用流程说明：

1.应用接入

所有调用密码应用服务接口的应用系统，需要在密码应用服务平台申请接入；密码应用服务平台审核通过后根据接入申请为其分配资源权限，以及下述信息：

- “syscode”：系统代码；
- “sysauthcode”：系统授权码，由应用负责其安全存储，用于通讯认证过程中的 hmac 计算；
- “secretcode”：一个计算密钥，二进制数据，长度为 32 字节。

2.应用系统拿到上述信息后，根据 message_header 的要求组织 header 头信息，每次调用的业务均需要带有上面的校验信息。

3.完成业务流程处理后，根据附录 B.2.2.1 报文结构要求对 message_header 组织响应 header 头信息，最终返回给服务调用方。

4.所有应用系统应安全保存 sysauthcode 和 secretcode，不得泄露。

A.2.2 数字证书身份认证应用示例

用户在登录业务系统前，首先应完成数字证书的绑定，即将数字证书的唯一标识与该用户在系统中的账户（或用户名）进行绑定，实现数字证书与系统用户及其权限的一一对应。数字证书绑定的操作流程见A.1.2.1。

证书用户完成数字证书绑定后，即可使用数字证书登录业务系统，证书登录认证处理流程如下：

a) 用户将数字证书载体（智能密码钥匙）插入计算机后，输入业务系统URL地址，访问系统；
b) 服务器端程序调用B.2.3.2.7随机数获取接口，产生一个随机数，并保存在会话（Session）中；
c) 服务器端将随机数传递给客户端；
d) 用户输入证书pin码，由数字证书载体（智能密码钥匙）验证证书pin码的正确性；
e) 验证证书pin成功后，使用智能密码钥匙对随机数签名；
f) 客户端将用户证书和签名值提交到服务器端；
g) 服务器端程序接收到上述数据后，通过密码应用服务接口调用密码应用服务平台验证用户证书和签名值的有效性。其中，验证证书的有效性包括：证书有效期、信任源以及是否被吊销；
h) 解析证书中的唯一标识，根据唯一标识获取用户权限，显示用户权限对应的操作页面。

数字证书身份认证处理流程如图A.7所示。

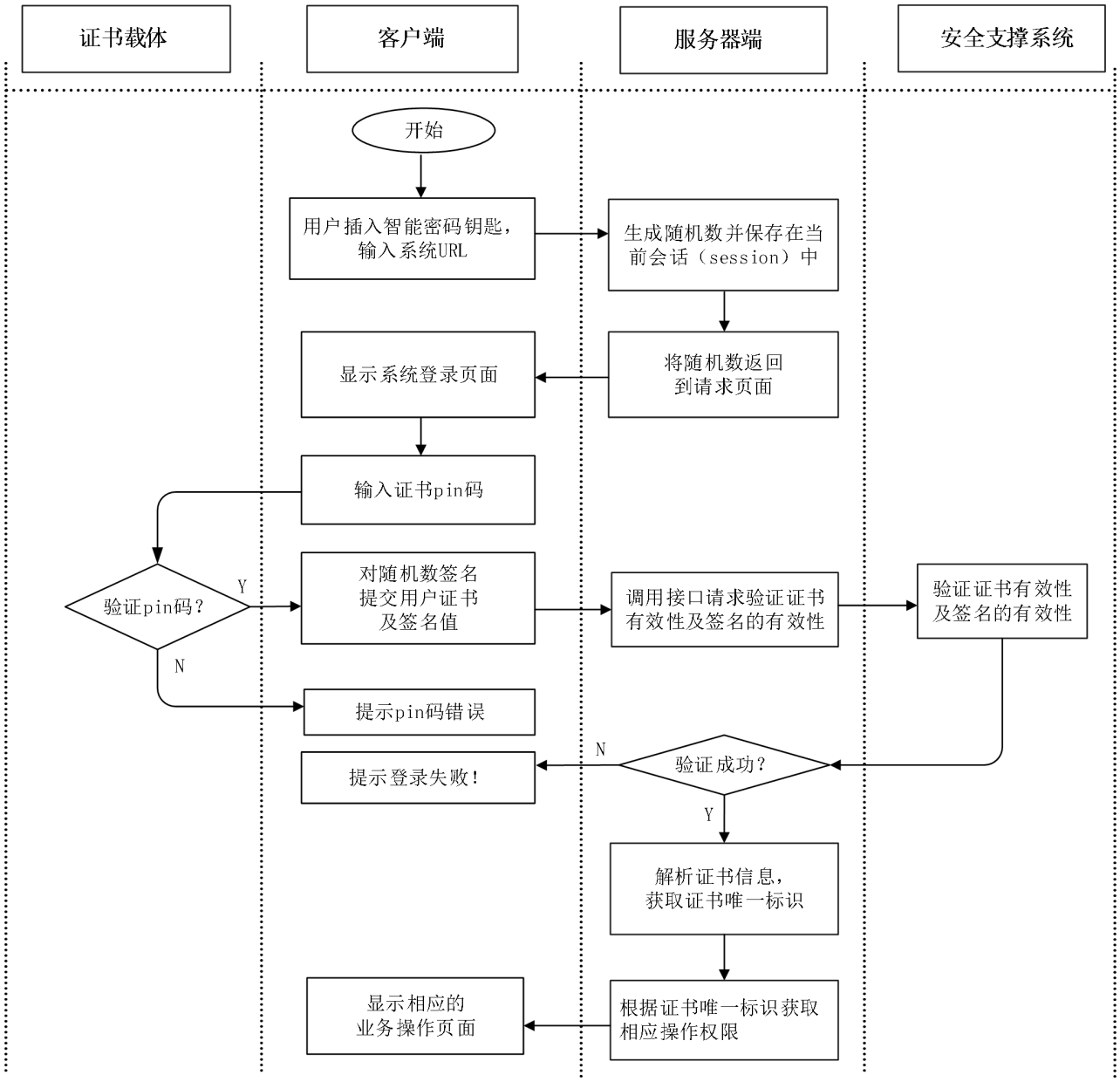

图 A.7 数字证书身份认证流程图

附 录 B
（规范性）
数字证书客户端与服务端应用接口技术要求

B.1客户端应用接口

a) 获取接口版本信息 SOF_GetVersion

原型： BSTR SOF_GetVersion()
描述： 获取控件的版本号。
参数： 无

返回值：	非空	成功
	空	失败

b) 设置签名算法 SOF_SetSignMethod

原型： long SOF_SetSignMethod（long SignMethod）
描述： 设置接口在签名运算时使用的签名算法，如 SM3、SHA1 算法等。

参数：	SignMethod[in]	签名算法标识
返回值：	0	成功
	其他	失败，详见附录 C

c) 获得当前签名算法 SOF_GetSignMethod

原型： long SOF_GetSignMethod()
描述： 获得控件签名使用的签名算法。
参数： 无

返回值：	非 0	当前接口使用的签名算法的预定义值
	0	没有设置签名算法

d) 设置加密算法 SOF_SetEncryptMethod

原型： long SOF_SetEncryptMethod（long EncryptMethod）
描述： 设置组件进行数据加解密时使用的对称算法，SM1、SM4 算法等。

参数：	EncryptMethod[in]	对称加解密算法标识
返回值：	0	成功
	其他	失败，详见附录 C

e) 获得加密算法 SOF_GetEncryptMethod

原型： long SOF_GetEncryptMethod()
描述： 获得控件使用的对称加解密算法。
参数： 无

返回值：	非 0	当前控件使用的加密算法的预定义值
	0	没有设置加密算法

f) 获得证书列表 SOF_GetUserList

原型： BSTR SOF_GetUserList()
描述： 取得当前已安装证书的用户列表。
参数： 无

返回值：	非空	用户列表字符串数据格式为：用户名 1‖ContainerName1&&& 用户名 2‖ ContainerNam21&&&…
	空	失败

备注： 根据证书应用的策略不同得到不同的证书列表。在证书列表中，用户名代表证书的 CN 项内容；ContainerName 代表证书和密钥对应的容器名，也

可以是代表证书实体身份的唯一的号码，通过 ContainerName 可以找到唯一的加密证书、签名证书，并使用对应的密钥。

g) 导出用户签名证书 SOF_ExportUserCert

原型： BSTR SOF_ExportUserCert（BSTR ContainerName）
描述： 根据证书容器名，获取 base64 编码的证书字符串。
参数： ContainerName[in] 证书容器名
返回值： 非空 Base64 编码的证书字符串
空 失败
备注： 默认导出签名证书，无签名证书时导出加密证书。

h) 校验证书口令 SOF_Login

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

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

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

j) 修改证书口令 SOF_ChangePassWd

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

k) 导出用户加密证书 SOF_ExportExChangeUserCert

原型： BSTR SOF_ExportExChangeUserCert（BSTR ContainerName）
描述： 根据证书容器名，获取 base64 编码的加密（交换）证书字符串。
参数： ContainerName[in] 证书容器名
返回值： 非空 Base64 编码的证书字符串
空 失败

l) 获得证书信息 SOF_GetCertInfo

原型： BSTR SOF_GetCertInfo（BSTR Base64EncodeCert,short Type）
描述： 获取证书内指定类型的信息。
参数： base64EncodeCert[in] Base64 编码的证书
Type[in] 获取信息的类型
返回值： 非空 证书内指定类型的信息
空 失败

m) 获得证书扩展信息 SOF_GetCertInfoByOid

原型： BSTR SOF_GetCertInfoByOid（BSTR Base64EncodeCert, BSTR Oid）
描述： 根据 OID 获取证书私有扩展项信息。
参数： base64EncodeCert[in] Base64 编码的证书
Oid[in] 私有扩展对象 ID

返回值： 非空 私有扩展对象 OID 对应的信息
空 失败

n) 获得设备信息 SOF_GetDeviceInfo

原型： BSTR SOF_GetDeviceInfo（BSTR ContainerName,long Type）
描述： 根据容器和类型代码获得设备信息。
参数： ContainerName[in] 证书容器名
Type[in] 信息类别
返回值： 非空 Type 对应的设备信息
空 失败

o) 验证证书有效性 SOF_ValidateCert

原型： long SOF_ValidateCert（BSTR Base64EncodeCert）
描述： 验证证书有效性。
参数： base64EncodeCert[in] Base64 编码的证书
返回值： 0 验证成功
其他 -1 证书不被信任
-2 超过有效期范围
-3 证书已作废
-4 证书已冻结
-5 证书未生效
-6 其他错误

备注： 基本的证书验证策略应包括：
a） 验证 CA 信任列表，各层都要进行签名和有效期验证；
b） 各层证书的有效期；
c） 各层证书的吊销状态。在特殊情况下（如：网络条件不允许），证书的吊销状态可采取灵活方式，由应用系统各层内部维护一个吊销列表，在证书登录认证时应用该吊销列表。验证证书有效性也可采取代理验证方式。

p) 数字签名 SOF_SignData

原型： BSTR SOF_SignData（BSTR ContainerName,BSTR InData）
描述： 对字符串数据进行数字签名，签名结果为数据类型 A
参数： ContainerName[in] 证书容器名
InData[in] 签名原文
返回值： 非空 Base64 编码的签名结果
空 失败

q) 验证签名 SOF_VerifySignedData

原型： BOOL VerifySignedData（BSTR Base64EncodeCert,BSTR InData,
BSTR SignValue）
描述： 验证数字签名，签名值为数据类型 A。
参数： base64EncodeCert[in] Base64 编码的签名者证书
InData[in] 签名原文
SignValue[in] Base64 编码的数据类型 A 签名值
返回值： TRUE 成功
FALSE 失败

r) 文件签名 SOF_SignFile

原型： BSTR SOF_SignFile（BSTR ContainerName,BSTR InFile）
描述： 根据文件路径，对指定文件中的数据进行数字签名，签名结果为数据类型 A。
参数： ContainerName[in] 证书容器名
InFile[in] 原文文件路径，包含文件名
返回值： 非空 Base64 编码的签名结果

空　　失败

s) 验证文件签名 SOF_VerifySignedFile

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

描述：验证文件的数字签名，签名值为数据类型 A。

参数：base64EncodeCert[in]　Base64 编码的签名者证书

InFile[in]　全路径文件名称

SignValue[in]　Base64 编码的数据类型 A 签名值

返回值：TRUE　成功

FALSE　失败

t) 加密数据 SOF_EncryptData

原型：BSTR SOF_EncryptData（BSTR Base64EncodeCert,BSTR Indata）

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

参数：base64EncodeCert[in]　Base64 编码的加密用的数据接收方数字证书

Indata[in]　待加密的明文

返回值：非空　加密后的数据类型 B 的密文，采用 Base64 编码

空　失败

u) 解密数据 SOF_DecryptData

原型：BSTR SOF_DecryptData（BSTR ContainerName, BSTR Indata）

描述：使用证书对应的私钥解密数字信封，密文数据格式为数据类型 B。

参数：ContainerName[in]　证书容器名

Indata[in]　待解密的 Base64 编码的数据类型 B 的密文

返回值：非空　解密后的明文

空　失败

v) 文件加密 SOF_EncryptFile

原型：BOOL SOF_EncryptFile（BSTR Base64EncodeCert,BSTR InFile, BSTR OutFile）

描述：使用证书加密文件，得到数据类型 B 的数字信封文件。

参数：base64EncodeCert[in]　Base64 编码的加密证书

InFile[in]　待加密的全路径文件名称

OutFile[in]　待生成的密文文件全路径名称

返回值：TRUE　成功

FALSE　失败

w) 文件解密 SOF_DecryptFile

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

描述：使用证书对应的私钥解密数据类型 B 的数字信封文件

参数：ContainerName[in]　证书容器名

InFile[in]　待解密的全路径文件名称

OutFile[in]　明文文件保存路径

返回值：TRUE　成功

FALSE　失败

x) 消息签名 SOF_SignMessage

原型：BSTR SOF_SignMessage（short flag, BSTR ContainerName, BSTR InData）

描述：对字符串数据进行数字签名，签名结果的格式为数据类型 B。

参数： flag[in] 是否带原文的标识，1：不带原文；0：带原文。
ContainerName[in] 证书容器名
InData[in] 签名明文
返回值： 非空 Base64 编码的签名结果
空 失败
备注： 签名结果包含：原文（可选）＋签名者证书＋签名值。

y) 验证消息签名 SOF_ VerifySignedMessage
原型： BOOL SOF_ VerifySignedMessage（BSTR MessageData, BSTR InData）
描述： 验证数字签名，签名值为数据类型 B。
参数： MessageData[in] Base64 编码的消息签名数据
InData[in] 原文。如果签名结果带原文，则本参数为空。
返回值： TRUE 成功
FALSE 失败

z) 解析消息签名 SOF_GetInfoFromSignedMessage
原型： BSTR SOF_GetInfoFromSignedMessage（BSTR MessageData, short type）
描述： 解析签名包内的信息，包括：原文、签名值、签名证书等信息。签名包为数据类型 B。
参数： MessageData[in] Base64 编码的签名包
type[in] 类型
返回值： 非空 解析出的信息
空 失败
备注： Type 为 1 时解析出原文；Type 为 2 时解析出 Base64 编码的签名者证书；Type 为 3 时解析出 Base64 编码的签名值。

aa) XML 数字签名 SOF_SignDataXML
原型： BSTR SOF_ SignDataXML（BSTR ContainerName，BSTR InData）
描述： 对 XML 数据进行数字签名，输出符合 RFC3275 的 XML 签名结果。
参数： ContainerName [in] 证书容器名
InData [in] XML 格式的签名原文
返回值： 非空 签名结果
空 失败
备注： XML 签名标准按照 RFC3275。

ab) 验证 XML 数字签名 SOF_VerifySignedDataXML
原型： BOOL SOF_ VerifySignedDataXML（BSTR InData）
描述： 验证 XML 签名。
参数： InData[in] XML 签名结果
返回值： TRUE 成功
FALSE 失败
备注： XML 签名标准按照 RFC3275。

ac) 解析 XML 签名数据 SOF_GetXMLSignatureInfo
原型： BSTR SOF_ GetXMLSignatureInfo（BSTR XMLSignedData，short type）
描述： 解析 XML 签名数据，获取签名值、XMl 原文、证书等信息。
参数： XMLSignedData [in] XML 格式的签名数据
type [in] 待解析的参数类型
返回值： 非空 根据 type 解析出各项对应的信息
空 失败
备注： type 参数意义如下：1：xml 原文；2：摘要；3：签名值；4：签名证书；5：摘要算法。XML 签名标准按照 RFC3275。

ad) 产生随机数 SOF_GenRandom

原型： BSTR SOF_GenRandom（short RandomLen）

描述： 产生指定长度的随机数。

参数： RandomLen[in] 待产生的随机数长度

返回值： 非空 Base64 编码的随机数值

空 失败

ae) 获取最新的错误信息 SOF_GetLastError

原型： long SOF_GetLastError（）

描述： 获取最新的错误代码。

参数： 无

返回值： 错误代码，详见附录 C。

B.2 服务端应用接口

B.2.1 COM 组件接口

COM 组件接口适用于应用服务器为 ASP 或 ASP.NET 以及其他用 C、C++开发的应用。

COM 组件接口包括以下接口函数：

a) 获取指定应用的实例 SOF_InitCertAppPolicy

原型： long SOF_InitCertAppPolicy(BSTR PolicyName)

描述： 根据应用策略名称设置应用符合的证书应用策略。该名称要和服务器配置文件对应。接口从配置文件中读取应用策略信息，包括使用的密钥和证书、信任的根证书、证书验证的策略、验证方式等。配置内容自行定义。

参数：	PolicyName[in]	应用策略名称
返回值：	0	成功
	其它	失败，详见附录 C

b) 设置签名算法 SOF_SetSignMethod

原型： long SOF_SetSignMethod(long signMethod)

描述： 设置 COM 组件签名运算使用的签名算法，主要使用 SM3withSM2 算法，兼容 RSA 算法但不推荐使用。

参数：	signMethod[in]	签名算法标识
返回值：	0	成功
	其他	失败，详见附录 C

c) 获得当前签名算法 SOF_GetSignMethod

原型： long SOF_GetSignMethod()

描述： 获得组件签名运算使用的签名算法主要使用 SM3withSM2 算法，兼容 RSA 算法但不推荐使用。

参数：	无	
返回值：	非 0	当前的签名算法的预定义值
	0	失败，没有设置算法

d) 设置加密算法 SOF_SetEncryptMethod

原型： long SOF_SetEncryptMethod(long EncryptMethod)

描述： 设置组件对数据加解密使用的对称算法，主要使用 SM1 和 SM4 算法，兼容 AES 算法但不推荐使用。

参数：	EncryptMethod[in]	对称密码算法标识
返回值：	0	成功
	其他	失败，详见附录 C

e) 获得加密算法 SOF_GetEncryptMethod

原型： long SOF_GetEncryptMethod()

描述： 获得组件使用的对称加解密算法，主要使用 SM1 和 SM4 算法，兼容 AES 算法但不推荐使用。

参数：	无	
返回值：	非 0	预先设置的加密算法的预定义值
	0	失败，没有设置算法

f) 获得服务器证书 SOF_GetServerCertificate

原型： BSTR SOF_GetServerCertificate(short CertUsage)

描述： 读取当前应用指定的服务器证书。

参数：	CertUsage[in]	证书用途，1：加密证书、2：签名证书
返回值：	非空	base64 编码的服务器证书

空　　失败

g) 产生随机数 SOF_GenRandom

原型：BSTR SOF_GenRandom(short RandomLen)

描述：产生指定长度的随机数。

参数：RandomLen[in]　　待产生的随机数长度（字节）

返回值：非空　　base64 编码的随机数值

空　　失败

h) 获得证书信息 SOF_GetCertInfo

原型：BSTR SOF_GetCertInfo(BSTR Base64EncodeCert,long type)

描述：根据指定类型，获取证书内的相关信息。

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

type[in]　　证书信息的类型

返回值：非空　　Type 对应的信息

空　　失败

i) 获得证书扩展信息 SOF_GetCertInfoByOid

原型：BSTR SOF_GetCertInfoByOid(BSTR Base64EncodeCert,BSTR oid)

描述：根据 OID 获取证书扩展项信息。

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

oid[in]　　私有扩展对象 ID，如“1.2.156.xxx”

返回值：非空　　证书私有扩展项信息

空　　失败

j) 验证证书有效性 SOF_ValidateCert

原型：long SOF_ValidateCert(BSTR Base64EncodeCert)

描述：根据应用的策略根据验证证书有效性。

最基本的证书验证策略应包括：

a） 验证 CA 信任列表，各层都要进行签名和有效期验证；

b） 各层证书的有效期；

c） 各层证书的吊销状态。在特殊情况下（如：网络条件不允许），证书的吊销状态可采取灵活方式，由应用系统内部维护一个吊销列表，在证书登录认证时应用该吊销列表。验证证书有效性也可采取代理验证方式。

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

返回值：0　　验证成功

其他　　-1 证书不被信任

-2 超过有效期范围

-3 证书已作废

-4 证书已冻结

-5 证书未生效

-6 其他错误

k) 数字签名 SOF_SignData

原型：BSTR SOF_SignData(BSTR InData)

描述：对字符串数据进行数字签名，返回的签名结果为数据类型 A。

参数：InData[in]　　待签名的数据原文

返回值：非空　　base64 编码的签名值

空　　失败

l) 验证签名 SOF_VerifySignedData

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

描述：验证数字签名。

参数：Base64EncodeCert[in]　　base64 编码的签名证书

	InData[in]	待验证的原文
	SignValue[in]	签名值，为 base64 编码的数据类型 A
返回值：	0	验证成功
	其他	验证失败，详见附录 C

m) 文件签名 SOF_SignFile

原型： BSTR SOF_SignFile(BSTR InFile)

描述： 对文件数字签名，得到 base64 编码的数据类型 A 的签名数据。

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

n) 验证文件签名 SOF_ VerifySignedFile

原型： long SOF_ VerifySignedFile(BSTR Base64EncodeCert, BSTR InFile, BSTR SignValue)

描述： 验证文件数字签名。

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

o) 加密数据 SOF_EncryptData

原型： BSTR SOF_EncryptData(BSTR Cert, BSTR InData)

描述： 使用证书对数据进行加密，得到数据类型 B 的密文数据。

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

p) 解密数据 SOF_DecryptData

原型： BSTR SOF_DecryptData (BSTR ContainerName,BSTR InData)

描述： 解密数据类型 B 的数字信封。

参数：	BSTR ContainerName[in]	证书容器名
	BSTR InData[in]	base64 编码的待解密密文数据
返回值：	非空	解密后的明文
	空	失败

q) 文件加密 SOF_EncryptFile

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

描述： 加密文件，得到数据类型 B 的密文文件。

参数：	BSTR Cert[in]	加密证书
	BSTR InFile[in]	待加密的明文文件路径
	BSTR OutFile[in]	密文文件保存路径，密文为数据类型 B
返回值：	0	成功
	其他	失败，详见附录 C

r) 文件解密 SOF_ DecryptFile

原型： long SOF_ DecryptFile(BSTR ContainerName, BSTR InFile,BSTR OutFile)

描述： 解密密文文件。

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

	BSTR OutFile[in]	明文文件保存路径
返回值：	0	成功
	其他	失败，详见附录 C

s) 消息签名 SOF_SignMassage

原型： BSTR SOF_SignMassage(BSTR InData)

描述： 对字符串数据进行数字签名，签名格式为数据类型 B。

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

t) 验证消息签名 SOF_VerifySignedMassage

原型： long SOF_VerifySignedMassage(BSTR SignedMassage)

描述： 验证消息签名包，签名格式为数据类型 B。

参数：	SignedMassage[in]	原文消息结构的 base64 编码的签名数据
返回值：	0	成功
	其他	失败，详见附录 C

u) 不带原文的消息签名 SOF_ SignMassageDetach

原型： BSTR SOF_ SignMassageDetach(BSTR InData)

描述： 对字符串数据进行数字签名，签名格式为不带原文的数据类型 B。

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

v) 验证不带原文的消息签名 SOF_VerifySignedMassageDetach

原型： long SOF_VerifySignedMassageDetach(BSTR InData ,BSTR SignedMassage)

描述： 验证消息签名包，签名格式为不带原文的数据类型 B。

参数：	InData[in]	原文数据
	SignedMassage[in]	不带原文消息结构的 base64 编码的签名数据
返回值：	0	成功
	其他	失败，详见附录 C

w) 解析消息签名 SOF_GetInfoFromSignedMassage

原型： BSTR SOF_GetInfoFromSignedMassage(BSTR SignedMassage,short type)

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

参数：	MassageData[in]	base64 编码的签名数据
	type[in]	类型 Type 为 1 时解析出原文；Type 为 2 时解析出 Base64 编码的签名者证书；Type 为 3 时解析出 Base64 编码的签名值。
返回值：	非空	解析出的信息
	空	失败

x) XML 数字签名 SOF_SignDataXML

原型： BSTR SOF_SignDataXML(BSTR InData)

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

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

y) 验证 XML 数字签名 SOF_VerifySignedDataXML

原型： long SOF_VerifySignedDataXML(BSTR InData)

描述： 验证 XML 格式的数字签名。
XML 签名标准为 RFC3275。

参数：	InData[in]	带签名值的 XML 数据
返回值：	0	成功
	其他	失败，详见附录 C

z) 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。XML 签名标准按照 RFC3275。

参数：	XMLSignedData[in]	XML 格式的签名数据
	type[in]	待解析的参数类型，type 参数意义如下：1：XML 原文；2：摘要；3：签名值；4：签名证书；5：摘要算法；6：签名算法。
返回值：	非空	获得的信息
	空	失败
	空	失败

aa) 获取最新的错误代码 SOF_GetLastError

原型： long SOF_GetLastError()

描述： 获取接口最新的错误代码。

参数： 无

返回值： 详见附录 C

B.2.2 Java 组件接口

Java 组件接口适用于应用服务器为 Java 环境的应用。

Java 组件接口包括以下接口函数：

a) 获取指定应用的实例 SOF_getInstance

原型： java.lang.Object SOF_getInstance(java.lang.String PolicyName)

描述： 初始化接口，通过应用别名获取实例，应用别名关联所配置的证书、密钥、信任证书链、算法类型、CRL 及证书验证策略等。用户如果有多应用需求，可同时获取多个实例对象满足不同的使用需求，不同的实例在调用方法时会有不同效果（如：不同密钥的签名，不同算法的加密，不同的证书验证策略）。

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

b) 设置签名算法 SOF_setSignMethod

原型： void SOF_setSignMethod(long signMethod)

描述： 设置 Java 组件签名运算使用的签名算法，主要使用 SM3withSM2 算法，兼容 RSA 算法但不推荐使用。

参数：	signMethod	签名算法标识
返回值：	无	

c) 获得当前签名算法 SOF_getSignMethod

原型： java.lang.long SOF_getSignMethod()

描述： 获得组件签名运算使用的签名算法，主要使用 SM3withSM2 算法，兼容 RSA 算法但不推荐使用。

参数： 无

返回值： 非 0 当前的签名算法的预定义值

0 没有设置算法

d) 设置加密算法 SOF_setEncryptMethod

原型： void SOF_setEncryptMethod(long encryptMethod)

描述： 设置组件对数据加解密使用的对称算法，主要使用 SM1 和 SM4 算法，兼容 AES 算法但不推荐使用。

参数： encryptMethod 对称密码算法标识

返回值： 无

e) 获得加密算法 SOF_getEncryptMethod

原型： java.lang.long SOF_getEncryptMethod()

描述： 获得组件使用的对称加解密算法，主要使用 SM1 和 SM4 算法，兼容 AES 算法但不推荐使用。

参数： 无

返回值： 非 0 当前控件使用的加密算法

0 没有设置算法

f) 获得服务器证书 SOF_getServerCertificate

原型： java.lang.String SOF_getServerCertificate()

描述： 读取当前应用指定的服务器证书。如果有签名证书则得到签名证书，否则得到加密证书。

参数： 无

返回值： 非空 base64 编码的服务器证书

空 失败

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

原型： java.lang.String SOF_getServerCertificateByUsage (short certUsage)

描述： 根据密钥用途，读取当前应用指定的服务器证书。

参数： certUsage 证书用途，1：加密证书、2：签名证书

返回值： 非空 base64 编码的服务器证书

空 失败

h) 产生随机数 SOF_genRandom

原型： java.lang.String SOF_genRandom(short RandomLen)

描述： 产生指定长度的随机数。

参数： RandomLen 待产生的随机数长度

返回值： 非空 base64 编码的随机数值

空 失败

i) 获得证书信息 SOF_getCertInfo

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

描述： 根据 type 解析证书内的相关信息。

参数： base64EncodeCert base64 编码的数字证书

type 获取证书信息的类型

返回值： 非空 type 对应的信息

空 失败

j) 获得证书扩展信息 SOF_getCertInfoByOid

原型： java.lang.String SOF_getCertInfoByOid(java.lang.String Base64EncodeCert, java.lang.String oid)

描述： 根据 OID 获取证书私有扩展项信息。

参数： Base64EncodeCert base64 编码的证书

oid 私有扩展对象 ID，如“1.2.156.xxx”

返回值： 非空 证书 oid 对应的值
空 失败

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

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

m) 验证签名 SOF_verifySignedData
原型： boolean SOF_verifySignedData(java.lang.String base64EncodeCert,
java.lang.String inData,
java.lang.String signValue)
描述： 验证数字签名。
参数： base64EncodeCert base64 编码的签名证书
inData 待验证的原文
signValue 签名值，为 base64 编码的数据类型 A
返回值： ture 验证成功
false 验证失败

n) 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 编码的数据类型 A 的签名数据。
参数： inFile 待签名的文件路径
返回值： 非空 base64 编码的签名数据
空 失败

o) 验证文件签名 SOF_ verifySignedFile
原型： boolean SOF_ verifySignedFile(java.lang.String base64EncodeCert,
java.lang.String inFile,
java.lang.String signValue)
描述： 验证文件数字签名。
参数： base64EncodeCert base64 编码的签名证书
inFile 待验证的原文路径
signValue 签名值，签名值为数据类型 A
返回值： ture 验证成功
false 验证失败

p) 加密数据 SOF_encryptData
原型： java.lang.String SOF_encryptData(java.lang.String Cert,
java.lang.String byte[] inData)
描述： 使用数字证书对数据进行加密，密文为数据类型 B 的数字信封格式。
参数： Cert 加密证书

	inData	待加密的明文
返回值：	非空	base64 编码格式的数据类型 B 的密文
	空	失败

q) 解密数据 SOF_decryptData

原型：	byte[] SOF_decryptData (java.lang.String ContainerName, java.lang.StringiInData)	
描述：	解密数据类型 B 的数字信封数据。	
参数：	ContainerName	解密密钥对应的证书唯一标识
	inData	base64 编码的待解密密文数据
返回值：	非空	解密后的明文
	空	失败

r) 文件加密 SOF_encryptFile

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

s) 文件解密 SOF_ decryptFile

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

t) 消息签名 SOF_signMassage

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

u) 验证消息签名 SOF_verifySignedMassage

原型：	boolean SOF_verifySignedMassage(java.lang.String SignedMassage)	
描述：	验证数字签名，签名格式为带原文数据类型 B。	
参数：	SignedMassage	base64 编码的签名数据包
返回值：	ture	成功
	false	失败

v) 不带原文的消息签名 SOF_signMassageDetach

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

w) 验证不带原文的消息签名 SOF_verifySignedMassageDetach

原型： boolean SOF_verifySignedMassageDetach(byte[].inData , java.lang.String SignedMassage)
描述： 验证签名格式为不带原文的数据类型 B 的数字签名。
参数： inData 原文数据
SignedMassage base64 编码的消息签名包
返回值： ture 成功
false 失败

x) 解析消息签名 SOF_getInfoFromSignedMassage
原型： byte[] SOF_getInfoFromSignedMassage(java.lang.String SignedMassage, short type)
描述： 解析数据类型 B 的签名包内的信息，可获得原文、签名值、签名证书等信息。
参数： SignedMassage 签名包
type type 定义为：1：原文；2：签名者证书； 3 签名值。
返回值： 非空 返回 type 对应的值
空 失败

y) XML 数字签名 SOF_signDataXML
原型： java.lang.String SOF_signDataXML(java.lang.String inData)
描述： 对 XML 数据进行数字签名，输出符合 RFC3275 的 XML 签名结果。
参数： inData XML 格式的签名原文
返回值： 非空 输出 XML 格式的签名结果
空 失败

z) 验证 XML 数字签名 SOF_verifySignedDataXML
原型： boolean SOF_verifySignedDataXML(java.lang.String XMLSignedData)
描述： 验证 XML 格式的数字签名。
XML 签名标准为 RFC3275。
参数： XMLSignedData 带签名值的 XML 数据
返回值： 0 成功
其他 失败

aa) 解析 XML 签名数据 SOF_getXMLSignatureInfo
原型： java.lang.String SOF_getXMLSignatureInfo(java.lang.String XMLSignedData,short type)
描述： 解析 XML 签名数据，获取签名值、XML 原文、证书等信息。
参数： XMLSignedData XML 格式的签名数据
type type 定义：1：XML 原文；2：摘要；3：签名值；4：签名证书；5：摘要算法；6：签名算法。
返回值： 非空 type 对应的信息
空 失败

ab) 获取最新的错误代码 SOF_getLastError
原型： java.lang.long SOF_getLastError()
描述： 获取接口最新的错误代码。
参数： 无
返回值： 详见附录 C

B.2.3 密码应用服务接口

密码应用服务接口形态为 HTTP restful，提供身份认证、数字签名、数据加解密、时间戳、电子印章、责任认定等功能。

身份认证服务接口，符合附录 B.2.3.2 的要求。
数字签名服务接口，符合附录 B.2.3.3 的要求。
数据加解密服务接口，符合附录 B.2.3.4 的要求。
电子印章服务接口，符合 LD/T 01.4-2022 要求。

B.2.3.1 报文结构

密码应用服务接口出参和入参采用 JSON 数据格式。报文由报文头（message_header）和报文体（message_content）两部分组成。

1) 请求报文头（message_header）

″message_header″:{
″syscode″:″系统代码″,
″businesstype″:″业务类型″,
″version″:″版本 1″,
″ctime″:″时间戳″//1970 年 1 月 1 日（UTC/GMT 的午夜）开始所经过的毫秒数,
″random″:″随机数″,
″hmac″:″计算‘时间戳||随机数||系统授权码||message_content’拼接信息的 SM3 hash 值，以此 hash 值为原文，再根据平台分配的 secretcode 计算 SM3-HMAC 值″
}

message_header 报文头说明：

	参数名称	类型	是否必须	描述
message_header	businesstype	String	是	业务类型字段，由各个接口确定
	syscode	String	是	系统代码，由密码应用服务平台提供
	version	String	是	默认版本 1.0，默认是 1.0
	ctime	String	是	时间戳//1970 年 1 月 1 日（UTC/GMT 的午夜）开始所经过的毫秒数，默认与服务器相差时间不超过 5 分钟，超过 5 分钟判定失效
	random	String	是	uuid 字符串，去除’-’后得到的 32 位字符串。
	hmac	String	是	由密码应用服务平台提供的系统授权码，原文组成方式为原文=时间戳||随机数||系统授权码||message_content，计算上述原文的国密 SM3 的 hash 值，此处称为 hash1。由密码应用服务平台提供的 secretcode，使用 secretcode 作为自定义密钥，对上面计算的 hash1 值使用 SM3-HMAC 算法，计算得到最终的 hmac 值。此 hmac 值采用 base64 编码的字符串进行传输。

样例：

″message_header″:{

″syscode″:″hrss_app_01″,
″businesstype″:″001″,
″version″:″1.0″,
″ctime″:″1613615186975″,
″random″:″20716e14b4c9448c9c07d92d5774b139″,
″hmac″:″P1LvCPDHTyeFbT3CGrYeoUcw86Je54s3GHBPg0LabC8=″
}

2) 响应报文头 message_header

″message_header″:{
″syscode″:″系统代码″,
″businesstype″:″业务类型″,
″version″:″版本 1″,
″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″
″errorCode″:″错误码″,
″errorInfo″:″错误信息″
}

message_header 报文头说明：

	参数名称	类型	是否必须	描述
message_header	businesstype	String	是	业务类型字段，由各个接口确定
	syscode	String	是	系统代码，由密码应用服务平台提供
	version	String	是	默认版本 1.0，默认是 1.0
	errorCode	String	是	错误码，0-代表成功 其他错误码详见附录
	errorInfo	String	是	错误信息与错误码匹配，0-代表成功 其他错误码详见附录″
	hmac	String	是	由密码应用服务平台提供的系统授权码，原文组成方式为原文=时间戳||随机数||系统授权码||message_content，时间戳和随机数由请求客户端上送获取。计算上述原文的国密 SM3 的 hash 值，此处称为 hash1。由密码应用服务平台提供的 secretcode，使用 secretcode 作为自定义密钥，对上面计算的 hash1 值使用 SM3-HMAC 算法，计算得到最终的 hmac 值。此 hmac 值采用 base64 编码的字符串进行传输。

样例：

″message_header″:{
″syscode″:″hrss_app_01″,
″businesstype″:″001″,
″version″:″1.0″,
"hmac":"y/aVYlD8Eock91Kyaoy4p79cg9MMLyRcmni8JfwvBqo=",

″errorCode″:″0″,
″errorInfo″:″成功″
}

B.2.3.2 身份认证服务接口

系统以 http 方式提供，请求参数和响应参数都以 JSON 方式提供。全部以 POST 方式提交，UTF-8 编码方式。

B. 2. 3. 2. 1 数字证书签发接口

a) 数字证书签发接口

功能简介	数字证书签发接口（DN 规则区别）（RA 接口服务，为行业电子签章系统等应用提供证书签发服务），证书有效开始时间是根据当前请求开始，截至时间根据证书模板 5 年后为止。证书密钥算法根据 P10 中的请求算法系统内部进行识别匹配。
接口说明	接口调用的传入参数和返回参数以 json 形式组成
请求 API	Post /cert/v1/cert

请求报文	{
	″message_header″:{
	″syscode″:″系统代码″,
	″businesstype″:″certSign″,
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	″version″:″版本 1″,
	″ctime″:″时间戳″//1970 年 1 月 1 日（UTC/GMT 的午夜）开始所经过的毫秒数,
	″random″:″随机数″
	},
	″message_content″:{
	″CertRequest″:″证书请求 P10″,
	″SubjectUniqueID″:″实体唯一标识″,
	″CertType″:″证书类型″,
	″Contact″:″联系方式″,
	″CertUse″:″证书用途″
	}
	}

响应报文	{
	″message_header″:{
	″version″:″版本 1″,
	″syscode″:″系统标识″,
	″businesstype″:″certSign″,
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	″errorCode″:″错误码″,
	″errorInfo″:″错误信息″
	}
	″message_content″:{
	″SignCert″:″签名证书″,
	″EncCert″:″加密证书″,
	″EnvelopedKey″:″保护的加密私钥″,
	″AdminPin″:″管理员 PIN″
	}

	}
备注	“证书请求 P10”： 此内容是 PKCS#10，通过 DER 编码后进行 base64，形成字符文本传输。 “实体唯一标识”：此内容是社保证书中扩展项，标识当前证书持证实体的唯一识别号，通过系统转换生成，以字符文本输入。 “实体唯一标识”的编码规则为： 用户编号（变长）+@+证书类型代码（1 位）+证件类型代码（2 位）+证件号码（变长） “证书类型”：3 类证书，1 人员证书，2 机构证书，3 设备证书 “签名证书”：返回签名证书，格式为 DER 编码，通过 base64 后的字符串文本； “加密证书”：返回加密证书，格式为 DER 编码，通过 base64 后的字符串文本； “保护的加密私钥”：按 GB/T 35291-2017 加密密钥对保护结构二进制格式，通过 base64 后的字符串文本。

请求参数说明

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	由接口文档定义的固定值，系统内部为了分别每一个不同业务定义
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	CertRequest	String	是	证书请求 P10
	SubjectUniqueID	String	是	实体唯一标识
	CertType	int	是	证书类型
	Contact	String	是	联系方式
	CertUse	String	是	证书用途

返回业务参数说明

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	由接口文档定义的固定值，系统内部为了分别每一个不同业务定义
	errorCode	String	是	错误码，0-代表成功 其他错误码详见附录
	errorInfo	String	是	错误说明
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	SignCert	String	是	签名证书
	EncCert	String	是	加密证书
	EnvelopedKey	String	是	保护的加密私钥
	AdminPin	String	是	管理员 PIN

b) 证书更新接口

功能简介	数字证书签发接口（DN 规则区别）（RA 接口服务，为行业电子签章系统等应用提供证书签发服务），证书有效开始时间是根据当前请求开始，截至时间根据证书模板 5 年后为止。证书密钥算法根据 P10 中的请求算法系统内部进行识别匹配。
接口说明	接口调用的传入参数和返回参数以 json 形式组成
请求 API	Post /cert/v1/cert
请求报文	{

	″message_header″:{
	″syscode″:″系统代码″,
	″businesstype″:″certUpdate″,
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	″version″:″版本 1″,
	″ctime″:″时间戳″//1970 年 1 月 1 日（UTC/GMT 的午夜）开始所经过的毫秒数,
	″random″:″随机数″
	}
	″message_content″:{
	″CertRequest″:″证书请求 P10″,
	″SubjectUniqueID″:″实体唯一标识″,
	″CertType″:″证书类型″,
	″Contact″:″联系方式″,
	″CertUse″:″证书用途″,
	″CertSN″:″更新前证书序列号″,
	″CertUpdateType″:″证书更新方式″
	}
	}

响应报文	{
	″message_header″:{
	″version″:″版本 1″,
	″syscode″:″系统标识″,
	″businesstype″:″certUpdate″,
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	″errorCode″:″错误码″,
	″errorInfo″:″错误信息″
	}
	″message_content″:{
	″SignCert″:″签名证书″,
	″EncCert″:″加密证书″,
	″EnvelopedKey″:″保护的加密私钥″,
	″AdminPin″:″管理员 PIN″
	}
	}

备注	“证书请求 P10”： 此内容是 PKCS#10，通过 DER 编码后进行 base64，形成字符文本传输。 “实体唯一标识“：此内容是社保证书中扩展项，标识当前证书持证实体的唯一识别号，通过系统转换生成，以字符文本输入。 “证书类型”： 3 类证书，1 人员证书，2 机构证书，3 设备证书 “签名证书”：返回签名证书，格式为 DER 编码，通过 base64 后的字符串文本； “加密证书”：返回加密证书，格式为 DER 编码，通过 base64 后的字符串文本； “保护的加密私钥”：按 GB/T 35291-2017 加密密钥对保护结构二进制格式，通过 base64 后的字符串文本。

请求参数说明

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	由接口文档定义的固定

				值，系统内部为了分别每一个不同业务定义
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	CertRequest	String	是	证书请求 P10
	SubjectUniqueID	String	是	实体唯一标识
	CertType	int	是	证书类型
	Contact	String	是	联系方式
	CertUse	String	是	证书用途
	CertSN	String	是	更新前证书序列号
	CertUpdateType	Int	是	″证书更新方式”，1 变更 3 续期

返回业务参数说明

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	由接口文档定义的固定值，系统内部为了分别每一个不同业务定义
	errorCode	String	是	错误码，0-代表成功 其他错误码详见附录
	errorInfo	String	是	错误说明
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	SignCert	String	是	签名证书
	EncCert	String	是	加密证书
	EnvelopedKey	String	是	保护的加密私钥
	AdminPin	String	是	管理员 PIN

c) 证书状态操作接口

功能简介	数字证书状态操作（RA 接口服务，为行业电子签章系统等应用提供证书状态更新服务）。
接口说明	接口调用的传入参数和返回参数以 json 形式组成
请求 API	Post /cert/v1/cert

请求报文	{
	″message_header″:{
	″syscode″:″系统代码″,
	″businesstype″:″certState″,
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	″version″:″版本 1″,
	″ctime″:″时间戳″//1970 年 1 月 1 日（UTC/GMT 的午夜）开始所经过的毫秒数,
	″random″:″随机数″
	},
	″message_content″:{
	″CertSN″:″证书序列号″,
	″StatuType″:″状态标识″
	}
	}

响应报文	{
	″message_header″:{
	″version″:″版本 1″,
	″syscode″:″系统标识″,
	″businesstype″:″certState ″,
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	″errorCode″:″错误码″,
	″errorInfo″:″错误信息″
	},
	″message_content″:{
	″RetCode″:″状态操作结果″
	}
	}

备注	

请求参数说明

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	由接口文档定义的固定值，系统内部为了分别每一个不同业务定义
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	CertSN	String	是	证书序列号
	StatuType	Int	是	证书状态：1=挂失，2=解挂，3 注销

返回业务参数说明

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	由接口文档定义的固定值，系统内部为了分别每一个不同业务定义
	errorCode	String	是	错误码，0-代表成功 其他错误码详见附录
	errorInfo	String	是	错误说明
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	RetCode	Int	是	状态操作结果

B. 2. 3. 2. 2 移动证书签发接口

a) 移动证书签发接口

功能简介	移动证书签发接口（DN 规则区别）（RA 接口服务，为移动证书协同签名系统等提供证书签发服务，如电子社保卡、地方参保企业）
接口说明	接口调用的传入参数和返回参数以 json 形式组成
请求 API	Post /cert/v1/mobilecert

请求报文	{
	″message_header″:{
	″syscode″:″系统代码″,

	″businesstype″:″ mobileCertSign″,
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	″version″:″版本 1″,
	″ctime″:″时间戳″//1970 年 1 月 1 日（UTC/GMT 的午夜）开始所经过的毫秒数,
	″random″:″随机数″
	},
	″message_content″:{
	″CertRequest″:″证书请求 P10″,
	″SubjectUniqueID″:″实体唯一标识″,
	″CertType″:″证书类型″
	}
	}

响应报文	{
	″message_header″:{
	″version″:″版本 1″,
	″syscode″:″系统标识″,
	″businesstype″:″mobileCertSign ″,
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	″errorCode″:″错误码″,
	″errorInfo″:″错误信息″
	}
	″message_content″:{
	″SignCert″:″签名证书″,
	″EncCert″:″加密证书″,
	″EnvelopedKey″:″保护的加密私钥″
	}
	}

备注	“证书请求 P10”： 此内容是 PKCS#10，通过 DER 编码后进行 base64，形成字符文本传输。 “实体唯一标识”：此内容是社保证书中扩展项，标识当前证书持证实体的唯一识别号，通过系统转换生成，以字符文本输入。 “签名证书”：返回签名证书，格式为 DER 编码，通过 base64 后的字符串文本； “加密证书”：返回加密证书，格式为 DER 编码，通过 base64 后的字符串文本； “保护的加密私钥”：按 GB/T 35291-2017 加密密钥对保护结构二进制格式，通过 base64 后的字符串文本。

请求参数说明

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	由接口文档定义的固定值，系统内部为了分别每一个不同业务定义
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	CertRequest	String	是	证书请求 P10
	SubjectUniqueID	String	是	实体唯一标识
	CertType	int	是	证书类型

返回业务参数说明

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	由接口文档定义的固定值，系统内部为了分别每一个不同业务定义
	errorCode	String	是	错误码，0-代表成功 其他错误码详见附录
	errorInfo	String	是	错误说明
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	SignCert	String	是	签名证书
	EncCert	String	是	加密证书
	EnvelopedKey	String	是	保护的加密私钥

b) 移动证书状态操作接口

功能简介	对移动证书状态进行操作
接口说明	接口调用的传入参数和返回参数以 json 形式组成
请求 API	Post /cert/v1/mobilecert

请求报文	{
	″message_header″:{
	″syscode″:″系统代码″,
	″businesstype″:″mobileCertState″,
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	″version″:″版本 1″,
	″ctime″:″时间戳″//1970 年 1 月 1 日（UTC/GMT 的午夜）开始所经过的毫秒数,
	″random″:″随机数″
	},
	″message_content″:{
	″CertSN″:″证书序列号″,
	″StatuType″:″状态标识″
	}
	}

响应报文	{
	″message_header″:{
	″version″:″版本 1″,
	″syscode″:″系统标识″,
	″businesstype″:″mobileCertState ″,
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	″errorCode″:″错误码″,
	″errorInfo″:″错误信息″
	},
	″message_content″:{
	″RetCode″:″状态操作结果″
	}
	}

备注	

请求参数说明

message_header	参数名称	类型	是否必须	描述

	businesstype	String	是	由接口文档定义的固定值，系统内部为了分别每一个不同业务定义
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	CertSN	String	是	证书序列号
	StatuType	Int	是	证书状态：1=挂失，2=解挂，3 注销

返回业务参数说明

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	由接口文档定义的固定值，系统内部为了分别每一个不同业务定义
	errorCode	String	是	错误码，0-代表成功 其他错误码详见附录
	errorInfo	String	是	错误说明
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	RetCode	Int	是	状态操作结果

B. 2. 3. 2. 3 三代卡证书接口

a) 三代卡证书签发接口

功能简介	三代卡证书签发接口（RA 接口服务，为三代卡提供证书签发接口，如三代卡自助服务系统）
接口说明	接口调用的传入参数和返回参数以 json 形式组成
请求 API	Post /cert/v1/cardcert

请求报文	{
	″message_header″:{
	″syscode″:″系统代码″,
	″businesstype″:″cardCertSign″,
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	″version″:″版本 1″,
	″ctime″:″时间戳″//1970 年 1 月 1 日（UTC/GMT 的午夜）开始所经过的毫秒数,
	″random″:″随机数″
	},
	″message_content″:{
	″SignPubKey″:″签名公钥″,
	″Name″:″姓名″,
	″SSNumber″:″社会保障号码″,
	″CardNum″:″卡号″,
	″AreaCode″:″发卡地行政区划代码″,
	″CertRequest″:″证书请求 P10″
	}}

响应报文	{
	″message_header″:{
	″version″:″版本 1″,
	″syscode″:″系统标识″,

	″businesstype″:″cardCertSign″,
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	″errorCode″:″错误码″,
	″errorInfo″:″错误信息″
	}
	″message_content″:{
	″SignCert″:″签名证书″,
	″EncCert″:″加密证书″,
	″EnvelopedKey″:″保护的加密私钥″,
	″MainManKey″:″主控密钥″,
	″AdminPin″:″管理员 PIN″
	}
	}

备注	“签名证书”：返回签名证书，格式为 DER 编码，通过 base64 后的字符串文本； “加密证书”：返回加密证书，格式为 DER 编码，通过 base64 后的字符串文本； “保护的加密私钥”：按 GB/T 35291-2017 加密密钥对保护结构二进制格式，通过 base64 后的字符串文本。

请求参数说明

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	由接口文档定义的固定值，系统内部为了分别每一个不同业务定义
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	SignPubKey	String	是	签名公钥
	Name	String	是	姓名
	SSNumber	String	是	社会保障号
	CardNum	String	是	卡号
	AreaCode	String	是	发卡地行政区划代码
	CertRequest	String	是	证书请求 P10

返回业务参数说明

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	由接口文档定义的固定值，系统内部为了分别每一个不同业务定义
	errorCode	String	是	错误码，0-代表成功 其他错误码详见附录
	errorInfo	String	是	错误说明
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	SignCert	String	是	签名证书
	EncCert	String	是	加密证书
	EnvelopedKey	String	是	保护的加密私钥
	MainManKey	String	是	主控密钥
	AdminPin	String	是	管理员 PIN

b) 三代卡证书查询接口

功能简介	三代卡证书查询接口
接口说明	接口调用的传入参数和返回参数以 json 形式组成
请求 API	Post /cert/v1/ cardcert

请求报文	{
	"message_header":{
	"syscode":"系统代码",
	"businesstype":"cardCertQuery ",
	"hmac":"计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值",
	"version":"版本 1",
	"ctime":"时间戳"//1970 年 1 月 1 日（UTC/GMT 的午夜）开始所经过的毫秒数,
	"random":"随机数"
	},
	"message_content":{
	"CertSN":"证书序列号",
	"Name":"姓名",
	"SSNumber":"社会保障号码",
	"CardNum":"卡号" ,
	"AreaCode":"发卡地行政区划代码"
	}
	}

响应报文	{
	"message_header":{
	"version":"版本 1",
	"syscode":"系统标识",
	"businesstype":"cardCertQuery",
	"hmac":"计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值",
	"errorCode":"错误码",
	"errorInfo":"错误信息"
	}
	"message_content":{
	"SignCert":"签名证书",
	"EncCert":"加密证书",
	"EnvelopedKey":"保护的加密私钥",
	"MainManKey":"主控密钥",
	"AdminPin":"管理员 PIN"
	}
	}

备注	“签名证书”：返回签名证书，格式为 DER 编码，通过 base64 后的字符串文本； “加密证书”：返回加密证书，格式为 DER 编码，通过 base64 后的字符串文本； “保护的加密私钥”：按 GB/T 35291-2017 加密密钥对保护结构二进制格式，通过 base64 后的字符串文本。

请求参数说明

	参数名称	类型	是否必须	描述
message_header	businesstype	String	是	由接口文档定义的固定值，系统内部为了分别每一个不同业务定义

	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	CertSN	String	是	证书序列号
	Name	String	是	姓名
	SSNumber	String	是	社会保障号
	CardNum	String	是	卡号
	AreaCode	String	是	发卡地行政区划代码

返回业务参数说明

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	由接口文档定义的固定值，系统内部为了分别每一个不同业务定义
	errorCode	String	是	错误码，0-代表成功 其他错误码详见附录
	errorInfo	String	是	错误说明
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	SignCert	String	是	签名证书
	EncCert	String	是	加密证书
	EnvelopedKey	String	是	保护的加密私钥
	MainManKey	String	是	主控密钥
	AdminPin	String	是	管理员 PIN

c) 三代卡状态操作接口

功能简介	三代卡证书状态操作接口（RA 接口服务，为三代卡提供证书签发接口，如三代卡自助服务系统）
接口说明	接口调用的传入参数和返回参数以 json 形式组成
请求 API	Post /cert/v1/cardcert

请求报文	{
	″message_header″:{
	″syscode″:″系统代码″,
	″businesstype″:″cardCertState″,
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	″version″:″版本 1″,
	″ctime″:″时间戳″//1970 年 1 月 1 日（UTC/GMT 的午夜）开始所经过的毫秒数,
	″random″:″随机数″
	}
	″message_content″:{
	″CertSN″:″证书序列号″,
	″SSNumber″:″社会保障号码″,
	″CardNum″:″卡号″,
	″AreaCode″:″发卡地行政区划代码″,
	″OperType″:″操作类型″
	}
	}

响应报文	{
	″message_header″:{
	″version″:″版本 1″,

	″syscode″:″系统标识″,
	″businesstype″:″cardCertState″,
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	″errorCode″:″错误码″,
	″errorInfo″:″错误信息″
	}
	″message_content″:{
	″Results″:″状态操作结果″
	}
	}

备注	

请求参数说明

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	由接口文档定义的固定值，系统内部为了分别每一个不同业务定义
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	CertSN	String	是	证书序列号
	SSNumber	String	是	社会保障号
	CardNum	String	是	卡号
	AreaCode	String	是	发卡地行政区划代码
	OperType	Int	是	证书状态：1=挂失，2=解挂，3 注销

返回业务参数说明

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	由接口文档定义的固定值，系统内部为了分别每一个不同业务定义
	errorCode	String	是	错误码，0-代表成功 其他错误码详见附录
	errorInfo	String	是	错误说明
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	Results	String	是	状态操作结果

B.2.3.2.4 公钥证书查询接口

功能简介	公钥证书查询接口（返回证书实体，用于安全邮件，点对点数据加密、裸签名验证）
接口说明	接口调用的传入参数和返回参数以 json 形式组成
请求 API	Post /cert/v1/ certauth

请求报文	{
	″message_header″:{
	″syscode″:″系统代码″,
	″businesstype″:″certQuery″,

	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	″version″:″版本 1″,
	″ctime″:″时间戳″//1970 年 1 月 1 日（UTC/GMT 的午夜）开始所经过的毫秒数,
	″random″:″随机数″
	}
	″message_content″:{
	″CertSN″:″证书序列号″,
	″SubjectUniqueID″:″实体唯一标识″,
	″CertDN″:″证书主题 DN 项″
	}
	}

响应报文	{
	″message_header″:{
	″version″:″版本 1″,
	″syscode″:″系统标识″,
	″businesstype″:″certQuery″,
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	″errorCode″:″错误码″,
	″errorInfo″:″错误信息″
	},
	″message_content″:{
	″SignCert″:″签名证书″
	}
	}

备注	“证书序列号”： 此内容是证书序列号，非必填项，三个参数必须包含一个。 “实体唯一标识“：是社保证书中扩展项，标识当前证书持证实体的唯一识别号，非必填项，三个参数必须包含一个。 “证书主题 DN 项”：证书 subjectDN，非必填项，三个参数必须包含一个。

请求参数说明

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	由接口文档定义的固定值，系统内部为了分别每一个不同业务定义
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	CertSN	String	是	证书序列号
	SubjectUniqueID	String	是	实体唯一标识
	CertDN	String	是	证书主题 DN 项

返回业务参数说明

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	由接口文档定义的固定值，系统内部为了分别每一个不同业务定义
	errorCode	String	是	错误码，0-代表成功 其他错误码详见附录
	errorInfo	String	是	错误说明

	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	SignCert	String	是	签名证书

B.2.3.2.5 证书状态查询接口

功能简介	证书状态查询接口（证书是否被吊销、冻结，密码系统间通过调用 LDAP/OCSP 系统服务接口，应用系统调用该 HTTP 服务）
接口说明	接口调用的传入参数和返回参数以 json 形式组成
请求 API	Post /cert/v1/certauth

请求报文	{
	"message_header":{
	"syscode":"系统代码",
	"businesstype":"certStateQuery",
	"hmac":"计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值",
	"version":"版本 1",
	"ctime":"时间戳"//1970 年 1 月 1 日（UTC/GMT 的午夜）开始所经过的毫秒数,
	"random":"随机数"
	},
	"message_content":{
	"CertSN":"证书序列号"
	}
	}

响应报文	{
	"message_header":{
	"version":"版本 1",
	"syscode":"系统标识",
	"businesstype":"certStateQuery ",
	"hmac":"计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值",
	"errorCode":"错误码",
	"errorInfo":"错误信息"
	},
	"message_content":{
	"CertStatus":"证书当前状态" ,
	"Cause":"原因"
	}
	}

备注	“证书当前状态”：0 正常，1 无效。 “原因”：非正常状态需要填写原因。

请求参数说明

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	由接口文档定义的固定值，系统内部为了分别每一个不同业务定义
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	CertSN	String	是	证书序列号

返回业务参数说明

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	由接口文档定义的固定值，系统内部为了分别每一个不同业务定义
	errorCode	String	是	错误码，0-代表成功 其他错误码详见附录
	errorInfo	String	是	错误说明
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	CertStatus	String	是	证书当前状态
	Cause	String	是	原因

B. 2. 3. 2. 6 **证书验证接口**

功能简介	验证证书是否由人力资源社会保障电子认证系统签发，状态是否正常，是否在有效期内。
接口说明	接口调用的传入参数和返回参数以 json 形式组成
请求 API	Post /cert/v1/certauth

请求报文	{
	″message_header″:{
	″syscode″:″系统代码″,
	″businesstype″:″checkCert″,
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	″version″:″版本 1″,
	″ctime″:″时间戳″//1970 年 1 月 1 日（UTC/GMT 的午夜）开始所经过的毫秒数，
	″random″:″随机数″
	}
	″message_content″:{
	″Base64edCert″:″Base64 编码证书″
	}
	}

响应报文	{
	″message_header″:{
	″version″:″版本 1″,
	″syscode″:″系统标识″,
	″businesstype″:″checkCert″,
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	″errorCode″:″错误码″,
	″errorInfo″:″错误信息″
	},
	″message_content″:{
	″VerifyResults″:″证书验证结果”
	}
	}

备注	

请求参数说明

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	由接口文档定义的固定值，系统内部为了分别每一个不同业务定义
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	Base64edCert	String	是	Base64 编码证书

返回业务参数说明

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	由接口文档定义的固定值，系统内部为了分别每一个不同业务定义
	errorCode	String	是	错误码，0-代表成功 其他错误码详见附录
	errorInfo	String	是	错误说明
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	VerifyResults	String	是	证书验证结果

B.2.3.2.7 **随机数获取接口**

功能简介	返回随机数
接口说明	接口调用的传入参数和返回参数以 json 形式组成
请求 API	POST /random/v1/generate

请求报文	{
	″message_header″:{
	″syscode″:″系统代码″,
	″businesstype″:″generateRandom″,
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	″version″:″版本 1″,
	″ctime″:″时间戳″//1970 年 1 月 1 日（UTC/GMT 的午夜）开始所经过的毫秒数,
	″random″:″随机数″
	},
	″message_content″:{
	″RadmonLen″:″随机数长度″
	}
	}

响应报文	{
	″message_header″:{
	″version″:″版本 1″,
	″syscode″:″系统标识″,
	″businesstype″:″generateRandom″,
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	″errorCode″:″错误码″,
	″errorInfo″:″错误信息″
	},
	″message_content″:{

	″Radmon″:″随机数值″ } }
备注	

请求参数说明

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	由接口文档定义的固定值，系统内部为了分别每一个不同业务定义
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	RadmonLen	Int	是	随机数长度

返回业务参数说明

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	由接口文档定义的固定值，系统内部为了分别每一个不同业务定义
	errorCode	String	是	错误码，0-代表成功 其他错误码详见附录
	errorInfo	String	是	错误说明
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	Radmon	String	是	随机数值

B. 2. 3. 2. 8 动态口令认证接口

a) 令牌绑定接口

功能简介	为第三方应用提供令牌绑定服务，将用户信息与令牌信息产生绑定关系
接口说明	接口调用的传入参数和返回参数以 json 形式组成
请求 API	POST /dynameicPassword/v1/card

请求报文	{
	″message_header″:{
	″syscode″:″系统代码″,
	″businesstype″:″bindCard″,
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	″version″:″版本 1″,
	″ctime″:″时间戳″//1970 年 1 月 1 日（UTC/GMT 的午夜）开始所经过的毫秒数，
	″random″:″随机数″
	},
	″message_content″:{
	″userId″:″用户 ID″,
	″cardSn″:″令牌序列号″,
	″mobile″:″手机号″,
	″email″:″邮箱″,
	″groupId″:″用户组编号″,
	″taticPassword″:″静态密码″
	}

	}

响应报文	{
	″message_header″:{
	″version″:″版本 1″,
	″syscode″:″系统标识″,
	″businesstype″:″bindCard″,
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	″errorCode″:″错误码″,
	″errorInfo″:″错误信息″
	},
	″message_content″:{
	}
	}

备注	

请求参数说明

message_header	参数名称	类型	是否必须	描述
	businesstype	String	Y	bindCard：令牌绑定
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	userId	String	Y	用户 ID
	cardSn	String	Y	令牌序列号
	mobile	String	N	手机号
	email	String	N	邮箱
	groupId	int	Y	用户组编号
	staticPassword	String	N	静态密码

返回业务参数说明

message_header	参数名称	类型	是否必须	描述
	businesstype	String	Y	bindCard：令牌绑定
	errorCode	String	Y	错误码，0-代表成功 其他错误码详见附录
	errorInfo	String	Y	错误说明
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述

b) 令牌注销接口

功能简介	为第三方应用提供用户令牌解绑服务，将用户信息与令牌信息解除绑定
接口说明	接口调用的传入参数和返回参数以 json 形式组成
请求 API	POST /dynameicPassword/v1/card

请求报文	{
	″message_header″:{
	″syscode″:″系统代码″,
	″businesstype″:″logoutCard″,
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,

	″version″:″版本 1″,
	″ctime″:″时间戳″//1970 年 1 月 1 日（UTC/GMT 的午夜）开始所经过的毫秒数,
	″random″:″随机数″
	},
	″message_content″:{
	″cardSn″:″令牌序列号″
	}
	}

响应报文	{
	″message_header″:{
	″version″:″版本 1″,
	″syscode″:″系统标识″,
	″businesstype″:″ logoutCard″,
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	“errorCode″:″错误码″,
	″errorInfo″:″错误信息″
	}
	″message_content″:{
	}
	}

备注	

请求参数说明

方式一：

message_header	参数名称	类型	是否必须	描述
	businesstype	String	Y	logoutCard：令牌注销
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	cardSn	String	Y	令牌序列号

方式二

message_header	参数名称	类型	是否必须	描述
	businesstype	String	Y	logoutCard：令牌注销
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	code	String	Y	响应代码：0-代表成功，其他数字代码详见响应码说明
	msg	String	Y	响应信息

返回业务参数说明

message_header	参数名称	类型	是否必须	描述
	businesstype	String	Y	logoutCard：令牌注销
	errorCode	String	Y	错误码，0-代表成功 其他错误码详见附录
	errorInfo	String	Y	错误说明
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述

c) 修改令牌绑定接口

功能简介	为第三方应用提供令牌绑定信息修改服务，将用户信息与令牌信息的绑定关系更新
接口说明	接口调用的传入参数和返回参数以 json 形式组成
请求 API	POST /dynameicPassword/v1/card

请求报文	{
	″message_header″:{
	″syscode″:″系统代码″,
	″businesstype″:″ updateBind″,
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	″version″:″版本 1″,
	″ctime″:″时间戳″//1970 年 1 月 1 日（UTC/GMT 的午夜）开始所经过的毫秒数,
	″random″:″随机数″
	},
	″message_content″:{
	″userId″:″用户 ID″,
	″cardSn″:″令牌序列号″,
	″mobile″:″手机号″,
	″email″:″邮箱″ ,
	″groupId″:″用户组编号″,
	″staticPassword″:″静态密码″,
	″preStaticPassword″:″上次的静态密码″
	}
	}

响应报文	{
	″message_header″:{
	″version″:″版本 1″,
	″syscode″:″系统标识″,
	″businesstype″:″ updateBind″,
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	″errorCode″:″错误码″,
	″errorInfo″:″错误信息″
	},
	″message_content″:{
	}
	}

备注	

请求参数说明

message_header	参数名称	类型	是否必须	描述
	businesstype	String	Y	updateBind：修改令牌绑定
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	userId	String	Y	用户 ID
	cardSn	String	Y	令牌序列号
	mobile	String	N	手机号

	email	String	N	邮箱
	groupId	int	Y	用户组编号
	staticPassword	String	N	静态密码
	preStaticPassword	String	N	上次的静态密码

返回业务参数说明

message_header	参数名称	类型	是否必须	描述
	businesstype	String	Y	updateBind：修改令牌绑定
	errorCode	String	Y	错误码，0-代表成功 其他错误码详见附录
	errorInfo	String	Y	错误说明
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述

d) 获取激活码接口

功能简介	获取多键令牌的激活码，激活令牌
接口说明	接口调用的传入参数和返回参数以 json 形式组成
请求 API	POST /dynameicPassword/v1/card

请求报文	{
	″message_header″:{
	″syscode″:″系统代码″,
	″businesstype″:″ getActivateCode″,
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	″version″:″版本 1″,
	″ctime″:″时间戳″//1970 年 1 月 1 日（UTC/GMT 的午夜）开始所经过的毫秒数,
	″random″:″随机数″
	},
	″message_content″:{
	″ cardSn″:″令牌序列号″
	}
	}

响应报文	{
	″message_header″:{
	″version″:″版本 1″,
	″syscode″:″系统标识″,
	″businesstype″:″ getActivateCode″,
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	″errorCode″:″错误码″,
	″errorInfo″:″错误信息″
	},
	″message_content″:{
	″activateCode″:″令牌激活码″
	}
	}

备注	

方式一：

message_header	参数名称	类型	是否必须	描述
	businesstype	String	Y	getActivateCode：获取激活码
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	cardSn	String	Y	令牌序列号

方式二

message_header	参数名称	类型	是否必须	描述
	businesstype	String	Y	getActivateCode：获取激活码
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	code	String	Y	响应代码：0-代表成功，其他数字代码详见响应码说明
	msg	String	Y	响应信息

返回业务参数说明

message_header	参数名称	类型	是否必须	描述
	businesstype	String	Y	getActivateCode：获取激活码
	errorCode	String	Y	错误码，0-代表成功 其他错误码详见附录
	errorInfo	String	Y	错误说明
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	activateCode	String	Y	令牌激活码

e) 令牌同步接口

功能简介	由于服务端和客户端时间存在偏差，令牌同步可以将相差不大的令牌修改时间偏差，使令牌能够验证
接口说明	接口调用的传入参数和返回参数以 json 形式组成
请求 API	POST /dynameicPassword/v1/card

请求报文	{
	″message_header″:{
	″syscode″:″系统代码
	″businesstype″:″ synchronizeCard″,
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	″version″:″版本 1″,
	″ctime″:″时间戳″//1970 年 1 月 1 日（UTC/GMT 的午夜）开始所经过的毫秒数,
	″random″:″随机数″
	},
	″message_content″:{
	″ cardSn″:″令牌序列号″, //
	″ firstPassword″:″当前时间的密码″, //
	″ secondPassword″:″下一分钟密码″
	}
	}

响应报文	{
	″message_header″:{
	″version″:″版本 1″,
	″syscode″:″系统标识″,

	″businesstype″:″ synchronizeCard″,
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	″errorCode″:″错误码″,
	″errorInfo″:″错误信息″
	},
	″message_content″:{
	}
	}

备注	

方式一：

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	synchronizeCard：令牌同步
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	cardSn	String	Y	令牌序列号
	firstPassword	String	Y	当前时间密码
	secondPassword	String	Y	下一分钟密码

方式二

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	synchronizeCard：令牌同步
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	userId	String	Y	用户 ID
	groupId	int	Y	用户组编号
	firstPassword	String	Y	当前时间密码
	secondPassword	String	Y	下一分钟密码

返回业务参数说明

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	getActivateCode：获取激活码
	errorCode	String	是	错误码，0-代表成功 其他错误码详见附录
	errorInfo	String	是	错误说明
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述

f) 令牌冻结接口

功能简介	冻结令牌，使令牌暂时无法使用。
接口说明	接口调用的传入参数和返回参数以 json 形式组成
请求 API	POST /dynameicPassword/v1/card

请求报文	{
	″message_header″:{
	″syscode″:″系统代码
	″businesstype″:″ lockCard″,
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,

	″version″:″版本 1″,
	″ctime″:″时间戳″//1970 年 1 月 1 日（UTC/GMT 的午夜）开始所经过的毫秒数,
	″random″:″随机数″
	},
	″message_content″:{
	″cardSn″:″令牌序列号″, //
	″intLockTime″:″锁定时间，单位分钟″ //
	}
	}

响应报文	{
	″message_header″:{
	″version″:″版本 1″,
	″syscode″:″系统标识″,
	″businesstype″:″ intLockTime″,
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	″errorCode″:″错误码″,
	″errorInfo″:″错误信息″
	},
	″message_content″:{
	}
	}

备注	

方式一：

message_header	参数名称	类型	是否必须	描述
	businesstype	String	Y	intLockTime：令牌冻结
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	cardSn	String	Y	令牌序列号
	intLockTime	Int	N	锁定时间单位分钟

方式二

message_header	参数名称	类型	是否必须	描述
	businesstype	String	Y	intLockTime：令牌冻结
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	userId	String	Y	用户 ID
	groupId	int	Y	用户组编号
	intLockTime	Int	N	锁定时间单位分钟

返回业务参数说明

message_header	参数名称	类型	是否必须	描述
	businesstype	String	Y	intLockTime：令牌冻结
	errorCode	String	Y	错误码，0-代表成功 其他错误码详见附录
	errorInfo	String	Y	错误说明
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述

g) 令牌解冻接口

功能简介	解冻令牌，使被冻结的令牌恢复正常使用功能。
接口说明	接口调用的传入参数和返回参数以 json 形式组成
请求 API	POST /dynameicPassword/v1/card

请求报文	{
	″message_header″:{
	″syscode″:″系统代码″,
	″businesstype″:″ unLockCard″,
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	″version″:″版本 1″,
	″ctime″:″时间戳″//1970 年 1 月 1 日（UTC/GMT 的午夜）开始所经过的毫秒数,
	″random″:″随机数″
	},
	″message_content″:{
	″ cardSn″:″令牌序列号″ //
	}
	}

响应报文	{
	″message_header″:{
	″version″:″版本 1″,
	″syscode″:″系统标识″,
	″businesstype″:″ unLockCard″,
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	″errorCode″:″错误码″,
	″errorInfo″:″错误信息″
	},
	″message_content″:{
	}
	}

备注	

方式一：

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	unLockCard：令牌解冻
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	cardSn	String	Y	令牌序列号

方式二

message_header	参数名称	类型	是否必须	描述
	businesstype	String	Y	unLockCard：令牌解冻
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	userId	String	Y	用户 ID
	groupId	int	Y	用户组编号

返回业务参数说明

message_header	参数名称	类型	是否必须	描述
	businesstype	String	Y	unLockCard：令牌解冻
	errorCode	String	Y	错误码，0-代表成功 其他错误码详见附录
	errorInfo	String	Y	错误说明
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述

h) 获取解锁码接口

功能简介	使令牌解锁。
接口说明	接口调用的传入参数和返回参数以 json 形式组成
请求 API	POST /dynameicPassword/v1/card

请求报文	{
	″message_header″:{
	″syscode″:″系统代码″,
	″businesstype″:″ getUnLockPin″,
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	″version″:″版本 1″,
	″ctime″:″时间戳″//1970 年 1 月 1 日（UTC/GMT 的午夜）开始所经过的毫秒数,
	″random″:″随机数″
	},
	″message_content″:{
	″cardSn″:″令牌序列号″, //
	″pur″:″解锁请求码″//
	}
	}

响应报文	{
	″message_header″:{
	″version″:″版本 1″,
	″syscode″:″系统标识″,
	″businesstype″:″ getUnLockPin″,
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	″errorCode″:″错误码″,
	″errorInfo″:″错误信息″
	}
	″message_content″:{
	″unLockPing″:″令牌解锁码″
	}
	}

备注	

方式一：

message_header	参数名称	类型	是否必须	描述

	businesstype	String	Y	getUnLockPin：获取解锁码
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	cardSn	String	Y	令牌序列号
	pur	String	Y	解锁请求码

方式二

message_header	参数名称	类型	是否必须	描述
	businesstype	String	Y	getUnLockPin：获取解锁码
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	userId	String	Y	用户 ID
	groupId	int	Y	用户组编号
	pur	String	Y	解锁请求码

返回业务参数说明

message_header	参数名称	类型	是否必须	描述
	businesstype	String	Y	getUnLockPin：获取解锁码
	errorCode	String	Y	错误码，0-代表成功 其他错误码详见附录
	errorInfo	String	Y	错误说明
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	unLockPing	String	Y	令牌解锁码

i) 清除失败次数接口

功能简介	当令牌的验证失败次数达到上限（日/累计），令牌将无法验证，该功能可清除失败次数，恢复令牌验证功能
接口说明	接口调用的传入参数和返回参数以 json 形式组成
请求 API	POST /dynameicPassword/v1/card

请求报文	{
	″message_header″:{
	″syscode″:″系统代码″,
	″businesstype″:″ clearErrorCounter″,
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	″version″:″版本 1″,
	″ctime″:″时间戳″//1970 年 1 月 1 日（UTC/GMT 的午夜）开始所经过的毫秒数,
	″random″:″随机数″
	}
	″message_content″:{
	″cardSn″:″令牌序列号″ //
	}
	}

响应报文	{
	″message_header″:{
	″version″:″版本 1″,

	″syscode″:″系统标识″,
	″businesstype″:″ clearErrorCounter″,
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	″errorCode″:″错误码″,
	″errorInfo″:″错误信息″
	},
	″message_content″:{
	}
	}

备注	

方式一：

message_header	参数名称	类型	是否必须	描述
	businesstype	String	Y	clearErrorCounter：清除令牌失败次数
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	cardSn	String	Y	令牌序列号

方式二

message_header	参数名称	类型	是否必须	描述
	businesstype	String	Y	clearErrorCounter：清除令牌失败次数
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	userId	String	Y	用户 ID
	groupId	int	Y	用户组编号

返回业务参数说明

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	clearErrorCounter：清除令牌失败次数
	errorCode	String	是	错误码，0-代表成功 其他错误码详见附录
	errorInfo	String	是	错误说明
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述

j) 令牌验证接口

功能简介	验证令牌的动态密码（时间/挑战）
接口说明	接口调用的传入参数和返回参数以 json 形式组成
请求 API	POST /dynameicPassword/v1/card

请求报文	{
	″message_header″:{
	″syscode″:″系统代码″,
	″businesstype″:″verifyPassword″,
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,

	″version″:″版本 1″,
	″ctime″:″时间戳″//1970 年 1 月 1 日（UTC/GMT 的午夜）开始所经过的毫秒数,
	″random″:″随机数″
	},
	″message_content″:{
	″cardSn″:″令牌序列号″, //
	″password″:″验证码″, //
	″challenge″:″挑战，非必要输入″, //
	″policyId″:″策略编号,动态口令服务器验证策略非必须输入″//
	}
	}

响应报文	{
	″message_header″:{
	″version″:″版本 1″,
	″syscode″:″系统标识″,
	″businesstype″:″ verifyPassword″,
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	″errorCode″:″错误码″,
	″errorInfo″:″错误信息″
	},
	″message_content″:{
	}
	}

备注	

方式一：

message_header	参数名称	类型	是否必须	描述
	businesstype	String	Y	verifyPassword：令牌验证
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	cardSn	String	Y	令牌序列号
	password	String	Y	验证码
	challenge	String	N	挑战,非必须输入
	policyId	int	N	策略编号,动态口令服务器验证策略非必须输入

方式二

message_header	参数名称	类型	是否必须	描述
	businesstype	String	Y	verifyPassword：令牌验证
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	userId	String	Y	用户 ID
	groupId	int	Y	用户组编号
	password	String	Y	验证码
	challenge	String	N	挑战,非必须输入
	policyId	int	N	策略编号,动态口令服务器验证策略非必须输入

返回业务参数说明

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	verifyPassword：令牌验证
	errorCode	String	是	错误码，0-代表成功 其他错误码详见附录
	errorInfo	String	是	错误说明
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述

B.2.3.3 数字签名与验证服务接口

B.2.3.3.1 移动协同签名

a) 是否启用协同签名接口

功能说明	用于查询是否启用的协同签名功能，该接口用于判断协同签名功能是否启用了。
接口说明	接口调用的传入参数和返回参数以 json 形式组成
请求 API	POST /verify/v1/collaborSign

请求报文	{
	″message_header″:{
	″syscode″:″系统代码″, //密码应用服务平台分配
	″businesstype″:″ getSignedWay″,
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	//对业务请求做签名（不是对业务数据），需要与应用协商使用
	″version″:″版本 1″,
	″ctime″:时间戳″,
	″random″:″随机数″
	},
	″message_content″:
	{
	″otpType″：″设备类型 1-ios 2-android″,
	″otpNum″：″8 位数字″
	}
	}

响应报文	{
	″message_header″:{
	″businesstype″:″ getSignedWay″,
	″syscode″:″系统标识″, //密码应用服务平台分配
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	″version″:″版本 1″,
	″errorCode″:″错误码″,
	″errorInfo″:″错误信息″
	}
	″message_content″:{
	″isCollaborative″:″是否协同签名，0：不是，1:是″,
	″description″:″各个字段描述信息″
	}
	}

备注	

请求参数说明：

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	getSignedWay：是否启用协同签名
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	otpType	String	是	设备类型，1：ios 2：android
	otpNum	String	是	8 位数字

返回业务参数说明

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	getSignedWay：是否启用协同签名
	errorCode	String	是	错误码，0-代表成功 其他错误码详见附录
	errorInfo	String	是	错误说明
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	isCollaborative	String	否	code 为 0 时返回，是否协同签名, 0：不是, 1:是
	description	String	否	code 为 0 时返回，各个字段描述信息

b) 生成服务端公私钥接口

功能说明	用于为业务接口提供生成服务端公私钥接口
接口说明	接口调用的传入参数和返回参数以 json 形式组成
请求 API	Post /verify/v1/collaborSign

请求报文	{
	"message_header":{
	"syscode":"系统代码", //密码应用服务平台分配
	"businesstype":"genenrateKey",
	"hmac":"计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值", //对业务请求做签名（不是对业务数据），需要与应用协商使用
	"version":"版本 1",
	"ctime":"时间戳",
	"random":"随机数"
	}
	"message_content":{
	"userName"："用户名",
	"imei"："移动端 IMEI",
	"isNew"："1-新生成公钥 0-不生成",
	"otpType"："设备类型 1-ios 2-android",
	"otpNum"："8 位数字"
	}
	}

响应报文	{

	″message_header″:{
	″businesstype″:″genenrateKey″,
	″syscode″:″系统标识″, //密码应用服务平台分配
	“hmac”:”计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值”,
	″version″:″版本 1″,
	″errorCode″:″错误码″,
	″errorInfo″:″错误信息″
	},
	″message_content″:{
	″keyLength″:″code 为 0 时返回，密钥长度″,
	″algorithm″:″code 为 0 时返回，算法″,
	″publicKey″:″code 为 0 时返回，公钥″,
	″description″:″code 为 0 时返回，各个字段描述信息″
	}
	}

备注	

请求参数说明：

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	genenrateKey：生成服务端公私钥
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	userName	String	是	用户名
	otpType	String	是	设备类型，1：ios 2：android
	otpNum	String	是	8 位数字
	imei	String	是	设备 IMEI 码
	isNew	String	是	1：新生成公私钥, 0：不生成

返回业务参数说明

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	getSignedWay：是否启用协同签名
	errorCode	String	是	错误码，0-代表成功 其他错误码详见附录
	errorInfo	String	是	错误说明
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	keyLength	String	否	code 为 0 时返回，密钥长度
	algorithm	String	否	code 为 0 时返回，算法
	publicKey	String	否	code 为 0 时返回，公钥
	description	String	否	code 为 0 时返回，各个字段描述信息

c) 获取签名值接口

功能说明	用于为业务接口提供获取签名值接口功能
接口说明	接口调用的传入参数和返回参数以 json 形式组成

请求 API	POST /verify/v1/collaborSign

请求报文	{
	″message_header″:{
	″syscode″:″系统代码″, //密码应用服务平台分配
	″businesstype″:″ getSignedVal″,
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	//对业务请求做签名（不是对业务数据），需要与应用协商使用
	″version″:″版本 1″,
	″ctime″:″时间戳″,
	″random″:″随机数″
	},
	″message_content″:{
	″userName″:″用户名″,
	″imei″:″设备 IMEI 码″,
	″otpType″:″设备类型，1：ios；2：android″,
	″otpNum″:″8 位数字″ ,
	″m″:″原文″ ,
	″pk″:″同态公钥″ ,
	″r1″:″椭圆曲线点 R1″ ,
	″ck″:″密文″ ,
	″session″:″会话 id″ ,
	″p1″:″客户端公钥″ ,
	″isP10″:″是否用于产生 P10 请求，0：不是；1：是″ ,
	″num″:″挑战信息″ }
	}

响应报文	{
	″message_header″:{
	″businesstype″:″getSignedVal″,
	″syscode″:″系统标识″, //密码应用服务平台分配
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	″version″:″版本 1″,
	″errorCode″:″错误码″,
	″errorInfo″:″错误信息″
	},
	″message_content″:{
	″e″:″code 为 0 时返回，摘要″,
	″r2″:″code 为 0 时返回，椭圆曲线点″,
	″ck″:″code 为 0 时返回，密文″,
	″session″:″code 为 0 时返回，会话 id″,
	″description″:″code 为 0 时返回，各个字段描述信息″}
	}

备注	

请求参数说明：

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	getSignedVal：获取签名值
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	userName	String	是	用户名

	otpType	String	是	设备类型，1：ios 2：android
	otpNum	String	是	8 位数字
	imei	String	是	设备 IMEI 码
	m	String	是	原文
	pk	String	是	同态公钥
	r1	String	是	椭圆曲线点 R1
	ck	String	是	密文
	session	String	是	会话 id
	p1	String	否	客户端公钥
	isP10	String	是	是否用于产生 P10 请求， 0：不是；1：是
	num	String	是	挑战信息

返回业务参数说明

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	getSignedVal：获取签名值
	errorCode	String	是	错误码，0-代表成 功 其他错误码详见附录
	errorInfo	String	是	错误说明
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	e	String	否	code 为 0 时返回，摘要
	r2	String	否	code 为 0 时返回，椭圆曲线点
	ck	String	否	code 为 0 时返回，密文
	session	String	否	code 为 0 时返回，会话 id
	description	String	否	code 为 0 时返回，各个字段描述信息

B.2.3.3.2 RAW 方式验签

功能说明	用于为业务接口提供获取 RAW 签名值验证接口功能。
接口说明	接口调用的传入参数和返回参数以 json 形式组成
请求 API	Post / sign /v1/raw

请求报文	{
	″message_header″:{
	″syscode″:″系统代码″, //密码应用服务平台分配
	″businesstype″:″ verifyRaw″,
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″, //对业务请求做签名（不是对业务数据），需要与应用协商使用
	″version″:″版本 1″,
	″ctime″:″时间戳″,
	″random″:″随机数″
	},
	″message_content″:{
	″signedText″:″签名结果″, //签名结果为数据类型 A
	″plainText″:″原文″,
	″tsaText″：″时间戳签名，没有时间戳时填 EMP″,

	″cert″:″用于验签名的证书″ } }

响应报文	{
	″message_header″:{
	″businesstype″:″ verifyRaw″,
	″syscode″:″系统标识″, //密码应用服务平台分配
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	″version″:″版本 1″,
	″errorCode″:″错误码″,
	″errorInfo″:″错误信息″
	},
	″message_content″:{
	}
	}

备注	

请求参数说明

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	verifyRaw： RAW 方式验证签名
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	plainText	String	是	原文
	signedText	String	是	签名结果
	cert	String	否	用于验签名的证书
	tsaText	String	否	时间戳签名，没有时间戳时填 EMP

返回业务参数说明

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	verifyRaw： RAW 方式验证签名
	errorCode	String	是	错误码，0-代表成功 其他错误码详见附录
	errorInfo	String	是	错误说明
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述

B.2.3.3.3 RAW 事后验签

功能说明	提供数据 RAW 方式事后签名
接口说明	接口调用的传入参数和返回参数以 json 形式组成
请求 API	Post / sign /v1/raw

请求报文	{
	″message_header″:{
	″syscode″:″系统代码″, //密码应用服务平台分配
	″businesstype″:″ verifyRawAfter″,
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算

	SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	//对业务请求做签名（不是对业务数据），需要与应用协商使用
	″version″:″版本 1″,
	″ctime″:″时间戳″,
	″random″:″随机数″
	},
	″message_content″:{
	″signedText″:″签名结果″,//签名结果为数据类型 A
	″plainText″:″原文″,
	″cert″:″用于验签名的证书″}
	}

响应报文	{
	″message_header″:{
	″businesstype″:″verifyRawAfter″,
	″syscode″:″系统标识″, //密码应用服务平台分配
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	″version″:″版本 1″,
	″errorCode″:″错误码″,
	″errorInfo″:″错误信息″
	},
	″message_content″:{
	}
	}

备注	

请求参数说明

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	verifyRawAfter：RAW 方式事后验签名
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	plaintext	String	是	原文
	signedText	String	是	签名结果
	cert	String	否	用于验签名的证书

返回业务参数说明

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	verifyRawAfter：RAW 方式事后验签名
	errorCode	String	是	错误码，0-代表成功 其他错误码详见附录
	errorInfo	String	是	错误说明
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	无	无	无	无

B.2.3.3.4 RAW 签名

功能说明	用于为业务接口提供获取 RAW 签名接口功能。
接口说明	接口调用的传入参数和返回参数以 json 形式组成

请求 API	Post /sign/v1/raw

请求报文	{
	″message_header″:{
	″syscode″:″系统代码″, //密码应用服务平台分配
	″businesstype″:″ signRaw″,
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	//对业务请求做签名（不是对业务数据），需要与应用协商使用
	″version″:″版本 1″,
	″ctime″:″时间戳″,
	″random″:″随机数″
	},
	″message_content″:{
	″plainText″：″原文″,
	″subject″：″用于签名的私钥对应的公钥证书的主题″,
	″digestAlg″：″摘要算法″,
	″useTsa″：″0-不请求 TSA 服务 1-请求 TSA 服务，空表示不请求″
	}
	}

响应报文	{
	″message_header″:{
	″businesstype″:″ signRaw″,
	″syscode″:″系统标识″, //密码应用服务平台分配
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	″version″:″版本 1″,
	″errorCode″:″错误码″,
	″errorInfo″:″错误信息″
	},
	″message_content″:{
	″signedData″:″ RAW 签名后的数据 base64 编码″//返回的签名结果为数据类型 A″
	}
	}

备注	

请求参数说明

message_header	参数名称	类型	是 否必须	描述
	businesstype	String	是	signRaw：RAW 方式签名
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是 否必须	描述
	plainText	String	是	原文
	subject	String	是	用于签名的私钥对应的公钥证书的主题
	digestAlg	String	否	摘要算法。 RSA 支持:SHA256, SHA384, SHA512 国密支持:SHA256, SM3 没有值时会采用服务器配置的默认签名算法
	useTsa	String	否	0-不请求 TSA 服务 1-请求 TSA 服务，

				空表示不请求

返回业务参数说明

message_header	参数名称	类型	是 否必须	描述
	businesstype	String	是	signRaw：RAW 方式签名
	errorCode	String	是	错误码，0-代表成功 其他错误码详见附录
	errorInfo	String	是	错误说明
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是 否必须	描述
	signedData	String	是	raw 签名后的数据 base64 编码

B.2.3.3.5 Attached 方式验签

功能说明	用于为业务接口提供获取 ATTACH 签名值验证接口功能
接口说明	接口调用的传入参数和返回参数以 json 形式组成
请求 API	Post / sign/v1/attach

请求报文	{
	″message_header″:{
	″syscode″:″系统代码″, //密码应用服务平台分配
	″businesstype″:″ verifyAttach″,
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″, //对业务请求做签名（不是对业务数据），需要与应用协商使用
	″version″:″版本 1″,
	″ctime″:″时间戳″,
	″random″:″随机数″
	},
	″message_content″:{
	″tsaText″：″时间戳签名，没有时间戳时填 EMP″,
	″signedText″：″签名结果″,//签名结果为数据类型 B
	″needCert″：“0-标明不需要返回用于验证签名的公钥证书，1-标明需要返回用于验证签名的公钥证书，空：表示不需要”}
	}

响应报文	{
	″message_header″:{
	″businesstype″:″ verifyAttach″,
	″syscode″:″系统标识″, //密码应用服务平台分配
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	″version″:″版本 1″,
	″errorCode″:″错误码″,
	″errorInfo″:″错误信息″
	},
	″message_content″:{
	″signCert″:″用于验证签名的公钥证书″
	}
	}

备注	

请求参数说明

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	verifyAttach：验证 attach 方式签名
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	plainText	String	是	原文
	subject	String	是	用于签名的私钥对应的公钥证书的主题
	digestAlg	String	否	摘要算法。 RSA 支持：SHA256, SHA384, SHA512 国密支持：SHA256, SM3
	useTsa	String	否	0-不请求 TSA 服务 1-请求 TSA 服务， 空表示不请求

返回业务参数说明

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	verifyAttach：验证 attach 方式签名
	errorCode	String	是	错误码，0-代表成功 其他错误码详见附录
	errorInfo	String	是	错误说明
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	signCert	String	否	用于验证签名的公钥证书

B.2.3.3.6 Attached 事后验签

功能说明	用于为业务提供数据 Attached 方式事后验签服务功能
接口说明	接口调用的传入参数和返回参数以 json 形式组成
请求 API	Post /sign/v1/attach

请求报文	{
	″message_header″:{
	″syscode″:″ signAttachAfter″, //密码应用服务平台分配
	″businesstype″:″verifyAttachAfter″,
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″, //对业务请求做签名（不是对业务数据），需要与应用协商使用
	″version″:″版本 1″,
	″ctime″:″时间戳″,
	″random″:″随机数″
	},
	″message_content″:{
	″signedText″:″签名结果″,//签名结果为数据类型 B
	″needCert″ :″ 0-标明不需要返回用于验证签名的公钥证书，1-标明需要返回用于验证签名的公钥证书，空：表示不需要″}

	}

响应报文	{
	″message_header″:{
	″businesstype″:″verifyAttachAfter″,
	″syscode″:″系统标识″, //密码应用服务平台分配
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	″version″:″版本 1″,
	″errorCode″:″错误码″,
	″errorInfo″:″错误信息″
	},
	″message_content″:{
	}
	}

备注	

请求参数说明

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	verifyAttachAfter ： Attached 方式事后验签
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	signedText	String	是	签名结果
	needCert	String	否	0-标明不需要返回用于验证签名的公钥证书 1-标明需要返回用于验证签名的公钥证书 空：表示不需要

返回业务参数说明

message_header	参数名称	类型	是 否必须	描述
	businesstype	String	是	verifyAttachAfter：Attached 方式事后验签
	errorCode	String	是	错误码，0-代表成功 其他错误码详见附录
	errorInfo	String	是	错误说明
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是 否必须	描述
	signCert	String	否	用于验证签名的公钥证书

B.2.3.3.7 Attached 方式签名

功能说明	各个接入应用提供数据 Attached 签名服务接口，返回的签名结果为数据类型 B。
接口说明	接口调用的传入参数和返回参数以 json 形式组成
请求 API	Post / sign/v1/attach

请求报文	{
	″message_header″:{
	″syscode″:″系统代码″, //密码应用服务平台分配

	″businesstype″:″signAttach″,
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,//对业务请求做签名（不是对业务数据），需要与应用协商使用
	″version″:″版本 1″,
	″ctime″:″时间戳″,
	″random″:″随机数″
	},
	″message_content″:{
	″plainText″：″原文″,
	″subject″：″用于签名的私钥对应的公钥证书的主题″,
	″digestAlg″：″摘要算法″,
	″useTsa″：″0-不请求 TSA 服务，1-请求 TSA 服务，空表示不请求″
	}
	}

响应报文	{
	″message_header″:{
	″businesstype″:″signAttach″,
	″syscode″:″系统标识″，//密码应用服务平台分配
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	″version″:″版本 1″,
	″errorCode″:″错误码″,
	″errorInfo″:″错误信息″
	},
	″message_content″:{
	″signedData″:″ Attache 签名后的数据″//返回的签名结果为数据类型 B
	}
	}

备注	

请求参数说明

message_header	参数名称	类型	是 否必须	描述
	businesstype	String	是	signAttach：attach 方式签名
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是 否必须	描述
	plainText	String	是	原文
	subject	String	是	用于签名的私钥对应的公钥证书的主题
	digestAlg	String	否	摘要算法。 RSA 支持:SHA256, SHA384, SHA512 国密支持:SHA256, SM3
	useTsa	String	否	0-不请求 TSA 服务 1-请求 TSA 服务，空表示不请求

返回业务参数说明

message_header	参数名称	类型	是 否必须	描述
	businesstype	String	是	signAttach：attach 方式签名
	errorCode	String	是	错误码，0-代表成功 其他错误码详见

				附录
	errorInfo	String	是	错误说明
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是 否必须	描述
	signedData	String	是	Attache 签名后的数据

B.2.3.3.8 Detached 方式验签

功能说明	提供数据 Detached 验签服务
接口说明	接口调用的传入参数和返回参数以 json 形式组成
请求 API	Post / sign/v1/ detached

请求报文	{
	″message_header″:{
	″syscode″:″系统代码″, //密码应用服务平台分配
	″businesstype″:″verifyDetached″,
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	//对业务请求做签名（不是对业务数据），需要与应用协商使用
	″version″:″版本 1″,
	″ctime″:″时间戳″,
	″random″:″随机数″
	},
	″message_content″:{
	″plainText″：″原文″,
	″signedText″：″签名结果″,//签名结果为数据类型 B
	″needCert″：″0-标明不需要返回用于验证签名的公钥证书，1-标明需要返回用于验证签名的公钥证书，空：表示不需要″,
	″tsaText″:″时间戳签名，没有时间戳时填 EMP 或留空″}
	}

响应报文	{
	″message_header″:{
	″businesstype″:″verifyDetached″,
	″syscode″:″系统标识″, //密码应用服务平台分配
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	″version″:″版本 1″,
	″errorCode″:″错误码″,
	″errorInfo″:″错误信息″
	},
	″message_content″:{
	″signCert″:″用于验证签名的公钥证书″
	}
	}

备注	

请求参数说明

message_header	参数名称	类型	是 否必须	描述
	businesstype	String	是	verifyDetached：Detached 方式验签
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			

message_content	参数名称	类型	是否必须	描述
	tsaText	String	否	时间戳签名，没有时间戳时填 EMP 或留空
	plainText	String	是	原文
	signedText	String	是	签名结果
	needCert	String	否	0-标明不需要返回用于验证签名的公钥证书 1-标明需要返回用于验证签名的公钥证书 空：表示不需要

返回业务参数说明

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	verifyDetached：Detached 方式验签
	errorCode	String	是	错误码，0-代表成功 其他错误码详见附录
	errorInfo	String	是	错误说明
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	signCert	String	否	用于验证签名的公钥证书

B.2.3.3.9 Detached 事后验签

功能说明	提供数据 Detached 方式事后验签
接口说明	接口调用的传入参数和返回参数以 json 形式组成
请求 API	Post / sign/v1/ detached

请求报文	{
	″message_header″:{
	″syscode″:″系统代码″, //密码应用服务平台分配
	″businesstype″:″verifyDetachedAfter″,
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	//对业务请求做签名（不是对业务数据），需要与应用协商使用
	″version″:″版本 1″,
	″ctime″:″时间戳″,
	″random″:″随机数″
	},
	″message_content″:{
	″plainText″：″原文″,
	″signedText″：″签名结果″,//签名结果为数据类型 B
	″needCert″：″0-标明不需要返回用于验证签名的公钥证书，1-标明需要返回用于验证签名的公钥证书，空：表示不需要″
	}

响应报文	{
	″message_header″:{
	″businesstype″:″verifyDetachedAfter″,
	″syscode″:″系统标识″, //密码应用服务平台分配
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,

	″version″:″版本 1″,
	″errorCode″:″错误码″,
	″errorInfo″:″错误信息″
	},
	″message_content″:{
	″signCert″:″用于验证签名的公钥证书″
	}
	}

备注	

请求参数说明

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	verifyDetachedAfter：Detached 方式事后验签
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	plainText	String	是	原文
	signedText	String	是	签名结果
	needCert	String	否	0-标明不需要返回用于验证签名的公钥证书 1-标明需要返回用于验证签名的公钥证书 空：表示不需要

返回业务参数说明

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	verifyDetachedAfter：Detached 方式事后验签
	errorCode	String	是	错误码，0-代表成功 其他错误码详见附录
	errorInfo	String	是	错误说明
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	signCert	String	否	用于验证签名的公钥证书

B.2.3.3.10 Dettach 签名

功能说明	提供数据 Detached 签名，返回的签名结果为数据类型 B。
接口说明	接口调用的传入参数和返回参数以 json 形式组成
请求 API	Post / sign/v1/ detached

请求报文	{
	″message_header″:{
	″syscode″:″系统代码″, //密码应用服务平台分配
	″businesstype″:″signDetached″,
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	//对业务请求做签名（不是对业务数据），需要与应用协商使用

	″version″:″版本 1″,
	″ctime″:″时间戳″,
	″random″:″随机数″
	},
	″message_content″:{
	″plainText″:″原文″,
	″subject″:″用于签名的私钥对应的公钥证书的主题″,
	″digestAlg″:″摘要算法″,
	″useTsa″:″0-不请求 TSA 服务，1-请求 TSA 服务，空表示不请求″}
	}

响应报文	{
	″message_header″:{
	″businesstype″:″signDetached″,
	″syscode″:″系统标识″, //密码应用服务平台分配
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	″version″:″版本 1″,
	″errorCode″:″错误码″,
	″errorInfo″:″错误信息″
	},
	″message_content″:{
	″signedData″:″ Detached 签名后的数据″//返回的签名结果为数据类型 B
	}
	}

备注	

请求参数说明

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	signDetached：Detached 方式签名
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	plainText	String	是	原文
	subject	String	是	用于签名的私钥对应的公钥证书的主题
	digestAlg	String	是	摘要算法。 RSA 支持: SHA256, SHA384, SHA512 国密支持:SHA256, SM3
	useTsa	String	否	0-不请求 TSA 服务 1-请求 TSA 服务， 空表示不请求

返回业务参数说明

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	signDetached：Detached 方式签名
	errorCode	String	是	错误码，0-代表成功 其他错误码详见附录
	errorInfo	String	是	错误说明
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			

message_content	参数名称	类型	是 否必须	描述
	signedData	String	是	Detached 签名后的数据

B.2.3.3.11 XML 数字封皮签名

功能说明	用于为业务接口提供 XML 数字封皮签名接口功能
接口说明	接口调用的传入参数和返回参数以 json 形式组成
请求 API	Post /sign/v1/xmlSign

请求报文	{
	″message_header″:{
	″syscode″:″系统代码″, //密码应用服务平台分配
	″businesstype″:″signXmlEnveloped″,
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″, //对业务请求做签名（不是对业务数据），需要与应用协商使用
	″version″:″版本 1″,
	″ctime″:″时间戳″,
	″random″:″随机数″
	},
	″message_content″:{
	″xml″:″ XML 文档 Base64 编码的 byte 字节数组″,
	″needCanonial″:″是否需要在客户端做规范化，0：不需要，1：需要，默认：不需要″,
	″signCertSubject″:″签名证书主题″,
	″sigId″:″ Signature 元素的 id(为空时默认为自动生成)″
	}
	}

响应报文	{
	″message_header″:{
	″businesstype″:″signXmlEnveloped″,
	″syscode″:″系统标识″, //密码应用服务平台分配
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	″version″:″版本 1″,
	″errorCode″:″错误码″,
	″errorInfo″:″错误信息″
	},
	″message_content″:{
	″xmlSignText″:″ Base64 编码包含签名信息的 xml 的字节数组″,
	″sigId″:″签名对应的 Signature 元素的 Id″
	}
	}

备注	

请求参数说明

message_header	参数名称	类型	是 否必须	描述
	businesstype	String	是	signXmlEnveloped：XML 数字签名
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是 否	描述

			必须	
	xml	String	是	XML 文档 Base64 编码的 byte 字节数组
	needCanonial	String	否	是否需要在客户端做规范化，0：不需要，1：需要，默认：不需要
	signCertSubject	String	是	签名证书主题
	sigId	String	否	Signature 元素的 id(为空时默认为自动生成)

返回业务参数说明

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	ssignXmlEnveloped：XML 数字签名
	errorCode	String	是	错误码，0-代表成功 其他错误码详见附录
	errorInfo	String	是	错误说明
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	xmlSignText	String	是	Base64 编码包含签名信息的 xml 的字节数组
	sigId	String	是	签名对应的 Signature 元素的 Id

B.2.3.3.12 XML 数字封内签名

功能说明	用于为业务接口提供 XML 数字封内签名接口功能
接口说明	接口调用的传入参数和返回参数以 json 形式组成
请求 API	Post / sign/v1/xmlSign

请求报文	{
	"message_header":{
	"syscode":"系统代码", //密码应用服务平台分配
	"businesstype":"signXmlEnveloping",
	"hmac":"计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值",
	//对业务请求做签名（不是对业务数据），需要与应用协商使用
	"version":"版本 1",
	"ctime":"时间戳",
	"random":"随机数"
	},
	"message_content":{
	"xml":" XML 文档 Base64 编码的 byte 字节数组",
	"needCanonial":"是否需要在客户端做规范化，0：不需要，1：需要，默认：不需要",
	"signCertSubject":"签名证书主题",
	"sigId":" Signature 元素的 id(为空时默认为自动生成)"}
	}

响应报文	{
	"message_header":{
	"businesstype":"signXmlEnveloping",
	"syscode":"系统标识", //密码应用服务平台分配

	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	″version″:″版本 1″,
	″errorCode″:″错误码″,
	″errorInfo″:″错误信息″
	},
	″message_content″:{
	″SignXML″:″加签名的 XML 文档″,
	″sigID″:″Signature 元素的 Id″}
	}

备注	

请求参数说明

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	signXmlEnveloping：封皮内数字签名
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	xml	String	是	XML 文档 Base64 编码的 byte 字节数组
	needCanonial	String	否	是否需要在客户端做规范化，0：不需要，1：需要，默认：不需要
	signCertSubject	String	是	签名证书主题
	sigId	String	否	Signature 元素的 id(为空时默认为自动生成)

返回业务参数说明

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	signXmlEnveloping：封皮内数字签名
	errorCode	String	是	错误码，0-代表成功 其他错误码详见附录
	errorInfo	String	是	错误说明
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	signedData	String	是	Attache 签名后的数据

B.2.3.3.13 XML 分离签名

功能说明	用于为业务接口提供 XML 分离签名接口功能。
接口说明	接口调用的传入参数和返回参数以 json 形式组成
请求 API	Post / sign/v1/xmlSign

请求报文	{
	″message_header″:{
	″syscode″:″系统代码″, //密码应用服务平台分配
	″businesstype″:″signXmlDetached″,
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	//对业务请求做签名（不是对业务数据），需要与应用协商使用

	″version″:″版本 1″
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值
	″ctime″:″时间戳″,
	″random″:″随机数″
	},
	″message_content″:{
	″xml″:″ XML 文档 Base64 编码的 byte 字节数组″,
	″needCanonial″:″是否需要在客户端做规范化，0：不需要，1：需要，默认：不需要″
	″signCertSubject″:″签名证书主题″,
	″sigId″:″ Signature 元素的 id(为空时默认为自动生成)″,
	″tbsId″:″ xml 文档内作为签名原文的元素的 ID，多个以逗号分隔″}
	}

响应报文	{
	″message_header″:{
	″businesstype″:″signXmlDetached″,
	″syscode″:″系统标识″，//密码应用服务平台分配
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	″version″:″版本 1″,
	″errorCode″:″错误码″,
	″errorInfo″:″错误信息″
	},
	″message_content″:{
	″SignXML″：″加签名的 XML 文档″,
	″sigID″：″Signature 元素的 Id″}
	}

备注	

请求参数说明

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	signXmlDetached：XML 分离签名
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	xml	String	是	XML 文档 Base64 编码的 byte 字节数组
	needCanonial	String	否	是否需要在客户端做规范化，0：不需要，1：需要，默认：不需要
	signCertSubject	String	是	签名证书主题
	sigId	String	否	Signature 元素的 id(为空时默认为自动生成)
	tbsId	String	是	xml 文档内作为签名原文的元素的 ID，多个以逗号分隔

返回业务参数说明

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	signXmlDetached：XML 分离签名
	errorCode	String	是	错误码，0-代表成功 其他错误码详见

				附录
	errorInfo	String	是	错误说明
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	xmlSignText	String	是	Base64 编码包含签名信息的 xml 的字节数组
	sigId	String	是	签名对应的 Signature 元素的 Id

B.2.3.3.14 XML 数字签名验证

功能说明	用于为业务接口提供 XML 数字签名验签接口功能
接口说明	接口调用的传入参数和返回参数以 json 形式组成
请求 API	Post /sign/v1/xmlSign

请求报文	{
	"message_header":{
	"syscode":"系统代码", //密码应用服务平台分配
	"businesstype":"verifyXml",
	"hmac":"计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值",
	//对业务请求做签名（不是对业务数据），需要与应用协商使用
	"version":"版本 1",
	"ctime":"时间戳",
	"random":"随机数"
	},
	"message_content":{
	"xml":" XML 文档 Base64 编码的 byte 字节数组",
	"needCanonial":"是否需要在客户端做规范化 0：不需要，1：需要，默认：不需要",
	"needCert":"是否返回签名证书信息 0：不需要，1：需要，默认：不需要"
	}
	}

响应报文	{
	"message_header":{
	"businesstype":"verifyXml",
	"syscode":"系统标识", //密码应用服务平台分配
	"hmac":"计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值",
	"version":"版本 1",
	"errorCode":"错误码",
	"errorInfo":"错误信息"
	},
	"message_content":[{
	" signCert":"签名证书",
	"signName":"签名区域名称",
	"verifyPass":"是否验签通过"}]
	}

备注	

请求参数说明

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	verifyXml：XML 验证签名
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	xml	String	是	XML 文档 Base64 编码的 byte 字节数组
	needCanonial	String	否	是否需要在客户端做规范化 0：不需要，1：需要，默认：不需要
	needCert	String	否	是否返回签名证书信息 0：不需要，1：需要，默认：不需要

返回业务参数说明

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	verifyXml：XML 验证签名
	errorCode	String	是	错误码，0-代表成功 其他错误码详见附录
	errorInfo	String	是	错误说明
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	signCert	String	否	签名证书
	signName	String	是	签名区域名称
	verifyPass	String	是	是否验签通过

B.2.3.3.15 XML 事后签名验证

功能说明	对给定的 xml 文档做普通事后验签名
接口说明	接口调用的传入参数和返回参数以 json 形式组成
请求 API	Post /sign/v1/xmlSign

请求报文	{
	″message_header″:{
	″syscode″:″系统代码″, //密码应用服务平台分配
	″businesstype″:″verifyXmlAfter″,
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	//对业务请求做签名（不是对业务数据），需要与应用协商使用
	″version″:″版本 1″,
	″ctime″:″时间戳″,
	″random″:″随机数″
	},
	″message_content″:{
	″xml″:″ XML 文档 Base64 编码的 byte 字节数组″,
	″needCanonial″:″是否需要在客户端做规范化 0：不需要，1：需要，默认：不需要″,
	″needCert″:″是否返回签名证书信息 0：不需要，1：需要，默认：不需要″
	}
	}

响应报文	{
	"message_header":{
	"businesstype":"verifyXmlAfter",
	"syscode":"系统标识", //密码应用服务平台分配
	"hmac":"计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值",
	"version":"版本 1",
	"errorCode":"错误码",
	"errorInfo":"错误信息"
	},
	"message_content":[{
	" signCert":"签名证书",
	"signName":"签名区域名称",
	"verifyPass":"是否验签通过"}]
	}

备注	

请求参数说明

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	verifyXmlAfter：XML 事后验证签名
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	xml	String	是	XML 文档 Base64 编码的 byte 字节数组
	needCanonial	String	否	是否需要在客户端做规范化 0：不需要，1：需要，默认：不需要
	needCert	String	否	是否返回签名证书信息 0：不需要，1：需要，默认：不需要

返回业务参数说明

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	verifyXmlAfter：XML 事后验证签名
	errorCode	String	是	错误码，0-代表成功 其他错误码详见附录
	errorInfo	String	是	错误说明
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	signCert	String	否	签名证书
	signName	String	是	签名区域名称
	verifyPass	String	是	是否验签通过

B.2.3.4 数据加解密服务接口

B.2.3.4.1 生成对称密钥接口

功能说明	用于为业务接口提供对称密钥对生成功能
接口说明	接口调用的传入参数和返回参数以 json 形式组成

请求 API	POST / key/v1/ keyOp

请求报文	{
	"message_header":{
	"syscode":"系统代码", //密码应用服务平台分配
	"businesstype":"generateSymmetricKey",
	"hmac":"计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值", //对业务请求做签名（不是对业务数据），需要与应用协商使用
	"version":"版本 1",
	"ctime":"时间戳",
	"random":"随机数"
	},
	"message_content":{
	"keyFlag":"在 LMK 下加密的密钥密文标识",
	"keyType":"密钥类型(相应密钥类型的密钥类型号)",
	"storeKeyIndex":"密钥存储索引,范围为：1<=storeKeyIndex<2048；其他值表示不存储",
	"storeKeyLabel":"密钥存储标签:仅当 1<=storeKeyIndex<2048 时生效，用于在密钥内部存储时标记密钥的标签说明，1-16 个 ASCII 字符"
	}
	}

响应报文	{
	"message_header":{
	"businesstype":"generateSymmetricKey",
	"syscode":"系统标识", //密码应用服务平台分配
	"hmac":"计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值",
	"version":"版本 1",
	"errorCode":"错误码",
	"errorInfo":"错误信息"
	},
	"message_content":{
	"lmkEncryptKey":"密钥在 LMK 下加密的密文",
	"lmkEncryptKeyCheck":"密钥校验值"
	}
	}

请求参数说明：

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	generateSymmetricKey 加注并锁定签名
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	keyType	String	是	密钥类型(相应密钥类型的密钥类型号),支持密钥类型代码和密钥类型名称两种格式，如“MDK”密钥可以传入“109”或“MDK”两种格式。 000/ZMK/KEK 001/ZPK 002/PVK/TPK/TMK

				003/TAK 008/ZAK 009/BDK 00A/ZEK/DEK 00B/TEK 011/KMC 109/MDK 10C/HMAC 209/MK-SMI 309/MK-SMC 402/CVK 409/MK-DAK 509/MK-DN
	keyFlag	String	是	在 LMK 下加密的密钥密文标识 :keyFlag Z - 单倍长 DES 密钥 X - 双倍长 3DES 密钥 Y - 三倍长 3DES 密钥 U - 双倍长的 3DES 算法密钥 T - 三倍长的 3DES 算法密钥 R - 16 字节 SM4 密钥 P - 16 字节 SM1 密钥 L - 16 字节 AES 密钥 M - AES-192 算法密钥 N - AES-256 算法密钥
	storeKeyIndex	String	是	密钥存储索引，范围为：1<=storeKeyIndex<2048；其他值表示不存储。
	storeKeyLabel	String	是	密钥存储标签：仅当 1<=storeKeyIndex<2048 时生效，用于在密钥内部存储时标记密钥的标签说明，1-16 个 ASCII 字符

响应业务参数说明：

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	generateSymmetricKey 加注并锁定签名
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	lmkEncryptKey	String	是	密钥在 LMK 下加密的密文
	lmkEncryptKeyCheck	String	是	密钥校验值

B.2.3.4.2 **更新对称密钥接口**

功能说明	用于为业务接口提供更新对称密钥信息功能
接口说明	接口调用的传入参数和返回参数以 json 形式组成
请求 API	POST /key/v1/keyOp
请求报文	{

	″message_header″:{
	″syscode″:″系统代码″, //密码应用服务平台分配
	″businesstype″:″updateSymmetricKey″,
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″, //对业务请求做签名（不是对业务数据），需要与应用协商使用
	″version″:″版本 1″,
	″ctime″:″时间戳″,
	″random″:″随机数″
	},
	″message_content″:{
	″srcKeyType″:″ 源密钥类型，支持密钥类型编号和密钥类型名称两种类型″,
	″srcKey″:″源密钥索引或密文″,
	″subKeyType″:″子密钥类型，支持密钥类型代码和密钥类型名称两种格式″,
	″subKeyAlgFlag″:″子密钥标识，支持标识 X、U、P、R、L、N″,
	″disperAlgType″:″分散算法模式″,
	″disperFactor″:″分散因子，长度为 n*m*2 的 HEX 字符串，n 为分散级数取值为 1-8，每级分散因子长度为 m*2H 即 m 字节。当 disperAlgType 为 0 时，m 为 8H 当 disperAlgType 为 1 或 2 或 3 时，m 为 16H；当 disperAlgType 为 4 时，分散因子为 n*16H″,
	″iv″:″初始向量，当 disperAlgType 为 5 时有效″,
	″storeKeyIndex″:″密钥存储索引, 范围 1=<storeKeyIndex<2048，其他值表示不存储″,
	″storeKeyLabel″:″ 密钥存储标签, 长度 1-16 当 1=<storeKeyIndex<2048 范围内时生效″
	}
	}

响应报文	{
	″message_header″:{
	″businesstype″:″updateSymmetricKey″,
	″syscode″:″系统标识″, //密码应用服务平台分配
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	″version″:″版本 1″,
	″errorCode″:″错误码″,
	″errorInfo″:″错误信息″
	},
	″message_content″:{
	″lmkEncryptKeyCheck″:″密钥校验值″,
	″lmkEncryptKey″:″密钥在 LMK 下加密的密文″
	}

备注	

请求参数说明：

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	updateSymmetricKey 加注并锁定签名
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	srcKeyType	String	是	srcKeyType - 源密钥类型，支持密钥类型编号和密钥类型名称两种类型，支持密钥类型代码

				和密钥类型名称两种格式，如“MDK”密钥可以传入“109”或“MDK”两种格式。支持类型： 000/ZMK/KEK； 109/MDK； 002/PVK/TPK/TMK； 007/EDK； 209/MK-SMI； 309/MK-SMC； 409/MK-DAK； 509/MK-DN； 00A/ZEK/DEK； 011/KMC； 008/ZAK；
	srcKey	String	是	srcKey - 源密钥索引或密文
	subKeyType	String	是	subKeyType - 子密钥类型，支持密钥类型代码和密钥类型名称两种格式，如“MDK”密钥可以传入“109”或“MDK”两种格式。支持类型： 000 - ZMK/KEK； 109 - MDK； 209 - MK-SMI； 309 - MK-SMC； 409 - MK-DAK； 509 - MK-DN； 00A - ZEK/DEK； 011 - KMC； 008 - ZAK；
	subKeyAlgFlag	String	是	子密钥标识，支持标识X、U、P、R、L、N
	disperAlgType	String	是	disperAlgType - 分散算法模式 0 - PBOC子密钥分散算法，8字节分散因子D，使用源密钥对16字节[D||D的非]采用源密钥的算法标识进行ECB模式加密 1 - ECB模式加密16字节分散因子 2 - ECB模式加密16字节分散因子，并复制扩展为32字节长度密钥（仅限

				subKeyAlgFlag 为 N） 3 – CBC 模式加密 16 字节分散因子 4 – ECB 模式加密分散因子，分散因子必须为 8 字节的倍数，且至少 16 字节。截取加密结果的前后各 8 字节作为子密钥（仅限 subKeyAlgFlag 为 X 或 U） 5 – CBC 模式加密分散因子，分散因子必须为 16 字节的倍数。截取加密结果的最后 16 字节作为子密钥（仅限源密钥和 subKeyAlgFlag 为 L）
	disperFactor	String	是	disperFactor – 分散因子，长度为 n*m*2 的 HEX 字符串，n 为分散级数取值为 1-8，每级分散因子长度为 m*2H 即 m 字节 当 disperAlgType 为 0 时，m 为 8H 当 disperAlgType 为 1 或 2 或 3 时，m 为 16H 当 disperAlgType 为 4 时，分散因子为 n*16H
	iv	String	是	IV – 初始向量，当 disperAlgType 为 5 时有效
	storeKeyIndex	String	是	密钥存储索引，范围 1=<storeKeyIndex<2048，其他值表示不存储。
	storeKeyLabel	String	是	密钥存储标签, 长度 1-16 当 1=<storeKeyIndex<2048 范围内时生效

响应业务参数说明：

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	updateSymmetricKey 加注并锁定签名
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	lmkEncryptKey	String	是	密钥在 LMK 下加密的密文
	lmkEncryptKeyCheck	String	是	密钥校验值

B.2.3.4.3 导出对称密钥接口

功能说明	用于为业务接口提供导出对称密钥接口功能
接口说明	接口调用的传入参数和返回参数以 json 形式组成
请求 API	POST /key/v1/keyOp

请求报文	{
	″message_header″:{
	″syscode″:″系统代码″, //密码应用服务平台分配
	″businesstype″:″exportSymmetricKey″,
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″, //对业务请求做签名（不是对业务数据），需要与应用协商使用
	″version″:″版本 1″,
	″ctime″:″时间戳″,
	″random″:″随机数″
	},
	″message_content″:{
	"zmkKey":″ZMK 密钥索引或密文″,
	"targetKeyType":″被导出的密钥类型，支持密钥类型名称和密钥类型编码两种格式″,
	"targetKey":″被导出的密钥索引或密文″,
	"targetKeyFlag":″被导出的密钥标识，支持标识 X、U、P、R、L、N"
	}
	}

响应报文	{
	″message_header″:{
	″businesstype″:″exportSymmetricKey″,
	″syscode″:″系统标识″, //密码应用服务平台分配
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	″version″:″版本 1″,
	″errorCode″:″错误码″,
	″errorInfo″:″错误信息″
	},
	″message_content″:{
	″zmkEncryptKeyCheck″:″AFF984320B28236A″,
	″zmkEncryptKey″:″L33E4F430B9DDDB6969F601CA0E253CC8″
	}

备注	

请求参数说明：

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	updateSymmetricKey 加注并锁定签名
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	zmkKey	String	是	ZMK 密钥索引或密文
	targetKey	String	是	被导出的密钥索引或密文
	targetKeyFlag	String	是	密钥标识，支持标识 X、U、P、R、L、N
	targetKeyType	String	否	密钥类型，支持密钥类型代码和密钥类型名称

				两种格式。

响应业务参数说明：

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	updateSymmetricKey 加注并锁定签名
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	zmkEncryptKey	String	是	ZMK 下加密的密钥密文
	zmkEncryptKeyCheck	String	是	密钥校验值

message_content 响应业务参数说明：

参数名称	类型	是否必须	描述
zmkEncryptKey	String	是	ZMK 下加密的密钥密文
zmkEncryptKeyCheck	String	是	密钥校验值

B.2.3.4.4 **导入对称密钥接口**

功能说明	用于为业务接口提供对称密钥导入接口功能
接口说明	接口调用的传入参数和返回参数以 json 形式组成
请求 API	POST /key/v1/keyOp

请求报文	{
	″message_header″:{
	″syscode″:″系统代码″, //密码应用服务平台分配
	″businesstype″:″importSymmetricKey″,
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″, //对业务请求做签名（不是对业务数据），需要与应用协商使用
	″version″:″版本 1″,
	″ctime″:″时间戳″,
	″random″:″随机数″
	
	},
	″message_content″:{
	"importedKeyType":"被导入的密钥类型，支持密钥类型名称和密钥类型编码两种格式, 如″109-MDK″密钥可以传入″MDK″或″109″两种格式。",
	"zmkKey":"ZMK 密钥索引或密文",
	"importKeyCipherByZmk":"ZMK 下加密的密钥密文",
	"importedKeyAlgFlag":"子密钥标识，支持标识 X、U、P、R、L、N",
	"storeKeyIndex":"密钥存储索引, 范围为：1<=storeKeyIndex<2048;其他值表示不存储。",
	"storeKeyLabel":"密钥存储标签:仅当 1<=storeKeyIndex<2048 时生效，用于在密钥内部存储时标记密钥的标签说明，0-16 个 ASCII 字符"
	}
	}

响应报文	{
	″message_header″:{
	″businesstype″:″importSymmetricKey″,
	″syscode″:″系统标识″, //密码应用服务平台分配
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算

	SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″，
	″version″:″版本 1″，
	″errorCode″:″错误码″，
	″errorInfo″:″错误信息″
	}，
	″message_content″:{
	″lmkEncryptKeyCheck″:″密钥校验值″，
	″lmkEncryptKey″:″密钥在 LMK 下加密的密文″
	}

备注	

请求参数说明：

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	importSymmetricKey 导入对称密钥
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	zmkKey	String	是	ZMK 密钥索引或密文
	importedKeyType	String	是	importedKeyType - 被导入的密钥类型，支持密钥类型名称和密钥类型编码两种格式，如″109-MDK″密钥可以传入″MDK″或″109″两种格式。 支持类型： 000 - ZMK/KEK； 001 - ZPK； 002 - PVK/TPK/TMK； 402 - CVK； 003 - TAK； 008 - ZAK； 009 - BDK； 109 - MDK； 209 - MK-SMI； 309 - MK-SMC； 409 - MK-DAK； 509 - MK-DN； 00A - ZEK/DEK； 00B - TEK； 10C - HMAC； 011 - KMC；
	importKeyCipherByZmk	String	是	ZMK 下加密的密钥密文
	importedKeyAlgFlag	String	是	子密钥标识，支持标识 X、U、P、R、L、N
	storeKeyIndex	String	是	密钥存储索引，范围为：1<=storeKeyIndex<2048；其他值表示不存储。

	storeKeyLabel	String	是	密钥存储标签：仅当1<=storeKeyIndex<2048 时生效，用于在密钥内部存储时标记密钥的标签说明，0-16个ASCII字符

响应业务参数说明：

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	importSymmetricKey 导入对称密钥
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	lmkEncryptKey	String	是	密钥在 LMK 下加密的密文
	lmkEncryptKeyCheck	String	是	密钥校验值

B.2.3.4.5 销毁对称密钥接口

功能说明	用于为业务接口提供对称密钥销毁功能
接口说明	接口调用的传入参数和返回参数以 json 形式组成
请求 API	POST /key/v1/keyOp

请求报文	{
	″message_header″:{
	″syscode″:″系统代码″,
	″businesstype″:″destroySymmetricKey″,
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	//对业务请求做签名（不是对业务数据），需要与应用协商使用,
	″version″:″版本 1″,
	″ctime″:″时间戳″,
	″random″:″随机数″
	},
	″message_content″:{
	″keyType″:″密钥类型：指定待删除密钥的类型。00：对称密钥，01：RSA 密钥对，02：SM2 密钥对。″,
	″keyIndex″:″密钥索引号：待删除的密钥的索引号取值范围（1=<keyIndex<2048）
	″}
	}

响应报文	{
	″message_header″:{
	″businesstype″:″destroySymmetricKey″,
	″syscode″:″系统标识″, //密码应用服务平台分配,
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	″version″:″版本 1″,
	″errorCode″:″错误码″,
	″errorInfo″:″错误信息″
	},
	″message_content″:{
	}

备注	

请求参数说明：

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	destroySymmetricKey 销毁对称密钥
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	keyType	String	是	keyType - 密钥类型：指定待删除密钥的类型 00：对称密钥 01：RSA 密钥对 02：SM2 密钥对
	keyIndex	String	是	密钥索引号 ：待删除的密钥的索引号取值范围（1=<keyIndex<2048）

响应业务参数说明：

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	destroySymmetricKey 销毁对称密钥
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述

B.2.3.4.6 **获取对称密钥信息接口**

功能说明	用于为业务接口提供获取对称密钥信息接口功能
接口说明	接口调用的传入参数和返回参数以 json 形式组成
请求 API	POST / key/v1/ keyOp

请求报文	{
	″message_header″:{
	″syscode″:″系统代码″, //密码应用服务平台分配
	″businesstype″:″getInfoSymmetricKey″,
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	//对业务请求做签名（不是对业务数据），需要与应用协商使用
	″version″:″版本 1″,
	″ctime″:″时间戳″,
	″random″:″随机数″
	
	},
	″message_content″:{
	″ifRefresh″:″指定是否需要刷新密钥缓存 0-需要 1-不需要″,
	″keyIndex″:″密钥索引号 ：密钥的索引号取值范围（1=<keyIndex<2048）″}
	}

响应报文	{
	″message_header″:{
	″businesstype″:″getInfoSymmetricKey″,
	″syscode″:″系统标识″, //密码应用服务平台分配
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算

	SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	″version″:″版本 1″,
	″errorCode″:″错误码″,
	″errorInfo″:″错误信息″
	},
	″message_content″:{
	″keyAlgFlag″:″密钥标识″,
	″keyCV″:″密钥校验值，校验通过后内部存储输入全，0 则不验证，直接存储覆盖″,
	″keyLabel″:″密钥标签 用于在密钥内部存储时标记密钥的标签说明， 0-16 个 ASCII 字符″,
	″keyType″:″密钥类型″,
	″updateTime″:″更新时间″
	}

备注	

请求参数说明：

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	getInfoSymmetricKey 获取对称密钥信息
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	ifRefresh	String	是	指定是否需要刷新密钥缓存 0-需要 1-不需要
	keyIndex	String	是	密钥索引号 ：密钥的索引号取值范围（1=<keyIndex<2048）

响应业务参数说明：

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	getInfoSymmetricKey 获取对称密钥信息
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	keyAlgFlag	String	是	密钥标识
	keyCV	String	是	校验值 ：密钥校验值，校验通过后内部存储输入全 0 则不验证，直接存储覆盖
	keyLabel	String	是	密钥标签 用于在密钥内部存储时标记密钥的标签说明， 0-16 个 ASCII 字符
	keyType	String	是	密钥类型
	updateTime	String	是	更新时间

B.2.3.4.7 **生成非对称密钥接口**

功能说明	用于为业务接口提供生成非对称密钥接口功能
接口说明	接口调用的传入参数和返回参数以 json 形式组成
请求 API	POST / key/v1/ keyOp

请求报文	{
	″message_header″:{
	″syscode″:″系统代码″, //密码应用服务平台分配
	″businesstype″:″generateAsymmetricKey″,
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″, //对业务请求做签名（不是对业务数据），需要与应用协商使用
	″version″:″版本 1″,
	″ctime″:″时间戳″,
	″random″:″随机数″
	
	},
	″message_content″:{
	"algorithmID":″密钥类型 RSA/SM2",
	"keyUseWay":″RSA 密钥用途。当密钥类型是 RSA，此项有效，SM2 时无效。0-签名密钥，1-密钥管理密钥，2-不限，建议使用此项。″,
	"keyLen":″RSA 密钥模长取值 1024﹣2048",
	"exponent":″RSA 公钥指数（取值 3 或 65537）″,
	"storeIndex":″(RSA/SM2)秘钥储存索引（取值 1-64 时储存在加密机内，取值为 0 时表示不储存）″,
	"keyLabel":″(RSA/SM2)密钥存储标签，当 storeIndex 取值在 1-64 之间是生效″}
	}

响应报文	{
	″message_header″:{
	″businesstype″:″generateAsymmetricKey″,
	″syscode″:″系统标识″, //密码应用服务平台分配
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	″version″:″版本 1″,
	″errorCode″:″错误码″,
	″errorInfo″:″错误信息″
	},
	″message_content″:{
	″encryptedPrivateKey″:″密钥校验值″,
	″pubKey″:″密钥在 LMK 下加密的密文″
	}

备注	

请求参数说明：

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	generateAsymmetricKey 生成非对称密钥
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	algorithmID	String	是	密钥类型 RSA/SM2
	keyUseWay	String	是	当密钥类型是 RSA，此项有效，SM2 时无效。keyUseWay - RSA 密钥用途 0 - 签名密钥 1 - 密钥管理密钥

				2 - 不限，建议使用此项″
	keyLen	String	否	RSA 密钥模长取值 1024 - 2048 且必须为 8 的倍数。
	exponent	String	否	RSA 公钥指数（取值 3 或 65537）″
	storeIndex	String	是	(RSA/SM2)秘钥储存索引 （取值 1-64 时储存在加密机内，取值为 0 时表示不储存）
	keyLabel	String	是	(RSA/SM2)密钥存储标签 当 storeIndex 取值在 1-64 之间是生效

响应业务参数说明：

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	generateAsymmetricKey 生成非对称密钥
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	pubKey	String	是	密钥在 LMK 下加密的密文
	encryptedPrivateKey	String	是	密钥校验值

B.2.3.4.8 更新非对称密钥接口

功能说明	用于为业务接口提供更新非对称密钥接口功能
接口说明	接口调用的传入参数和返回参数以 json 形式组成
请求 API	POST /key/v1/keyOp

请求报文	{
	″message_header″:{
	″syscode″:″系统代码″, //密码应用服务平台分配
	″businesstype″:″updateAsymmetricKey″,
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″, //对业务请求做签名（不是对业务数据），需要与应用协商使用
	″version″:″版本 1″,
	″ctime″:″时间戳″,
	″random″:″随机数″
	},
	″message_content″:{
	″algorithmID″:″密钥类型 RSA/SM2",
	″keyUseWay″:″RSA 密钥用途，0-签名密钥，1-密钥管理密钥，2-不限，建议使用此项″,
	″keyLen″:″RSA 密钥模长取值 1024 - 2048",
	″exponent″:″RSA 公钥指数（取值 3 或 65537）″,
	″storeIndex″:″(RSA/SM2)秘钥储存索引（取值 1-64 时储存在加密机内，取值为 0 时表示不储存）″,

	″keyLabel″:″(RSA/SM2)密钥存储标签，当 storeIndex 取值在 1-64 之间是生效″ } }

响应报文	{
	″message_header″:{
	″businesstype″:″updateAsymmetricKey″,
	″syscode″:″系统标识″, //密码应用服务平台分配
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	″version″:″版本 1″,
	″errorCode″:″错误码″,
	″errorInfo″:″错误信息″
	},
	″message_content″:{
	″encryptedPrivateKey″:″密钥校验值″,
	″pubKey″:″密钥在 LMK 下加密的密文″
	}

备注	

请求参数说明：

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	updateAsymmetricKey 更新非对称密钥
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	algorithmID	String	是	密钥类型 RSA/SM2
	keyUseWay	String	是	keyUseWay - RSA 密钥用途″+ ″0 - 签名密钥″+ ″1 - 密钥管理密钥″+ ″2 - 不限，建议使用此项″
	keyLen	String	否	RSA 密钥模长取值 1024 - 2048 且必须为 8 的倍数。
	exponent	String	否	RSA 公钥指数（取值 3 或 65537）″
	storeIndex	String	是	(RSA/SM2)秘钥储存索引 （取值 1-64 时储存在加密机内，取值为 0 时表示不储存）
	keyLabel	String	是	(RSA/SM2)密钥存储标签 当 storeIndex 取值在 1-64 之间是生效

响应业务参数说明：

	参数名称	类型	是否必须	描述

message_header	businesstype	String	是	updateAsymmetricKey 更新非对称密钥
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	pubKey	String	是	密钥在 LMK 下加密的密文
	encryptedPrivateKey	String	是	密钥校验值

B.2.3.4.9 导出公钥接口

功能说明	用于为业务接口提供导出非对称公钥接口功能
接口说明	接口调用的传入参数和返回参数以 json 形式组成
请求 API	POST / key/v1/keyOp

请求报文	{
	″message_header″:{
	″syscode″:″系统代码″, //密码应用服务平台分配
	″businesstype″:″exportAsymmetricKey″,
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″, //对业务请求做签名（不是对业务数据），需要与应用协商使用
	″version″:″版本 1″,
	″ctime″:″时间戳″,
	″random″:″随机数″
	},
	″message_content″:{
	″algorithmID″:″密钥类型：RSA/ECC",
	″storeIndex″：″密钥存储索引值 RSA/ECC(1-64)″}
	}

响应报文	{
	″message_header″:{
	″businesstype″:″exportAsymmetricKey″,
	″syscode″:″系统标识″, //密码应用服务平台分配
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	″version″:″版本 1″,
	″errorCode″:″错误码″,
	″errorInfo″:″错误信息″
	}
	″message_content″:{
	″pubKey″:″公钥 16 进制字符串信息″
	}

备注	

请求参数说明：

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	exportAsymmetricKey 导出公钥
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	algorithmID	String	是	密钥类型：RSA/ECC RSA:RSA 密钥

				ECC:SM2 密钥
	storeIndex	String	是	密钥存储索引值 RSA/ECC(1-64)

响应业务参数说明：

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	exportAsymmetricKey 导出公钥
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	pubKey	String	是	公钥 16 进制字符串信息

B.2.3.4.10 **数据加密接口**

功能说明	用于为业务接口提供数据加密接口功能
接口说明	接口调用的传入参数和返回参数以 json 形式组成
请求 API	POST / key/v1/keyOp

请求报文	{
	″message_header″:{
	″syscode″:″系统代码″, //密码应用服务平台分配
	″businesstype″:″encData″,
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″, //对业务请求做签名（不是对业务数据），需要与应用协商使用
	″version″:″版本 1″,
	″ctime″:″时间戳″,
	″random″:″随机数″
	},
	″message_content″:{
	″encKeyType″:″加密密钥类型：指定加密密钥的类型 00：对称密钥，01：RSA 密钥、02：SM2 密钥″,
	″inData″:″输入的明文″,
	″algType″:″当 encKeyType=00 时有效：加密算法模式″,
	″keyType″:″当 encKeyType=00 时有效：keyType 用于加密数据的源密钥类型，支持密钥类型名称和密钥类型编码两种格式。″,
	″key″:″用户加密数据的密钥的索引或密文;传入参数数据类型为 int 型时，通过密钥索引调用密钥。当 encKeyType=00 时：（取值需为 1<=key<2048 之间），传入参数类型为 String 时，按 LMK 加密的密钥密文处理；当 encKeyType=01 时：取值需为 1-64 之间；当 encKeyType=02 时：取值需为 1-64 之间。″,
	″disperFactor″:″当 encKeyType=00 时有效，密钥分散因子 n 级分散因子进行串联，且每级分散因子必须为 16 个字节。″,
	″sessionType″:″会话密钥产生模式，当 encKeyType=00 时有效″,
	″sessionFactor″:″sessionType 为 1 时，该域为 8 字节（16H）；sessionType 为 2 时，该域为 16 字节（32H）；sessionType 为 5 时，该域为 16 字节（32H）″,
	″padFlag″:″PAD 填充标识，取值范围：00 - 05 或 10 - 11，当 encKeyType=00 时有效。″,
	″iv″:″对称密钥时有效：初始向量，仅当 algType 取值为 01/02/03/4 时需存在该域″
	}
	}

响应报文	{

	″message_header″:{
	″businesstype″:″encData″,
	″syscode″:″系统标识″, //密码应用服务平台分配
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	″version″:″版本 1″,
	″errorCode″:″错误码″,
	″errorInfo″:″错误信息″
	},
	″message_content″:{
	″encData″:″加密密文″
	}

备注	

请求参数说明：

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	encData 数据加密
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	encKeyType	String	是	encKeyType - 加密密钥类型：指定加密密钥的类型 00：对称密钥，01：RSA 密钥、02：SM2 密钥
	key	String	是	用户加密数据的密钥的索引或密文;传入参数数据类型为 int 型时，通过密钥索引调用密钥 当 encKeyType =00 时：（取值需为 1<=key<2048 之间） 传入参数类型为 String 时，按 LMK 加密的密钥密文处理 当 encKeyType =01 时：取值需为 1-64 之间 当 encKeyType =02 时：取值需为 1-64 之间
	algType	String	是	当 encKeyType =00 时有效：加密算法模式 0 - ECB 模式加密 1 - CBC 模式加密 2 - CFB 模式加密 3 - OFB 模式加密 4 - CRT 模式加密 （16 字节分组长度处理）
	keyType	String	是	当 encKeyType =00 时有效：keyType 用于加密数据的源密钥类型，支持密钥类型名称和密钥类型编码两种格式。例如 MDK 可以传″109″和″MDK″两种格式。 309 - MK-SMC; 00A - ZEK/DEK;

				00B - TEK;
	key	String	是	当 encKeyType =00 时有效 key - 用户加密数据的密钥的索引或密文 ;传入参数数据类型为 int 型时，通过密钥索引调用密钥（取值需为 1-2048 之间）传入参数类型为 String 时，按 LMK 加密的密钥密文处理 当 encKeyType =01 或者 02 时,密钥索引取值 1-64
	disperFactor	String	是	当 encKeyType =00 时有效，密钥分散因子 n 级分散因子进行串联，且每级分散因子必须为 16 个字节
	sessionType	String	是	当 encKeyType =00 时有效，对称密钥时有效：sessionType - 会话密钥产生模式 0 - 不产生会话密钥; 1 - ECB 模式加密 8 字节会话密钥因子，得 8 字节会话密钥; 2 - ECB 模式加密 16 字节会话密钥因子，得 16 字节会话密钥; 3 - 密钥的左右 8 字节异或，得 8 字节会话密钥; 4 - 取密钥的左 8 字节作为会话密钥; 5 - CBC 模式加密 16 字节会话密钥因子，得 16 字节会话密钥
	sessionFactor	String	是	当 encKeyType =00 时有效, sessionType 为 1 时，该域为 8 字节（16H） sessionType 为 2 时，该域为 16 字节（32H） sessionType 为 5 时，该域为 16 字节（32H）
	padFlag	String	是	当 encKeyType =00 时 padFlag - PAD 填充标识，取值范围：00 - 05 或 10 - 11 0 - PBOC 2.0 填充模式 1 - ISO/IEC 9797-1 的 PADDING 模式 2 2 - ISO/IEC 9797-1 的 PADDING 模式 1 3 - ANSI X9.23

				4 – PKCS#5 5 – NoPadding 模式 PBOC3.0 11– 左填充+ISO/IEC 9797-1 当 encKeyType =01 或者 02 时 padFlag 取值为 0 或者 1：0：不填充 1：PKCS#1 V1.5 方法（EMSA-PKCS1-v1_5）
	inData	String	是	输入的明文数据
	iv	String	是	对称密钥时有效：初始向量，仅当 algType 取值为 01/02/03/4 时需存在该域 若密钥算法为 128 分组，该域为 16 字节（32H）；若密钥算法为 64 分组，该域为 8 字节（16H）；

响应业务参数说明：

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	encData 数据加密
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	encData	String	是	加密密文

B.2.3.4.11 **数据解密接口**

功能说明	用于为业务提供数据解密功能
接口说明	接口调用的传入参数和返回参数以 json 形式组成
请求 API	POST / key/v1/ keyOp

请求报文	{
	"message_header":{
	"syscode":"系统代码", //密码应用服务平台分配
	"businesstype":"decData",
	"hmac":"计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值",//对业务请求做签名（不是对业务数据），需要与应用协商使用
	"version":"版本 1",
	"ctime":"时间戳",
	"random":"随机数"
	
	},
	"message_content":{
	"decKeyType":"解密密钥类型：指定解密密钥的类型 00：对称密钥，01：RSA 密钥、02：SM2 密钥",
	"inData":"输入的密文数据",
	"algType":"当 decKeyType=00 时有效，加密算法模式。",
	"keyType":"用于加密数据的源密钥类型，支持密钥类型名称和密钥类型编码两种格式，当 decKeyType=00 时有效。",
	"key":"用户加密数据的密钥的索引或密文",
	"disperFactor":"当 decKeyType=00 时有效，密钥分散因子 n 级分散因子进行串联，且每级分散因子必须为 16 个字节",

	″sessionType″:″会话密钥产生模式，当 decKeyType=00 时有效″, ″sessionFactor″:″会话密钥因子，当 decKeyType=00 时有效″, ″padFlag″:″当 decKeyType=01 或者 02 时有效，0：不填充 1：PKCS#1V1.5 方法（EMSA-PKCS1-v1_5）″, ″iv″:″初始向量，仅当 algType 取值为 01/02/03/4 时需存在该域″} }

响应报文	{
	″message_header″:{
	″businesstype″:″decData″,
	″syscode″:″系统标识″, //密码应用服务平台分配
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	″version″:″版本 1″,
	″errorCode″:″错误码″,
	″errorInfo″:″错误信息″
	},
	″message_content″:{
	″decData″:″解密明文″
	}

备注	

请求参数说明：

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	decData 数据解密
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	decKeyType	String	是	decKeyType- 解密密钥类型：指定解密密钥的类型 00：对称密钥，01：RSA 密钥、02：SM2 密钥
	algType	String	是	当 decKeyType=00 时有效，加密算法模式″ 0 - ECB 模式加密 1 - CBC 模式加密 2 - CFB 模式加密 3 - OFB 模式加密 4 - CRT 模式加密（16 字节分组长度处理）″
	keyType	String	是	当 decKeyType=00 时有效，用于加密数据的源密钥类型，支持密钥类型名称和密钥类型编码两种格式。例如 MDK 可以传\″109\″和\″MDK\″两种格式。 keyType - 密钥类型，支持类型包括： MK-SMC 、 ZEK 、 DEK、TEK
	key	String	是	用户加密数据的密钥的索引或密文;传入参数数据类型为 int 型时，通过密钥索引调用密钥 对称密钥：（取值需为 1<=key<2048 之间）

				传入参数类型为 String 时，按 LMK 加密的密钥密文处理 RSA 密钥：取值需为 1-64 之间 SM2 密钥：取值需为 1-64 之间
	disperFactor	String	是	当 decKeyType=00 时有效，密钥分散因子 n 级分散因子进行串联，且每级分散因子必须为 16 个字节
	sessionType	String	是	当 decKeyType=00 时有效，会话密钥产生模式， 0 - 不产生会话密钥 1 - ECB 模式加密 8 字节会话密钥因子，得 8 字节会话密钥 2 - ECB 模式加密 16 字节会话密钥因子，得 16 字节会话密钥 3 - 密钥的左右 8 字节异或，得 8 字节会话密钥 4 - 取密钥的左 8 字节作为会话密钥 5 - CBC 模式加密 16 字节会话密钥因子，得 16 字节会话密钥
	sessionFactor	String	是	当 decKeyType=00 时有效，会话密钥因子， sessionType 为 1 时，该域为 8 字节（16H） sessionType 为 2 时，该域为 16 字节（32H） sessionType 为 5 时，该域为 16 字节（32H）
	padFlag	String	是	当 decKeyType=01 或者 02 时有效，0：不填充 1：PKCS#1 V1.5 方法（EMSA-PKCS1-v1_5） 当 decKeyType=00 时有效， PAD 填充标识，取值范围：00 - 05 或 10 - 110 - PBOC 2.0 填充模式 1 - ISO/IEC 9797-1 的 PADDING 模式 2 2 - ISO/IEC 9797-1 的 PADDING 模式 1 3 - ANSI X9.23 4 - PKCS#5 5 - NoPadding 模式 PBOC3.0 11- 左填充+ISO/IEC 9797-1
	inData	String	是	输入的密文数据

	iv	String	是	初始向量，仅当 algType 取值为 01/02/03/4 时需存在该域

响应业务参数说明：

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	decData 数据解密
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	decData	String	是	解密明文

B.2.3.4.12 计算数据摘要接口

功能说明	用于为业务提供计算摘要数据功能
接口说明	接口调用的传入参数和返回参数以 json 形式组成
请求 API	POST / key/v1/ keyOp

请求报文	{
	″message_header″:{
	″syscode″:″系统代码″, //密码应用服务平台分配
	″businesstype″:″digestData″,
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,//对业务请求做签名（不是对业务数据），需要与应用协商使用
	″version″:″版本 1″,
	″ctime″:″时间戳″,
	″random″:″随机数″
	},
	″message_content″:{
	″hashAlgFlag″:″ HASH 算法标识″,
	″data″:″待计算摘要数据 长度取值范围″,
	″id″:″用户 ID，当且仅当 hashAlgFlag 取值为 20 时存在″,
	″pubKey″:″ SM2 算法公钥 当且仅当 hashAlgFlag 取值为 20 时存在 ，采取格式为对公钥的 x 和 y 进行 ASN1 格式 Der 编码（x, y）得到字符串 z，拼接 x，y 即是 z+x+y, 其中 z, x, y 均为 16 进制字符串″}
	}

响应报文	{
	″message_header″:{
	″businesstype″:″digestData″,
	″syscode″:″系统标识″, //密码应用服务平台分配
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	″version″:″版本 1″,
	″errorCode″:″错误码″,
	″errorInfo″:″错误信息″
	},
	″message_content″:{
	″digestData″:″摘要值″
	}

备注	

请求参数说明：

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	digestData 计算摘要数据
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	hashAlgFlag	String	是	HASH 算法标识 1：SHA1″ + ″ 2：MD5″ + ″ 3：ISO 10118-2″ + ″ 5：SHA-224″ + ″ 6：SHA-256″ + ″ 7：SHA-384″ + ″ 8：SHA-512″ + ″ 20：SM3-256″
	data	String	是	待计算摘要数据 长度取值范围（0-4096 字节）
	id	String	是	ID - 用户 ID 当且仅当 hashAlgFlag 取值为 20 时存在 若 ID 取值为 null，则 HSM 默认采用国密局发布的默认 ID：”1234567812345678” 且下个域 pubKey 取值为空
	pubKey	String	是	SM2 算法公钥 当且仅当 hashAlgFlag 取值为 20 时存在 ，采取格式为对公钥的 x 和 y 进行 ASN1 格式 Der 编码（x, y）得到字符串 z，拼接 x，y 即是 z+x+y, 其中 z, x, y 均为 16 进制字符串

响应业务参数说明：

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	digestData 计算数据摘要
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	digestData	String	是	摘要值

B.2.3.4.13 生成 MAC 接口

功能说明	用于为业务提供生成 MAC 功能
接口说明	接口调用的传入参数和返回参数以 json 形式组成
请求 API	POST / key/v1/ keyOp

请求报文	{
	″message_header″:{
	″syscode″:″系统代码″, //密码应用服务平台分配
	″businesstype″:″caculateMac″,
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″, //对业

	务请求做签名（不是对业务数据），需要与应用协商使用
	″version″:″版本 1″,
	″ctime″:″时间戳″,
	″random″:″随机数″
	}
	″message_content″:{
	″plain″:″明文″,
	″algType″:″对称算法″,
	″key″:″对称密钥对象″,
	″macLength″:″ MAC 结果长度″
	}
	}

响应报文	{
	″message_header″:{
	″businesstype″:″caculateMac″,
	″syscode″:″系统标识″, //密码应用服务平台分配
	″hmac″:″计算时间戳||随机数||系统授权码||message_content 拼接的信息计算 SM3 的 hash 为原文，根据平台分配的 secretcode 计算原文的 SM3-HMAC 值″,
	″version″:″版本 1″,
	″errorCode″:″错误码″,
	″errorInfo″:″错误信息″
	},
	″message_content″:{
	″macData″:″计算出的 MAC 值″
	}

备注	

请求参数说明：

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	caculateMac 生成 MAC
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	algType	String	是	对称算法类型
	plain	String	是	明文
	key	String	是	密钥对象
	macLength	int	是	MAC 结果长度。基于对称密码算法生成消息鉴别码时，一般对消息使用 CBC 模式进行加密，取密文的最后一个分组作为消息鉴别码。CBC-MAC 是最为广泛使用的消息认证算法之一，同时它也是一个 ANSI 标准（X9.17）。CBC-MAC 实际上就是对消息使用 CBC 模式进行加密，取密文的最后一块作为认证码。

响应业务参数说明：

message_header	参数名称	类型	是否必须	描述
	businesstype	String	是	caculateMac 生成 MAC
	其他公共参数说明，请查看附录 B.2.3.1 的 message_header 说明			
message_content	参数名称	类型	是否必须	描述
	macData	String	是	计算出的 MAC 值

附 录 C
（规范性）
高级应用接口相关标识

C.1证书应用接口相关标识

```
#define CREATE_CERTCHAIN_ERROR      13000    //无法构成证书链
#define CERT_PERIOD_OVER            13001    //证书已过期
#define CERT_IN_CRL                 13002    //证书在 CRL 列表中
#define LOCAL_CRL_LOSE              13003    //本地找不到 CRL 列表
#define VERIFY_CERTSIGN_ERROR       13004    //证书签名不通过
#define ABERROR                     20000    //异常未知错误
#define ADVINIT_ERROR               20001    //初始化错误
#define ADVEND_ERROR                20002    //释放错误
#define SET_HARDWARE_ERROR          20003    //设置硬件介质错误
#define USER_LOGIN_ERROR            20004    //用户登录失败
#define CHANGE_PIN_ERROR            20005    //修改 PIN 码失败
#define USER_LOGOUT_ERROR           20006    //用户登录退出失败
#define BASE64_ENCODE_ERROR         20007    //BASE64 加密错误
#define BASE64_DECODE_ERROR         20008    //BASE64 解密错误
#define GET_CERT_ERROR              20009    //获取证书错误
#define CHECK_CERT_ERROR            20010    //验证证书错误
#define GET_CERTINFO_ERROR          20011    //获取证书详细信息错误
#define SEAL_ENVELOPE_ERROR         20012    //生成数字信封错误
#define FSEAL_ENVELOPE_ERROR        20013    //对文件作数字信封处理错误
#define SEAL_ENVELOPEEX_ERROR       20014    //批量生成数字信封错误
#define FSEAL_ENVELOPEEX_ERROR      20015    //对文件批量生成数字信封错误
#define OPEN_ENVELOPE_ERROR         20016    //拆解数字信封错误
#define FOPEN_ENVELOPE_ERROR        20017    //从文件拆解数字信封错误
#define SIGN_DATA_ERROR             20018    //数字签名错误
#define FSIGN_DATA_ERROR            20019    //对文件数字签名错误
#define SIGN_DATAEX_ERROR           20020    //数字签名错误
#define FSIGN_DATAEX_ERROR          20021    //对文件数字签名错误
#define VERIFY_SIGN_ERROR           20022    //数字验签错误
#define FVERIFY_SIGN_ERROR          20023    //从文件数字验签错误
#define HASH_DATA_ERROR             20024    //哈希运算失败
#define GEN_RANDOM_ERROR            20025    //生成随机数错误
#define SYMM_ENCRYPT_ERROR          20026    //对称加密错误
#define FSYMM_ENCRYPT_ERROR         20027    //对文件作对称加密错误
#define SYMM_DECRYPT_ERROR          20028    //对称解密错误
#define FSYMM_DECRYPT_ERROR         20029    //对文件作对称解密错误
#define GETCERT_LDAP_ERROR          20030    //从 LDAP 服务器查询证书错误
#define GET_CRL_ERROR               20031    //从 LDAP 服务器获取 CRL 列表错误
```

```
#define READ_FILE_ERROR                 20032   //读取本地文件错误
#define WRITE_FILE_ERROR                20033   //写文件错误
#define MFC_INIT_ERROR                  20034   //MFC 初始化错误
#define MAXFILESIZE_ERROR               20035   //读取文件超过最大值
#define GETALGO_FROMPA_ERROR            20041   //通过 PA 获取算法标识失败
#define OPEN_FILE_ERROR                 20042   //打开配置文件失败
#define CONFIG_FILE_EMPTY               20043   //配置文件内容为空
#define PARAM_KEY_NOFOUND               20044   //匹配键值失败
#define GET_PARAMVALUE_ERROR            20045   //从配置文件获取信息失败
#define INDATA_IS_NULL                  20501   //输入数据为空
#define INCERT_IS_NULL                  20502   //输入证书信息为空
#define CLEARTEXT_IS_NULL               20503   //原文信息为空
#define SIGNDATA_IS_NULL                20504   //签名数据信息为空
#define SYMMKEY_IS_NULL                 20505   //对称密钥数据为空
#define CERTNUM_IS_ZERO                 20506   //证书个数为零
#define FILEPATH_IS_NULL                20507   //文件路径为空
#define SEARCHVALUE_IS_NULL             20508   //搜索字串为空
#define FILE_IS_NULL                    20509   //文件为空
#define WRITEDATA_IS_NULL               20510   //待写的数据为空
#define INFILE_IS_NULL                  20512   //输入文件路径为空
#define OUTFILE_IS_NULL                 20513   //输出文件路径为空
#define CLEARFILE_IS_NULL               20514   //原文文件路径为空
#define SIGNFILE_IS_NULL                20515   //签名数据文件路径为空
#define LOGINPIN_IS_NULL                20516   //登录 PIN 为空
#define KEYLABEL_IS_NULL                20517   //私钥标签为空
#define OLDPASSWD_IS_NULL               20518   //旧 PIN 码为空
#define NEWPASSWD_IS_NULL               20519   //新 PIN 码为空
#define OWNERCERT_IS_NULL               20520   //自己证书为空
#define OTHERCERT_IS_NULL               20521   //对方证书为空
#define PARAMKEY_IS_NULL                20522   //查找键名为空
#define GET_CERT_TYPE_ERROR             20601   //证书类型错误
#define UNKOWN_ERROR                    20700   //未知错误
```