

签名验签服务器

HTTP接口说明

声 明

版权声明：

本文档的版权属深圳奥联信息安全技术有限公司（以下简称“奥联信安”）所有。

本文档的版权受到中华人民共和国国家法律和国际公约的保护。未经书面许可，任何单位和个人不得以任何形式或通过任何途径非法使用、拷贝、修改、扩散本文档的全部或部分内容。

警告和承诺：

我们努力使本文档尽可能的完备和准确，但疏漏和缺陷之处在所难免。任何人或实体由于本文档提供的信息造成的任何损失或损害，奥联信息安全技术有限公司不承担任何义务或责任。

深圳奥联信息安全技术有限公司保留未经通知用户对本文档内容进行修改的权利。

反馈信息：

如果您对本文档有任何疑问、意见或建议，请与我们联系。对您的帮助，我们十分感激。

深圳奥联信息安全技术有限公司

网 站：[www.myibc.n](http://www.fisherman-it.com/)et

邮 箱：[service@myibc.n](mailto:support@fisherman-it.com)et

地 址：深圳市宝安区宝兴路6号海纳百川大厦B座16楼

读者对象

本文档提供签名验签服务器HTTP接口说明，并随购买的产品一并附给用户。也可为解决方案经理或技术经理、项目经理提供工作参考。

目 录

[第 1 章 简介 8](#_Toc3975)

[第 2 章 导出证书 9](#_Toc27567)

[2.1 功能说明 9](#_Toc28)

[2.2 接口说明 9](#_Toc9878)

[2.3 请求消息 9](#_Toc11413)

[参数说明： 9](#_Toc69)

[请求示例： 9](#_Toc7775)

[2.4 响应消息 10](#_Toc23042)

[返回值说明： 10](#_Toc239)

[返回JSON示例： 10](#_Toc27541)

[第 3 章 解析证书 12](#_Toc1679)

[3.1 功能说明 12](#_Toc31059)

[3.2 接口说明 12](#_Toc6307)

[3.3 请求消息 12](#_Toc19403)

[参数说明： 12](#_Toc22329)

[请求示例： 13](#_Toc32171)

[3.4 响应消息 14](#_Toc18902)

[返回值说明： 14](#_Toc6596)

[返回JSON示例： 14](#_Toc29658)

[第 4 章 验证证书有效性 16](#_Toc4712)

[4.1 功能说明 16](#_Toc20734)

[4.2 接口说明 16](#_Toc9328)

[4.3 请求消息 16](#_Toc8877)

[参数说明： 16](#_Toc2984)

[请求示例： 16](#_Toc32081)

[4.4 响应消息 17](#_Toc23419)

[返回值说明： 17](#_Toc15554)

[返回JSON示例： 17](#_Toc9740)

[第 5 章 单包数字签名 19](#_Toc380)

[5.1 功能说明 19](#_Toc11564)

[5.2 接口说明 19](#_Toc9296)

[5.3 请求消息 19](#_Toc29629)

[参数说明： 19](#_Toc15513)

[请求示例： 20](#_Toc31629)

[5.4 响应消息 21](#_Toc30028)

[返回值说明： 21](#_Toc6829)

[返回JSON示例： 21](#_Toc3931)

[第 6 章 单包验证数字签名 22](#_Toc27852)

[6.1 功能说明 22](#_Toc9274)

[6.2 接口说明 22](#_Toc18464)

[6.3 请求消息 22](#_Toc4492)

[参数说明： 22](#_Toc6881)

[请求示例： 23](#_Toc32580)

[6.4 响应消息 24](#_Toc32106)

[返回值说明： 24](#_Toc15290)

[返回JSON示例： 24](#_Toc26225)

[第 7 章 多包数字签名初始化 26](#_Toc17801)

[7.1 功能说明 26](#_Toc6650)

[7.2 接口说明 26](#_Toc22829)

[7.3 请求消息 26](#_Toc28914)

[参数说明： 26](#_Toc18615)

[请求示例： 27](#_Toc18071)

[7.4 响应消息 27](#_Toc22044)

[返回值说明： 27](#_Toc529)

[返回JSON示例： 28](#_Toc19676)

[第 8 章 多包数字签名更新 29](#_Toc9706)

[8.1 功能说明 29](#_Toc15904)

[8.2 接口说明 29](#_Toc23632)

[8.3 请求消息 29](#_Toc6777)

[参数说明： 29](#_Toc28221)

[请求示例： 29](#_Toc32488)

[8.4 响应消息 30](#_Toc17652)

[返回值说明： 30](#_Toc30425)

[返回JSON示例： 30](#_Toc20610)

[第 9 章 多包数字签名结束 32](#_Toc17199)

[9.1 功能说明 32](#_Toc17182)

[9.2 接口说明 32](#_Toc30368)

[9.3 请求消息 32](#_Toc20631)

[参数说明： 32](#_Toc31211)

[请求示例： 32](#_Toc23096)

[9.4 响应消息 33](#_Toc24231)

[返回值说明： 33](#_Toc29384)

[返回JSON示例： 33](#_Toc24007)

[第 10 章 多包数字验证签名初始化 35](#_Toc17298)

[10.1 功能说明 35](#_Toc1799)

[10.2 接口说明 35](#_Toc25238)

[10.3 请求消息 35](#_Toc4308)

[参数说明： 35](#_Toc27469)

[请求示例： 36](#_Toc29812)

[10.4 响应消息 36](#_Toc30390)

[返回值说明： 36](#_Toc8315)

[返回JSON示例： 37](#_Toc13433)

[第 11 章 多包数字验证签名更新 38](#_Toc7117)

[11.1 功能说明 38](#_Toc13917)

[11.2 接口说明 38](#_Toc1463)

[11.3 请求消息 38](#_Toc18554)

[参数说明： 38](#_Toc1008)

[请求示例： 39](#_Toc17339)

[11.4 响应消息 39](#_Toc30081)

[返回值说明： 39](#_Toc30128)

[返回JSON示例： 40](#_Toc14803)

[第 12 章 多包数字验证签名结束 41](#_Toc20421)

[12.1 功能说明 41](#_Toc5515)

[12.2 接口说明 41](#_Toc24919)

[12.3 请求消息 41](#_Toc19153)

[参数说明： 41](#_Toc31544)

[请求示例： 42](#_Toc26928)

[12.4 响应消息 42](#_Toc4267)

[返回值说明： 42](#_Toc14657)

[返回JSON示例： 43](#_Toc18651)

[第 13 章 消息签名 44](#_Toc2731)

[13.1 功能说明 44](#_Toc3408)

[13.2 接口说明 44](#_Toc8013)

[13.3 请求消息 44](#_Toc1284)

[参数说明： 44](#_Toc1160)

[请求示例： 45](#_Toc10482)

[13.4 响应消息 46](#_Toc29698)

[返回值说明： 46](#_Toc18013)

[返回JSON示例： 46](#_Toc24409)

[第 14 章 验证消息签名 48](#_Toc9257)

[14.1 功能说明 48](#_Toc9891)

[14.2 接口说明 48](#_Toc27624)

[14.3 请求消息 48](#_Toc14250)

[参数说明： 48](#_Toc15519)

[请求示例： 49](#_Toc1091)

[14.4 响应消息 49](#_Toc15425)

[返回值说明： 49](#_Toc28727)

[返回JSON示例： 50](#_Toc26228)

# 简介

本文档依据信息安全技术签名验签服务器技术规范GB/T 38629-2020，通过HTTP协议完成接口调用(附录A)。

# 导出证书

## 功能说明

导出证书

## 接口说明

接口协议：HTTP协议

请求方式：POST application/json

接口服务地址： /ExportCert

## 请求消息

参数说明：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **子节点** | **父节点** | **节点类型** | **说明** | **必填** |
| version |  | Integer | 协议版本.描述当前语法的版本，当前版本号为V1，用整型值0表示 | 是 |
| reqType |  | String | 请求类型，应为exportCert | 是 |
| reqTime |  | String | 请求时间，带时区格式的时间字符串。采用GeneralizedTime语法，格式要求：YYYYMMDDhhmm[ss[.s...]]{Z} + hhmm|-hhmm}  例子：20131001120000Z+0800 | 是 |
| request |  | Object | 请求包 | 是 |
| identification | request | Integer | Base64编码的证书标志 | 是 |

请求示例：

{

"version": 0,

"reqType": "exportCert",

"reqTime": "带时区格式的时间字符串",

"request": {

"identification": "Base64编码的证书标志"

}

}

## 响应消息

返回值说明：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **子节点** | **父节点** | **节点类型** | **说明** | **必填** |
| version |  | Integer | 协议版本.描述当前语法的版本，当前版本号为V1，用整型值0表示 | 是 |
| respType |  | String | 响应类型，当为exportCert | 是 |
| respTime |  | String | 响应时间，带时区格式的时间字符串。采用GeneralizedTime语法，格式要求：YYYYMMDDhhmm[ss[.s...]]{Z} + hhmm|-hhmm}  例子：20131001120000Z+0800 | 是 |
| respond |  | Object | 响应包 | 是 |
| respValue | respond | Integer | 响应码，0表示成功，非0表示失败 | 是 |
| cert | respond | String | Base64编码的导出的DER格式证书 | 是 |

返回JSON示例：

{

"version": 0,

"respType": "exportCert",

"respTime": "带时区格式的时间字符串",

"respond": {

"respValue": "响应码",

"cert": "Base64编码的签名的结果"

}

}

# 解析证书

## 功能说明

解析证书

## 接口说明

接口协议：HTTP协议

请求方式：POST application/json

接口服务地址： /ParseCert

## 请求消息

参数说明：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **子节点** | **父节点** | **节点类型** | **说明** | **必填** |
| version |  | Integer | 协议版本.描述当前语法的版本，当前版本号为V1，用整型值0表示 | 是 |
| reqType |  | String | 请求类型，应为parseCert | 是 |
| reqTime |  | String | 请求时间，带时区格式的时间字符串。采用GeneralizedTime语法，格式要求：YYYYMMDDhhmm[ss[.s...]]{Z} + hhmm|-hhmm}  例子：20131001120000Z+0800 | 是 |
| request |  | Object | 请求包 | 是 |
| infoType | request | Integer | 要解析证书信息的类型，应遵循GB/T 33560-2017中6.3.4的规定 | 是 |
| cert | request | String | Base64编码的要解析的DER格式数字证书 | 是 |



请求示例：

{

"version": 0,

"reqType": "parseCert",

"reqTime": "带时区格式的时间字符串",

"request": {

"infoType": "要解析证书信息的类型",

"cert": "Base64编码的要解析的数字证书"

}

}

## 响应消息

返回值说明：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **子节点** | **父节点** | **节点类型** | **说明** | **必填** |
| version |  | Integer | 协议版本.描述当前语法的版本，当前版本号为V1，用整型值0表示 | 是 |
| respType |  | String | 响应类型，当为parseCert | 是 |
| respTime |  | String | 响应时间，带时区格式的时间字符串。采用GeneralizedTime语法，格式要求：YYYYMMDDhhmm[ss[.s...]]{Z} + hhmm|-hhmm}  例子：20131001120000Z+0800 | 是 |
| respond |  | Object | 响应包 | 是 |
| respValue | respond | Integer | 响应码，0表示成功，非0表示失败 | 是 |
| info | respond | String | Base64编码的获取的解析到的证书信息 | 否 |

返回JSON示例：

{

"version": 0,

"respType": "parseCert",

"respTime": "带时区格式的时间字符串",

"respond": {

"respValue": "响应码",

"info": "Base64编码的获取的证书信息"

}

}

# 验证证书有效性

## 功能说明

验证证书有效性

## 接口说明

接口协议：HTTP协议

请求方式：POST application/json

接口服务地址： /ValidateCert

## 请求消息

参数说明：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **子节点** | **父节点** | **节点类型** | **说明** | **必填** |
| version |  | Integer | 协议版本.描述当前语法的版本，当前版本号为V1，用整型值0表示 | 是 |
| reqType |  | String | 请求类型，应为validateCert | 是 |
| reqTime |  | String | 请求时间，带时区格式的时间字符串。采用GeneralizedTime语法，格式要求：YYYYMMDDhhmm[ss[.s...]]{Z} + hhmm|-hhmm}  例子：20131001120000Z+0800 | 是 |
| request |  | Object | 请求包 | 是 |
| cert | request | String | Base64编码的要验证有效性的DER格式数字证书 | 是 |
| ocsp | request | String | 是否获取证书的OCSP状态，默认值FALSE。TRUE：是。FALSE:否。 | 否 |

请求示例：

{

"version": 0,

"reqType": "validateCert",

"reqTime": "带时区格式的时间字符串",

"request": {

"cert": "Base64编码的要验证有效性的数字证书",

"ocsp": "true"

}

}

## 响应消息

返回值说明：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **子节点** | **父节点** | **节点类型** | **说明** | **必填** |
| version |  | Integer | 协议版本.描述当前语法的版本，当前版本号为V1，用整型值0表示 | 是 |
| respType |  | String | 响应类型，当为validateCert | 是 |
| respTime |  | String | 响应时间，带时区格式的时间字符串。采用GeneralizedTime语法，格式要求：YYYYMMDDhhmm[ss[.s...]]{Z} + hhmm|-hhmm}  例子：20131001120000Z+0800 | 是 |
| respond |  | Object | 响应包 | 是 |
| respValue | respond | Integer | 响应码，0表示成功，非0表示失败 | 是 |
| state | respond | Integer | 证书的OCSP状态标志 | 否 |

返回JSON示例：

{

"version": 0,

"respType": "validateCert",

"respTime": "带时区格式的时间字符串",

"respond": {

"respValue": "响应码",

"state": "证书的OCSP状态标志"

}

}

# 单包数字签名

## 功能说明

数字签名

## 接口说明

接口协议：HTTP协议

请求方式：POST application/json

接口服务地址： /SignData

## 请求消息

参数说明：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **子节点** | **父节点** | **节点类型** | **说明** | **必填** |
| version |  | Integer | 协议版本.描述当前语法的版本，当前版本号为V1，用整型值0表示 | 是 |
| reqType |  | String | 请求类型，应为signData | 是 |
| reqTime |  | String | 请求时间，带时区格式的时间字符串。采用GeneralizedTime语法，格式要求：YYYYMMDDhhmm[ss[.s...]]{Z} + hhmm|-hhmm}  例子：20131001120000Z+0800 | 是 |
| request |  | Object | 请求包 | 是 |
| signMethod | request | Integer | 签名算法类型，应遵循GB/T 33560-2017中6.2.4的规定 | 是 |
| keyIndex | request | Integer | 私钥索引值 | 是 |
| keyValue | request | String | Base64编码的签名者私钥权限标识码。需要Base64编码 | 是 |
| signerIDLen | request | Integer | 签名者的ID长度，当signMethod为SGD\_SM3\_SM2时有效 | 否 |
| signerID | request | String | 签名者的ID值，当signMethod为SGD\_SM3\_SM2时有效。需要Base64编码 | 否 |
| inDataLen | request | Integer | 待签名的数据原文长度 | 是 |
| inData | request | String | Base64编码的待签名的数据原文 | 是 |

表1签名算法标识

|  |  |  |
| --- | --- | --- |
| 标签 | 标识符 | 描述 |
| SGD\_SM3\_SM2 | 0x00020201 | 基于SM3算法和SM2算法的签名 |

请求示例：

{

"version": 0,

"reqType": "signData",

"reqTime": "带时区格式的时间字符串",

"request": {

"signMethod": "签名算法类型",

"keyIndex": "私钥的索引",

"keyValue": "私钥权限标识码",

"inDataLen": "待签名的数据原文长度",

"inData": "Base64编码的待签名的数据原文"

}

}

## 响应消息

返回值说明：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **子节点** | **父节点** | **节点类型** | **说明** | **必填** |
| version |  | Integer | 协议版本.描述当前语法的版本，当前版本号为V1，用整型值0表示 | 是 |
| respType |  | String | 响应类型，当为signData | 是 |
| respTime |  | String | 响应时间，带时区格式的时间字符串。采用GeneralizedTime语法，格式要求：YYYYMMDDhhmm[ss[.s...]]{Z} + hhmm|-hhmm}  例子：20131001120000Z+0800 | 是 |
| respond |  | Object | 响应包 | 是 |
| respValue | respond | Integer | 响应码，0表示成功，非0表示失败 | 是 |
| signature | respond | String | Base64编码的签名的结果 | 是 |

返回JSON示例：

{

"version": 0,

"respType": "signData",

"respTime": "带时区格式的时间字符串",

"respond": {

"respValue": "响应码",

"signature": "Base64编码的签名的结果"

}

}

# 单包验证数字签名

## 功能说明

验证单包数字签名

## 接口说明

接口协议：HTTP协议

请求方式：POST application/json

接口服务地址： /VerifySignedData

## 请求消息

参数说明：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **子节点** | **父节点** | **节点类型** | **说明** | **必填** |
| version |  | Integer | 协议版本.描述当前语法的版本，当前版本号为V1，用整型值0表示 | 是 |
| reqType |  | String | 请求类型，应为verifySignedData | 是 |
| reqTime |  | String | 请求时间，带时区格式的时间字符串。采用GeneralizedTime语法，格式要求：YYYYMMDDhhmm[ss[.s...]]{Z} + hhmm|-hhmm}  例子：20131001120000Z+0800 | 是 |
| request |  | Object | 请求包 | 是 |
| signMethod | request | Integer | 签名算法标识，应遵循GB/T 33560-2017中6.2.4的规定 | 是 |
| type | request | Integer | 验证数字签名时使用证书或证书序列号，1表示使用证书，2表示使用证书序列号 | 是 |
| cert | request | String | Base64编码的证书，type取值1时有效 | 否 |
| certSN | request | String | Base64编码的证书序列号，type取值2时有效 | 否 |
| signerIDLen | request | Integer | 签名者的ID长度，当signMethod为SGD\_SM3\_SM2时有效。 | 否 |
| signerID | request | String | 签名者的ID值，当signMethod为SGD\_SM3\_SM2时有效。需要Base64编码 | 否 |
| inDataLen | request | Integer | 待签名的数据原文长度 | 是 |
| inData | request | String | Base64编码的待签名的数据原文 | 是 |
| signature | request | String | Base64编码的签名结果 | 是 |
| verifyLevel | request | Integer | 表示证书验证级别，0：验证时间，1:验证时间和根证书签名，2:验证时间、根证书签名和CRL | 是 |

请求示例：

{

"version": 0,

"reqType": "verifySignedData",

"reqTime": "带时区格式的时间字符串",

"request": {

"signMethod": "签名算法类型",

"type": "验证数字签名时使用证书或证书序列号，1表示使用证书，2表示使用证书序列号",

"cert": "Base64编码的证书，type取值1时有效",

"certSN": "Base64编码的证书序列号，type取值2时有效",

"signerIDLen": "签名者的ID长度，当signMethod为SGD\_SM3\_SM2时有效",

"signerID": "Base64编码的签名者的ID值，当signMethod为SGD\_SM3\_SM2时有效",

"inDataLen": "待签名的数据原文长度",

"inData": "Base64编码的待签名的数据原文",

"signature": "Base64编码的签名结果",

"verifyLevel": "表示证书验证级别，0：验证时间，1:验证时间和根证书签名，2:验证时间、根证书签名和CRL"

}

}

## 响应消息

返回值说明：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **子节点** | **父节点** | **节点类型** | **说明** | **必填** |
| version |  | Integer | 协议版本.描述当前语法的版本，当前版本号为V1，用整型值0表示 | 是 |
| respType |  | String | 响应类型，当为verifySignedData | 是 |
| respTime |  | String | 响应时间，带时区格式的时间字符串。采用GeneralizedTime语法，格式要求：YYYYMMDDhhmm[ss[.s...]]{Z} + hhmm|-hhmm}  例子：20131001120000Z+0800 | 是 |
| respond |  | Object | 响应包 | 是 |
| respValue | respond | Integer | 响应码，0表示成功，非0表示失败 | 是 |

返回JSON示例：

{

"version": "v1",

"respType": "verifySignedData",

"respTime": "带时区格式的时间字符串",

"respond": {

"respValue": "响应码"

}

}

# 多包数字签名初始化

## 功能说明

多包数字签名初始化

## 接口说明

接口协议：HTTP协议

请求方式：POST application/json

接口服务地址： /SignDataInit

## 请求消息

参数说明：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **子节点** | **父节点** | **节点类型** | **说明** | **必填** |
| version |  | Integer | 协议版本.描述当前语法的版本，当前版本号为V1，用整型值0表示 | 是 |
| reqType |  | String | 请求类型，应为signDataInit | 是 |
| reqTime |  | String | 请求时间，带时区格式的时间字符串。采用GeneralizedTime语法，格式要求：YYYYMMDDhhmm[ss[.s...]]{Z} + hhmm|-hhmm}  例子：20131001120000Z+0800 | 是 |
| request |  | Object | 请求包 | 是 |
| signMethod | request | Integer | 签名算法类型，应遵循GB/T 33560-2017中6.2.4的规定 | 是 |
| signerPublicKey | request | String | Base64编码签名者的公钥，当signMethod为SGD\_SM3\_SM2时有效 | 否 |
| signerIDLen | request | Integer | 签名者的ID长度，当signMethod为SGD\_SM3\_SM2时有效 | 否 |
| signerID | request | String | Base64编码签名者的ID值，当signMethod为SGD\_SM3\_SM2时有效 | 否 |

请求示例：

{

"version": 0,

"reqType": "signDataInit",

"reqTime": "带时区格式的时间字符串",

"request": {

"signMethod": "签名算法类型",

"signerPublicKey": "Base64编码签名者的公钥，当signMethod为SGD\_SM3\_SM2时有效",

"signerIDLen": "签名者的ID长度，当signMethod为SGD\_SM3\_SM2时有效",

"signerID": "Base64编码签名者的ID值，当signMethod为SGD\_SM3\_SM2时有效"

}

}

## 响应消息

返回值说明：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **子节点** | **父节点** | **节点类型** | **说明** | **必填** |
| version |  | Integer | 协议版本.描述当前语法的版本，当前版本号为V1，用整型值0表示 | 是 |
| respType |  | String | 响应类型，当为signDataInit | 是 |
| respTime |  | String | 响应时间，带时区格式的时间字符串。采用GeneralizedTime语法，格式要求：YYYYMMDDhhmm[ss[.s...]]{Z} + hhmm|-hhmm}  例子：20131001120000Z+0800 | 是 |
| respond |  | Object | 响应包 | 是 |
| respValue | respond | Integer | 响应码，0表示成功，非0表示失败 | 是 |
| SessionID | respond | String | Base64编码的会话标识 | 否 |

返回JSON示例：

{

"version": 0,

"respType": "signDataInit",

"respTime": "带时区格式的时间字符串",

"respond": {

"respValue": 响应码,

"SessionID": "Base64编码的会话标识"

}

}

# 多包数字签名更新

## 功能说明

多包数字签名更新

## 接口说明

接口协议：HTTP协议

请求方式：POST application/json

接口服务地址： /SignDataUpdate

## 请求消息

参数说明：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **子节点** | **父节点** | **节点类型** | **说明** | **必填** |
| version |  | Integer | 协议版本.描述当前语法的版本，当前版本号为V1，用整型值0表示 | 是 |
| reqType |  | String | 请求类型，应为signDataUpdate | 是 |
| reqTime |  | String | 请求时间，带时区格式的时间字符串。采用GeneralizedTime语法，格式要求：YYYYMMDDhhmm[ss[.s...]]{Z} + hhmm|-hhmm}  例子：20131001120000Z+0800 | 是 |
| request |  | Object | 请求包 | 是 |
| SessionID | request | String | Base64编码的会话标识 | 是 |
| inDataLen | request | Integer | 数据明文长度 | 是 |
| inData | request | String | Base64编码的数据明文 | 是 |

请求示例：

{

"version": 0,

"reqType": "signDataUpdate",

"reqTime": "带时区格式的时间字符串",

"request": {

"SessionID": "Base64编码的会话标识",

"inDataLen": "数据明文长度",

"inData": "Base64编码的数据明文"

}

}

## 响应消息

返回值说明：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **子节点** | **父节点** | **节点类型** | **说明** | **必填** |
| version |  | Integer | 协议版本.描述当前语法的版本，当前版本号为V1，用整型值0表示 | 是 |
| respType |  | String | 响应类型，当为signDataUpdate | 是 |
| respTime |  | String | 响应时间，带时区格式的时间字符串。采用GeneralizedTime语法，格式要求：YYYYMMDDhhmm[ss[.s...]]{Z} + hhmm|-hhmm}  例子：20131001120000Z+0800 | 是 |
| respond |  | Object | 响应包 | 是 |
| respValue | respond | Integer | 响应码，0表示成功，非0表示失败 | 是 |
| SessionID | respond | String | Base64编码的会话标识 | 否 |

返回JSON示例：

{

"version": 0,

"respType": "signDataUpdate",

"respTime": "带时区格式的时间字符串",

"respond": {

"respValue": "响应码",

"SessionID": "Base64编码的会话标识"

}

}

# 多包数字签名结束

## 功能说明

多包数字签名结束

## 接口说明

接口协议：HTTP协议

请求方式：POST application/json

接口服务地址： /SignDataFinal

## 请求消息

参数说明：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **子节点** | **父节点** | **节点类型** | **说明** | **必填** |
| version |  | Integer | 协议版本.描述当前语法的版本，当前版本号为V1，用整型值0表示 | 是 |
| reqType |  | String | 请求类型，应为signDataFinal | 是 |
| reqTime |  | String | 请求时间，带时区格式的时间字符串。采用GeneralizedTime语法，格式要求：YYYYMMDDhhmm[ss[.s...]]{Z} + hhmm|-hhmm}  例子：20131001120000Z+0800 | 是 |
| request |  | Object | 请求包 | 是 |
| SessionID | request | String | Base64编码的会话标识 | 是 |
| keyIndex | request | Integer | 签名者私钥索引值，如十进制1表示索引值为1的密钥 | 是 |
| keyValue | request | String | Base64编码的私钥权限标识码 | 是 |

请求示例：

{

"version": 0,

"reqType": "signDataFinal",

"reqTime": "带时区格式的时间字符串",

"request": {

"SessionID": "Base64编码的会话标识",

"keyIndex": 签名者私钥索引,

"keyValue": "Base64编码的私钥权限标识码"

}

}

## 响应消息

返回值说明：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **子节点** | **父节点** | **节点类型** | **说明** | **必填** |
| version |  | Integer | 协议版本.描述当前语法的版本，当前版本号为V1，用整型值0表示 | 是 |
| respType |  | String | 响应类型，当为signDataFinal | 是 |
| respTime |  | String | 响应时间，带时区格式的时间字符串。采用GeneralizedTime语法，格式要求：YYYYMMDDhhmm[ss[.s...]]{Z} + hhmm|-hhmm}  例子：20131001120000Z+0800 | 是 |
| respond |  | Object | 响应包 | 是 |
| respValue | respond | Integer | 响应码，0表示成功，非0表示失败 | 是 |
| signature | respond | String | Base64编码签名值。数据的结构应遵循GB/T 35276 | 否 |

返回JSON示例：

{

"version": "v1",

"respType": "signedDataFinal",

"respTime": "带时区格式的时间字符串",

"respond": {

"respValue": "响应码",

"signature": "Base64编码签名值"

}

}

# 多包数字验证签名初始化

## 功能说明

多包数字签名初始化

## 接口说明

接口协议：HTTP协议

请求方式：POST application/json

接口服务地址： /VerifySignedDataInit

## 请求消息

参数说明：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **子节点** | **父节点** | **节点类型** | **说明** | **必填** |
| version |  | Integer | 协议版本.描述当前语法的版本，当前版本号为V1，用整型值0表示 | 是 |
| reqType |  | String | 请求类型，应为verifySignedDataInit | 是 |
| reqTime |  | String | 请求时间，带时区格式的时间字符串。采用GeneralizedTime语法，格式要求：YYYYMMDDhhmm[ss[.s...]]{Z} + hhmm|-hhmm}  例子：20131001120000Z+0800 | 是 |
| request |  | Object | 请求包 | 是 |
| signMethod | request | Integer | 签名算法类型，应遵循GB/T 33560-2017中6.2.4的规定 | 是 |
| signerPublicKey | request | String | Base64编码签名者的公钥，当signMethod为SGD\_SM3\_SM2时有效 | 否 |
| signerIDLen | request | Integer | 签名者的ID长度，当signMethod为SGD\_SM3\_SM2时有效 | 否 |
| signerID | request | String | Base64编码签名者的ID值，当signMethod为SGD\_SM3\_SM2时有效 | 否 |

请求示例：

{

"version": 0,

"reqType": "verifySignedDataInit",

"reqTime": "带时区格式的时间字符串",

"request": {

"signMethod": "签名算法类型",

"signerPublicKey": "Base64编码签名者的公钥，当signMethod为SGD\_SM3\_SM2时有效",

"signerIDLen": 签名者的ID长度，当signMethod为SGD\_SM3\_SM2时有效,

"signerID": "Base64编码签名者的ID值，当signMethod为SGD\_SM3\_SM2时有效"

}

}

## 响应消息

返回值说明：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **子节点** | **父节点** | **节点类型** | **说明** | **必填** |
| version |  | Integer | 协议版本.描述当前语法的版本，当前版本号为V1，用整型值0表示 | 是 |
| respType |  | String | 响应类型，当为verifySignedDataInit | 是 |
| respTime |  | String | 响应时间，带时区格式的时间字符串。采用GeneralizedTime语法，格式要求：YYYYMMDDhhmm[ss[.s...]]{Z} + hhmm|-hhmm}  例子：20131001120000Z+0800 | 是 |
| respond |  | Object | 响应包 | 是 |
| respValue | respond | Integer | 响应码，0表示成功，非0表示失败 | 是 |
| SessionID | respond | String | Base64编码的会话标识 | 否 |

返回JSON示例：

{

"version": 0,

"respType": "verifySignedDataInit",

"respTime": "带时区格式的时间字符串",

"respond": {

"respValue": "响应码",

"SessionID": "Base64编码的会话标识"

}

}

# 多包数字验证签名更新

## 功能说明

多包数字验证签名更新

## 接口说明

接口协议：HTTP协议

请求方式：POST application/json

接口服务地址： /VerifySignedDataUpdate

## 请求消息

参数说明：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **子节点** | **父节点** | **节点类型** | **说明** | **必填** |
| version |  | Integer | 协议版本.描述当前语法的版本，当前版本号为V1，用整型值0表示 | 是 |
| reqType |  | String | 请求类型，应为verifySignedDataUpdate | 是 |
| reqTime |  | String | 请求时间，带时区格式的时间字符串。采用GeneralizedTime语法，格式要求：YYYYMMDDhhmm[ss[.s...]]{Z} + hhmm|-hhmm}  例子：20131001120000Z+0800 | 是 |
| request |  | Object | 请求包 | 是 |
| SessionID | request | String | Base64编码的会话标识 | 否 |
| inDataLen | request | Integer | 数据明文长度 | 否 |
| inData | request | String | Base64编码的数据明文 | 否 |

请求示例：

{

"version": 0,

"reqType": "verifySignedDataUpdate",

"reqTime": "带时区格式的时间字符串",

"request": {

"SessionID": "Base64编码的会话标识",

"inDataLen": "数据明文长度",

"inData": "Base64编码的数据明文"

}

}

## 响应消息

返回值说明：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **子节点** | **父节点** | **节点类型** | **说明** | **必填** |
| version |  | Integer | 协议版本.描述当前语法的版本，当前版本号为V1，用整型值0表示 | 是 |
| respType |  | String | 响应类型，当为verifySignedDataUpdate | 是 |
| respTime |  | String | 响应时间，带时区格式的时间字符串。采用GeneralizedTime语法，格式要求：YYYYMMDDhhmm[ss[.s...]]{Z} + hhmm|-hhmm}  例子：20131001120000Z+0800 | 是 |
| respond |  | Object | 响应包 | 是 |
| respValue | respond | Integer | 响应码，0表示成功，非0表示失败 | 是 |
| SessionID | respond | String | Base64编码的会话标识 | 否 |

返回JSON示例：

{

"version": 0,

"respType": "verifySignedDataUpdate",

"respTime": "带时区格式的时间字符串",

"respond": {

"respValue": 响应码,

"SessionID": "Base64编码的会话标识"

}

}

# 多包数字验证签名结束

## 功能说明

多包数字验证签名结束

## 接口说明

接口协议：HTTP协议

请求方式：POST application/json

接口服务地址： /VerifySignedDataFinal

## 请求消息

参数说明：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **子节点** | **父节点** | **节点类型** | **说明** | **必填** |
| version |  | Integer | 协议版本.描述当前语法的版本，当前版本号为V1，用整型值0表示 | 是 |
| reqType |  | String | 请求类型，应为verifySignedDataFinal | 是 |
| reqTime |  | String | 请求时间，带时区格式的时间字符串。采用GeneralizedTime语法，格式要求：YYYYMMDDhhmm[ss[.s...]]{Z} + hhmm|-hhmm}  例子：20131001120000Z+0800 | 是 |
| request |  | Object | 请求包 | 是 |
| type | request | Integer | 验证数字签名时使用证书或证书序列号，1表示使用证书，2表示使用证书序列号 | 是 |
| cert | request | String | Base64编码的证书，type取值1时有效 | 否 |
| certSN | request | String | Base64编码的证书序列号，type取值2时有效 | 否 |
| SessionID | request | String | Base64编码的会话标识 | 是 |
| signature | request | String | Base64编码的签名结果 | 是 |
| verifyLevel | request | Integer | 表示证书验证级别，0：验证时间，1:验证时间和根证书签名，2:验证时间、根证书签名和CRL | 是 |

请求示例：

{

"version": "v1",

"reqType": "verifySignedDataFinal",

"reqTime": "带时区格式的时间字符串",

"request": {

"type": "验证数字签名时使用证书或证书序列号，1表示使用证书，2表示使用证书序列号",

"cert": "Base64编码的证书，type取值1时有效",

"certSN": "Base64编码的证书序列号，type取值2时有效",

"SessionID": "Base64编码的会话标识",

"signature": "Base64编码的签名结果",

"verifyLevel": "表示证书验证级别，0：验证时间，1:验证时间和根证书签名，2:验证时间、根证书签名和CRL"

}

}

## 响应消息

返回值说明：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **子节点** | **父节点** | **节点类型** | **说明** | **必填** |
| version |  | Integer | 协议版本.描述当前语法的版本，当前版本号为V1，用整型值0表示 | 是 |
| respType |  | String | 响应类型，应为verifySignedDataFinal | 是 |
| respTime |  | String | 响应时间，带时区格式的时间字符串。采用GeneralizedTime语法，格式要求：YYYYMMDDhhmm[ss[.s...]]{Z} + hhmm|-hhmm}  例子：20131001120000Z+0800 | 是 |
| respond |  | Object | 响应包 | 是 |
| respValue | respond | Integer | 响应码，0表示成功，非0表示失败 | 是 |

返回JSON示例：

{

"version": 0,

"respType": "verifySignedDataFinal",

"respTime": "带时区格式的时间字符串",

"respond": {

"respValue": 响应码,

}

}

# 消息签名

## 功能说明

消息签名

## 接口说明

接口协议：HTTP协议

请求方式：POST application/json

接口服务地址： /SignMessage

## 请求消息

参数说明：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **子节点** | **父节点** | **节点类型** | **说明** | **必填** |
| version |  | Integer | 协议版本.描述当前语法的版本，当前版本号为V1，用整型值0表示 | 是 |
| reqType |  | String | 请求类型，应为signMessage | 是 |
| reqTime |  | String | 请求时间，带时区格式的时间字符串。采用GeneralizedTime语法，格式要求：YYYYMMDDhhmm[ss[.s...]]{Z} + hhmm|-hhmm}  例子：20131001120000Z+0800 | 是 |
| request |  | Object | 请求包 | 是 |
| signMethod | request | Integer | 签名算法类型，应遵循GB/T 33560-2017中6.2.4的规定 | 是 |
| keyIndex | request | Integer | 私钥索引值 | 是 |
| keyValue | request | String | Base64编码的签名者私钥权限标识码 | 是 |
| signerIDLen | request | Integer | 签名者的ID长度，当signMethod为SGD\_SM3\_SM2时有效 | 否 |
| signerID | request | String | Base64编码签名者的ID值，当signMethod为SGD\_SM3\_SM2时有效 | 否 |
| inDataLen | request | Integer | 待签名的数据原文长度 | 是 |
| inData | request | String | Base64编码的待签名的数据原文 | 是 |
| originalText | request | String | 是否附加原文选项。TRUE:是。FALSE:否 | 否 |
| certificateChain | request | String | 是否附加证书链选项。TRUE:是。FALSE:否 | 否 |
| crl | request | String | 是否附加黑名单选项。TRUE:是。FALSE:否 | 否 |

请求示例：

{

"version": 0,

"reqType": "signMessage",

"reqTime": "带时区格式的时间字符串",

"request": {

"signMethod": 签名算法类型标识,

"keyIndex": 私钥的索引,

"keyValue": "私钥权限标识码",

"signerIDLen": 签名者的ID长度，当signMethod为SGD\_SM3\_SM2时有效,

"signerID": "Base64编码签名者的ID值，当signMethod为SGD\_SM3\_SM2时有效",

"inDataLen": 待签名的数据原文长度,

"inData": "Base64编码的待签名的数据原文",

"originalText": "是否附加原文选项",

"certificateChain": "是否附加证书链选项"

"crl": "是否附加黑名单选项"

}

}

## 响应消息

返回值说明：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **子节点** | **父节点** | **节点类型** | **说明** | **必填** |
| version |  | Integer | 协议版本.描述当前语法的版本，当前版本号为V1，用整型值0表示 | 是 |
| respType |  | String | 响应类型，当为signMessage | 是 |
| respTime |  | String | 响应时间，带时区格式的时间字符串。采用GeneralizedTime语法，格式要求：YYYYMMDDhhmm[ss[.s...]]{Z} + hhmm|-hhmm}  例子：20131001120000Z+0800 | 是 |
| respond |  | Object | 响应包 | 是 |
| respValue | respond | Integer | 响应码，0表示成功，非0表示失败 | 是 |
| signedMessage | respond | String | Base64编码的消息签名数据，消息的结构应遵循GB/T 35275 | 否 |

返回JSON示例：

{

"version": 0,

"respType": "signMessage",

"respTime": "带时区格式的时间字符串",

"respond": {

"respValue": "响应码",

"signedMessage": "Base64编码的消息签名数据"

}

}

# 验证消息签名

## 功能说明

验证消息签名

## 接口说明

接口协议：HTTP协议

请求方式：POST application/json

接口服务地址： /VerifySignedMessage

## 请求消息

参数说明：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **子节点** | **父节点** | **节点类型** | **说明** | **必填** |
| version |  | Integer | 协议版本.描述当前语法的版本，当前版本号为V1，用整型值0表示 | 是 |
| reqType |  | String | 请求类型，应为verifySignedMessage | 是 |
| reqTime |  | String | 请求时间，带时区格式的时间字符串。采用GeneralizedTime语法，格式要求：YYYYMMDDhhmm[ss[.s...]]{Z} + hhmm|-hhmm}  例子：20131001120000Z+0800 | 是 |
| request |  | Object | 请求包 | 是 |
| inDataLen | request | Integer | 待签名的数据原文长度 | 是 |
| inData | request | String | Base64编码的待签名的数据原文 | 是 |
| signerIDLen | request | Integer | 签名者的ID长度，当signMethod为SGD\_SM3\_SM2时有效 | 否 |
| signerID | request | String | 签名者的ID值，当signMethod为SGD\_SM3\_SM2时有效。需要Base64编码 | 否 |
| signedMessage | request | String | Base64编码的消息签名数据，消息的结构应遵循GB/T 35275 | 是 |

请求示例：

{

"version": 0,

"reqType": "verifySignedMessage",

"reqTime": "带时区格式的时间字符串",

"request": {

"inDataLen": 待签名的数据原文长度,

"inData": "Base64编码的待签名的数据原文",

"signerIDLen": 签名者的ID长度，当signMethod为SGD\_SM3\_SM2时有效,

"signerID": "Base64编码签名者的ID值，当signMethod为SGD\_SM3\_SM2时有效",

"signedMessage": "Base64编码的签名结果"

}

}

## 响应消息

返回值说明：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **子节点** | **父节点** | **节点类型** | **说明** | **必填** |
| version |  | Integer | 协议版本.描述当前语法的版本，当前版本号为V1，用整型值0表示 | 是 |
| respType |  | String | 响应类型，当为verifySignedMessage | 是 |
| respTime |  | String | 响应时间，带时区格式的时间字符串。采用GeneralizedTime语法，格式要求：YYYYMMDDhhmm[ss[.s...]]{Z} + hhmm|-hhmm}  例子：20131001120000Z+0800 | 否 |
| respond |  | Object | 响应包 | 是 |
| respValue | respond | Integer | 响应码，0表示成功，非0表示失败 | 是 |

返回JSON示例：

{

"version": 0,

"respType": "verifySignedMessage",

"respTime": "带时区格式的时间字符串",

"respond": {

"respValue": 响应码

}

}

