格尔 PKI 安全中间件

接口规范(精简版)

格尔软件股份有限公司

2020 年 6 月

修订记录

日期	作者	版本	内容
2020-06-08	KOAL	V1.0	新建版本
2020-10-20	KOAL	V1.1	getAllCert 接口添加可变参数 includeSN； 添加新接口 genRandom
2020-11-13	KOAL	V1.2	增加文件操作接口
2020-12-09	KOAL	V1.2.1	更新 p7 签名接口
2020-12-22	KOAL	V1.2.2	更新获取证书列表接口响应数据
2021-01-27	KOAL	V1.2.3	更新平台、key 支持信息，新增签发证书接口； 更新生成证书请求接口
2021-11-11	KOAL	V1.2.4	新增获取系统信息、设置可信驱动、获取登录临时参数、获取缓存 PIN 码接口；删除介质支持情况、删除发证接口下的导入证书、导入 pfx 证书、获取证书、指纹初始化接口
2021-11-30	KOAL	V1.2.5	新增获取初始化状态、获取 PROVIDER 状态、导入 pfx 证书、是否存在指纹接口，更新获取证书列表接口

目 录

1 概述

1.1 前言

格尔 PKI 安全中间件作为中间层，为底层与应用之间的交互搭建了“桥梁"，为应用提供了统一基础接口操作 SKF、CSP 接口的 USBKEY。作为应用系统的组件，格尔 PKI 安全中间件 SDK（以下简称 SDK）提供了 RSA/SM2 密码的应用接口，支持 PKCS#7 相关密码应用功能，同时提供 RSA/SM2 证书更新、秘钥更新、换发等相关证书接口，支持 PKCS#10 请求构造、证书解析等相关功能。多个外部应用通过 RPC 的方式调用同一套接口，无需关注 USBKEY 的操作过程，便可实现其相应的业务功能。

1.2 范围

本文描述了应用系统通过 RPC 的方式调用 SDK 访问 USBKEY 的开放接口以及集成开发规范，适用于所有涉及与 USBKEY 涉及交互过程的应用系统。

1.3 引用标准

本文档遵循或引用以下标准规范：

[1] RFC5210 Internet X.509 Public Key Infrastructure Certificate Management Protocol (CMP)

[2] RFC4211 Internet X.509 Public Key Infrastructure Certificate Request Message Format (CRMF)

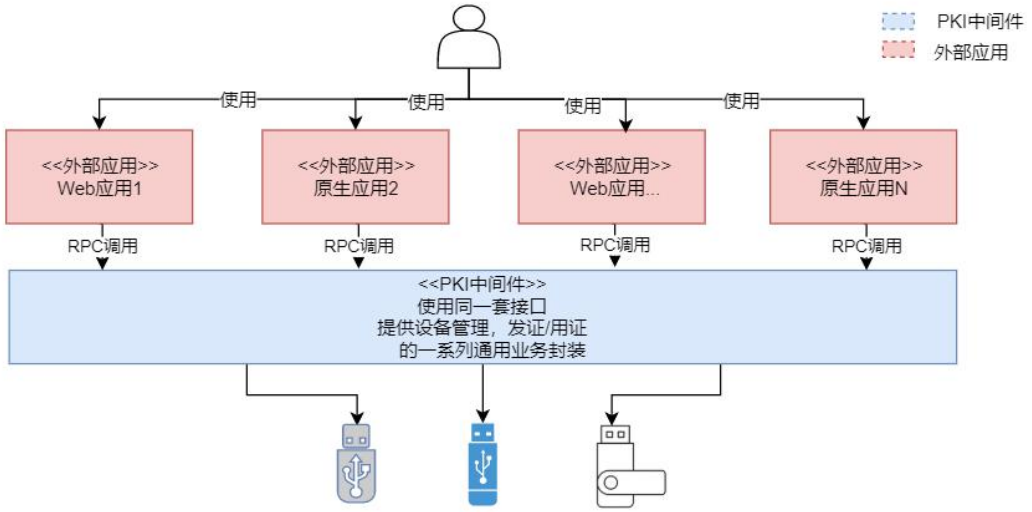
[3] ISO/IEC 8824/8825:2002 Abstract Syntax Notation One (ASN.1) and ASN.1 Encoding Rules

[4] GM/T 0010-2012 SM2 密码算法加密签名消息语法规范

[5] GM/T 0015 基于 SM2 密码算法的数字证书格式规范

2 结构模型

2.1 进程关系图



3 适用范围

中间件适配支持了多个平台、多个介质厂商以及多种浏览器，对外提供了安装包、自检程序等。

3.1 平台

CPU 架构	操作系统	安装包类型
i386	Windows xp sp3、windows7_x32	exe
x86_64	windows(7、10) x64	exe
	ubuntu 16.04	deb
	uos	deb
	centOS(7.5_64bit)	rpm
	兆芯+中科方德	deb
	兆芯+中标麒麟	rpm
mips64	龙芯+中标麒麟	rpm
	uos	deb
arm64	飞腾+银河麒麟	deb
	uos	deb

3.2 介质

目前中间件已支持多个普通 key、二代 key、指纹 key，如有需要可询问对接人员。

3.3 浏览器

支持 IE 浏览器（8+）、火狐、谷歌浏览器。

4 数据类型定义

4.1 基本数据类型

类型名称	描述	定义
INT8	有符号 8 位整数	
INT16	有符号 16 位整数	
INT32	有符号 32 位整数	
UINT8	无符号 8 位整数	
UINT16	无符号 16 位整数	
UINT32	无符号 32 位整数	
BOOL	布尔类型，取值为 TRUE 或 FALSE	
INT64	有符号的 longlong 型证书	typedef long long INT64

4.2 常量定义

4.2.1消息类型

（1）类型定义

```
enum enMsgType {
 enMsg_Login = 0x01,
 enMsg_LogOut = 0x02,
 enMsg_NotifyerStart = 0x0FFF0000,
 enMsg_NotifyerDevin = 0x0FFF0001,
 enMsg_NotifyerDevOut = 0x0FFF0002,
 enMsg_NotifyerDevModify = 0x0FFF0003,
 enMsg_NotifyerSessionClose = 0x0FFF0004,
}
```

（2）数据项描述

常量名	取值	描述
enMsg_Login	0x01	登录
enMsg_LogOut	0x02	注销
enMsg_NotifyerStart	0x0FFF0000	推送消息获取
enMsg_NotifyerDevin	0x0FFF0001	设备插入
enMsg_NotifyerDevOut	0x0FFF0002	设备拔出

enMsg_NotifyerDevModify	0x0FFF0003	设备信息更改
enMsg_NotifyerSessionClose	0x0FFF0004	Session 关闭

4.2.2PIN 类型

（1）类型定义

```
enum enPinType {
    PIN_ADMIN = 0,
    PIN_USER
}
```

（3）数据项描述

常量名	取值	描述
PIN_ADMIN	0	管理员 PIN 类型
PIN_USER	1	用户 PIN 类型

4.3 复合类型定义

4.3.1请求数据

（1）类型定义

```
struct msgRequest {
    INT32   reqid ;
    INT32   msgType;
    INT32   version;
    INT64   extend;
    string  jsonBody;
}
```

（2）数据项描述

常量名	类型	意义
reqid	INT32	请求 ID。登录请求 reqid = 0，登录成功后每次请求该字段递增
msgType	INT32	请求类型
version	INT32	消息版本。jsonbody 模板的版本
extend	INT64	扩展。备用字段，根据不同的 msgType 定义可能不一样
jsonBody	string	json 消息体

4.3.2响应数据

（1）类型定义

```
struct msgResponse {
    INT32  respid ;
    INT32  msgType;
    INT32  version;
    INT32  errCode ;
    INT64  extend;
    string jsonBody;
}
```

（2）数据项描述

常量名	类型	意义
respid	INT32	请求 ID，客户端主动请求的响应消息，该字段为请求结构体 msgRequest 的 reqid 字段
msgType	INT32	请求类型
version	INT32	消息版本。jsonbody 模板的版本
errCode	INT32	错误码，0 为执行成功，非 0 为失败
extend	INT64	扩展码
jsonBody	string	json 消息体

4.3.3会话数据

（1）类型定义

```
struct sessionTicket {
   INT64  sessionID;
   string ticket;
}
```

（2）数据项描述

常量名	类型	意义
sessionID	INT64	会话唯一标识
ticket	string	会话票据

5 公共接口

中间件服务运行过程中，应用可通过调用公共接口注册（注销）会话票据、获取中间件版本及介质的插拔通知。在公共接口中登录接口属于基础接口，即应用必须调用登录接口与中间件建立会话连接后方可处理其他操作。当应用业务处理结束后，需调用注销接口注销。需要注意的是，当连接超时之后，会话会被置为失效状态。

5.1 登录

5.1.1 接口

接口描述	msgResponse login(msgRequest req);

5.1.2 请求

msgRequest

通用类参数	参数说明	取值
respid	请求 ID，登录请求 reqid=0，登录成功后每次请求该字段递增	0
msgType	请求类型	0x01
version	消息版本，jsonBody 模板的版本	0x01
extend	扩展码	0x00
jsonbody	可变参数，为 json 格式	见下表

可变参数	参数说明	备注
appName	应用名称,若是网页的话，为 host 字段	由 KOAL 分配
appID	应用 ID	由 KOAL 分配
token	应用令牌	由 KOAL 分配

josnBody 示例：

```
{
   "appName":"",
```


格尔软件股份有限公司/上海格尔安全科技有限公司
上海市静安区江场西路 299 弄 5 号中铁时代广场 4 号楼 6 楼 www.koal.com

	"appID":"", "token":"" }

5.1.3 响应

msgResponse

通用类参数	参数说明	取值
respid	请求 ID，登录请求 reqid=0，登录成功后每次请求该字段递增	0
msgType	请求类型	0x01
version	消息版本，jsonBody 模板的版本	0x01
extend	扩展码	0x00
errCode	错误码，0 为执行成功，其他错误码转换为 16 进制见附录 9.1	0 或非 0
jsonbody	可变参数，为 json 格式	见下表

成功：

可变参数	参数说明	备注
sessionID	会话 id	
ticket	会话 ticket	
notifyPort	端口	
timeout	接口调用超时时间（单位：秒）	默认 30s

jsonBody 示例：

```
{
    "sessionID": "1592816002006",
    "ticket": "k7xUJ+RuGgd4jepABhLcyDGINOJQCHax",
    "notifyPort": "18080",
    "timeout": "30"
}
```

失败：

	可变参数 | 参数说明 | 备注
	msg | 接口调用失败时，错误信息提示 |
	
	jsonBody 示例：
	{
	"msg": "failed"
	}

5.2 注销

应用程序注销中间件的应用会话 session。

5.2.1接口

接口描述	msgResponse logout(sessionTicket tk);

5.2.2请求

sessionTicket	通用类参数	参数说明	取值
	sessionID	会话 id	取 5.1 章节登录接口的响应值
	ticket	会话 ticket	取 5.1 章节登录接口的响应值

5.3 获取推送消息

中间件可推送消息供上层应用使用，目前支持的消息类型有设备插入、设备拔出。

5.3.1接口

接口描述	msgResponse getNotify(sessionTicket tk, msgRequest req);

5.3.2请求

sessionTicket	通用类参数	参数说明	取值
	sessionID	会话 id	取 5.1 章节登录接口的响应值

	ticket	会话 ticket	取 5.1 章节登录接口的响应值

msgRequest

通用类参数	参数说明	取值
respid	请求 ID，登录请求 reqid=0，登录成功后每次请求该字段递增	1
msgType	请求类型	0x0FFF0000
version	消息版本，jsonBody 模板的版本	0x01
extend	扩展码	0x00

5.3.3响应

msgResponse

通用类参数	参数说明	取值
respid	请求 ID，登录请求 reqid=0，登录成功后每次请求该字段递增	1
msgType	请求类型，0x0FFF0001 为设备插入，0x0FFF0002 为设备拔出，0x0FFF0003 为设备信息修改推送（暂不支持），0x0FFF0004 为 session 关闭（暂不支持）	0x0FFF0001 0x0FFF0002 0x0FFF0003 0x0FFF0004
version	消息版本，jsonBody 模板的版本	0x01
extend	扩展码	0x00
errCode	错误码，0 为执行成功，其他错误码转换为 16 进制见附录 9.1	0 或非 0
jsonbody	可变参数，为 json 格式	见下表

成功：

可变参数	参数说明	备注
devID	系统自定义的设备的编号	
devNumber	设备编号，设备自带	
devLable	设备标签，可以用户设置	
providerName	提供商名称	

jsonBody 示例：

```
{
    "devID": "6005_Fnw2vXXgW1CNMc323osMUynM8Bl",
```

<table>
<tr><td></td><td>
"devNumber": "28C34DFB80",
"devLable": "ePass Token",
"providerName": "FEITIAN InterPass3000 CSP For NhBank V1.0"
}

失败：

<table>
<tr><th>可变参数</th><th>参数说明</th><th>备注</th></tr>
<tr><td>msg</td><td>接口调用失败时，错误信息提示</td><td></td></tr>
</table>

jsonBody 示例：
{
"msg": "failed"
}
</td></tr>
</table>

5.4 版本获取

5.4.1接口

接口描述	msgResponse getVersions(sessionTicket tk,msgRequest req);

5.4.2请求

<table>
<tr><td>sessionTicket</td><td>
<table>
<tr><th>通用类参数</th><th>参数说明</th><th>取值</th></tr>
<tr><td>sessionID</td><td>会话 id</td><td>取 5.1 章节登录接口的响应值</td></tr>
<tr><td>ticket</td><td>会话 ticket</td><td>取 5.1 章节登录接口的响应值</td></tr>
</table>
</td></tr>
<tr><td>msgRequest</td><td>
<table>
<tr><th>通用类参数</th><th>参数说明</th><th>取值</th></tr>
<tr><td>respid</td><td>请求 ID，登录请求 reqid=0，登录成功后每次请求该字段递增</td><td>1</td></tr>
<tr><td>msgType</td><td>请求类型</td><td>0x02</td></tr>
<tr><td>version</td><td>消息版本，jsonBody 模板的版本</td><td>0x01</td></tr>
<tr><td>extend</td><td>扩展码</td><td>0x00</td></tr>
</table>
</td></tr>
</table>

5.4.3响应

msgResponse

通用类参数	参数说明	取值
respid	请求 ID，登录请求 reqid=0，登录成功后每次请求该字段递增	1
msgType	请求类型	0x02
version	消息版本，jsonBody 模板的版本	0x01
extend	扩展码	0x00
errCode	错误码，0 为执行成功，其他错误码转换为 16 进制见附录 9.1	0 或非 0
jsonbody	可变参数，为 json 格式	见下表

成功：

可变参数	参数说明	备注
KPKIVersion	主干版本	
pluginVersion.deviceOperator	设备插件版本	
pluginVersion.enrollPlugin	发证插件版本	
pluginVersion.kmailPlugin	邮件插件版本	
pluginVersion.signxPlugin	用证插件版本	

jsonBody 示例：

```
{
    "KPKIVersion": "",
    "pluginVersion": {
        "deviceOperator": "",
        "enrollPlugin": "",
        "kmailPlugin": "",
        "signxPlugin": ""
    }
}
```

失败：

	可变参数 | 参数说明 | 备注
	msg | 接口调用失败时，错误信息提示 |
	
	jsonBody 示例：
	{
	"msg": "failed"
	}

5.5 获取系统信息

5.5.1接口

接口描述	msgResponse getSysInfo(sessionTicket tk,msgRequest req);

5.5.2请求

sessionTicket	通用类参数	参数说明	取值
	sessionID	会话 id	取 5.1 章节登录接口的响应值
	ticket	会话 ticket	取 5.1 章节登录接口的响应值
msgRequest	通用类参数	参数说明	取值
	respid	请求 ID，登录请求 reqid=0，登录成功后每次请求该字段递增	1
	msgType	请求类型	0x03
	version	消息版本，jsonBody 模板的版本	0x01
	extend	扩展码	0x00

5.5.3响应

msgResponse	通用类参数	参数说明	取值
	respid	请求 ID，登录请求 reqid=0，登录成功后每次请求该字段递增	1
	msgType	请求类型	0x03
	version	消息版本，jsonBody 模板的版本	0x01
	extend	扩展码	0x00

errCode	错误码，0 为执行成功，其他错误码转换为 16 进制见附录 9.1	0 或非 0
jsonbody	可变参数，为 json 格式	见下表

成功：

可变参数	参数说明	备注
sysName	系统名称	
arch	系统架构	

jsonBody 示例：

```
{
    "sysName":"Windows 10 Enterprise",
    "arch":"x64"
}
```

失败：

可变参数	参数说明	备注
msg	接口调用失败时，错误信息提示	

jsonBody 示例：

```
{
    "msg": "failed"
}
```

5.6 设置可信驱动

可信驱动信息由集成应用从驱动方获取，同时需要传入与系统对应的可信驱动信息，系统信息可调用 5.5 章节所示接口，最终选择并传入相应的驱动可信信息。

windows 系统需要传入驱动对应的 64 位库和 32 位库。

5.6.1接口

接口描述	msgResponse setTrustedDrives(sessionTicket tk,msgRequest req);

5.6.2请求

sessionTicket

通用类参数	参数说明	取值
sessionID	会话 id	取 5.1 章节登录接口的响应值
ticket	会话 ticket	取 5.1 章节登录接口的响应值

msgRequest

通用类参数	参数说明	取值
respid	请求 ID，登录请求 reqid=0，登录成功后每次请求该字段递增	1
msgType	请求类型	0x04
version	消息版本，jsonBody 模板的版本	0x01
extend	扩展码	0x00
jsonbody	可变参数，为 json 格式	见下表

可变参数	参数说明	备注
name	驱动名称	
path	驱动路径、名称	驱动厂商提供
hash	驱动库的 sha1 值	驱动厂商提供
comment	备注。可传入当前系统的名称、架构	中文建议 utf8
trustedMode	驱动可信模式: 1 表示模式一，集成应用传入可信驱动信息对比校验; 其他模式待定	

jsonBody 示例：

```
{
    "trustedMode":"1",
    "drives":[
        {
            "name":"Longmai",
            "path":"c:\windows\system32\shuttlecsp11_3000GM",
```

<table>
<tr><td></td><td>"hash":"5dfbc5210f86b7b8d17b95d528c53067046d594e",
"comment":"windows 32bit longmai ukey",
},
]
}</td></tr>
</table>

5.6.3响应

<table>
<tr><td>msgResponse</td><td>
<table>
<tr><th>通用类参数</th><th>参数说明</th><th>取值</th></tr>
<tr><td>respid</td><td>请求 ID，登录请求 reqid=0，登录成功后每次请求该字段递增</td><td>1</td></tr>
<tr><td>msgType</td><td>请求类型</td><td>0x04</td></tr>
<tr><td>version</td><td>消息版本，jsonBody 模板的版本</td><td>0x01</td></tr>
<tr><td>extend</td><td>扩展码</td><td>0x00</td></tr>
<tr><td>errCode</td><td>错误码，0 为执行成功，其他错误码转换为 16 进制见附录 9.1</td><td>0 或非 0</td></tr>
<tr><td>jsonbody</td><td>可变参数，为 json 格式</td><td>见下表</td></tr>
</table>

成功：

<table>
<tr><th>可变参数</th><th>参数说明</th><th>备注</th></tr>
<tr><td>name</td><td>驱动名称</td><td></td></tr>
<tr><td>Path</td><td>驱动路径、名称</td><td></td></tr>
<tr><td>hash</td><td>驱动库的摘要值</td><td>sha1</td></tr>
<tr><td>status</td><td>可信结果 true/false</td><td></td></tr>
<tr><td>trustedMode</td><td>驱动可信模式:
1.模式一，集成应用传入可信驱动信息对比校验;
其他模式待定</td><td></td></tr>
</table>

jsonBody 示例：
{
"trustedMode":"1",
"drives":[
</td></tr>
</table>

```
        {
            "name":"Longmai",
            "path":"c:\windows\system32\shuttlecsp11_3000GM",
            "hash":"5dfbc5210f86b7b8d17b95d528c53067046d594e",
            "status":"true",
},
    ]
}
```


注：存在需要设置多个驱动的场景，在此仅举例一个

失败：

可变参数	参数说明	备注
msg	接口调用失败时，错误信息提示	

jsonBody 示例：


```
{
    "msg": "failed"
}
```


5.7 获取登录临时参数

5.7.1接口

接口描述	msgResponse getLoginTempParam(sessionTicket tk,msgRequest req);

5.7.2 请求

sessionTicket	通用类参数	参数说明	取值
	sessionID	会话 id	取 5.1 章节登录接口的响应值
	ticket	会话 ticket	取 5.1 章节登录接口的响应值

msgRequest	**通用类参数**	**参数说明**	**取值**
	respid	请求 ID，登录请求 reqid=0，登录成功后每次请求该字段递增	1
	msgType	请求类型	0x05
	version	消息版本，jsonBody 模板的版本	0x01
	extend	扩展码	0x00

5.7.3 响应

msgResponse

通用类参数	**参数说明**	**取值**
respid	请求 ID，登录请求 reqid=0，登录成功后每次请求该字段递增	1
msgType	请求类型	0x05
version	消息版本，jsonBody 模板的版本	0x01
extend	扩展码	0x00
errCode	错误码，0 为执行成功，其他错误码转换为 16 进制见附录 9.1	0 或非 0
jsonbody	可变参数，为 json 格式	见下表

成功：

可变参数	**参数说明**	**备注**
appName	应用名称	
appID	应用 ID	
token	会话 Token	

jsonBody 示例：

```
{
   "appName":"",
   "appID":"",
   "token":""
}
```

失败：

可变参数	参数说明	备注
msg	接口调用失败时，错误信息提示	

jsonBody 示例：

```
{
    "msg": "failed"
}
```

6 设备管理接口

设备管理接口作为业务处理的基础接口，主要涉及与介质交互，可为应用提供设备认证、列证、列应用、列容器、列设备、PIN 码修改、验证及解锁等一系列功能。

6.1 校验 PIN 码

6.1.1接口

接口描述	msgResponse verifyPIN(sessionTicket tk, msgRequest req);

6.1.2请求

sessionTicket

通用类参数	参数说明	取值
sessionID	会话 id	取 5.1 章节登录接口的响应值
ticket	会话 ticket	取 5.1 章节登录接口的响应值

msgRequest

通用类参数	参数说明	取值
respid	请求 ID，登录请求 reqid=0，登录成功后每次请求该字段递增	0x01
msgType	请求类型	0x18
version	消息版本，jsonBody 模板的版本	0x01
extend	扩展码	0x00
jsonbody	可变参数，为 json 格式	见下表

可变参数	参数说明	备注
devID	设备 ID	
appName	应用名称	
PINType	PIN 类型	
PIN	PIN 码	
isCachedPIN	是否为密文缓存 PIN，0 为不是，1 为是	默认 0

	jsonBody 示例：
	{
	"devID":"6009_gnqIoO99FHlJQUudLE7eQYaxHlb",
	"appName":"GM3000ECC",
	"PINType":"1",
	"PIN":"123456",
	"isCachedPIN":"0"
	}

6.1.3响应

msgResponse

通用类参数	参数说明	取值
respid	请求 ID，登录请求 reqid=0，登录成功后每次请求该字段递增	0x01
msgType	请求类型	0x18
version	消息版本，jsonBody 模板的版本	0x01
extend	扩展码	0x00
errCode	错误码，0 为执行成功，其他错误码转换为 16 进制见附录 9.1	0 或非 0
jsonbody	可变参数，为 json 格式	见下表

可变参数	参数说明	备注
devID	设备 ID	
appName	应用名称	
PINType	PIN 类型	
pulRetryCount	错误后返回的剩余重试次数	
msg	错误消息提示	

jsonBody 示例：

{

"devID": "6009_gnqIoO99FHlJQUudLE7eQYaxHlb",

"appName": "GM3000ECC",

"PINType": "1",

"pulRetryCount": "",

"msg": "successful"

	}

6.2 获取缓存 PIN 码

6.2.1接口

接口描述	msgResponse getCachedPIN(sessionTicket tk, msgRequest req);

6.2.2请求

sessionTicket

通用类参数	参数说明	取值
sessionID	会话 id	取 5.1 章节登录接口的响应值
ticket	会话 ticket	取 5.1 章节登录接口的响应值

msgRequest

通用类参数	参数说明	取值
respid	请求 ID，登录请求 reqid=0，登录成功后每次请求该字段递增	0x01
msgType	请求类型	0x51
version	消息版本，jsonBody 模板的版本	0x01
extend	扩展码	0x00
jsonbody	可变参数，为 json 格式	见下表

可变参数	参数说明	备注
devID	设备 ID	
appName	应用名称	
PINType	PIN 类型	

jsonBody 示例：

```
{
    "devID":"",
    "devSerNum":"",
    "PINType":""
}
```

6.2.3响应

msgResponse

通用类参数	参数说明	取值
respid	请求 ID，登录请求 reqid=0，登录成功后每次请求该字段递增	0x01
msgType	请求类型	0x51
version	消息版本，jsonBody 模板的版本	0x01
extend	扩展码	0x00
errCode	错误码，0 为执行成功，其他错误码转换为 16 进制见附录 9.1	0 或非 0
jsonbody	可变参数，为 json 格式	见下表

成功：

可变参数	参数说明	备注
devID	设备 ID	
devSerNum	设备序列号	
devVidPid	设备 vidpid	
appName	应用名称	
PINType	PIN 类型	
PIN	PIN 缓存	
createdTime	PIN 缓存时间戳	

jsonBody 示例：

```
{
    "devID":"",
    "devSerNum":"",
    "devVidPid":"",
    "appName":"",
    "PINType":"",
    "PIN":"",
    "createdTime":""
}
```

失败：

可变参数	参数说明	备注
msg	接口调用失败时，错误信息提示	

jsonBody 示例：

```
{
    "msg": "failed"
}
```

6.3 清除应用安全状态

6.3.1接口

接口描述	msgResponse clearAppSecState(sessionTicket tk, msgRequest req);

6.3.2请求

sessionTicket

通用类参数	参数说明	取值
sessionID	会话 id	取 5.1 章节登录接口的响应值
ticket	会话 ticket	取 5.1 章节登录接口的响应值

msgRequest

通用类参数	参数说明	取值
respid	请求 ID，登录请求 reqid=0，登录成功后每次请求该字段递增	0x01
msgType	请求类型	0x3b
version	消息版本，jsonBody 模板的版本	0x01
extend	扩展码	0x00
jsonbody	可变参数，为 json 格式	见下表

可变参数	参数说明	备注
devID	设备 ID	
appName	应用名称	

jsonBody 示例：

```
{
```

	"devID":"6009_gnqIoO99FHlJQUudLE7eQYaxHlb", "appName":"GM3000ECC" }

6.3.3响应

msgResponse

通用类参数	参数说明	取值
respid	请求 ID，登录请求 reqid=0，登录成功后每次请求该字段递增	0x01
msgType	请求类型	0x3b
version	消息版本，jsonBody 模板的版本	0x01
extend	扩展码	0x00
errCode	错误码，0 为执行成功，其他错误码转换为 16 进制见附录 9.1	0 或非 0

失败：

可变参数	参数说明	备注
msg	接口调用失败时，错误信息提示	

jsonBody 示例：

```
{
    "msg": "failed"
}
```

6.4 导出数字证书

6.4.1接口

接口描述	msgResponse exportCertificate(sessionTicket tk, msgRequest req);

6.4.2请求

sessionTicket

通用类参数	参数说明	取值
sessionID	会话 id	取 5.1 章节登录接口的响应值
ticket	会话 ticket	取 5.1 章节登录接口的响应值

msgRequest

通用类参数	参数说明	取值
respid	请求 ID，登录请求 reqid=0，登录成功后每次请求该字段递增	0x01
msgType	请求类型	0x22
version	消息版本，jsonBody 模板的版本	0x01
extend	扩展码	0x00
jsonbody	可变参数，为 json 格式	见下表

可变参数	参数说明	备注
devID	设备 ID	
appName	应用名称	
containerName	容器名称	
signFlag	1 表示签名证书，0 表示加密证书	

jsonBody 示例：

```
{
    "devID":"6009_gnqIoO99FHlJQUudLE7eQYaxHlb",
    "appName":"GM3000ECC",
    "containerName":"con1",
    "signFlag":"1"
}
```

6.4.3响应

msgResponse

通用类参数	参数说明	取值
respid	请求 ID，登录请求 reqid=0，登录成功后每次请求该字段递增	0x01
msgType	请求类型	0x22

version	消息版本，jsonBody 模板的版本	0x01
extend	扩展码	0x00
errCode	错误码，0 为执行成功，其他错误码转换为 16 进制见附录 9.1	0 或非 0
jsonbody	可变参数，为 json 格式	见下表

成功：

可变参数	参数说明	备注
devID	设备 ID	
appName	应用名称	
containerName	容器名称	
signFlag	1 表示签名证书，0 表示加密证书	
cert	证书内容	base64 编码

jsonBody 示例：

```
{
    "devID":"6009_gnqIoO99FHlJQUudLE7eQYaxHlb",
    "appName":"GM3000ECC",
    "containerName":"con1",
    "signFlag":"1",
    "cert":"xxx"
}
```

失败：

可变参数	参数说明	备注
msg	接口调用失败时，错误信息提示	

jsonBody 示例：

```
{
    "msg": "failed"
}
```

6.5 导出公钥

6.5.1接口

接口描述	msgResponse exportPublicKey(sessionTicket tk, msgRequest req);

6.5.2请求

sessionTicket

通用类参数	参数说明	取值
sessionID	会话 id	取 5.1 章节登录接口的响应值
ticket	会话 ticket	取 5.1 章节登录接口的响应值

msgRequest

通用类参数	参数说明	取值
respid	请求 ID，登录请求 reqid=0，登录成功后每次请求该字段递增	0x01
msgType	请求类型	0x23
version	消息版本，jsonBody 模板的版本	0x01
extend	扩展码	0x00
jsonbody	可变参数，为 json 格式	见下表

可变参数	参数说明	备注
devID	设备 ID	
appName	应用名称	
containerName	容器名称	
signFlag	1 表示签名公钥，0 表示加密公钥	

jsonBody 示例：

```
{
    "devID":"6009_gnqIoO99FHlJQUudLE7eQYaxHlb",
    "appName":"GM3000ECC",
    "containerName":"con1",
    "signFlag":"1"
}
```

6.5.3响应

msgResponse

通用类参数	参数说明	取值
respid	请求 ID，登录请求 reqid=0，登录成功后每次请求该字段递增	0x01
msgType	请求类型	0x23
version	消息版本，jsonBody 模板的版本	0x01
extend	扩展码	0x00
errCode	错误码，0 为执行成功，其他错误码转换为 16 进制见附录 9.1	0 或非 0
jsonbody	可变参数，为 json 格式	见下表

成功：

可变参数	参数说明	备注
devID	设备 ID	
appName	应用名称	
containerName	容器名称	
signFlag	1 表示签名公钥，0 表示加密公钥	
publicKey	公钥内容	base64 编码

jsonBody 示例：

```
{
    "devID":"6009_gnqIoO99FHlJQUudLE7eQYaxHlb",
    "appName":"GM3000ECC",
    "containerName":"con1",
    "signFlag":"1",
    "publicKey":"xxx"
}
```

失败：

可变参数	参数说明	备注
msg	接口调用失败时，错误信息提示	

	jsonBody 示例：
	{
	"msg": "failed"
	}

6.6 获取证书列表

6.6.1接口

接口描述	msgResponse getAllCert(sessionTicket tk, msgRequest req);

6.6.2请求

sessionTicket

通用类参数	参数说明	取值
sessionID	会话 id	取 5.1 章节登录接口的响应值
ticket	会话 ticket	取 5.1 章节登录接口的响应值

msgRequest

通用类参数	参数说明	取值
respid	请求 ID，登录请求 reqid=0，登录成功后每次请求该字段递增	0x01
msgType	请求类型	0x28
version	消息版本，jsonBody 模板的版本	0x01
extend	扩展码	0x00
jsonbody	可变参数，为 json 格式	见下表

可变参数	参数说明	备注
includeSN	展示证书序列号数组 SN 下的证书，josn 数组格式	可选

jsonBody 示例：

```
{
    "includeSN":[
        "2025000000000000000000001B",
        "2025000000000000000000001C"
    ]
}
```

6.6.3响应

msgResponse

通用类参数	参数说明	取值
respid	请求 ID，登录请求 reqid=0，登录成功后每次请求该字段递增	0x01
msgType	请求类型	0x28
version	消息版本，jsonBody 模板的版本	0x01
extend	扩展码	0x00
errCode	错误码，0 为执行成功，其他错误码转换为 16 进制见附录 9.1	0 或非 0
jsonbody	可变参数，为 json 格式	见下表

成功：

可变参数	参数说明	备注
devID	设备 ID	
devName	设备名称	
manufacturer	设备厂商	
devProvider	设备 provider	
devSn	设备序列号	
appName	应用名称	
containerName	容器名称	
hasPIN	是否有 PIN 码	
hasFinger	是否具有指纹	0：非指纹 key； 1：指纹 key，应用未录入指纹； 2：指纹 key，存在指纹； 0xff：未知类型
signFlag	证书用途，1 表示签名，0 表示加密	
subjectName	使用者,所有子项,无数据时返回{}	
issuerName	颁发者,所有子项,无数据时返回{}	
SN	序列号	
actionDate	生效时间	
validDate	失效时间	
certType	证书类型	
keyUsage	秘钥类型,0 表示加密,1 表示签名,2 表示签名加密	

jsonBody 示例：


```
{
    "certs":[
        {
            "devID": "7726_wfThjIlu6Lvud5goIuY1\/0NG8k\/",
            "devName": "93A101CFABB76FB71669EDD81EF0CDE",
            "manufacturer": "Longmai",
```

```
                "devProvider": "Longmai",
                "devSn": "93A101CFABB76FB71669EDD81EF0CDE",
                "appName": "KOAL_NDS",
                "containerName": "KOAL_NDS_LIC",
                "hasPIN": "1",
                "hasFinger": "1",
                "signFlag": "0",
                "subjectName": {
                    "C": "CN",
                    "CN": "xxx",
                     .....
                },
                "issuerName": {
                    "CN": "xxx"
                },
                "SN": "01224F91F88D15",
                "actionDate": "2021-01-14 14:15:41",
                "validDate": "2022-01-14 14:15:41",
                "certType": "RSA",
                "keyUsage": "2"
            }
        ]
}
```


备注：存在同时有较多证书的情况，在此仅列举一个

失败：

可变参数	参数说明	备注
msg	接口调用失败时，错误信息提示	

jsonBody 示例：


```
{
    "msg": "failed"
}
```

6.7 验证指纹

6.7.1接口

接口描述	msgResponse verifyFinger(sessionTicket tk,msgRequest req);

6.7.2请求

sessionTicket

通用类参数	参数说明	取值
sessionID	会话 id	取 5.1 章节登录接口的响应值
ticket	会话 ticket	取 5.1 章节登录接口的响应值

msgRequest

通用类参数	参数说明	取值
respid	请求 ID，登录请求 reqid=0，登录成功后每次请求该字段递增	0x01
msgType	请求类型	0x32
version	消息版本，jsonBody 模板的版本	0x01
extend	扩展码	0x00
jsonbody	可变参数，为 json 格式	见下表

可变参数	参数说明	备注
devID	设备 ID	
appName	设备名称	
type	指纹类型	

jsonBody 示例：

```
{
    "devID":"6009_gnqIoO99FHlJQUudLE7eQYaxHlb",
    "appName":"xxx",
    "type":"1",
}
```

6.7.3响应

msgResponse

通用类参数	参数说明	取值
respid	请求 ID，登录请求 reqid=0，登录成功后每次请	0x01

	求该字段递增	
msgType	请求类型	0x32
version	消息版本，jsonBody 模板的版本	0x01
extend	扩展码	0x00
errCode	错误码，0 为执行成功，其他错误码转换为 16 进制见附录 9.1	0 或非 0
jsonbody	可变参数，为 json 格式	见下表

成功：

可变参数	参数说明	备注
devID	设备 ID	
appName	应用名称	
type	指纹类型	
pulRetryCount	错误后返回的剩余重试次数	
msg	错误信息提示	

jsonBody 示例：

```
{
    "devID":"6009_gnqIoO99FHlJQUudLE7eQYaxHlb",
    "appName":"App1",
    "type":"1",
    "pulRetryCount":""
    "msg":"successful"
}
```

6.8 生成随机数

6.8.1接口

接口描述	msgResponse genRandom(sessionTicket tk, msgRequest req);

6.8.2请求

sessionTicket

通用类参数	参数说明	取值
sessionID	会话 id	取 5.1 章节登录接口的响应值
ticket	会话 ticket	取 5.1 章节登录接口的响应值

msgRequest

通用类参数	参数说明	取值
respid	请求 ID，登录请求 reqid=0，登录成功后每次请求该字段递增	0x01
msgType	请求类型	0x42
version	消息版本，jsonBody 模板的版本	0x01
extend	扩展码	0x00
jsonbody	可变参数，为 json 格式	见下表

可变参数	参数说明	备注
devID	设备 ID	
randomLen	要产生的随机数长度	字节

jsonBody 示例：

```
{
    "devID":"6009_gnqIoO99FHlJQUudLE7eQYaxHlb",
    "randomLen":"16"
}
```

6.8.3响应

msgResponse

通用类参数	参数说明	取值
respid	请求 ID，登录请求 reqid=0，登录成功后每次请求该字段递增	0x01
msgType	请求类型	0x42
version	消息版本，jsonBody 模板的版本	0x01
extend	扩展码	0x00
errCode	错误码，0 为执行成功，其他错误码转换为 16 进制见附录 9.1	0 或非 0
jsonbody	可变参数，为 json 格式	见下表

成功：

可变参数	参数说明	备注
b64Random	Base64 编码的随机数，json 格式	

jsonBody 示例：

```
{
    "b64Random": "iQjTbWshJOkWrEVXlyDs5g=="
}
```

失败：

可变参数	参数说明	备注
msg	接口调用失败时，错误信息提示	

jsonBody 示例：

```
{
    "msg": " failed"
}
```

7 用证接口

用证接口主要包括普通签名验签接口、PKCS#7 签名验签接口、组（解）PKCS#7 数字信封接口、解析证书接口等，应用可通过调用接口实现用证、证书详情获取等业务。

7.1 数据签名

7.1.1接口

接口描述	msgResponse signData(sessionTicket tk, msgRequest req);

7.1.2请求

sessionTicket

通用类参数	参数说明	取值
sessionID	会话 id	取 5.1 章节登录接口的响应值
ticket	会话 ticket	取 5.1 章节登录接口的响应值

msgRequest

通用类参数	参数说明	取值
respid	请求 ID，登录请求 reqid=0，登录成功后每次请求该字段递增	0x01
msgType	请求类型	0x10
version	消息版本，jsonBody 模板的版本	0x01
extend	扩展码	0x00
jsonbody	可变参数，为 json 格式	见下表

可变参数	参数说明	备注
devID	设备 ID	
appName	应用名称	
conName	容器名称	
srcData	源数据	base64 编码
isBase64SrcData	是否为 base64 编码源数据，1 表示是，0 表示否	默认使用 1
type	1 表示 PM-BD 签名，2 表示 SM2/RSA 签名，3 SSL 建链定制签名，4 表示银行二代 key 签名(需要屏显报文)	默认使用 2
mdType	摘要类型，1 表示 MD5，2 表示 SHA1，3 表示 SM3，4 表示 SHA256	摘要与证书类型关系 (SM2-SM3 RSA-SHA1\MD5\SHA256)

jsonBody 示例：

```
{
    "devID":"6009_gnqIoO99FHlJQUudLE7eQYaxHlb",
    "appName":"GM3000ECC",
    "conName":"con1",
```

<table>
<tr><td></td><td>"srcData":"MTIzNDU2NzgxMjM0NTY3OA==",
"isBase64SrcData":"1",
"type":"2",
"mdType":"2"
}</td></tr>
</table>

7.1.3响应

msgResponse

通用类参数	参数说明	取值
respid	请求 ID，登录请求 reqid=0，登录成功后每次请求该字段递增	0x01
msgType	请求类型	0x10
version	消息版本，jsonBody 模板的版本	0x01
extend	扩展码	0x00
errCode	错误码，0 为执行成功，其他错误码转换为 16 进制见附录 9.1	0 或非 0
jsonbody	可变参数，为 json 格式	见下表

成功：

可变参数	参数说明	备注
b64signData	签名数据	base64 编码

jsonBody 示例：

```
{
    "b64signData":"xxx"
}
```

失败：

可变参数	参数说明	备注
msg	接口调用失败时，错误信息提示	base64 编码

	jsonBody 示例：
	{
	"msg": "failed"
	}

7.2 数据签名-PKCS#7

7.2.1接口

接口描述	msgResponse signMessage(sessionTicket tk, msgRequest req);

7.2.2请求

sessionTicket	**通用类参数**	**参数说明**	**取值**
	sessionID	会话 id	取 5.1 章节登录接口的响应值
	ticket	会话 ticket	取 5.1 章节登录接口的响应值
msgRequest	**通用类参数**	**参数说明**	**取值**
	respid	请求 ID，登录请求 reqid=0，登录成功后每次请求该字段递增	0x01
	msgType	请求类型	0x12
	version	消息版本，jsonBody 模板的版本	0x01
	extend	扩展码	0x00
	jsonbody	可变参数，为 json 格式	见下表

可变参数	参数说明	备注
devID	设备 ID	
appName	应用名称	
conName	容器名称	
srcData	待签数据	base64 编码
attachData	0 表示 detached 方式签名，1 表示 attached	
mdType	指定的摘要类型，"1"-MD5 "2"-SHA1 "3"-SM3 "4"-SHA256	
signwithSM2Std	用于 sm2 签名，1 表示使用 SM2 规范，0 表示使用 RFC 规范	默认 0
isSignedWithKey2G	是否使用银行二代 key 签名（需要屏显报文，0 表示不使用，1 表示使用	默认 0
noAttr	不携带属性，1 表示是，0 表示否	默认 0

jsonBody 示例：

```
{
    "devID":"6009_gnqIoO99FHlJQUudLE7eQYaxHlb",
    "appName":"GM3000ECC",
    "conName":"con1",
    "srcData":"MTIzNDU2NzgxMjM0NTY3OA==",
    "attachData":"0",
    "mdType":"3",
    "signwithSM2Std":"0",
    "isSignedWithKey2G":"0",
    "noAttr":"0"
}
```

7.2.3响应

msgResponse	通用类参数	参数说明	取值
	respid	请求 ID，登录请求 reqid=0，登录成功后每次请	0x01

	求该字段递增	
msgType	请求类型	0x12
version	消息版本，jsonBody 模板的版本	0x01
extend	扩展码	0x00
errCode	错误码，0 为执行成功，其他错误码转换为 16 进制见附录 9.1	0 或非 0
jsonbody	可变参数，为 json 格式	见下表

成功：

可变参数	参数说明	备注
signData	签名数据	base64 编码

jsonBody 示例：

```
{
    "signData":"xxx"
}
```

失败：

可变参数	参数说明	备注
msg	接口调用失败时，错误信息提示	

jsonBody 示例：

```
{
    "msg": "failed"
}
```

7.3 解析证书

7.3.1接口

接口描述	msgResponse parseCert(sessionTicket tk, msgRequest req);

7.3.2请求

sessionTicket

通用类参数	参数说明	取值
sessionID	会话 id	取 5.1 章节登录接口的响应值
ticket	会话 ticket	取 5.1 章节登录接口的响应值

msgRequest

通用类参数	参数说明	取值
respid	请求 ID，登录请求 reqid=0，登录成功后每次请求该字段递增	0x01
msgType	请求类型	0x17
version	消息版本，jsonBody 模板的版本	0x01
extend	扩展码	0x00
jsonbody	可变参数，为 json 格式	见下表

可变参数	参数说明	备注
cert	证书内容	base64 编码

jsonBody 示例：

```
jsonBody    =   {
      "cert":"xxx"
}
```

7.3.3响应

msgResponse

通用类参数	参数说明	取值
respid	请求 ID，登录请求 reqid=0，登录成功后每次请求该字段递增	0x01
msgType	请求类型	0x17
version	消息版本，jsonBody 模板的版本	0x01
extend	扩展码	0x00
errCode	错误码，0 为执行成功，其他错误码转换为 16 进制见附录 9.1	0 或非 0
jsonbody	可变参数，为 json 格式	见下表

成功：

可变参数	参数说明	备注
version	版本	
certType	证书类型	
SN	序列号	
actionDate	生效时间	
validDate	失效时间	
subject	使用者，所有 subject 子项，无数据时返回{}	
issuer	颁发者。所有 issuer 子项，无数据时返回{}	
keyUsage	证书密钥用途，0 表示加密证书，1 表示签名证书，2 表示证书既可签名又可加密，其他值表示异常	
subjectPublicKeyInfo.Algorithm	公钥算法	
subjectPublicKeyInfo.Param	公钥参数	
subjectPublicKeyInfo.BitLen	公钥长度	
subjectPublicKeyInfo.Content	公钥内容	
subjectPublicKeyInfo.Exponent	公钥指数	
extensions	证书扩展项，所有扩展项包括每项的内容、是否为关键扩展，无数据时返回{}	
signatureInfo.sigAlg	签名算法	
signatureInfo.signature	证书签名	
hash	证书指纹	

jsonBody 示例：


```
{
    "version": "V3",
    "certType": "SM2",
    "SN": "5DE10000000000000000003F",
```

```
"actionDate": "2019-10-31 00:00:00",
"validDate": "2022-10-30 23:59:59",
"subject": {
    "C": "CN",
    "CN": "onlinetestno1"
},
"issuer": {
    "C": "CN",
    "CN": "local_ca_sm"
},
"keyUsage": "1",
"subjectPublicKeyInfo": {
    "Algorithm": "id-ecPublicKey",
    "Param": "1.2.156.10197.1.301",
    "BitLen": "256",
    "Content": "04:78:02:df:da:de:46:95:e4:7b:d3:76:30:50:
                22:5b:54:04:ab:52:2a:c7:1b:4d:fe:42:82:a1:
                93:83:b2:d1:ca:a0:70:94:91:31:24:28:54:1c:
                17:0e:a7:ab:1c:7c:3f:c0:38:f4:3e:75:0c:76:
                7c:5c:ac:ed:dd:be:68:2d:c0",
    "Exponent": ""
},
"extensions": {
    "Netscape Cert Type": {
        "data": "SSL Client",
        "critical": "0"
    },
    "X509v3 Authority Key Identifier": {
        "data":
"keyid:FD:F2:45:7B:0C:B9:99:F0:42:8E:DE:AF:C2:F3:8A:F8:CA:F2:4B:91\n",
        "critical": "0"
    },
    "X509v3 Extended Key Usage": {
        "data": "TLS Web Client Authentication, E-mail Protection",
```

```
                "critical": "0"
            },
            "X509v3 Key Usage": {
                "data": "Digital Signature, Non Repudiation",
                "critical": "1"
            },
            "X509v3 Subject Key Identifier": {
                "data":
"DC:D6:8E:FF:DF:75:FF:6C:E4:25:C9:60:07:B3:27:EB:76:BE:FC:6C",
                "critical": "0"
            }
        },
        "signatureInfo": {
            "sigAlg": "CN GM ECDSA Sign/Verify with SM3",
            "signature": "30:44:02:20:2f:a0:cc:df:8c:75:9d:0d:ed:44:
                          f6:6c:16:10:af:7c:c8:10:94:8d:a4:2f:c1:ac:
                          97:4d:72:a9:37:97:8e:aa:02:20:5f:98:0b:46:
                          02:d4:01:48:a4:42:ed:5f:02:66:b9:76:1f:e2:
                          ae:c2:7c:1f:db:87:d8:a7:c2:cc:b9:a5:6a:b1"
        },
        "hash": "d083df724da37f1e65441a2d2c519941229ac693"
}
```


失败：

可变参数	参数说明	备注
msg	接口调用失败时，错误信息提示	

jsonBody 示例：


```
{
    "msg": "failed"
}
```

7.4 组 P7 数字信封

7.4.1接口

接口描述	msgResponse envelopeEncrypt(sessionTicket tk, msgRequest req);

7.4.2请求

sessionTicket

通用类参数	参数说明	取值
sessionID	会话 id	取 5.1 章节登录接口的响应值
ticket	会话 ticket	取 5.1 章节登录接口的响应值

msgRequest

通用类参数	参数说明	取值
respid	请求 ID，登录请求 reqid=0，登录成功后每次请求该字段递增	0x01
msgType	请求类型	0x18
version	消息版本，jsonBody 模板的版本	0x01
extend	扩展码	0x00
jsonbody	可变参数，为 json 格式	见下表

可变参数	参数说明	备注
srcData	明文数据	base64 编码
cert	证书	base64 编码
cihperType	指定的算法类型，"0"-DES(不支持),"1"-3DES,"2"-AES,"3"-SM4,"4"-AES128,"5"-DES-ECB(不支持),"6"-3DES-ECB(不支持),"7"-AES_ECB,"8"-SM4_ECB,"9"-AES128_ECB	

jsonBody 示例：

```
jsonBody    =  {
      "srcData":"MTIzNDU2NzgxMjM0NTY3OA==",
      "cert":"xxx",
      "cihperType":"0"
}
```

7.4.3响应

msgResponse

通用类参数	参数说明	取值
respid	请求 ID，登录请求 reqid=0，登录成功后每次请求该字段递增	0x01
msgType	请求类型	0x18
version	消息版本，jsonBody 模板的版本	0x01
extend	扩展码	0x00
errCode	错误码，0 为执行成功，其他错误码转换为 16 进制见附录 9.1	0 或非 0
jsonbody	可变参数，为 json 格式	见下表

成功：

可变参数	参数说明	备注
envelopeData	数字信封	base64 编码

jsonBody 示例：

```
{
    "envelopeData":"xxx"
}
```

失败：

可变参数	参数说明	备注
msg	接口调用失败时，错误信息提示	base64 编码

jsonBody 示例：

```
{
    "msg": "failed"
}
```

7.5 解 P7 数字信封

7.5.1接口

接口描述	msgResponse envelopeDecrypt(sessionTicket tk, msgRequest req);

7.5.2请求

sessionTicket

通用类参数	参数说明	取值
sessionID	会话 id	取 5.1 章节登录接口的响应值
ticket	会话 ticket	取 5.1 章节登录接口的响应值

msgRequest

通用类参数	参数说明	取值
respid	请求 ID，登录请求 reqid=0，登录成功后每次请求该字段递增	0x01
msgType	请求类型	0x19
version	消息版本，jsonBody 模板的版本	0x01
extend	扩展码	0x00
jsonbody	可变参数，为 json 格式	见下表

可变参数	参数说明	备注
devID	设备 ID	
appName	应用名称	
conName	容器名称	
srcData	信封数据	base64 编码

jsonBody 示例：

```
{
    "devID":"6009_gnqIoO99FHlJQUudLE7eQYaxHlb",
    "appName":"GM3000ECC",
    "conName":"con1",
    "srcData":"xxx"
}
```

7.5.3响应

msgResponse

通用类参数	参数说明	取值
respid	请求 ID，登录请求 reqid=0，登录成功后每次请求该字段递增	0x01
msgType	请求类型	0x19
version	消息版本，jsonBody 模板的版本	0x01
extend	扩展码	0x00
errCode	错误码，0 为执行成功，其他错误码转换为 16 进制见附录 9.1	0 或非 0
jsonbody	可变参数，为 json 格式	见下表

成功：

可变参数	参数说明	备注
plainData	明文	base64 编码

jsonBody 示例：

```
{
    "plainData":"MTIzNDU2NzgxMjM0NTY3OA=="
}
```

失败：

可变参数	参数说明	备注
msg	接口调用失败时，错误信息提示	

jsonBody 示例：

```
{
    "msg": "failed"
}
```

8 附录

8.1 错误代码说明

序号	错误代码	说明
1	0	成功
2	1	session 不存在
3	2	登录状态已注销
4	3	已经注册了 Notify
5	4	消息类型错误
6	5	消息 jsonBody 无效/缺少参数
7	6	app 已经登录
8	7	超时
9	8	登录参数未授权认证
10	0x0A000001	失败
11	0x0A000003	不支持的服务
12	0x0A000006	参数不正确
13	0x0A00001B	密钥未发现
14	0x0A00001C	证书未发现
15	0x0A000023	设备已移除
16	0x0A000024	PIN 不正确
17	0x0A000025	PIN 被锁死
18	0x0A000027	PIN 长度错误
19	0x0A00002A	PIN 类型错误
20	0x0A00002C	应用已经存在
21	0x0A00002D	用户没有登录
22	0x0A00002E	应用不存在
23	0x0A00002F	文件已存在

24	0x0A000031	文件不存在
25	0x0A000032	已达到最大可管理容器数
26	0x0B000035	容器不存在
27	0x0B000036	容器已存在