

移动安全认证系统

Android SDK

接入规范

v2.0.2

目 录

1	文档说明.....	- 4 -
1.1	编制说明.....	- 4 -
1.2	参考资料.....	- 4 -
1.3	适用范围.....	- 4 -
2	接口列表.....	- 4 -
3	集成流程说明.....	- 5 -
4	数据格式定义.....	- 5 -
4.1	用户信息数据格式.....	- 5 -
5	接口定义.....	- 6 -
5.1	MKEY 服务端接口.....	- 6 -
5.1.1	应用授权.....	- 6 -
5.2	MKEYAPI 接口.....	- 7 -
5.2.1	SDK 初始化.....	- 7 -
5.2.2	接口初始化.....	- 8 -
5.2.3	申请证书.....	- 9 -
5.2.4	更新证书.....	- 10 -
5.2.5	注销证书.....	- 11 -
5.2.6	读取证书容器标识列表.....	- 12 -
5.2.7	读取证书.....	- 13 -
5.2.8	读取证书项.....	- 13 -
5.2.9	读取证书 OID 项.....	- 14 -
5.2.10	数字签名.....	- 15 -
5.2.11	验证签名.....	- 16 -
5.2.12	验证密码.....	- 17 -
5.2.13	修改密码.....	- 18 -
5.2.14	解锁密码.....	- 19 -
6	示例代码.....	- 20 -
6.1	参数签名示例.....	- 20 -
7	错误码定义.....	- 21 -

版本历史记录

版本号	更新描述
1.0.0	基础版
2.0.0	PIN 码作为接口参数，增加解锁密码功能
2.0.1	修改应用授权接口地址，增加版本历史记录
2.0.2	增加读取证书容器标识列表接口

1 文档说明

1.1 编制说明

移动安全认证系统简称MKEY。

本文档描述MKEY产品SDK接入规范。

1.2 参考资料

无。

1.3 适用范围

本规范适用于MKEY产品SDK接入规范，描述应用APP与MKEY产品SDK的交互接口说明。

2 接口列表

接口分类	接口名称	接口说明	备注
接口	initSDK	初始化SDK	需要先调用该接口
	getInstance	获取SDK实例	
	applyCert	申请证书	
	updateCert	更新证书	
	revokeCert	注销证书	
	getCertContList	读取证书容器标识列表	
	getCert	读取证书	
	getCertInfo	获取证书项	
	getCertInfoByOid	获取证书OID项	
	signature	数字签名	
	verifySignature	验证签名	
	verifyPin	验证密码	
	changePin	修改密码	
	unlockPin	解锁密码	

3 集成流程说明

- 1、将 libs 目录下的 jar 文件拷贝到应用 app/libs 目录下
- 2、将 libs 目录下的 jniLibs 文件夹拷贝到应用 app/src/main 目录下
- 3、在 gradle 根节点添加如下代码：

```
android {  
    repositories {  
        flatDir {  
            dirs 'libs'  
        }  
    }  
}
```

- 4、配置 AndroidManifest.XML

```
<!-- 网络权限 -->  
<uses-permission android:name="android.permission.INTERNET" />  
<!-- SD卡读写权限 -->  
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />  
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

4 数据格式定义

4.1 用户信息数据格式

传入的用户信息数据格式为json结构, 格式如下:

```
{  
    "name": "张三",  
    "idNo": "310111000xxxxxxxx9876",  
    "mobile": "139xxxxxxxx",  
    "province": "上海市",  
    "city": "上海市",  
    "organization": "",  
    "organizationUnit": "",  
    "email": "test@custletech.com"  
}
```

参数说明:

参数名称	字段类型	说明	必填
name	String	姓名	是
idNo	String	身份证号	是
mobile	String	手机号	是
province	String	省份	否
city	String	城市	否
organization	String	单位	否
organizationUnit	String	部门	否
email	String	电子邮箱	否

5 接口定义

5.1 MKEY 服务端接口

5.1.1 应用授权

5.1.1.1 功能说明

APP 后台调用服务端接口，获取应用授权码，应用授权码作为其他接口调用的凭据。

5.1.1.2 接口说明

接口协议：HTTP协议

请求方式：POST

接口服务地址：https://ip:port/sdk/v1/authorize/app

5.1.1.3 请求消息

消息流向：客户端→服务器

参数说明：

	参数名	参数说明
必选	businessCode	机构账户（平台分配）
必选	appId	机构对应应用ID（平台分配）
必选	date	请求时间，时间格式为yyyyMMddHHmmss

必选	sign	使用应用私钥对参数 “businessCode##appId##date”进行签名，得到 签名值Base64Encode后进行URLEncode(UTF-8) 传输。应用私钥由平台分配，签名示例代码请见 MKeySDKDemo程序与 参数签名示例 。
----	------	---

5.1.1.4 响应消息

消息流向：服务器→客户端

返回JSON示例：

```
{
  "ret":0,
  "msg":"success",
  "data": {
    "code":"a2be-22e3-3d99-v022"
  }
}
```

返回值说明：

字段	类型	说明
ret	Integer	结果码，0为成功，其他为失败
msg	String	结果描述
code	String	应用授权码

5.2 MKEYAPI 接口

5.2.1 SDK 初始化

5.2.1.1 接口描述

本接口用于初始化 SDK 地址。

5.2.1.2 接口定义

```
public static void initSDK(String url, String contCode)
```

参数说明:

参数名称	字段类型	说明	必填
url	String	SDK地址	是
contCode	String	证书容器标识, 用于区分多套证书, 自定义	是

5.2.1.3 调用示例

```
MKeyApi.initSDK (url, contCode);
```

5.2.2 接口初始化

5.2.2.1 接口描述

本接口用于初始化 MKEYAPI。

5.2.2.2 接口定义

```
public static MKeyApi getInstance(Context context, String appId, String appCode, String userInfo)
```

参数说明:

参数名称	字段类型	说明	必填
context	Context	上下文	是
appId	String	机构对应应用ID (平台分配)	是
appCode	String	应用授权码, 5.1.1.4中获取	是
userInfo	String	用户信息, 详见4.1。	是

注: appCode 可在读取证书容器标识列表、读取证书、读取证书项、读取证书 OID 项、验证签名、验证密码、修改密码等接口传空。

userInfo 可在读取证书容器标识列表接口传空。

5.2.2.3 调用示例

```
MKeyApi.getInstance(context, appId, appCode, userInfo);
```


5.2.3 申请证书

5.2.3.1 接口描述

本接口用于申请 RSA 或 SM2 证书。

5.2.3.2 接口定义

```
public void applyCert(String pin, MKeyApiCallback callback)
```

参数说明：

参数名称	字段类型	说明	必填
pin	String	证书密码	是

响应结果：

MKeyApiResult（使用 get 方法获取属性值）			
参数名称	字段类型	说明	备注
code	String	错误码	
msg	String	错误描述	

5.2.3.3 调用示例

```
MKeyApi.getInstance(context, appId, appCode, userInfo).applyCert(pin, new MKeyA
piCallback() {
    @Override
    public void onMKeyApiCallback (MKeyApiResult result) {
        Log.e("DYRZLOG", result.getCode());
        Log.e("DYRZLOG", result.getMsg());
    }
});
```

5.2.4 更新证书

5.2.4.1 接口描述

本接口用于证书更新。

5.2.4.2 接口定义

```
public void updataCert(String pin, MKeyApiCallback callback)
```

参数说明：

参数名称	字段类型	说明	必填
pin	String	证书密码	是

响应结果：

MKeyApiResult（使用 get 方法获取属性值）			
参数名称	字段类型	说明	备注
code	String	错误码	
msg	String	错误描述	

5.2.4.3 调用示例

```
MKeyApi.getInstance(context, appId, appCode, userInfo).updataCert(new MKeyApiCallback () {
    @Override
    public void onMKeyApiCallback (MKeyApiResult result) {
        Log.e("DYRZLOG", result.getCode());
        Log.e("DYRZLOG", result.getMsg());
    }
});
```

5.2.5 注销证书

5.2.5.1 接口描述

本接口用于证书注销删除。

5.2.5.2 接口定义

```
public void revokeCert(String pin, MKeyApiCallback callback)
```

参数说明：

参数名称	字段类型	说明	必填
pin	String	证书密码	是

响应结果：

MKeyApiResult（使用 get 方法获取属性值）			
参数名称	字段类型	说明	备注
code	String	错误码	
msg	String	错误描述	

5.2.5.3 调用示例

```
MKeyApi.getInstance(context, appId, appCode, userInfo).revokeCert(pin, new MKeyApiCallback() {  
    @Override  
    public void onMKeyApiCallback(MKeyApiResult result) {  
        Log.e("DYZZLOG", result.getCode());  
        Log.e("DYZZLOG", result.getMsg());  
    }  
});
```

5.2.6 读取证书容器标识列表

5.2.6.1 接口描述

本接口用于读取证书列表，列表数据为 json 格式，如 {"idNo1":["code1","code2"],"idNo2":["code1","code2"]}。

5.2.6.2 接口定义

```
public void getCertContCodeList(MKeyApiCallback callback)
```

响应结果：

MKeyApiResult（使用 get 方法获取属性值）			
参数名称	字段类型	说明	备注
code	String	错误码	
msg	String	错误描述	
data	String	证书容器标识列表，json格式。可在SDK初始化接口中设置对应标识	

5.2.6.3 调用示例

```
MKeyApi.getInstance(context, appId, "", userInfo).getCertContCodeList(new MKeyApiCallback() {
    @Override
    public void onMKeyApiCallback (MKeyApiResult result) {
        Log.e("DYZLOG", result.getCode());
        Log.e("DYZLOG", result.getMsg());
        Log.e("DYZLOG", result.getData());
    }
});
```

5.2.7 读取证书

5.2.7.1 接口描述

本接口用于读取证书，证书为 Base64 编码格式。

5.2.7.2 接口定义

```
public void getCert(MKeyApiCallback callback)
```

响应结果：

MKeyApiResult（使用 get 方法获取属性值）			
参数名称	字段类型	说明	备注
code	String	错误码	
msg	String	错误描述	
data	String	Base64证书	

5.2.7.3 调用示例

```
MKeyApi.getInstance(context, appId, "", userInfo).getCert(new MKeyApiCallback()
{
    @Override
    public void onMKeyApiCallback (MKeyApiResult result) {
        Log.e("DYRZLOG", result.getCode());
        Log.e("DYRZLOG", result.getMsg());
        Log.e("DYRZLOG", result.getData());
    }
});
```

5.2.8 读取证书项

5.2.8.1 接口描述

本接口用于读取证书基本项。

5.2.8.2 接口定义

`public void getCertInfo(MKeyApiCallback callback)`

响应结果：

MKeyApiResult（使用 get 方法获取属性值）			
参数名称	字段类型	说明	备注
code	String	错误码	
msg	String	错误描述	
data	String	Base64证书	

5.2.8.3 调用示例

```

MKeyApi.getInstance(context, appId, "", userInfo).getCertInfo(new MKeyApiCallback() {
    @Override
    public void onMKeyApiCallback (MKeyApiResult result) {
        Log.e("DYZZLOG", result.getCode());
        Log.e("DYZZLOG", result.getMsg());
        Log.e("DYZZLOG", result.getData());
    }
});

```

5.2.9 读取证书 OID 项

5.2.9.1 接口描述

本接口用于读取证书 OID 项。

5.2.9.2 接口定义

`public void getCertInfoByOid(String strOid, MKeyApiCallback callback)`

参数说明：

参数名称	字段类型	说明	必填
strOid	String	证书oid标识	是

响应结果：

MKeyApiResult（使用 get 方法获取属性值）			
参数名称	字段类型	说明	备注
code	String	错误码	
msg	String	错误描述	
data	String	Base64证书	

5.2.9.3 调用示例

```
MKeyApi.getInstance(context, appId, "", userInfo).getCertInfoByOid(strOid, new
MKeyApiCallback() {
    @Override
    public void onMKeyApiCallback (MKeyApiResult result) {
        Log.e("DYRZLOG", result.getCode());
        Log.e("DYRZLOG", result.getMsg());
        Log.e("DYRZLOG", result.getData());
    }
});
```

5.2.10 数字签名

5.2.10.1 接口描述

本接口用于对数据进行数字签名，签名值为 Base64 编码值。

5.2.10.2 接口定义

```
public void signature(String signSrc, String pin, MKeyApiCallback callback)
```

参数说明：

参数名称	字段类型	说明	必填
signSrc	String	签名原文，如需解Base64后签名，需在原文Base64字符串增加“KSBASE64:”前缀	是
pin	String	证书密码	是

响应结果：

MKeyApiResult（使用 get 方法获取属性值）			
参数名称	字段类型	说明	备注
code	String	错误码	
msg	String	错误描述	
data	String	Base64签名值	

5.2.10.3 调用示例

```
MKeyApi.getInstance(context, appId, context, appCode, userInfo).Signature (signSrc, new MKeyApiCallback () {
    @Override
    public void onMKeyApiCallback (MKeyApiResult result) {
        Log.e("DYZZLOG", result.getCode());
        Log.e("DYZZLOG", result.getMsg());
        Log.e("DYZZLOG", result.getData());
    }
});
```

5.2.11 验证签名

5.2.11.1 接口描述

本接口用于对签名值进行验证。

5.2.11.2 接口定义

```
public void verifySignature(String signSrc, String signData, MKeyApiCallback callback)
```

参数说明：

参数名称	字段类型	说明	必填
signSrc	String	签名原文	是
signData	String	签名值	是

响应结果：

MKeyApiResult（使用 get 方法获取属性值）			
参数名称	字段类型	说明	备注
code	String	错误码	
msg	String	错误描述	

5.2.11.3 调用示例

```

MKeyApi.getInstance(context, appId, "", userInfo).verifySignature(signSrc, signData, new MKeyApiCallback() {
    @Override
    public void onMKeyApiCallback(MKeyApiResult result) {
        Log.e("DYRZLOG", result.getCode());
        Log.e("DYRZLOG", result.getMsg());
    }
});

```

5.2.12 验证密码

5.2.12.1 接口描述

本接口用于对证书密码验证。

5.2.12.2 接口定义

```
public void verifyPin(String pin, MKeyApiCallback callback)
```

参数说明：

参数名称	字段类型	说明	必填
pin	String	证书密码	是

响应结果：

MKeyApiResult（使用 get 方法获取属性值）			
参数名称	字段类型	说明	备注
code	String	错误码	

msg	String	错误描述	
-----	--------	------	--

5.2.12.3 用示例

```
MKeyApi.getInstance(context, appId, "", userInfo).verifyPin(pin, new MKeyApiCallback() {
    @Override
    public void onMKeyApiCallback (MKeyApiResult result) {
        Log.e("DYRZLOG", result.getCode());
        Log.e("DYRZLOG", result.getMsg());
    }
});
```

5.2.13 修改密码

5.2.13.1 接口描述

本接口用于对证书密码修改。

5.2.13.2 接口定义

```
public void changePin(String oldPin, String newPin, MKeyApiCallback callback)
```

参数说明：

参数名称	字段类型	说明	必填
oldPin	String	证书密码	是
newPin	String	新的证书密码	是

响应结果：

MKeyApiResult（使用 get 方法获取属性值）			
参数名称	字段类型	说明	备注
code	String	错误码	
msg	String	错误描述	

5.2.13.3 调用示例

```
MKeyApi.getInstance(context, appId, "", userInfo).changePin(oldPin, newPin, new
    MKeyApiCallback() {
        @Override
        public void onMKeyApiCallback (MKeyApiResult result) {
            Log.e("DYZZLOG", result.getCode());
            Log.e("DYZZLOG", result.getMsg());
        }
    });
```

5.2.14 解锁密码

5.2.14.1 接口描述

本接口用于对证书密码解锁重置。

5.2.14.2 接口定义

```
public void unlockPin(final String adminPin, final String newPin, final
    MKeyApiCallback callback)
```

参数说明：

参数名称	字段类型	说明	必填
adminPin	String	管理员密码	是
newPin	String	新的证书密码	是

响应结果：

MKeyApiResult（使用 get 方法获取属性值）			
参数名称	字段类型	说明	备注
code	String	错误码	
msg	String	错误描述	

5.2.14.3 调用示例

```
MKeyApi.getInstance(context, appId, appCode, userInfo).unlockPin(adminPin, newPin, new MKeyApiCallback() {
    @Override
    public void onMKeyApiCallback (MKeyApiResult result) {
        Log.e("DYRZLOG", result.getCode());
        Log.e("DYRZLOG", result.getMsg());
    }
});
```

6 示例代码

6.1 参数签名示例

➤ JAVA 版

```
/**
 * 使用私钥进行签名
 * @param privateKey 私钥
 * @param bData 参数数据
 * @return BASE64 编码格式签名值
 */
private static String signature(String privateKey, byte[] bData) {
    String result = null;
    try {
        PKCS8EncodedKeySpec keySpec = new PKCS8EncodedKeySpec(
            Base64.decode(privateKey));
        KeyFactory keyFactory = KeyFactory.getInstance("RSA");
        PrivateKey priKey = keyFactory.generatePrivate(keySpec);

        Signature oSig = Signature.getInstance("SHA256WithRSA");
        oSig.initSign(priKey);
        oSig.update(bData);
        byte[] signature = oSig.sign();
        if (signature != null && signature.length > 0) {
            result = new String (Base64.encode(signature));
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

```

return result;
}

```

➤ C# .NET 版

```

var rsa = new RSACryptoServiceProvider();
// privateKey 为应用私钥
rsa.FromXmlString(privateKey);
// data 为参数数据
byte[] signBytes =
    rsa.SignData(Encoding.UTF8.GetBytes(data), CryptoConfig.CreateFromName("SHA256"));
// BASE64 编码格式签名值
result = Convert.ToBase64String(signBytes);

```

7 错误码定义

错误码	错误描述
0	成功
1	取消
2	参数错误
10	网络错误
11	暂不支持错误
12	异常错误
13	用户认证错误
14	应用鉴权错误
100	证书申请错误
101	证书请求文件生成错误
102	密钥错误
103	证书保存错误
104	证书获取错误
105	证书项获取错误
106	证书 OID 获取错误
107	证书删除错误
108	证书更新错误
200	密码验证错误
201	密码验证错误，剩余输入 1 次
202	密码验证错误，剩余输入 2 次
203	密码验证错误，剩余输入 3 次
204	密码验证错误，剩余输入 4 次
205	密码验证错误，剩余输入 5 次
210	密码锁死错误

211	密码更新错误
212	密码解锁错误
213	密码异常
300	数字签名错误
301	签名密钥错误
302	验证签名错误