移动安全认证系统
iOS SDK

接入规范 v2.0.2

目 录

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

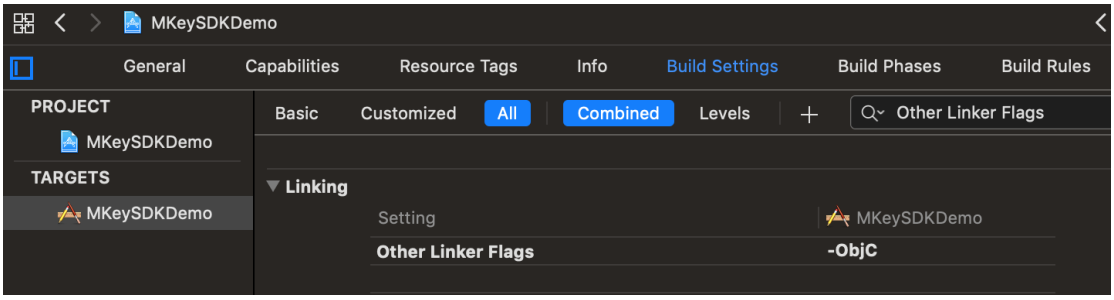
接口分类	接口名称	接口说明	备注
接口	initSDK	初始化SDK	需要先调用该接口
	getInstance	获取SDK实例	
	applyCert	申请证书	
	updataCert	更新证书	
	revokeCert	注销证书	
	getCertContCodeList	读取证书容器标识列表	
	getCert	读取证书	
	getCertInfo	获取证书项	
	getCertInfoByOid	获取证书OID项	
	signature	数字签名	
	verifySignature	验证签名	
	verifyPin	验证密码	
	changePin	修改密码	

	unlockPin	解锁密码	

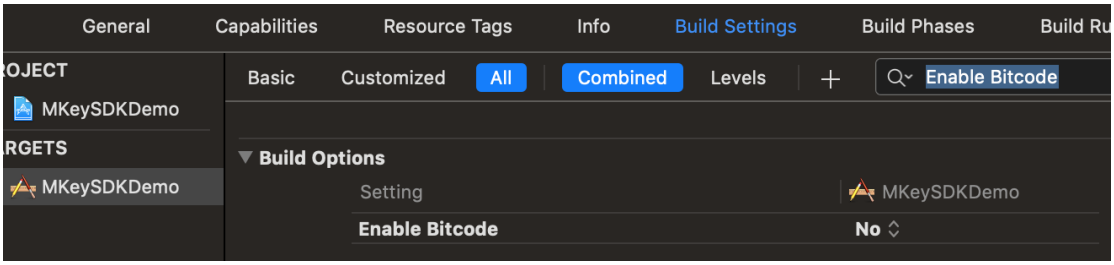
3 集成流程说明

1、 将 MkeySDK 文件夹导入到工程

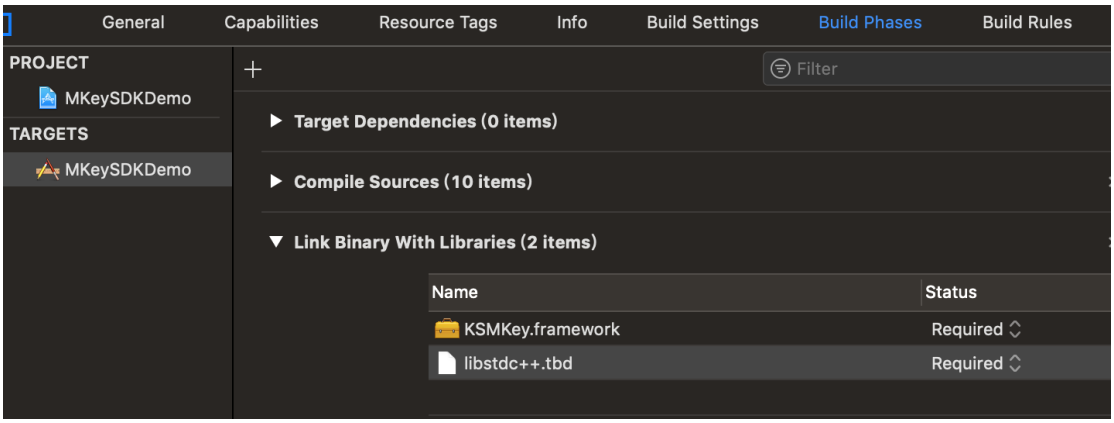
2、 在 TARGETS -> Build Settings -> Other Linker Flags 添加-ObjC



3、 将 TARGETS -> Build Settings -> Enable Bitcode 设置为 No



4、 在 TARGETS -> Build Phases -> Link Binary With Libraries 添加 libstdc++.tbd 库



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：客户端一>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：服务器一>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

+ (void)initSDK:(NSString *)url contCode:(NSString *)code

参数说明:

参数名称	字段类型	说明	必填
url	NSString	SDK地址	是
code	NSString	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: code];

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

+ (instancetype)getInstance:appId appCode:(NSString *)appCode userInfo:(NSString *)userInfo

参数说明:

参数名称	字段类型	说明	必填
appId	NSString	机构对应应用ID（平台分配）	是
appCode	NSString	应用授权码，5.1.1.4中获取	是
userInfo	NSString	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:appId appCode:appCode userInfo: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

- (void)applyCert:(NSString *)pin resultBlock:(void (^)(MResultBean *resultBean))resultBlock

参数说明：

参数名称	字段类型	说明	必填
pin	NSString	证书密码	是

响应结果：

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

5.2.3.3调用示例

```
[[MKeyApi getInstance:appId appCode:appCode userInfo:userInfo] applyCert:pin
resultBlock:^(ResultBean *resultBean) {
        NSLog(@"%lu", (long)resultBean.code);
        NSLog(@"%@", resultBean.msg);
    }];
```

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

- (void)updataCert:(NSString *)pin resultBlock:(void (^)(ResultBean *resultBean))resultBlock

参数说明：

参数名称	字段类型	说明	必填
pin	NSString	证书密码	是

响应结果：

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

5.2.4.3调用示例

```
[[MKeyApi getInstance:appId appCode:appCode userInfo:userInfo] updataCert:pin
resultBlock:^(ResultBean *resultBean) {
        NSLog(@"%lu", (long)resultBean.code);
        NSLog(@"%@", resultBean.msg);
    }];
```

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

- (void)revokeCert:(NSString *)pin resultBlock:(void (^)(MResultBean *resultBean))resultBlock

响应结果：

ResultBean（使用 get 方法获取属性值）			
参数名称	**字段类型**	**说明**	**备注**

code	NSInteger	错误码	
msg	NSString	错误描述	

5.2.5.3调用示例

```
[[MKeyApi getInstance:appCode userInfo:userInfo] revokeCert:pin resultBlock:^(ResultBean
*resultBean) {
    NSLog(@"%lu", (long)resultBean.code);
    NSLog(@"%@", resultBean.msg);
}];
```

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

- (void)getCertContCodeList:(void (^)(ResultBean *resultBean))resultBlock

响应结果：

ResultCertBean（使用 get 方法获取属性值）			
参数名称	**字段类型**	**说明**	**备注**
code	NSInterger	错误码	
msg	NSString	错误描述	
data	NSString	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:appId appCode:@"" userInfo:@""] getCertContCodeList:^(ResultBean
*resultBean) {
    NSLog(@"%lu", (long)resultBean.code);
    NSLog(@"%@", resultBean.msg);
    NSLog(@"%@", resultBean.data);
}];
```

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

- (void)getCert:(void (^)(ResultBean *resultBean))resultBlock

响应结果：

ResultCertBean（使用 get 方法获取属性值）			
参数名称	**字段类型**	**说明**	**备注**
code	NSInterger	错误码	
msg	NSString	错误描述	
data	NSString	Base64证书值	

5.2.7.3调用示例

```
[[MKeyApi getInstance:appId appCode:@"" userInfo:userInfo] getCert:^(ResultBean *resultBean)
{
    NSLog(@"%lu", (long)resultBean.code);
```

```
    NSLog(@"%@", resultBean.msg);
    NSLog(@"%@", resultBean.data);
}];
```

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

- (void)getCertInfo:(void (^)(ResultBean *resultBean))resultBlock

响应结果：

ResultCertBean（使用 get 方法获取属性值）			
参数名称	**字段类型**	**说明**	**备注**
code	NSInterger	错误码	
msg	NSString	错误描述	
data	NSString	Base64证书值	

5.2.8.3调用示例

```
[[MKeyApi getInstance:appId appCode:@"" userInfo:userInfo] getCertInfo:^(ResultBean
*resultBean) {
    NSLog(@"%lu", (long)resultBean.code);
    NSLog(@"%@", resultBean.msg);
    NSLog(@"%@", resultBean.data);
}];
```

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

- (void)getCertInfoByOid:(NSString *)strOid resultBlock:(void (^)(ResultBean *resultBean))resultBlock

参数说明：

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

响应结果：

ResultCertBean（使用 get 方法获取属性值）			
参数名称	**字段类型**	**说明**	**备注**
code	NSInterger	错误码	
msg	NSString	错误描述	
data	NSString	Base64证书值	

5.2.9.3调用示例

```
[[MKeyApi getInstance:appId appCode:@"" userInfo:userInfo] getCertInfoByOid:strOid
resultBlock:^(ResultBean *resultBean) {
    NSLog(@"%lu", (long)resultBean.code);
    NSLog(@"%@", resultBean.msg);
    NSLog(@"%@", resultBean.data);
}];
```

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

- (void)signature:(NSString *)signSrc pin:(NSString *)pin resultBlock:(void (^)(ResultBean * resultBean))resultBlock

参数说明：

参数名称	字段类型	说明	必填
signSrc	NSString	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	NSString	证书密码	是

响应结果：

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

5.2.10.3 调用示例

```
[[MKeyApi getInstance:appId appCode:appCode userInfo:userInfo] signature:signSrc pin:pin resultBlock:^(ResultBean *resultBean) {
    NSLog(@"%lu", (long)resultBean.code);
    NSLog(@"%@", resultBean.msg);
    NSLog(@"%@", resultBean.data);
```

}];

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

- (void)verifySignature:(NSString *)signSrc signData:(NSString *)signData resultBlock:(void (^)(ResultBean *resultBean))resultBlock

参数说明：

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

响应结果：

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

5.2.11.3 调用示例

```
[[MKeyApi getInstance:appId appCode:@"" userInfo:userInfo] verifySignature:signSrc
signData:signData resultBlock:^(ResultBean *resultBean) {
    NSLog(@"%lu", (long)resultBean.code);
    NSLog(@"%@", resultBean.msg);
```

}];

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

- (void)verifyPin:(NSString *)pin resultBlock:(void (^)(ResultBean *resultBean))resultBlock

参数说明：

参数名称	字段类型	说明	必填
pin	NSString	证书密码	是

响应结果：

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

5.2.12.3 用示例

```
[[MKeyApi getInstance:appId appCode:@"" userInfo:userInfo] verifyPin:pin
resultBlock:^(ResultBean *resultBean) {
    NSLog(@"%lu", (long)resultBean.code);
    NSLog(@"%@", resultBean.msg);
}];
```

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

- (void)changePin:(NSString *)oldPin newPin:(NSString *)newPin resultBlock::(void (^)(ResultBean *resultBean))resultBlock

参数说明：

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

响应结果：

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

5.2.13.3 调用示例

```
[[MKeyApi getInstance:appId appCode:appCode userInfo:userInfo] changePin:pin
newPin:newPin resultBlock :^(ResultBean *resultBean) {
    NSLog(@"%lu", (long)resultBean.code);
    NSLog(@"%@", resultBean.msg);
}];
```

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。

5.2.14.2 接口定义

- (void)unlockPin:(NSString *)adminPin newPin:(NSString *)newPin resultBlocks:(void (^)(MResultBean *resultBean))resultBlock

参数说明：

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

响应结果：

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

5.2.14.3 调用示例

```
[[MKeyApi getInstance:appId appCode:@"" userInfo:userInfo] unlockPin:adminPin
newPin:newPin resultBlock :^(ResultBean *resultBean) {
    NSLog(@"%lu", (long)resultBean.code);
    NSLog(@"%@", resultBean.msg);
}];
```

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
109	证书注销错误
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	密码解锁错误
211	密码更新错误

212	密码锁死错误
213	密码异常错误
300	数字签名错误
301	签名密钥错误
302	验证签名错误