

# CIU98\_B V2 Host SDK[V1.2]

## -Stm32l433

### 用户使用手册

V1.1



北京中电华大电子设计有限责任公司  
CEC Huada Electronic Design Co.,Ltd

2021 年 12 月

# 声 明

本文档的版权属北京中电华大电子设计有限责任公司所有。任何未经授权对本文档进行复印、印刷和出版发行的行为，都将被视为是对北京中电华大电子设计有限责任公司版权的侵害。北京中电华大电子设计有限责任公司保留对此行为诉诸法律的权力。

北京中电华大电子设计有限责任公司保留未经通知用户对本手册内容进行修改的权利。

本文档并未以暗示、反言或其他形式转让本公司以及任何第三方的专利、商标、版权、所有权等任何权利或许可。本公司不承担因使用、复制、修改、散布等行为导致的任何法律责任。

## 变更记录

版本	修改描述	日期
V1.0	初稿	2021-7-3
V1.1	1) 修改参考资料中的文档名，添加“V2”、“Stm32l433”字样； 2) 修改第4章，将接口抽象层描述改为协议抽象层描述； 3) 添加7.5小节介绍接口的切换方法；	2021-12-13

## 目 录

1	引言 .....	1
2	参考资料 .....	1
3	概述 .....	1
4	整体架构 .....	1
5	目录结构 .....	2
6	基于 STM32L433 的 DEMO 使用说明 .....	3
6.1	基于 STM32L433 的硬件环境搭建 .....	4
6.2	主端安全 SDK 代码移植说明 .....	6
7	API 应用示例说明 .....	6
7.1	通信接口示例 .....	6
7.2	应用示例 .....	7
7.2.1	通信演示 .....	7
7.2.2	设备认证演示 .....	7
7.2.3	密钥管理演示 .....	8
7.2.4	密码算法演示 .....	9
7.3	文件演示 .....	10
7.3.1	写文件 writefile_test .....	10
7.3.2	读文件 writefile_test .....	10
7.4	PIN 操作类接口演示 .....	10
7.4.1	修改 pin 演示 .....	10
7.4.2	重置 pin 演示 .....	11
7.4.3	验证 pin 演示 .....	11
7.5	注意事项 .....	11
8	常用错误码含义 .....	11
附录:	.....	12
	日志打印 .....	12

## 1 引言

本文档主要介绍 CIU98\_B V2 Host SDK[V1.2]（以下简称主端安全 SDK）的整体架构、目录结构、API 应用示例说明以及相关注意事项，帮助客户快速使用主端安全 SDK。

## 2 参考资料

《CIU98\_B V2 Host SDK[V1.2]-Stm32l433 API 接口说明.chm》 V1.1

《CIU98\_B V2 Host SDK[V1.2]-Stm32l433 用户移植手册.doc》 V1.1

## 3 概述

本主端安全 SDK 包括通信、控制、设备认证、密钥管理、PIN 相关操作、密码运算、文件管理、信息管理和 Loader 升级接口功能。

支持的通信接口为 SPI 和 I2C，支持的算法有 3DES/AES/ECC256/RSA/SM2/SM3/SM4/SHA/系列。其中 ECC 算法的 ECDSA、ECIES、ECDH 符合 SEC 规范，曲线参数支持 nistp256r1。

本 SDK 代码空间统计如下：

接口	RAM	FLASH
I2C	3K	20.1K
SPI	3K	20.9K

说明：

- 1、此空间统计不包含 demo。
- 2、RAM 空间仅包含栈空间（此 SDK 未使用堆），用户可以配置“CIU98\_B\_V2\_Host\_SDK[V1.2.0-release]-Stm32l433/src/util/util.h”中宏定义“DEQUE\_MAX\_SIZE”来根据平台具体情况自定义配置所占 RAM 空间。
- 3、在本 SDK demo 工程中，默认空间配置文件 startup\_stm32l433.s 中栈空间为 0x2E00（1.5kB），包含 demo 栈空间，用户可根据实际应用调整此处大小，但需保证大于 3kB。

## 4 整体架构

本 SDK 采用分层设计，从上到下依次分为 APP（客户应用）、APIs Layer（应用编程接口）、Command Layer（命令层）、Protocol Abstraction Layer（协议抽象层）、Link Protocol Layer（链路通信协议层）和 Hardware Portable Layer（硬件适配层）6 部分组成，整体架构图如下：

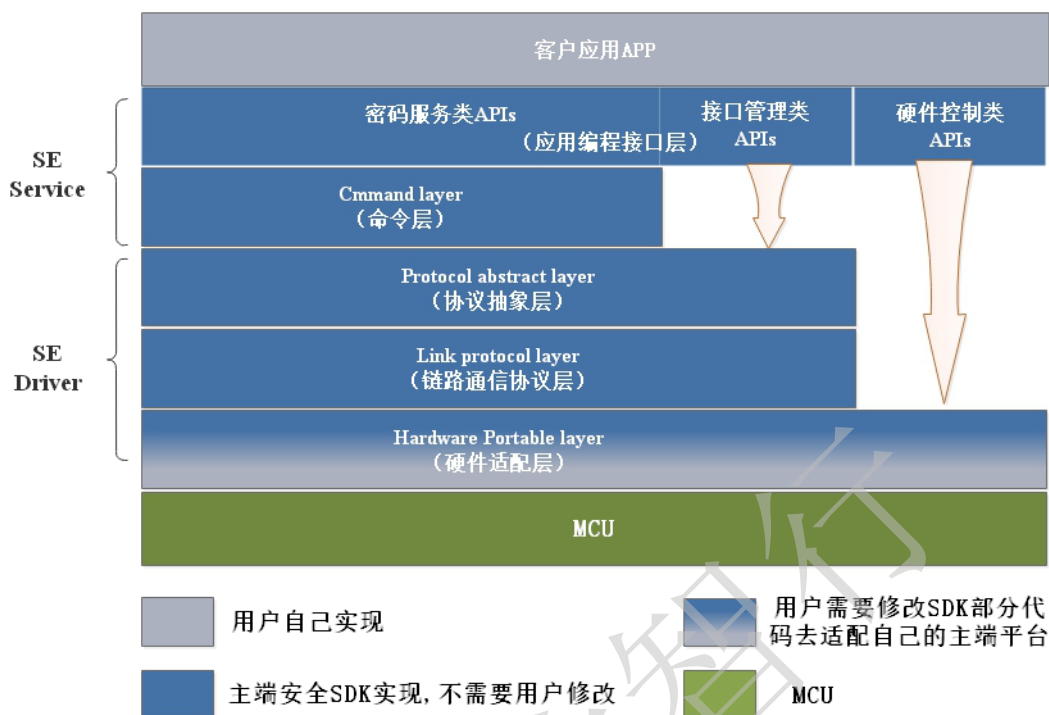


图 1 整体架构图

- ✓ APIs Layer (应用编程接口层): 封装了 SE 对外提供的所有应用功能函数接口, 包括通信、控制、鉴权、密钥管理、PIN 操作、文件管理、密码运算等功能函数接口, 方便用户开发应用。
- ✓ Command Layer (命令层): 封装与 COS 对应的应用命令, 为上层应用功能提供基础命令支持, 易于上层应用功能接口开发和移植。
- ✓ Protocol Abstraction Layer (协议抽象层): 定义统一的抽象接口, 屏蔽链路通信协议层不同接口协议的实现差异, 封装了底层驱动协议实现细节。
- ✓ Link Protocol Layer (链路通信协议层): 实现 HED 各链路通信接口的通信协议。
- ✓ Hardware Portable Layer (硬件适配层): 屏蔽相同硬件接口的不同 MCU 实现细节及差异, 提供统一硬件访问接口, 并依据所定义的函数接口实现在不同 MCU 环境下的硬件功能适配和移植

## 5 目录结构

主端安全 SDK 为标准 C 语言编写的应用接口源码。

主端安全 SDK 的源码目录结构和各目录含义如图 2、图 3 所示:

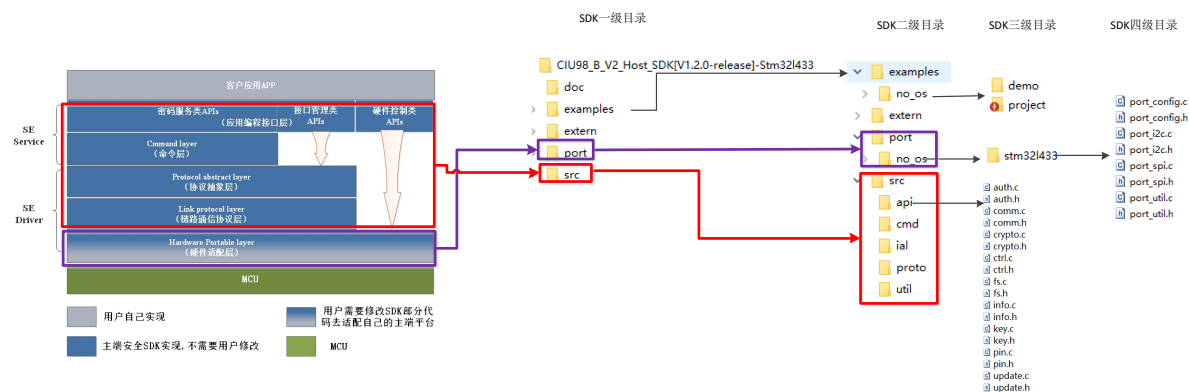


图 2 主端安全 SDK 目录结构图

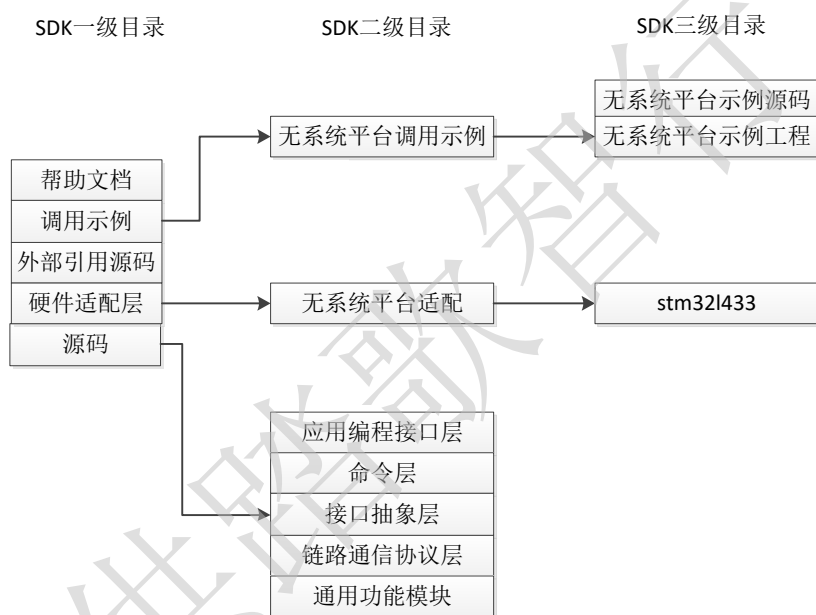


图 3 主端安全 SDK 目录结构含义

客户需要关注的文件夹为 src/api、port 和 examples，其中：

src/api 为应用编程接口，为上层应用程序调用的接口，具体参考《CIU98\_B Host SDK[V1.2]API 接口说明.chm》。

examples 为 demo 示例，为 app 中的接口调用例程，具体参考本文档“6 API 应用示例说明”。

port 为硬件适配层，为参考无操作系统硬件平台（STM32L433）的硬件适配层示例，客户可根据不同硬件平台进行修改，具体接口参考《CIU98\_B V2 Host SDK[V1.2]-Stm32l433 API 接口说明.chm》，移植过程参考文档《CIU98\_B V2 Host SDK[V1.2]-Stm32l433 用户移植手册 V1.1》。

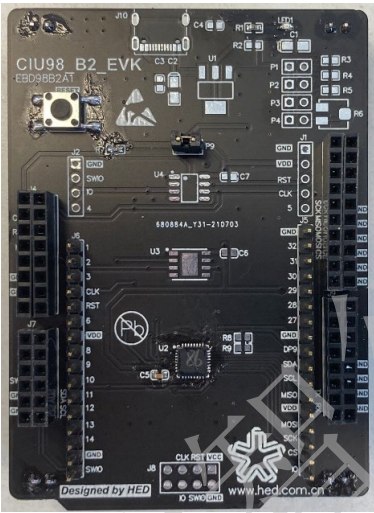
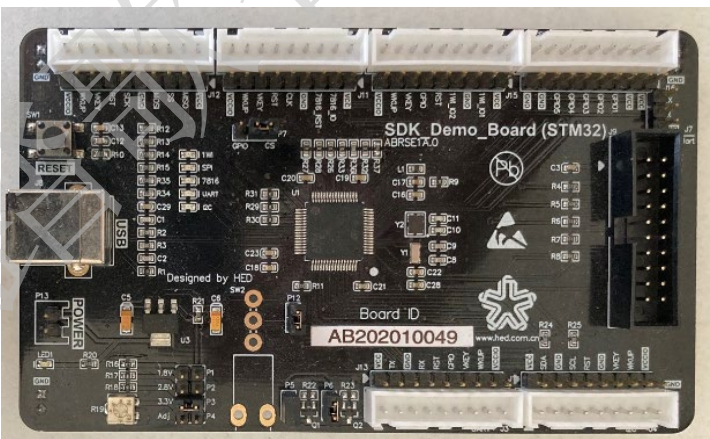
## 6 基于 STM32L433 的 Demo 使用说明

在使用基于 STM32L433 硬件平台的 demo 时，首先完成硬件连接环境的搭建，其次完

成主端安全 SDK 代码的移植，具体过程参考 6.1 和 6.2。

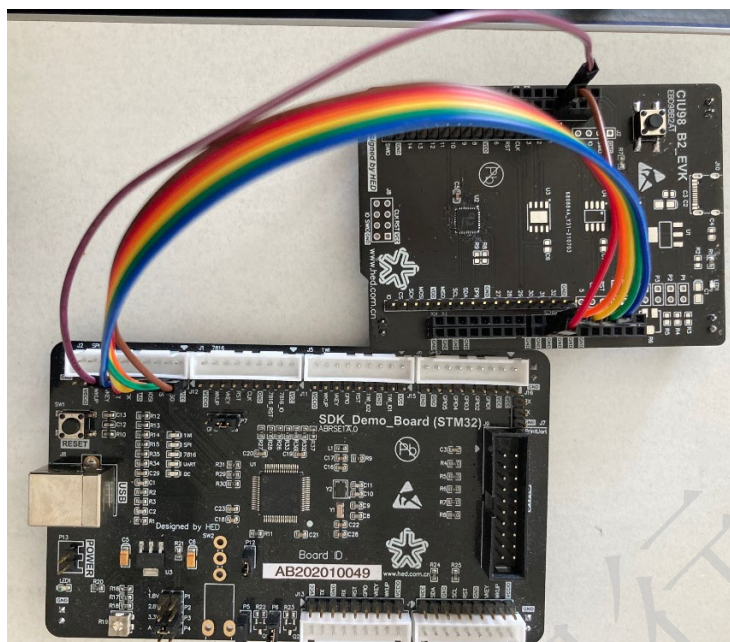
## 6.1 基于 STM32L433 的硬件环境搭建

所需硬件如下表所示：

序号	硬件电路	图片
1	CIU98_B2 开发板	
2	基于 STM32L433 的 SDK Demo 板	

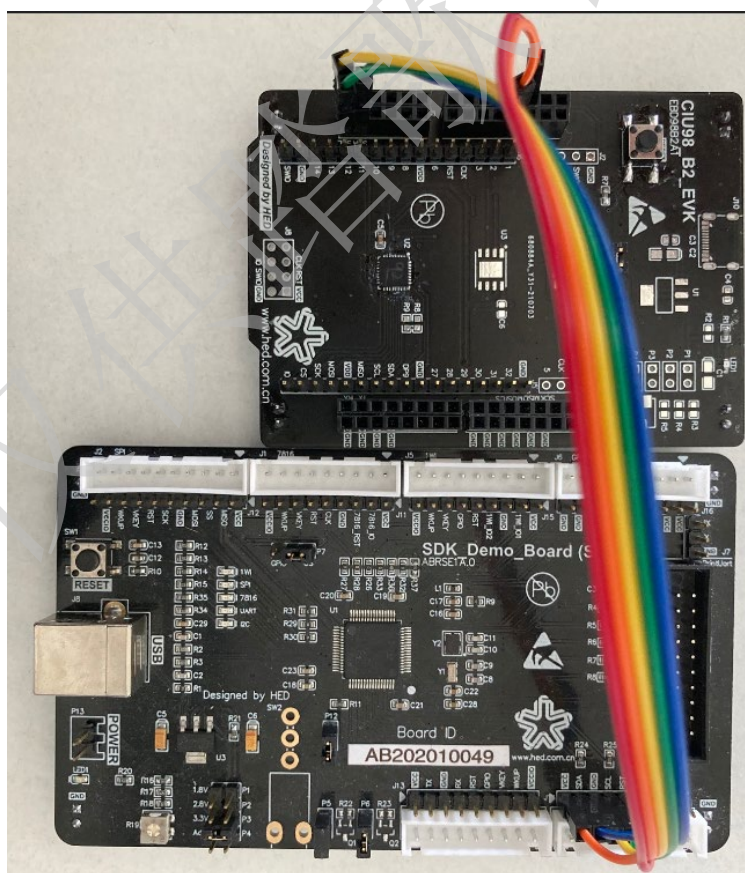
SPI 通信接口的硬件电路连接如下图所示：





连接线分别为：VCC、GND、RST、CS、SCK、MOSI、MISO。根据丝印标识连接。

I2C 通信接口的硬件电路连接如下图所示：



连接线分别为：VCC、GND、RST、SDA、SCL。根据丝印标识连接。

## 6.2 主端安全 SDK 代码移植说明

参考《CIU98\_B V2 Host SDK[V1.2]-Stm32l433 用户移植手册 V1.1》完成代码移植

## 7 API 应用示例说明

在完成硬件环境搭建和主端安全 SDK 代码移植之后，即可参照本部分介绍的“API 应用示例说明”调用主端安全 SDK 提供的接口，具体的接口使用说明参考《CIU98\_B V2 Host SDK[V1.2]-Stm32l433 API 接口说明.chm》。

主端安全 SDK 接口使用调用流程如下：

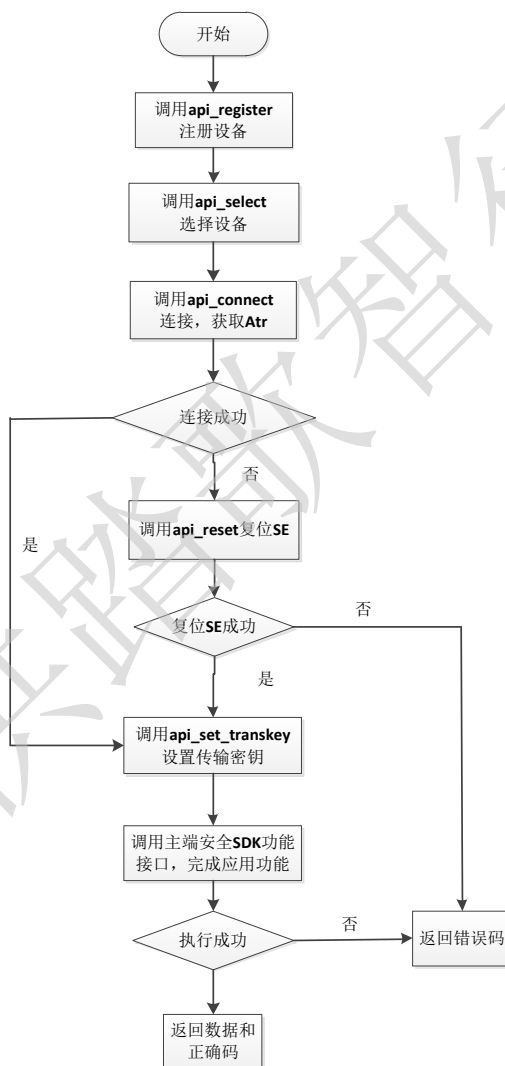


图 4 主端安全 SDK 接口使用调用流程

### 7.1 通信接口示例

examples/no\_os/project/driver为演示通信接口的keil工程文件，演示SPI/I2C通信接口功能。在完成硬件环境搭建和代码移植之后，可以首先运行本接口示例确保通信正常。

## 7.2 应用示例

应用示例是在保证通信的基础上对应用功能的演示。

examples/no\_os/demo/app 中为调用 api 接口的示例文件，包括构建本地演示所用软算法及应用测试文件。具体说明如下：

addins	第三方算法库及软算法封装
auth_test.c	设备认证示例
auth_test.h	
comm_test.c	通信示例
comm_test.h	密码运算示例
crypto_test.c	控制示例
crypto_test.h	文件管理示例
ctrl_test.c	信息管理示例
ctrl_test.h	密钥管理示例
fs_test.c	PIN相关操作示例
fs_test.h	
info_test.c	
info_test.h	
key_test.c	
key_test.h	
pin_test.c	
pin_test.h	

图 5 SDK 应用示例文件

examples/no\_os/project/xxx/app 为演示 api 接口的 keil 工程文件。

examples/no\_os/project/xxx/app 下的 user 目录为 main 入口文件，分别对应上图的应用示例，用户可自行选择运行所需示例。

### 7.2.1 通信演示

通信演示文件为 examples/no\_os/demo/app/comm\_test。

通信主测试函数为 comm\_test。应用示例中的通信仅演示了单一 SPI。

通信接口注意事项：

通信顺序为注册 api\_register - 选择 api\_select - 连接 api\_connect。

### 7.2.2 设备认证演示

安全认证演示文件为 examples/no\_os/demo/app/auth\_test。

安全认证主测试函数为 auth\_test，对设备认证接口 api\_ext\_auth 进行了测试。

本示例为本地演示示例，在实际中客户需要自行实现下面（1）-（2）。

（1）调用 api\_get\_random 向 SE 申请一个随机数。

（2）用设备密钥对此随机数加密，产生认证数据。若随机数长度不满足分组长度，则

在结尾补0参与运算。

(3) 调用api\_ext\_auth进行设备认证。

### 7.2.3 密钥管理演示

密钥管理演示文件为 examples/no\_os/demo/app/key\_test。

密钥管理主测试函数为 key\_test，其中分别提供更新设备主控密钥、更新传输密钥、产生非对称密钥对、产生对称密钥、密钥协商、导入密钥、导出密钥、删除密钥、获取密钥信息演示示例。

#### 7.2.3.1 更新设备主控密钥演示

本示例为本地演示示例，在实际中客户需要自行实现 (1) - (2) 过程。

(1) 获得设备权限。

(2) 更新主控密钥。

#### 7.2.3.2 更新传输密钥演示

api\_set\_trans\_key 接口用于传输密钥的初始化。传输密钥可用于加密保护 SDK 和 SE 之间的数据传输。本示例为本地演示示例，在实际中客户需要自行实现 (1) - (2) 过程。

(1) 调用api\_verify\_pin接口获得安全文件的写权限。

(2) 调用 api\_set\_transkey 设置传输密钥。

#### 7.2.3.3 产生非对称密钥对演示

api\_generate\_keypair 接口用于产生非对称固定/临时公私钥对。如果为固定密钥，需要满足安全文件的写权限。

本示例为本地演示示例，在实际中客户需要自行实现 (1) - (2) 过程。

(1) 如果更新固定密钥，则调用api\_verify\_pin接口获得安全文件的写权限。若更新临时密钥则忽略此步。

(2) 调用 api\_generate\_keypair 生成密钥对。

#### 7.2.3.4 产生对称密钥演示

api\_generate\_symkey 接口用于生成一个对称应用密钥。生成固定密钥需要获得安全文件的写权限。本示例为本地演示示例，在实际中客户需要自行实现 (1) - (2) 过程。

(1) 如果更新固定密钥，则调用api\_verify\_pin接口获得安全文件的写权限。若更新临时密钥则忽略此步。

(2) 调用 api\_generate\_symke 生成密钥对。

#### 7.2.3.5 密钥协商演示

api\_generate\_shared\_key 接口用于产生 ECC/SM2 协商密钥，其中 ECC 协商中 ECDH 符合 IEEE 1363 规范，支持 OPEN SSL 应用。本示例为本地演示示例，在实际中客户需要自行实现 (1) 过程。

(1) 调用api\_generate\_shared\_key进行密钥协商。



### 7.2.3.6 导入密钥演示

`api_import_key` 接口用于应用密钥的明文或密文导入。导入固定密钥时，密钥 KID 为：01~EF。导入临时密钥时，密钥 KID 为：F0~FF。导入固定应用密钥时，需要获得安全文件的写权限。本示例为本地演示示例，在实际中客户需要自行实现（1）-（2）过程。

（1）如果导入固定应用密钥时，则调用 `api_verify_pin` 接口获得安全文件的写权限。若导入临时密钥则忽略此步。

（2）调用 `api_import_key` 导入密钥。

### 7.2.3.7 导出密钥演示

`api_export_key` 用于以明文或密文的形式导出对称临时应用密钥和非对称临时应用私钥，及明文导出固定/临时非对称应用公钥。本示例为本地演示示例，在实际中客户需要自行实现（1）-（2）过程。

（1）如果导出固定应用密钥时，则调用 `api_verify_pin` 接口获得安全文件的读权限。若导出临时密钥则忽略此步。

（2）调用 `api_export_key` 导出密钥。

### 7.2.3.8 删除密钥演示

`api_del_key` 接口用于删除指定密钥标识符 KID 的应用密钥。删除固定应用密钥时，需要获得安全文件的写权限。本示例为本地演示示例，在实际中客户需要自行实现（1）-（2）过程。

（1）如果删除固定应用密钥时，则调用 `api_verify_pin` 接口获得安全文件的写权限。若导出临时密钥则忽略此步。

（2）调用 `api_del_key` 删除指定 kid 的密钥。

### 7.2.3.9 获取密钥信息演示

`api_get_key_info` 接口用于读取密钥信息。本示例为本地演示示例，在实际中客户需要自行实现（1）-（2）过程。

（1）调用 `api_get_key_info` 获得应用密钥信息或管理密钥信息。

## 7.2.4 密码算法演示

密码运算演示文件为 `examples/no_os/demo/app/crypto_test`。

密码运算主测试函数为 `crypto_test`，其中分别提供对称算法运算、非对称算法运算和 HASH 运算等。

密码运算注意点如下：

1. 无权限控制。
2. 对称算法加解密计算，选择 `PADDING_NOPADDING` 补位，客户需自行处理补位/去补位，保证输入数据为分组长度整数倍；选择 `PADDING_PKCS7` 补位，对输入数据无限

制，由 SDK 处理按 PKCS7 规则补位/去补位。

3. 对称算法 MAC、验证 MAC 计算，选择 PADDING\_NOPADDING 补位，客户需自行处理补位，保证输入数据为分组长度整数倍；选择 PADDING\_ISO9797\_M1/ M2，由 SDK 处理按 ISO9797\_M1/ M2 规则补位。

4. 对称算法加密数据不超过 6k 字节。

5. 非对称算法加密数据不超过 4k 字节。

6. 非对称算法签名和验签时，输入数据未哈希时，需要指明哈希算法类型。输入数据已哈希时，哈希算法类型为 ALG\_NONE。当为 RSA 算法且算法类型 ALG\_NONE 时，数据需按照 PKCS1 补位。

7. SM2 算法签名或验签算法不包含 ZA 计算，需要 ZA 数据需先调用 api\_sm2\_get\_zs 接口，得到 ZA 数据，然后将 ZA 数据+待签名数据作为源数据，进行签名或验签计算。

### 7.3 文件演示

文件管理演示文件为 examples/no\_os/demo/app/fs\_test。

文件管理主测试函数为 fs\_test，其中分别提供选择文件、读写文件。

#### 7.3.1 写文件 writefile\_test

api\_write\_file 接口用于完成写文件操作。需要满足文件的写权限。本示例为本地演示示例，在实际中客户需要自行实现（1）-（2）过程。

（1）调用 api\_verify\_pin 接口获得文件的写权限。若文件无写权限要求则忽略此步。

（2）调用 api\_write\_file 进行文件写操作。

#### 7.3.2 读文件 readfile\_test

api\_read\_file 接口用于完成读文件操作。需要满足文件的读权限。本示例为本地演示示例，在实际中客户需要自行实现（1）-（2）过程。

（1）调用 api\_verify\_pin 接口获得文件的读权限。若文件无读权限要求则忽略此步。

（2）调用 api\_read\_file 进行文件读操作。

### 7.4 pin 操作类接口演示

pin 操作类演示文件为 examples/no\_os/demo/app/pin\_test。

pin 操作类接口演示提供了修改 pin、重置 pin 和验证 pin 演示示例。

#### 7.4.1 修改 pin 演示

api\_change\_pin 接口用于修改管理员 PIN 和用户 PIN。本示例为本地演示示例，在实际中客户需要自行实现（1）过程。

（1）调用 api\_change\_pin 接口修改 pin 值。

### 7.4.2 重置 pin 演示

`api_reload_pin` 接口用于重置用户 PIN。本示例为本地演示示例，在实际中客户需要自行实现（1）过程。

（1）调用 `api_reload_pin` 接口重置用户 pin。

### 7.4.3 验证 pin 演示

`api_verify_pin` 接口用于验证 PIN。本示例为本地演示示例，在实际中客户需要自行实现（1）过程。

（1）调用 `api_verify_pin` 接口验证 pin。

## 7.5 注意事项

示例源码都是以 SPI 接口为例实现的，若需要改变为 I2C 接口修改说明如下：

```
//--- 1. 注册及选择设备 ---
ret = api_register(PERIPHERAL_SPI, SPI_PERIPHERAL_SE0);
if(ret != SE_SUCCESS)
{
    LOGE("fail");
    return ret;
}

//--- 2. 连接, 获取atr ---
ret = api_select(PERIPHERAL_SPI, SPI_PERIPHERAL_SE0);
if(ret != SE_SUCCESS)
{
    LOGE("fail");
    return ret;
}
```

修改为: `api_register(PERIPHERAL_I2C, I2C_PERIPHERAL_SE0);`

修改为: `api_select(PERIPHERAL_I2C, I2C_PERIPHERAL_SE0);`

## 8 常用错误码含义

错误码	含义
0x30000002	连接超时，请检查硬件是否正确连接，适配层是否正确适配。
0x30000007	通信错误，请尝试对 SE 重新上电。
0x100063Cx	验证 pin 错，请检查 pin 数据是否正确。
0x10006985	使用条件不满足，请检查是否在执行该操作前进行了安全认证，例如：pin 认证或者设备认证。
0x10006A84	空间不足，请检查数据长度是否超过了文件支持的最大长度。
0x10006A88	kid 不存在，请检查 kid 是否存在。
0x10006582	SE 程序跑飞，请检查输入数据的正确性。
0x20000001	输入参数错，检查输入参数是否合法。

附录：

## 日志打印

可以通过 LOGE("");打印日志用于调试，其打印结果如下图所示：

```
E - demo/cmd_api_test/api_digest_test.c(171) - main: failed to api_verify_pin
```

上图中的日志报错，指明对应文件为 `api_digest_test.c`，错误行数为 171 行，错误函数为 `api_verify_pin`。若需要看返回错误码，只需要按 `printf` 的语法规则，将出错函数的返回值 `ret` 打印出来即可。

例：`printf(“%04x”,ret);`