**SJJ1310金融交易接口**

**使用说明文档**

****

**北京江南天安科技有限公司**

**2014年8月**

声明

本手册由北京江南天安科技有限公司编写。天安科技TASS保留对本手册的所有权和解释权，任何公司和个人未经允许，不得擅自使用、复制、修改、传播本手册的内容。天安科技保留有对本手册进行重新修订的权利，随时可能对本手册中出现的错误、与最新资料不符之处、命令的更新等做必要的修改，这些不再另行通知，但全部编入新版手册内。

北京江南天安科技有限公司

二〇一四年八月

**版本历史**

|  |  |  |  |
| --- | --- | --- | --- |
| **手册版本** | **时间** | **适应加密机** | **更新说明** |
| V1.1.1 | 20150724 | SJJ1310 | 更改接口命名规范 |
| V1.1 | 20150724 | SJJ1310 | 1.初始化模块增加以配置文件方式初始化函数  2.通讯模块做了相应的调整 |
| V1.0 | 20150723 | SJJ1310 | 初稿 |

**目 录**

[1. 编程指南 3](#_Toc416275747)

[1.1. 接口使用环境 3](#_Toc416275748)

[1.2. 多算法支持 3](#_Toc416275749)

[1.3. 数据类型缩写约定 3](#_Toc416275750)

[2. 接口初始化 4](#_Toc416275751)

[2.1. 打开密码设备 4](#_Toc416275752)

[2.2. 关闭密码设备 4](#_Toc416275753)

[2.3. 创建与密码设备的会话 4](#_Toc416275754)

[2.4. 关闭与密码设备已建立的会话 5](#_Toc416275755)

[3. 密钥管理接口 6](#_Toc416275756)

[3.1. ZMK密钥分散生成ZMK密钥 6](#_Toc416275757)

[3.2. 随机产生ZMK 6](#_Toc416275758)

[3.3. 产生PIK 7](#_Toc416275759)

[3.4. 产生MAK 8](#_Toc416275760)

[3.5. 产生ZEK 9](#_Toc416275761)

[3.6. 分散产生子密钥并用ZMK导出 9](#_Toc416275762)

[3.7. 分散产生子密钥并用RSA公钥导出 10](#_Toc416275763)

[4. PIN加密接口 11](#_Toc416275764)

[4.1. 解密PIN 11](#_Toc416275765)

[5. MAC计算接口 12](#_Toc416275766)

[5.1. 产生ANSIX9.19MAC 12](#_Toc416275767)

[6. 解密磁道 13](#_Toc416275768)

[6.1. 加密磁道 13](#_Toc416275769)

[6.2. 解密磁道 14](#_Toc416275770)

[7. 错误代码 15](#_Toc416275771)

[7.1. 加密机错误 15](#_Toc416275772)

[7.2. 接口层错误 17](#_Toc416275773)

# 编程指南

## 接口使用环境

接口运行环境：SUSE系统11.0 32 位。

其他要求：如需连接多台加密机请提前部署并启动负载均衡软件。具体使用方法请参照《天安密码服务平台用户使用说明V3.2》。

## 多算法支持

接口中提供密钥算法参数，应用系统可根据算法参数指定算法。

密码机支持多种对称密码算法，使用时根据其算法标识使用相应的密码算法。

|  |  |
| --- | --- |
| **算法标识** | **注释** |
| Z | 单倍长的DES算法密钥； |
| X | 双倍长的3DES算法密钥； |
| Y | 三倍长的3DES算法密钥； |

## 数据类型缩写约定

以下章节中涉及到参数长度说明时使用到的缩写均遵循此规则。

|  |  |
| --- | --- |
| n | 可变长度域 |
| A | 字母数字字符，包括任何ASCII字符 |
| H | 十六进制字符，’0’ – ‘9’和’A’ – ‘F’ |
| N | 十进制数字字符，’0’ – ‘9’ |
| B | 二进制字符（字节）， X’00 to X’FF |

# 接口初始化

## 打开密码设备

* 函数说明

phDeviceHandle由函数初始化并填写内容。

* 函数原型

int SDF\_OpenDevice (void \*\*phDeviceHandle, char \*ipaddr, int port );

* 参数说明

|  |  |  |
| --- | --- | --- |
| **名称** | **说明** | **备注** |
| **phDeviceHandle** | 返回设备句柄 |  |
| **ipaddr** | 密码设备IP |  |
| **port** | 密码设备端口 |  |
| 返回值 |  | 0成功，非0失败 |

## 关闭密码设备

* 函数说明

关闭密码设备，并释放相关资源。

* 函数原型

int SDF\_CloseDevice (void \*hDeviceHandle);

* 参数说明

|  |  |  |
| --- | --- | --- |
| **名称** | **说明** | **备注** |
| **hDeviceHandle** | 已打开的设备句柄 |  |
| 返回值 |  | 0成功，非0失败 |

## 创建与密码设备的会话

* 函数说明

创建与密码设备的会话。

* 函数原型

int SDF\_OpenSession (void \*hDeviceHandle, void \*\*phSessionHandle/\*OUT\*/);

* 参数说明

|  |  |  |
| --- | --- | --- |
| **名称** | **说明** | **备注** |
| hDeviceHandle | 已打开的设备句柄 |  |
| phSessionHandle | 返回与密码设备建立的新会话句柄 |  |
| 返回值 |  | 成功返回0，失败返回其他。 |

## 关闭与密码设备已建立的会话

* 函数说明

关闭与密码设备已建立的会话，并释放相关资源。

* 函数原型

int SDF\_CloseSession(void \*hSessionHandle);

* 参数说明

|  |  |  |
| --- | --- | --- |
| **名称** | **说明** | **备注** |
| hSessionHandle | 与密码设备已建立的会话句柄 |  |
| 返回值 |  | 成功返回0，失败返回其他。 |

## 通过制定配置文件的方式初始化接口

* 函数说明

通过制定配置文件的方式初始化接口。

* 函数原型

int SDF\_HsmApiInit (char \*pcConfigFilePath);

* 参数说明

|  |  |  |
| --- | --- | --- |
| **名称** | **说明** | **备注** |
| pcConfigFilePath | 配置文件路径 |  |
| 返回值 |  | 成功返回0，失败返回其他。 |

# 密钥管理接口

## ZMK密钥分散生成ZMK密钥，并使用ZMK加密保护导出

* 函数说明

由一个ZMK分散生成另外一下子密钥，并通过ZMK密钥加密保护导出。

注：对称密钥部分需要支持SM4算法。下同。

* 函数原型

int Tass\_Disper\_Zmk( void \*hSessionHandle,

int iKeyIdx,

char \*pcKey\_LMK,

char \*pcDisData,

int iZmkIdx,

char \* pcZmkKey\_LMK,

char \*pcZmk\_ZMK/\*OUT\*/,

char \*pcZmk\_Lmk/\*OUT\*/,

char \*pcZmkCv/\*OUT\*/ );

* 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **长度&类型** | **备注** |
| hSessionHandle | 与设备建立的会话句柄 |  |  |
| iKeyIdx | 密钥索引 |  | 此参数固定为0 |
| pcKey\_LMK | 密钥密文 | 1A+32H | LMK加密的ZMK密文1A为算法标识 |
| pcDisData | 分散参数 | 16H | 8个字节的分散因子 |
| pcZmk\_Lmk | ZMK密文 | 1A + 32 H | LMK下加密的ZMK，用来本地保存。 |
| pcZmkCv | 密钥校验值 | 8H | 4个字节 |
| pcZmk\_ZMK | ZMK保护 分散的子密钥导出 |  |  |

## 随机产生ZMK

* 函数说明

随机生成ZMK密钥，并使用一个ZMK保护导出，同时生成新ZMK的本地密文。

* 函数原型

int Tass\_Generate\_Zmk( void \*hSessionHandle,

int iKeyIdx,

char \*pcKey\_LMK,

char cZmk\_Scheme,

char \*pcZmk\_Zmk/\*OUT\*/,

char \*pcZmk\_Lmk/\*OUT\*/,

char \*pcZmkCv/\*OUT\*/ );

* 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **长度&类型** | **备注** |
| hSessionHandle | 与设备建立的会话句柄 |  |  |
| iKeyIdx | 密钥索引 |  | 此参数固定为0 |
| pcKey\_LMK | 密钥密文 | 1A+32H | LMK加密的KEK密文1A为算法标识（参考1.2多算法支持） |
| cZmk\_Scheme | 新ZMK算法标识 | 1A | 固定为X |
| pcZmk\_Zmk | ZMK加密的  ZMK密文 | 1A + 32 H | 被ZMK加密的ZMK，用来分发给密钥使用机构。 |
| pcZmk\_Lmk | LMK加密的ZMK密文 | 1A + 32 H | 被LMK加密的ZMK，用来本地保存。 |
| pcZmkCv | 密钥的校验值 | 8H | 4个字节 |

## 产生PIK

* 函数说明

随机产生指定密钥使用机构（ATM、POS等）的PIN加密工作密钥，输出分发和应用系统（P端）存储（如需要）的密钥和校验值。

* 函数原型

int Tass\_Generate\_Pik( void \*hSessionHandle,

int iKeyIdx,

char \*pcKey\_LMK,

char cPik\_Scheme,

char \*pcPik\_Zmk/\*OUT\*/,

char \*pcPik\_Lmk/\*OUT\*/,

char \*pcPikCv/\*OUT\*/ );

* 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **长度&类型** | **备注** |
| hSessionHandle | 与设备建立的会话句柄 |  |  |
| iKeyIdx | 密钥索引 |  | 此参数固定为0 |
| pcKey\_LMK | PIK的保护密钥ZMK | 1A+32H | LMK加密的ZMK密文。 |
| cPik\_Scheme | 新ZMK算法标识 | 1A | 参考1.2多算法支持 |
| pcPik\_Zmk | ZMK加密的PIK密文 | 16 H /  1A + 32 H | 被ZMK加密的PIK，用来分发给密钥使用机构 |
| pcPik\_Lmk | LMK加密的PIK密文 | 16 H /  1A + 32 H | LMK下加密的PIK |
| pcPikCv | PIK密钥的校验值 | 8H | 4个字节 |

## 产生MAK

* 函数说明

随机产生指定机构（ATM、POS等）的MAC计算工作密钥，输出分发和应用系统存储（如需要）的密钥和校验值。

* 函数原型

int ass\_Generate\_Mak ( void \*hSessionHandle,

int iKeyIdx,

char \*pcKey\_LMK,

char cMak\_Scheme,

char \*pcMak\_Zmk/\*OUT\*/,

char \*pcMak\_Lmk/\*OUT\*/,

char \*pcMakCv/\*OUT\*/ );

* 参数说明

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **名称** | **说明** | **类型** | **长度** | **备注** |
| hSessionHandle | 与设备建立的会话句柄 |  |  |  |
| iKeyIdx | 密钥索引 | int |  | 此参数固定为0 |
| pcKey\_LMK | MAK的保护密钥ZMK | char\* | 32H/  1A+32H | LMK加密的ZMK密文。 |
| cMak\_Scheme | 新ZMK算法标识 | char | 1A | 参考1.2多算法支持 |
| pcMak\_Zmk | ZMK加密的MAK密文 | char\* | 16 H /  1A + 32 H | 被ZMK加密的MAK，用来分发给密钥使用机构 |
| pcMak\_Lmk | LMK加密的MAK密文 | char\* | 16 H /  1A + 32 H | 本地保存的MAK。 |
| pcMakCv | MAK密钥的校验值 | char\* | 8H | 4个字节 |

## 产生ZEK

* 函数说明

随机产生指定密钥使用机构（ATM、POS等）的ZEK工作密钥，输出分发和应用系统存储（如需要）的密钥和校验值。

* 函数原型

int Tass\_Generate\_Zek( void \*hSessionHandle,

int iKeyIdx,

char \*pcKey\_LMK,

char cZek\_Scheme,

char \*pcZek\_Zmk/\*OUT\*/,

char \*pcZek\_Lmk/\*OUT\*/,

char \*pcZekCv/\*OUT\*/ );

* 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **长度&类型** | **备注** |
| hSessionHandle | 与设备建立的会话句柄 |  |  |
| iKeyIdx | 密钥索引 |  | 此参数固定为0 |
| pcKey\_LMK | ZEK的保护密钥ZMK | 32H/  1A+32H | LMK加密的ZMK密文。 |
| cZek\_Scheme | 新ZMK算法标识 | 1A | 参考1.2多算法支持 |
| pcZek\_Zmk | ZMK加密的ZEK密文 | 16 H /  1A + 32 H | 被ZMK加密的ZEK，用来分发给密钥使用机构 |
| pcZek\_Lmk | LMK加密的ZEK密文 | 16 H /  1A + 32 H / | 本地保存的ZEK。 |
| pcZekCv | ZEK密钥的校验值 | 8H | 4个字节 |

## 分散产生子密钥并用RSA公钥导出

* 函数说明

将ZMK分散产生子密钥，然后用保护密钥将子密钥加密保护导出。

* 函数原型

int Tass\_DeriveKeyExportedByRsa(void \*hSessionHandle,

char \*pcZmkCipher\_Lmk,

char \*pcPublicKey,

char \*pcDisData,

char \*pcSubkeyCipher\_Tk/\*out\*/,

char \*pcSubkeyCipher\_Lmk,/\*out\*/

char \*pcSubkeyCv/\*out\*/ );

* 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **长度&类型** | **备注** |
| hSessionHandle | 与设备建立的会话句柄 |  |  |
| pcZmkCipher\_Lmk | 待分散的ZMK | 1A+32H |  |
| pcPublicKey | 保护密钥 | n\*2H | DER编码的RSA公钥 |
| pcDisData | 分散因子 | 16H | 8个字节的分散因子 |
| pcSubkeyCipher\_Tk | 子密钥密文 | n\*2H | 保护密钥加密的子密钥密文，公钥加密前对数据进行了PKCS填充方式。 |
| pcSubkeyCipher\_Lmk | LMK加密的子密钥密文 | 1A+32H |  |
| pcSubkeyCv | 子密钥校验值 | 8H | 4个字节 |

## 

## 随机数生成接口

* 函数说明

产生随机数。

* 函数原型

int Tass\_GenRandData(void \*hSessionHandle,

int randlen

char \*randdata/\*out\*/ );

* 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **长度&类型** | **备注** |
| hSessionHandle | 与设备建立的会话句柄 |  |  |
| randlen | 随机数长度 |  |  |
| randdata | 生成的随机数 |  |  |

## 随机生成RSA/SM2密钥对，并加密导出

* 函数说明

随机生成RSA密钥对，并使用ZMK加密导出。

* 函数原型

int Tass\_GenRSAKey(void \*hSessionHandle,

int RsaLen

int zmkIndex

char \*zmk\_Lmk

char \*zmk\_disData

int mode

char \*Rsa\_D\_ZMK/\*out\*/

char \* Rsa\_P\_ZMK /\*out\*/

char \* Rsa\_Q\_ZMK /\*out\*/

char Rsa\_DP\_ZMK /\*out\*/

char \* Rsa\_DQ\_ZMK /\*out\*/

char \* Rsa\_QINV\_ZMK /\*out\*/

char \* Rsa\_N /\*out\*/

char \* Rsa\_E /\*out\*/

char \* Rsa\_LMK /\*out\*/

);

SM2生成需求如下：

int Tass\_GenSm2Key(void \*hSessionHandle,

int zmkIndex

char \*zmk\_Lmk

char \*zmk\_disData

int mode

char \*SM2\_D\_ZMK/\*out\*/

char \* SM2\_PUBKEY /\*out\*/

char \* SM2\_LMK /\*out\*/

);

* 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **长度&类型** | **备注** |
| hSessionHandle | 与设备建立的会话句柄 |  |  |
| char \*zmk\_Lmk | 保护RSA密钥分量的保护密钥 | 为NULL时，不保护导出，只输出RSA本地密文 |  |
| char \*zmk\_disData | ZMK分散参数 | NULL时不分散 |  |
| Mode | 加密算法模式 | ECB/CBC等 |  |
| char \*Rsa\_D\_ZMK/\*out\*/  char \* Rsa\_P\_ZMK /\*out\*/  char \* Rsa\_Q\_ZMK /\*out\*/  char Rsa\_DP\_ZMK /\*out\*/  char \* Rsa\_DQ\_ZMK /\*out\*/  char \* Rsa\_QINV\_ZMK /\*out\*/ | ZMK保护密钥存在时，使用ZMK加密导出的RSA密钥分量 |  |  |
| char \* Rsa\_N /\*out\*/  char \* Rsa\_E /\*out\*/ | 公钥成份，无需加密，直接导出 |  |  |
| char \* Rsa\_LMK /\*out\*/ | RSA本地密文 |  |  |

## RSA/SM2公钥加密运算接口

* 函数说明

产生随机数。

* 函数原型

int Tass\_PubKey\_Oper(void \*hSessionHandle,

int keytype

char \*indata

char \*RSAPubKeyE,

char \*RSAPubKeyN,

char \*SM2\_PUBKEY,

char \*OutData

);

* 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **长度&类型** | **备注** |
| hSessionHandle | 与设备建立的会话句柄 |  |  |
| keytype | 识别RSA或SM2 |  |  |
| char \*indata | 输入数据需要为公钥等长数据，内部不负责数据处理 |  |  |
| char \*RSAPubKeyE,  char \*RSAPubKeyN, | RSA公钥 | 为SM2时此项为NULL |  |
| char \*SM2\_PUBKEY | SM2公钥 | 为RSA时此项为空 |  |

## RSA/SM2私钥解密运算接口

* 函数说明

产生随机数。

* 函数原型

int Tass\_PRIVATE\_Oper(void \*hSessionHandle,

int keytype

char \* Rsa\_LMK

char \* SM2\_LMK

char \* indata

char \* outdata

);

* 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **长度&类型** | **备注** |
| hSessionHandle | 与设备建立的会话句柄 |  |  |
| keytype | 识别SM2或RSA |  |  |
| char \* Rsa\_LMK | RSA本地密钥密文 |  |  |
| char \* SM2\_LMK | SM2本地密钥密文 |  |  |
| Indata | 外部送入数据保障其合法性 | 等长于公钥 |  |
| char \* outdata | 私钥解密后数据 |  |  |

# PIN加密接口

## 解密PIN

* 函数说明

把按ANSIX9.8格式组织的PIN的明文用指定的PIK进行解密。

* 函数原型

int Tass\_Decrypt\_PIN ( void \*hSessionHandle,

int iKeyIdx,

char \*pcKey\_LMK,

char \*pcPinBlk,

int iPinBlkFmt,

char \*pcPan,

char \*pcPinText/\*OUT\*/);

* 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **长度&类型** | **备注** |
| hSessionHandle | 与设备建立的会话句柄 |  |  |
| iKeyIdx | ZPK密钥索引 |  | 0-2048(当此参数为0的时候，采用pcKey\_LMK)。 |
| pcKey\_LMK | ZPK密钥密文 | 16 H /  1 A + 32 H /  1 A + 48 H | LMK加密的ZPK密文。 |
| pcPinBlk | PIN密文 | n\*2H | 加密后的PIN |
| iPinBlkFmt | PINBLOCK格式 |  | 此参数固定为1，即为ANSIX9.8 |
| pcPan | 用户账号 | 12N | 卡号为去掉最后一位校验位，并从右向左取12位。 |
| pcPinText | PIN明文 | nN |  |

# MAC计算接口

## 产生ANSIX9.19MAC

* 函数说明

根据输入的MAC数据采用标准的ANSIX9.19算法产生MAC。

* 函数原形

int Tass\_GenANSI\_Mac ( void \*hSessionHandle,

int iKeyIdx,

char \*pcKey\_LMK,

int iMacDataLen,

char \*pcMacData,

char \*pcMac/\*OUT\*/);

* 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **长度&类型** | **备注** |
| hSessionHandle | 与设备建立的会话句柄 |  |  |
| iKeyIdx | 密钥索引 |  | 此参数固定为0 |
| pcKey**\_**LMK | MAC密钥密文 | 1A+32H | LMK加密的ZAK密文  1A为算法标识（参考1.2多算法支持） |
| iMacDataLen | MAC数据长度 |  | 待计算MAC数据的字符个数 |
| pcMacData | 待计算MAC数据 | n\*2H |  |
| pcMac | 计算出来的MAC值 | 16H |  |

# 解密磁道

## 加密磁道(注：请开放数据加解密接口。)

* 函数说明

使用ZEK加密磁道数据。

* 函数原型

int Tass\_EncryptTrackData ( void \*hSessionHandle,

int iKeyIdx,

char \*pcKey\_LMK,

char \*pcTrackText,

int iTrackTextLen,

int iAlgId,

int iPadFlg,

char \*pcIV,

char \*pcTrackCipher /\*OUT\*/);;

* 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **长度&类型** | **备注** |
| hSessionHandle | 与设备建立的会话句柄 |  |  |
| iKeyIdx | ZEK密钥索引 |  | 该参数固定为0 |
| pcKey\_LMK | ZEK密钥密文 | 16 H /  1 A + 32 H /  1 A + 48 H | LMK加密的ZEK密文。 |
| pcTrackText | 磁道密文 | n\*2H | 磁道数据密文 |
| iTrackTextLen | 磁道密文长度 |  | 磁道数据密文长度 |
| iAlgId | 解密模式 |  | 1 - ECB  2 - CBC |
| iPadFlg | 数据填充标识 |  | ANSI X9.19 - 0  ANSI X9.23 - 1  PBOC MAC - 2 |
| pcIV | 初始化IV | 16H | 当算法标识为2时存在 |
| pcTrackCipher | 磁道密文 | n\*2H | 磁道数据明文 |

## 解密磁道

* 函数说明

使用ZEK解密磁道数据密文。

* 函数原型

int Tass\_DecryptTrackData ( void \*hSessionHandle,

int iKeyIdx,

char \*pcKey\_LMK,

char \*pcTrackCipher,

int iTrackCipherLen,

int iAlgId,

int iPadFlg,

char \*pcIV,

char \*pcTrackText /\*OUT\*/);

* 参数说明

|  |  |  |  |
| --- | --- | --- | --- |
| **名称** | **说明** | **长度&类型** | **备注** |
| hSessionHandle | 与设备建立的会话句柄 |  |  |
| iKeyIdx | ZEK密钥索引 |  | 该参数固定为0 |
| pcKey\_LMK | ZEK密钥密文 | 16 H /  1 A + 32 H /  1 A + 48 H | LMK加密的ZEK密文。 |
| pcTrackCipher | 磁道密文 | n\*2H | 磁道数据密文 |
| iTrackCipherLen | 磁道密文长度 |  | 磁道数据密文长度 |
| iAlgId | 解密模式 |  | 1 - ECB  2 - CBC |
| iPadFlg | 数据填充标识 |  | ANSI X9.19 - 0  ANSI X9.23 - 1  PBOC MAC - 2 |
| pcIV | 初始化IV | 16H | 当算法标识为2时存在 |
| pcTrackText | 磁道明文 | n\*2H | 磁道数据明文 |

# 错误代码

## 加密机错误

|  |  |  |
| --- | --- | --- |
|  | **代码** | **具体描述** |
| 0x00000000 | 00 | 没有错误。 |
| 0x00000001 | 01 | 验证失败。/输入密钥奇偶校验错误警告。 |
| 0x00000002 | 02 | 密钥长度错误（不符合算法要求）。 |
| 0x00000004 | 04 | 无效密钥类型错误。 |
| 0x00000005 | 05 | 无效密钥长度标识。 |
| 0x0000000A | 10 | 源密钥奇偶校验错。 |
| 0x0000000B | 11 | 目的密钥奇偶校验错。/密钥为全0。 |
| 0x0000000C | 12 | 用户存储区域的内容无效。重启、掉电或重写。 |
| 0x0000000D | 13 | 主密钥奇偶校验错。 |
| 0x0000000E | 14 | LMK对02-03加密下的PIN失效。 |
| 0x0000000F | 15 | 无效的输入数据（无效的格式，无效的字符，或者没有提供足够的数据）。 |
| 0x00000010 | 16 | 控制台或打印机没有准备好或者没有连接好。 |
| 0x00000011 | 17 | HSM不在授权状态，或者不能清除PIN输出，或者两种情况都有。 |
| 0x00000012 | 18 | 没有装载文档格式定义。 |
| 0x00000013 | 19 | 指定的Diebold表无效。 |
| 0x00000014 | 20 | PIN块没有包含有效值/解密出来的PIN块无效。出现类似情况多是由于密文和密钥不匹配导致的。 |
| 0x00000015 | 21 | 无效的索引值，或者索引/块数导致了溢出。 |
| 0x00000016 | 22 | 无效的账号。 |
| 0x00000017 | 23 | 无效的PIN块格式代码。 |
| 0x00000018 | 24 | PIN长度小于4或大于12。 |
| 0x00000019 | 25 | 十进制表错误。 |
| 0x0000001A | 26 | 无效的密钥方案。 |
| 0x0000001B | 27 | 不匹配的密钥长度。 |
| 0x0000001C | 28 | 无效的密钥类型。 |
| 0x0000001D | 29 | 密钥函数被禁止。 |
| 0x0000001E | 30 | 参考数无效。 |
| 0x0000001F | 31 | 没有足够的请求入口以提供批量处理。 |
| 0x00000021 | 33 | LMK密钥转换存储区被破坏。 |
| 0x00000028 | 40 | 无效的固件校验和。 |
| 0x00000029 | 41 | 内部的硬件/软件错：RAM已坏，无效的错误代码，等等。 |
| 0x0000002A | 42 | DES错误。 |
| 0x0000002E | 46 | 数据长度错，不是8的倍数。RSA私钥解密时会出现。多由于待解密的密文前没有加明文长度标识导致的。 |
| 0x0000002F | 47 | DSP错误；报告给管理员。 |
| 0x00000031 | 49 | 私钥错误；报告给管理员。 |
| 0x0000003C | 60 | 无此命令。 |
| 0x0000004A | 74 | 无效摘要信息语法（仅仅无哈希模式）。 |
| 0x0000004B | 75 | 无效公钥/私钥对。 |
| 0x0000004C | 76 | 公钥长度错误。 |
| 0x0000004D | 77 | 明文数据块错误。 |
| 0x0000004E | 78 | 私钥长度错误。 |
| 0x0000004F | 79 | 哈希算法对象标识错误。 |
| 0x00000050 | 80 | 数据长度错误。MAC数据（或其它数据）的长度大于或小于指定的长度。 |
| 0x00000051 | 81 | 证书偏移值与长度错。 |
| 0x00000052 | 82 | 无效的密钥索引号，即索引号超出加密机要求的索引范围。 |
| 0x0000005A | 90 | HSM接收的请求信息中的数据奇偶校验错。 |
| 0x0000005B | 91 | 纵向冗余校验（LRC）字符不匹配对输入数据所计算出的值  （当HSM接收到一个透明的异步包时）。 |
| 0x0000005C | 92 | 计数值（命令/数据域）不在有效范围内，或者不正确  （当HSM接收到一个透明的异步包时）。 |
| 0x00000061 | 97 | 解密出错（该错误代码是33指令报，自行添加） |
| 0x00000062 | 98 | RSA公钥加密失败。由于待加密的数据长度超过密钥模长。 |

## 接口层错误

接口错误代码：

|  |  |  |
| --- | --- | --- |
|  | **错误代码** | **具体描述** |
| 0x80000000 | 0 | 没有错误 |
| 0x80000100 | 2147483904 | 函数执行过程中，该会话已经关闭。 |
| 0x80000111 | 2147483921 | 设备句柄无效 |
| 0x80000001 | 2147483649 | 未知错误 |
| 0x80000200 | 2147484160 | 连接加密机失败 |
| 0x80000211 | 2147484177 | socket读取超时 |
| 0x80000214 | 2147484180 | socket写超时 |
| 0x80000220 | 2147484192 | 响应报文错误 |
| 0x80000033 | 2147483699 | 产生密钥失败 |
| 0x80000230 | 2147484208 | 参数为NULL |
| 0x80000231 | 2147484209 | 参数长度无效 |
| 0X80000232 | 2147484210 | 参数值无效 |
| 0X80000233 | 2147484211 | 与加密机通讯失败 |