ICS 35.040
L 80
备案号:

中华人民共和国密码行业标准

GM/T 0063—2018

智能密码钥匙密码应用接口检测规范

Cryptography application interface test specification for cryptographic smart token

（报批稿）

××××-××-××发布　　××××-××-××实施

国家密码管理局　发布

目　　次

前　言

本标准依据 GB/T 1.1-2009 给出的规则起草。

请注意本文件的某些内容可能涉及专利。本文件的发布机构不承担识别这些专利的责任。

本标准由密码行业标准化技术委员会提出并归口。

本标准起草单位：深圳市文鼎创数据科技有限公司、国家密码管理局商用密码检测中心、北京握奇智能科技有限公司、天地融科技股份有限公司、飞天诚信科技股份有限公司。

本标准主要起草人：刘伟丰、周国良、吴玲玲、伍友良、董静、李大为、罗鹏、汪雪林、张渊、李勃、牟宁波、李成伟、朱鹏飞、莫凡。

智能密码钥匙密码应用接口检测规范

1 范围

本标准规定了智能密码钥匙密码应用接口检测环境、检测内容和检测方法。

本标准适用于智能密码钥匙密码应用接口检测，也可用于指导智能密码钥匙的研制和使用。

2 规范性引用文件

下列文件对于本文件的应用是必不可少的。凡是注日期的引用文件，仅注日期的版本适用于本文件。凡是不注日期的引用文件，其最新版本（包括所有的修改单）适用于本文件。

GB/T 25064-2010 信息安全技术 公钥基础设施电子签名格式规范

GB/T 32905-2016 信息安全技术 SM3密码杂凑算法

GB/T 32907-2016 信息安全技术 SM4分组密码算法

GB/T 32915-2016 信息安全技术 二元序列随机性检测规范

GB/T 32918-2016 信息安全技术 SM2椭圆曲线公钥密码算法

GB/T 33560-2017 信息安全技术 密码应用标识规范

GB/T 35275-2017 信息安全技术 SM2密码算法加密签名消息语法规范

GB/T 35276-2017 信息安全技术 SM2密码算法使用规范

GB/T 35291-2017 信息安全技术 智能密码钥匙应用接口规范

GM/T 0014-2012 数字证书认证系统密码协议规范

GM/T 0015-2012 基于SM2密码算法的数字证书格式规范

GM/T 0017-2012 智能密码钥匙密码应用接口数据格式规范

GM/T 0027-2014 智能密码钥匙技术规范

GM/T 0031-2014 安全电子签章密码应用技术规范

3 术语和定义

下列术语和定义适用于本文件。

3.1

容器 container

密码设备中用于保存密钥和证书所划分的唯一性存储空间。

3.2

应用 application

包括容器和文件的一种结构，具备独立的权限管理。

3.3

设备 device

本标准中将智能密码钥匙统称为设备。

3.4

设备认证 device authentication
应用程序对智能密码钥匙的认证。

3.5

设备标签 device label
设备的别名，可以由用户进行设定并存储于设备内部。

3.6

消息鉴别码 message authentication code
又称消息认证码，是消息鉴别算法的输出。

3.7

管理员PIN administrator PIN
管理员的PIN，为ASCII字符串。

3.8

用户PIN user PIN
用户的PIN，为ASCII字符串。

3.9

SM2算法 SM2 algorithm
由GB/T 32918定义的一种公钥密码算法。

3.10

SM3算法 SM3 algorithm
由GB/T 32905定义的一种密码杂凑算法。

3.11

SM4算法 SM4 algorithm
由GB/T 32907定义的一种分组密码算法。

3.12

验证设备 verify equipment
用于密码算法基准运算的检测仪器或设备。

3.13

参考数据 reference data
用于判断密码算法实现正确性的一组数据，包括源数据和目标数据。

3.14

RSA算法 Rivest-Shamir-Adleman algorithm（RSA）
一种基于大整数因子分解问题的公钥密码算法。

4 缩略语

下列缩略语适用于本文件。

ID	Identifier	标识符
MAC	Message Authentication Code	消息鉴别码
PIN	Personal Identification Number	个人身份识别码
RA	Registration Authority	数字证书注册中心

5 送检材料说明

表1列出了送检厂商按照国家密码管理主管部门检测要求提交相关文档资料，文档资料应包含但不限于以下内容。

表 1 送检材料说明

类别	说明
设备（硬件）	适配库文件的设备不少于声明所支持的最大设备数目的两倍。
软件工具	包含检测接口库文件的智能密码钥匙产品安装包。
《资料总体说明》	1. 送检厂商名称、固定设备信息和支持算法等相关信息。 2. 待测接口库名称和版本号，设备检测依赖的操作系统版本。 3. 提交的文档资料和工具清单。 4. 送检样品的初始 PIN 码。 5. 必要的操作说明和注意事项。
《技术工作总结报告》	以结构图的形式，说明整个产品的框架结构，包括产品的各子系统的构成、各子系统的功能和各子系统的实现原理，并附以详细的文字说明。
《不存在隐式通道的声明》	送检厂商应提供产品中涉及密码的部分不存在隐式通道的声明文件。
《密码自测试或自评估报告》	送检厂商应提供检测接口库文件的自测试或自评估报告。
特别说明	1. 资料与送检样品一致性说明：所有提交的受测件样品硬件在技术指标、安全功能、实际使用指令参数上应与技术资料中文本描述一致；软件工具在使用方法、算法实现过程上应与技术资料中文本描述一致。 2. 送检厂商不予提供或认为不必要提供的资料和工具应在上述说明文件中逐项加以说明。

6 检测环境

6.1 检测环境拓扑图

智能密码钥匙密码应用接口检测环境参考拓扑图，如图1所示：

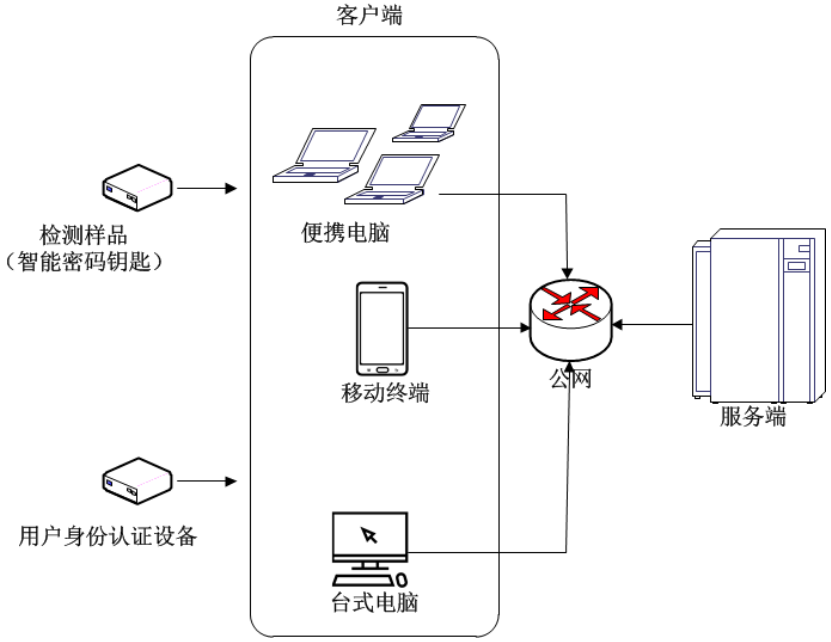

图 1　智能密码钥匙密码应用接口检测环境参考拓扑图

6.2 检测仪器

检测仪器应符合适用的国家及行业标准，检测仪器名称见表 2 所示：

表 2　检测仪器列表

仪器名称	备注
客户端	用于运行操作系统及检测平台客户端软件
服务端	用于运行操作系统及检测平台服务端
用户身份认证设备	用于认证用户身份

6.3 检测软件

检测软件应符合适用的国家及行业标准，检测软件名称见表 3 所示：

表 3　检测软件列表

软件名称	备注
检测平台软件	用于执行检测的软件工具
操作系统	用于运行检测平台软件的操作系统

7 检测内容

7.1 应用功能检测

智能密码钥匙应用功能检测的目的是检测密码应用接口在典型应用场景下的适用性。应用功能检测的检测内容包括：

——证书申请
——证书下载
——证书更新
——证书导入
——数字签名
——数字信封生成
——数字信封解封

7.2 接口功能检测

智能密码钥匙密码应用接口功能检测的目的是检测密码应用接口实现和运行的正确性。功能检测的检测内容包括：

——设备管理
——访问控制
——应用管理
——文件管理
——容器管理
——密码服务

7.3 安全性检测

智能密码钥匙安全性检测的目的是检测智能密码钥匙应用接口在设计和实现过程中的安全性。安全性检测的检测内容包括：

——权限分类
——权限使用
——设备认证
——PIN 码安全要求
——密钥安全要求
——随机数安全要求

7.4 兼容性检测

智能密码钥匙兼容性检测的目的是检测智能密码钥匙在不同系统下应用接口是否能正常使用。兼容性检测的检测内容包括：

——系统兼容性
——交错兼容性

7.5 互操作性检测

智能密码钥匙互操作性检测的目的是检测智能密码钥匙与其他样品的互操作性。

8 检测方法

8.1 应用功能检测

8.1.1 证书申请

检测目的：

检测智能密码钥匙应用接口实现是否支持 GM/T 0014 规定的客户端与 RA 之间的证书申请协议。

检测条件：

设备已连接，预定应用已打开，预定容器已存在。

检测过程：

a) SM2 证书申请

步骤1. 调用 SKF_OpenApplication 接口打开预定应用；

步骤2. 调用 SKF_OpenContainer 接口打开预定容器；

步骤3. 调用 SKF_VerifyPIN 接口验证用户 PIN；

步骤4. 调用 SKF_GenECCKeyPair 接口，在预定容器生成 SM2 签名密钥对；

步骤5. 调用 SKF_ExportPublicKey 接口导出 SM2 签名密钥对的公钥；

步骤6. 调用 SKF_ECCSignData 接口计算签名。输入数据为待签数据根据 GB/T 35276 经过 SM2 签名预处理的结果。待签数据为 GM/T 0014 规定的 CertReqMessages 消息，其中 publicKey 字段为步骤 5 所导出的公钥。

b) RSA 证书申请

步骤1. 调用 SKF_OpenApplication 接口打开预定应用；

步骤2. 调用 SKF_OpenContainer 接口打开预定容器；

步骤3. 调用 SKF_VerifyPIN 接口验证用户 PIN；

步骤4. 调用 SKF_GenRSAKeyPair 接口，在预定容器生成 RSA 签名密钥对，密钥长度不少于 2048 位；

步骤5. 调用 SKF_ExportPublicKey 接口导出 RSA 签名密钥对的公钥；

步骤6. 调用 SKF_RSASignData 接口计算签名。输入数据为 GM/T 0014 规定的 CertReqMessages 消息，其中 publicKey 字段为步骤 5 所导出的公钥。

通过标准：

能够得到签名。

8.1.2 证书下载

检测目的：

检测智能密码钥匙应用接口实现是否支持 GM/T 0014 规定的客户端与 RA 之间的证书下载协议。

检测条件：

设备已连接，预定应用已打开，预定容器中存在签名密钥对。

检测过程：

步骤1. 调用 SKF_OpenApplication 接口打开预定应用；

步骤2. 调用 SKF_OpenContainer 接口打开预定容器；

步骤3. 调用 SKF_VerifyPIN 接口验证用户 PIN；

步骤4. 调用 SKF_ImportCertificate 接口，向预定容器导入数字证书。数字证书包含预定容器中的签名公钥，格式应符合 GM/T 0015；

步骤5. 调用 SKF_ExportCertificate 接口，从预定容器导出数字证书。

通过标准：

步骤 5 导出的数字证书与步骤 4 导入的数字证书相同。

8.1.3 证书更新

检测目的：

检测智能密码钥匙应用接口实现是否支持证书更新。

检测条件：

设备已连接，预定应用已打开，预定容器中存在签名密钥对。

检测过程：

a) SM2 证书更新

步骤1. 调用 SKF_OpenApplication 接口打开预定应用；

步骤2. 调用 SKF_VerifyPIN 接口验证用户 PIN；

步骤3. 调用 SKF_CreateContainer 接口，在预定应用创建容器；

步骤4. 调用 SKF_GenECCKeyPair 接口,在步骤 3 所创建的容器生成 SM2 签名密钥对；

步骤5. 调用 SKF_ExportPublicKey 接口导出步骤 4 所生成的 SM2 签名密钥对的公钥；

步骤6. 调用 SKF_OpenContainer 接口打开预定容器；

步骤7. 调用 SKF_ECCSignData 接口，使用预定容器的签名密钥对计算签名。输入数据为待签数据根据 GB/T 35276 经过 SM2 签名预处理的结果。待签数据为 GM/T 0014 规定的 CertReqMessages 消息，其中 publicKey 字段为步骤 5 所导出的公钥。

b) RSA 证书更新

步骤1. 调用 SKF_OpenApplication 接口打开预定应用；

步骤2. 调用 SKF_VerifyPIN 接口验证用户 PIN；

步骤3. 调用 SKF_CreateContainer 接口，在预定应用创建容器；

步骤4. 调用 SKF_GenRSAKeyPair 接口,在步骤 3 所创建的容器生成 RSA 签名密钥对,密钥长度不少于 2048 位；

步骤5. 调用 SKF_ExportPublicKey 接口导出步骤 4 所生成的 RSA 签名密钥对的公钥，公钥长度不少于 2048 位；

步骤6. 调用 SKF_OpenContainer 接口打开预定容器；

步骤7. 调用 SKF_RSASignData 接口计算签名。输入数据为 GM/T 0014 规定的 CertReqMessages 消息，其中 publicKey 字段为步骤 5 所导出的公钥。

通过标准：

能够得到签名。

8.1.4 证书导入

检测目的：

检测智能密码钥匙应用接口实现是否支持数字证书导入。

检测条件：

设备已连接，预定应用已打开，预定容器中存在签名密钥对。

检测过程：

a) SM2 加密证书导入

步骤1. 调用 SKF_OpenApplication 接口打开预定应用；
步骤2. 调用 SKF_OpenContainer 接口打开预定容器；
步骤3. 调用 SKF_VerifyPIN 接口验证用户 PIN；
步骤4. 调 SKF_ ImportECCKeyPair 接口在预定容器导入 SM2 加密密钥对，数据应符合 GB/T 35291；
步骤5. 调用 SKF_ImportCertificate 接口，向预定容器导入加密数字证书。数字证书包含预定容器中的加密公钥，格式应符合 GM/T 0015。

b) SM2 签名证书导入
步骤1. 调用 SKF_OpenApplication 接口打开预定应用；
步骤2. 调用 SKF_OpenContainer 接口打开预定容器；
步骤3. 调用 SKF_VerifyPIN 接口验证用户 PIN；
步骤4. 调用 SKF_ImportCertificate 接口，向预定容器导入签名数字证书。数字证书包含预定容器中的签名公钥，格式应符合 GM/T 0015。

c) RSA 加密证书导入
步骤1. 调用 SKF_OpenApplication 接口打开预定应用；
步骤2. 调用 SKF_OpenContainer 接口打开预定容器；
步骤3. 调用 SKF_VerifyPIN 接口验证用户 PIN；
步骤4. 调 SKF_ ImportRSAKeyPair 接口在预定容器导入 RSA 加密密钥对，密钥长度不少于 2048 位；
步骤5. 调用 SKF_ImportCertificate 接口，向预定容器导入加密数字证书。数字证书包含预定容器中的加密公钥，格式应符合 GB/T 25064。

d) RSA 签名证书导入
步骤1. 调用 SKF_OpenApplication 接口打开预定应用；
步骤2. 调用 SKF_OpenContainer 接口打开预定容器；
步骤3. 调用 SKF_VerifyPIN 接口验证用户 PIN；
步骤4. 调用 SKF_ImportCertificate 接口，向预定容器导入签名数字证书。数字证书包含预定容器中的签名公钥，格式应符合 GB/T 25064。

通过标准：

从预定容器导出数字证书，能够导出数字证书和相匹配的公钥，所导出的数字证书与导入的数字证书相同。

8.1.5 数字签名

检测目的：

检测智能密码钥匙应用接口实现是否支持 GM/T 0031 规定的电子签章应用。

检测条件：

设备已连接，预定应用已打开，预定容器中存在签名密钥对和签名证书。

检测过程：

a) SM2 数字签名
步骤1. 调用 SKF_OpenApplication 接口打开预定应用；
步骤2. 调用 SKF_OpenContainer 接口打开预定容器；
步骤3. 调用 SKF_VerifyPIN 接口验证用户 PIN；
步骤4. 调用 SKF_ECCSignData 接口，生成电子签章签名值。输入数据为待签数据根据 GB/T 35276 经过 SM2 签名预处理的结果。待签数据符合 GM/T

0031 规定的电子签章数据格式要求，其中包含的数字证书为预定容器中的签名证书。

b) RSA 数字签名

步骤1. 调用 SKF_OpenApplication 接口打开预定应用；

步骤2. 调用 SKF_OpenContainer 接口打开预定容器；

步骤3. 调用 SKF_VerifyPIN 接口验证用户 PIN；

步骤4. 调用 SKF_RSASignData 接口，生成电子签章签名值。输入数据符合 GM/T 0031 规定的电子签章数据格式要求，其中包含的数字证书为预定容器中的签名证书。

通过标准：

得到电子签章签名值，应遵循GM/T 0031打包形成电子签章数据，电子签章数据应能按照GM/T 0031的规定验证通过。

8.1.6 数字信封生成

检测目的：

检测智能密码钥匙应用接口实现是否支持生成数字信封。

检测条件：

设备已连接，预定应用已打开，预定容器已存在。

检测过程：

a) SM2 数字信封

步骤1. 调用 SKF_OpenApplication 接口打开预定应用；

步骤2. 调用 SKF_OpenContainer 接口打开预定容器；

步骤3. 调用 SKF_VerifyPIN 接口验证用户 PIN；

步骤4. 调用 SKF_ECCExportSessionKey 接口，调用时传入预定 SM2 公钥，加密会话密钥并导出其密文；

步骤5. 调用 SKF_EncryptInit 接口，进行加密初始化。调用时传入的密钥句柄为步骤 4 中输出的密钥句柄；

步骤6. 调用 SKF_Encrypt 接口，对预定数据进行加密。调用时输入的算法标识应符合 GB/T 33560 的规定；

步骤7. 将预定 SM2 公钥、步骤 4 中导出的会话密钥、步骤 6 中输入的算法标识和输出的加密结果以及其他所需信息按照 GB/T 35275 的规定组成数字信封。

b) RSA 数字信封

步骤1. 调用 SKF_OpenApplication 接口打开预定应用；

步骤2. 调用 SKF_OpenContainer 接口打开预定容器；

步骤3. 调用 SKF_VerifyPIN 接口验证用户 PIN；

步骤4. 调用 SKF_RSAExportSessionKey 接口，生成会话密钥并导出。调用时传入预定 RSA 公钥；

步骤5. 调用 SKF_EncryptInit 接口，进行加密初始化。调用时传入的密钥句柄为步骤 4 中输出的密钥句柄；

步骤6. 调用 SKF_Encrypt 接口，对预定数据进行加密；

步骤7. 将预定 RSA 公钥、步骤 4 中导出的会话密钥、步骤 6 中输出的加密结果以及其他所需信息按照 PKCS#7 的规定组成数字信封。

通过标准：

得到数字信封且能使用预定私钥解封得到预定数据。

8.1.7 数字信封解封

检测目的：

检测智能密码钥匙应用接口实现适用于解封数字信封。

检测条件：

设备已连接，预定应用已打开，预定容器中存在与数字信封对应的加密密钥对。

检测过程：

步骤1. 调用 SKF_OpenApplication 接口打开预定应用；

步骤2. 调用 SKF_OpenContainer 接口打开预定容器；

步骤3. 调用 SKF_VerifyPIN 接口验证用户 PIN；

步骤4. 调用 SKF_ImportSessionKey 接口导入会话密钥，输入数据为数字信封中的 RecipientInfo::encryptedKey 字段；

步骤5. 调用 SKF_DecryptInit 接口，进行解密初始化。调用时传入的密钥句柄为步骤 4 中输出的密钥句柄，算法标识来自数字信封中的 EncryptedContentInfo::contentEncryptionAlgorithm 字段；

步骤6. 调用 SKF_Decrypt 接口，对数字信封中的 EncryptedContentInfo::encryptedContent 字段解密。

通过标准：

解密结果与预定数据相同。

8.2 接口功能检测

8.2.1 设备管理

8.2.1.1 等待设备插拔事件

检测目的：

检测是否能获取到设备插拔事件和设备名称。

检测条件：

无。

检测过程：

——正常情况检测

步骤1. 启动线程，调用 SKF_WaitForDevEvent 接口；

步骤2. 插入设备；

步骤3. 拔出设备。

——异常情况检测

- 使用非法参数调用本接口，应返回错误码。
- 分配给设备名称所占用空间小于返回数据长度，应返回错误码。
- 插拔不能被本接口识别的设备不会影响到该接口功能。

通过标准：

正常情况检测步骤 2 和步骤 3 均可通过 SKF_WaitForDevEvent 接口返回正确的设备名称、名称长度和事件类型。

异常情况检测得到预期结果。

8.2.1.2 取消等待设备插拔事件

检测目的：

检测取消等待设备插拔事件功能。

检测条件：

无。

检测过程：

步骤1. 在一个进程中创建两个线程 A 和 B；

步骤2. 线程 A 调用 SKF_WaitForDevEvent 接口，线程 A 阻塞；

步骤3. 线程 B 调用 SKF_CancelWaitForDevEvent 接口。

通过标准：

步骤 3 执行完成后，步骤 2 中的线程 A 解除阻塞。

8.2.1.3 枚举设备

检测目的：

检测是否能获取当前系统中的设备列表。

检测条件：

无。

检测过程：

——正常情况检测

步骤1. 没有插入设备，调用 SKF_EnumDev 接口，设置 bPresent=FALSE，获取驱动支持的设备名称列表，设备列表总数为 N；

步骤2. 插入 1~N 个设备，调用 SKF_EnumDev 接口，设置 bPresent=TRUE，获取当前已插入的设备名称列表。

——异常情况检测

- 使用非法参数调用本接口，应返回错误码。
- 分配给设备名称列表所占用空间小于返回数据长度，应返回错误码。

通过标准：

正常情况检测获取到的设备名称列表与插入的设备信息相符，并且在步骤 2 中设置 szNameList=NULL 时，应能通过 pulSize 返回所需的内存空间大小。

异常情况检测得到预期结果。

8.2.1.4 连接设备

检测目的：

检测接口的连接设备功能。

检测条件：

预定名称的设备已插入。

检测过程：

——正常情况检测

步骤1. 调用 SKF_ConnectDev 接口，连接预定名称的设备；

步骤2. 调用 SKF_Transmit 接口，发送符合 GM/T 0017 的指令；

步骤3. 调用 SKF_DisconnectDev 接口，断开连接；

步骤4. 再次执行步骤 2。

——异常情况检测

使用非法参数调用本接口，应返回错误码。

通过标准：

正常情况检测步骤 2 中，返回结果数据应符合 GM/T 0017 的规定；步骤 4 中 SKF_Transmit 接口，返回错误码。

异常情况检测得到预期结果。

8.2.1.5 断开连接

检测目的：

检测是否能成功断开一个已经连接的设备，并释放句柄。

检测条件：

无。

检测过程：

——正常情况检测

本检测项作为 8.2.1.4、8.2.1.10 的一部分进行检测。

——异常情况检测

使用非法参数调用本接口，应返回错误码。

通过标准：

正常情况检测和异常情况检测均得到预期结果。

8.2.1.6 获取设备状态

检测目的：

检测接口能否获取设备是否存在的状态。

检测条件：

无。

检测过程：

——正常情况检测

步骤1. 调用 SKF_GetDevState 接口，获取预定名称设备的状态。

——异常情况检测

使用非法参数调用本接口，应返回错误码。

通过标准：

正常情况检测获取到的设备状态与预定名称设备的当前连接状态相符。

异常情况检测得到预期结果。

8.2.1.7 设置设备标签

检测目的：

检测是否能正确设置设备标签。

检测条件：

设备已连接。

检测过程：

——正常情况检测

步骤1. 调用 SKF_GetDevInfo 接口，获取设备的一些特征信息，记录其中输出的设备标签；

步骤2. 调用 SKF_SetLabel 接口，设置设备标签，所设置的设备标签与步骤 1 中所记录的设备标签不同；

步骤3. 再次执行步骤 1。

——异常情况检测

使用非法参数调用本接口，应返回错误码。

通过标准：

正常情况检测步骤 3 所记录的设备标签与步骤 2 设置设备标签相同。

异常情况检测得到预期结果。

8.2.1.8 获取设备信息

检测目的：

检测是否能正确获取设备的一些特征信息。

检测条件：

设备已连接。

检测过程：

——正常情况检测

本检测项作为 8.2.1.7 的一部分进行检测。

——异常情况检测

使用非法参数调用本接口，应返回错误码。

通过标准：

正常情况检测所获取到的厂商信息和支持的算法等与表 1 的《资料总体说明》所描述的信息匹配。

异常情况检测得到预期结果。

8.2.1.9 锁定设备

检测目的：

检测接口锁定设备功能。

检测条件：

设备已连接。

检测过程：

——正常情况检测

步骤1. 创建两个线程 A 和 B；

步骤2. 线程 A 调用 SKF_LockDev 接口，使用预定时间值，锁定设备；

步骤3. 线程 A 调用 SKF_Transmit 接口，发送符合 GM/T 0017 的指令，返回结果数据应符合 GM/T 0017 的规定；

步骤4. 线程 B 调用 SKF_Transmit 接口，发送符合 GM/T 0017 的指令，在预定时间内，应不成功；

步骤5. 线程 A 调用 SKF_UnlockDev 接口，解锁设备；

步骤6. 线程 B 调用 SKF_Transmit 接口，发送符合 GM/T 0017 的指令，返回结果数据应符合 GM/T 0017 的规定。

——异常情况检测

使用非法参数调用本接口，应返回错误码。

通过标准：

正常情况检测和异常情况检测均得到预期结果。

8.2.1.10 解锁设备

检测目的：

检测接口解锁设备功能。

检测条件：

设备已连接。

检测过程：

——正常情况检测

步骤1. 调用 SKF_LockDev 接口，超时时间使用预定值，锁定设备；

步骤2. 调用 SKF_DisconnectDev 接口，应不成功；

步骤3. 调用 SKF_UnlockDev 接口，解锁设备；

步骤4. 调用 SKF_DisconnectDev 接口，应成功。

——异常情况检测

使用非法参数调用本接口，应返回错误码。

通过标准：

正常情况检测和异常情况检测均得到预期结果。

8.2.1.11 设备命令传输

检测目的：

检测设备命令传输功能。

检测条件：

设备已连接。

检测过程：

——正常情况检测

本检测项作为 8.2.1.9 的一部分进行检测。

——异常情况检测

使用非法参数调用本接口，应返回错误码。

通过标准：

正常情况检测和异常情况检测均得到预期结果。

8.2.2 访问控制

8.2.2.1 修改设备认证密钥

检测目的：

检测是否能正确修改设备认证密钥。

检测条件：

设备已连接。

检测过程：

——正常情况检测

步骤1. 使用原设备认证密钥，调用 SKF_DevAuth 接口，完成设备认证；

步骤2. 使用与原设备认证密钥不同的新设备认证密钥，调用 SKF_ChangeDevAuthKey 接口，修改设备认证密钥；

步骤3. 使用原设备认证密钥，调用 SKF_DevAuth 接口，应不成功；

步骤4. 使用原设备认证密钥，调用 SKF_ChangeDevAuthKey 接口，应不成功；

步骤5. 使用新设备认证密钥，调用 SKF_DevAuth 接口，应成功；

步骤6. 使用原设备认证密钥，调用 SKF_ChangeDevAuthKey 接口，应成功。

——异常情况检测

使用非法参数调用本接口，应返回错误码。

通过标准：

正常情况检测和异常情况检测均得到预期结果。

8.2.2.2 设备认证

检测目的：

检测是否能正确操作设备。

检测条件：

设备已连接。

检测过程：

——正常情况检测

步骤1. 调用 SKF_GetDevInfo 接口，获取设备信息，得到设备认证所使用的对称算法；

步骤2. 按照 GB/T 35291-2017 的 8.2.3 完成设备认证。

——异常情况检测

- 使用非法参数调用本接口，应返回错误码。
- 使用错误的设备认证密钥组织正确的认证数据，调用 SKF_DevAuth 接口，应返回错误码。
- 使用正确的设备认证密钥组织错误的认证数据，调用 SKF_DevAuth 接口，应返回错误码。

通过标准：

正常情况检测和异常情况检测均得到预期结果。

8.2.2.3 修改 PIN

检测目的：

检测是否能正确修改预定应用的管理员 PIN 或用户 PIN。

检测条件：

设备已连接，预定应用已打开，用户 PIN 和管理员 PIN 没有锁死。

检测过程：

——正常情况检测

步骤1. 使用正确的原 PIN，调用 SKF_ChangePIN 接口，设置新的 PIN，且新 PIN 应与原 PIN 不同；

步骤2. 使用正确的新 PIN，调用 SKF_VerifyPIN 接口，校验 PIN，应成功；

步骤3. 修改 PIN 成功后，调用 SKF_GetPINInfo 获取当前剩余重试次数应与最大重试次数相同。

——异常情况检测

- 使用非法参数调用本接口，应返回错误码。
- 使用长度小于 6 位的 PIN，应返回错误码。
- 使用错误的原 PIN 修改，应返回错误码，同时返回错误重试次数。
- 使用错误的原 PIN 修改，直至 PIN 码锁死；再次使用正确的 PIN 修改，应返回错误码。

通过标准：

正常情况检测和异常情况检测均得到预期结果。

8.2.2.4 获取 PIN 信息

检测目的：

检测是否能正确获取预定应用下的 PIN 信息。

检测条件：

设备已连接，预定应用已打开。

检测过程：

——正常情况检测

步骤1. 调用 SKF_GetPINInfo 接口，获取 PIN 信息；

步骤2. 使用正确的 PIN，调用 SKF_ChangePIN 接口，修改 PIN；

步骤3. 调用 SKF_GetPINInfo 接口，获取 PIN 信息；

步骤4. 使用错误的 PIN，调用 SKF_VerifyPIN 接口，获取 PIN 码重试次数；

步骤5. 调用 SKF_GetPINInfo 接口，获取 PIN 信息；

步骤6. 使用正确的 PIN，调用 SKF_VerifyPIN 接口，获取 PIN 码重试次数；

步骤7. 调用 SKF_GetPINInfo 接口，获取 PIN 信息；

步骤8. 使用错误的 PIN，调用 SKF_VerifyPIN 接口，直至 PIN 码锁死；

步骤9. 调用 SKF_GetPINInfo 接口，获取 PIN 信息。

——异常情况检测

使用非法参数调用本接口，应返回错误码。

通过标准：

正常情况检测步骤 1、步骤 3、步骤 5、步骤 7 和步骤 9 应遵循 GB/T 35291。

异常情况检测可得到预期结果。

8.2.2.5 校验 PIN

检测目的：

检测是否能正确校验预定应用的管理员 PIN 或用户 PIN。

检测条件：

设备已连接，预定应用已打开，用户 PIN 和管理员 PIN 没有锁死。在预定应用中创建文件需要管理员权限。

检测过程：

——正常情况检测

a) 校验用户 PIN：

步骤1. 使用正确的用户 PIN，执行 8.1.1 证书申请，应成功；

步骤2. 使用错误的用户 PIN，执行 8.1.1 证书申请，应不成功。

b) 校验管理员 PIN：

步骤1. 调用 SKF_CreateFile 接口，创建文件，应不成功；

步骤2. 使用正确的管理员 PIN，调用 SKF_VerifyPIN 接口；

步骤3. 调用 SKF_CreateFile 接口，创建文件，应成功；

步骤4. 使用错误的管理员 PIN，调用 SKF_VerifyPIN 接口；

步骤5. 调用 SKF_CreateFile 接口，创建文件，应不成功。

——异常情况检测

- 使用非法参数调用本接口，应返回错误码。
- 使用长度小于 6 位的 PIN，应返回错误码。
- 使用错误的 PIN 认证，应返回错误码，同时返回错误重试次数。

- 使用错误的 PIN 认证，直至 PIN 码锁死；再次使用正确的 PIN 认证，应返回错误码。
- 在步骤 3 调用 SKF_CreateFile 接口之前，调用 SKF_ClearSecureState 接口，步骤 3 应不成功。

通过标准：

正常情况检测和异常情况检测均得到预期结果。

8.2.2.6 解锁 PIN

检测目的：

检测是否能正确解锁预定应用已锁定的用户 PIN。

检测条件：

设备已连接，预定应用已打开，用户 PIN 已锁死，管理员 PIN 没有锁死。

检测过程：

——正常情况检测

步骤1. 使用用户 PIN，调用 SKF_VerifyPIN 接口，应不成功；

步骤2. 使用正确的管理员 PIN，调用 SKF_UnblockPIN 接口，解锁用户 PIN；

步骤3. 使用步骤 2 中设置的用户 PIN，执行 8.2.2.5 的用户 PIN 校验。

——异常情况检测

- 使用非法参数调用本接口，应返回错误码。
- 使用长度小于 6 位的管理员 PIN，应返回错误码。
- 使用错误的管理员 PIN，应返回错误码，同时返回错误重试次数。
- 使用错误的管理员 PIN，直至 PIN 码锁死；再次使用正确的管理员 PIN，应返回错误码。

通过标准：

正常情况检测和异常情况检测均得到预期结果。

8.2.2.7 清除应用安全状态

检测目的：

检测是否能正确清除预定应用的安全状态。

检测条件：

设备已连接，预定应用已打开。

检测过程：

本检测项作为8.2.2.5的一部分进行检测。

通过标准：

检测得到预期结果。

8.2.3 应用管理

8.2.3.1 创建应用

检测目的：

检测是否能正确在设备中创建应用。

检测条件：

设备已连接，设备权限已获得。

检测过程：

——正常情况检测

步骤1. 调用 SKF_EnumApplication 接口，获得设备中的应用名称列表；
步骤2. 调用 SKF_GetDevInfo 接口，获取设备信息；
步骤3. 调用 SKF_CreateApplication 接口，创建与步骤 1 的列表中名称不同的应用；
步骤4. 调用 SKF_GetDevInfo 接口，获取设备信息，设备信息中的用户可用空间大小应不大于步骤 2 中所获得用户可用空间大小；
步骤5. 调用 SKF_EnumApplication 接口，获得设备中的应用名称列表，此列表包括步骤 1 和步骤 3 中的应用名称。

——异常情况检测

- 使用非法参数调用本接口，应返回错误码。
- 创建的应用空间大于设备剩余空间，应返回错误码。
- 设备权限未获得，应返回错误码。

通过标准：

正常情况检测和异常情况检测均得到预期结果。

参考 8.2.2 进一步验证所创建应用的有效性。

8.2.3.2 枚举应用

检测目的：

检测是否能正确枚举设备中存在的所有应用。

检测条件：

设备已连接。

检测过程：

——正常情况检测

本检测项作为 8.2.3.1 和 8.2.3.3 的一部分进行检测。

——异常情况检测

使用非法参数调用本接口，应返回错误码。

通过标准：

正常情况检测在 8.2.3.1 和 8.2.3.3 的步骤 1 中设置 szAppName=NULL 时，应能通过 pulSize 返回所需的内存空间大小。

异常情况检测可得到预期结果。

8.2.3.3 删除应用

检测目的：

检测是否能正确删除设备中预定的应用。

检测条件：

设备已连接，设备权限已获得，预定应用已打开。

检测过程：

——正常情况检测

步骤1. 调用 SKF_EnumApplication 接口，获得设备中的应用名称列表，名称列表应包含预定应用名称；
步骤2. 调用 SKF_GetDevInfo 接口，获取设备信息；
步骤3. 使用预定应用名称，调用 SKF_DeleteApplication 接口；
步骤4. 调用 SKF_GetDevInfo 接口，获取设备信息，设备信息中的用户可用空间大小应不小于步骤 2 中所获得用户可用空间大小；

步骤5. 调用 SKF_EnumApplication 接口，获得设备中的应用名称列表,此列表不包含预定应用名称。

——异常情况检测

- 使用非法参数调用本接口，应返回错误码。
- 设备权限未获得，应返回错误码。

通过标准：

正常情况检测和异常情况检测均得到预期结果。

8.2.3.4 打开应用

检测目的：

检测是否能正确打开设备中预定的应用。

检测条件：

设备已连接，预定应用已存在。

检测过程：

——正常情况检测

步骤1. 调用 SKF_OpenApplication 接口，打开预定应用，获得应用句柄；

步骤2. 使用步骤 1 获得应用句柄，调用 SKF_GetPINInfo 接口，应获得 PIN 信息；

步骤3. 调用 SKF_CloseApplication 接口，关闭预定应用；

步骤4. 使用步骤 1 获得应用句柄，调用 SKF_GetPINInfo 接口，应不成功。

——异常情况检测

使用非法参数调用本接口，应返回错误码。

通过标准：

正常情况检测和异常情况检测均得到预期结果。

8.2.3.5 关闭应用

检测目的：

检测是否能正确关闭设备中已打开的应用，并释放应用句柄。

检测条件：

设备已连接，预定应用已打开。

检测过程：

——正常情况检测

步骤1. 调用 SKF_CloseApplication 接口，关闭预定应用；

步骤2. 调用 SKF_OpenApplication 接口，打开预定应用，获得应用句柄；

步骤3. 调用 SKF_CreateFile 接口，创建文件，应不成功；

步骤4. 调用 SKF_VerifyPIN 接口，验证用户 PIN；

步骤5. 调用 SKF_CloseApplication 接口，关闭预定应用；

步骤6. 调用 SKF_OpenApplication 接口，打开预定应用，获得应用句柄；

步骤7. 调用 SKF_CreateFile 接口，创建文件，应成功；

步骤8. 调用 SKF_UnblockPIN 接口，解锁用户 PIN，应不成功；

步骤9. 调用 SKF_VerifyPIN 接口，验证管理员 PIN；

步骤10. 调用 SKF_CloseApplication 接口，关闭预定应用；

步骤11. 调用 SKF_OpenApplication 接口，打开预定应用，获得应用句柄；

步骤12. 调用 SKF_UnblockPIN 接口，解锁用户 PIN，应成功。

——异常情况检测

使用非法参数调用本接口，应返回错误码。

通过标准：

正常情况检测和异常情况检测均得到预期结果。

8.2.4 文件管理

8.2.4.1 创建文件

检测目的：

检测是否能正确在预定应用下创建文件。

检测条件：

设备已连接，预定应用已打开，安全状态已满足。

检测过程：

——正常情况检测

步骤1. 调用 SKF_EnumFiles 接口，枚举设备预定应用中已存在的文件；

步骤2. 调用 SKF_GetDevInfo 接口，获取设备信息；

步骤3. 调用SKF_CreateFile接口,创建一个和步骤1文件名称不相同的文件；

步骤4. 调用 SKF_GetDevInfo 接口，获取设备信息，设备信息中的用户可用空间大小应不大于步骤 2 中所获得用户可用空间大小；

步骤5. 调用 SKF_EnumFiles 接口，获得设备预定应用下的文件名称列表,此列表包括步骤 1 和步骤 3 中的文件名称。

——异常情况检测

- 使用非法参数调用本接口，应返回错误码。
- 创建的应用空间大于设备剩余空间，应返回错误码。
- 安全状态不满足，应返回错误码。
- 文件的名称长度大于 32 字节，应返回错误码。

通过标准：

正常情况检测和异常情况检测均得到预期结果。

8.2.4.2 删除文件

检测目的：

检测是否能正确删除预定应用下的文件。

检测条件：

设备已连接，安全状态已满足，预定应用已打开，预定文件已存在。

检测过程：

——正常情况检测

步骤1. 调用 SKF_EnumFiles 接口，枚举设备中预定应用中已存在的文件；

步骤2. 调用 SKF_GetDevInfo 接口，获取设备信息；

步骤3. 使用预定文件名称，调用 SKF_DeleteFile 接口；

步骤4. 调用 SKF_GetDevInfo 接口，获取设备信息，设备信息中的用户可用空间大小应不小于步骤 2 中所获得用户可用空间大小；

步骤5. 调用 SKF_EnumFiles 接口，获得设备中预定应用下的文件名称列表，此列表不包含步骤 3 删除的文件。

——异常情况检测

- 使用非法参数调用本接口，应返回错误码。

- 安全状态不满足，应返回错误码。

通过标准：

正常情况检测和异常情况检测均得到预期结果。

8.2.4.3 枚举文件

检测目的：

检测是否能正确枚举预定应用下存在的所有文件。

检测条件：

设备已连接，预定应用已打开。

检测过程：

——正常情况检测

本检测项作为 8.2.4.1 和 8.2.4.2 的一部分进行检测。

——异常情况检测

使用非法参数调用本接口，应返回错误码。

通过标准：

正常情况检测在 8.2.4.1 和 8.2.4.2 的步骤 1 中设置 szFileList=NULL 时，应能通过 pulSize 返回文件信息所需要的空间大小。

异常情况检测得到预期结果。

8.2.4.4 获取文件属性

检测目的：

检测是否能正确获取预定应用下预定文件的属性信息。

检测条件：

设备已连接，预定应用已打开，安全状态已满足。

检测过程：

——正常情况检测

步骤1. 调用 SKF_CreateFile 接口，创建文件；

步骤2. 调用 SKF_GetFileInfo 接口，获取文件属性。

——异常情况检测

使用非法参数调用本接口，应返回错误码。

通过标准：

正常情况检测步骤 2 获取到的文件属性与步骤 1 建立的文件属性相符。

异常情况检测得到预期结果。

8.2.4.5 读文件

检测目的：

检测是否能正确从预定文件的预定位置读取预定长度的数据。

检测条件：

设备已连接，安全状态已满足，预定应用已打开，预定文件已写入预定数据。

检测过程：

——正常情况检测

调用 SKF_ReadFile 接口，读取文件数据。

——异常情况检测

- 使用非法参数调用本接口，应返回错误码。

- 创建的应用空间大于设备剩余空间，应返回错误码。
- 安全状态不满足，应返回错误码。
- 文件的名称长度大于 32 字节，应返回错误码。

通过标准：

正常情况检测读出的数据应与预定数据一致。

异常情况检测得到预期结果。

8.2.4.6 写文件

检测目的：

检测是否能正确向预定文件的预定位置写入预定长度的数据。

检测条件：

设备已连接，安全状态已满足，预定应用已打开，预定文件已存在。

检测过程：

——正常情况检测

调用 SKF_WriteFile 接口，向文件写入预定数据。

——异常情况检测

- 使用非法参数调用本接口，应返回错误码。
- 创建的应用空间大于设备剩余空间，应返回错误码。
- 安全状态不满足，应返回错误码。
- 文件的名称长度大于 32 字节，应返回错误码。

通过标准：

正常情况检测步骤 1 后插拔设备，再读取文件数据应与预定数据一致。

异常情况检测得到预期结果。

8.2.5 容器管理

8.2.5.1 创建容器

检测目的：

检测是否能正确在预定应用下建立预定名称的容器，并返回容器句柄。

检测条件：

设备已连接，预定应用已打开。

检测过程：

——正常情况检测

步骤1. 调用 SKF_EnumContainer 接口，枚举预定应用中的容器列表；

步骤2. 调用 SKF_VerifyPIN 接口验证用户 PIN；

步骤3. 调用 SKF_CreateContainer 接口，创建与步骤 1 的列表中名称不同的容器；

步骤4. 调用 SKF_EnumContainer 接口，获得设备预定应用中的容器名称列表，此列表包括步骤 1 和步骤 3 中的应用名称。

——异常情况检测

- 使用非法参数调用本接口，应返回错误码。
- 安全状态不满足，应返回错误码。

通过标准：

正常情况检测和异常情况检测均得到预期结果。

8.2.5.2 删除容器

检测目的：

检测是否能正确删除预定应用下预定名称的容器，并释放容器相关的资源。

检测条件：

设备已连接，预定应用已打开。

检测过程：

——正常情况检测

步骤1. 调用 SKF_EnumContainer 接口，枚举预定应用中的容器列表；

步骤2. 调用 SKF_VerifyPIN 接口验证用户 PIN；

步骤3. 调用 SKF_DeleteContainer 接口；

步骤4. 调用 SKF_EnumContainer 接口，获得设备预定应用中的容器名称列表，此列表不包含步骤 3 删除的容器名称。

——异常情况检测

- 使用非法参数调用本接口，应返回错误码。
- 安全状态不满足，应返回错误码。

通过标准：

正常情况检测和异常情况检测均得到预期结果。

8.2.5.3 打开容器

检测目的：

检测是否能正确打开预定应用下的预定容器，并获取容器句柄。

检测条件：

设备已连接，预定应用已打开，预定容器已存在。

检测过程：

——正常情况检测

步骤1. 调用 SKF_OpenContainer 接口，打开预定容器；

步骤2. 调用 SKF_GetContainerType 接口，获取容器类型。

——异常情况检测

使用非法参数调用本接口，应返回错误码。

通过标准：

正常情况检测步骤 2 所获取的容器类型与容器实际情况相符。

异常情况检测得到预期结果。

8.2.5.4 关闭容器

检测目的：

检测是否能正确关闭预定应用下已打开的容器，并释放容器句柄相关资源。

检测条件：

设备已连接，预定应用已打开，预定容器已打开。

检测过程：

——正常情况检测

步骤1. 调用 SKF_GetContainerType 接口，获取容器类型，应成功；

步骤2. 调用 SKF_CloseContainer 接口，关闭预定容器；

步骤3. 调用 SKF_GetContainerType 接口，获取容器类型，应返回错误码。

——异常情况检测

使用非法参数调用本接口，应返回错误码。

通过标准：

正常情况检测和异常情况检测均得到预期结果。

8.2.5.5 枚举容器

检测目的：

检测是否能正确枚举预定应用下存在的所有容器。

检测条件：

设备已连接，预定应用已打开。

检测过程：

——正常情况检测

本检测项作为 8.2.5.1 和 8.2.5.2 的一部分进行检测。

——异常情况检测

使用非法参数调用本接口，应返回错误码。

通过标准：

正常情况检测在 8.2.5.1 和 8.2.5.2 的步骤 1 中设置 szContainerList=NULL 时，应能通过 pulSize 返回容器信息所需要的空间大小。

异常情况检测得到预期结果。

8.2.5.6 获取容器类型

检测目的：

检测是否能正确获取预定应用下的预定容器的相关信息。

检测条件：

设备已连接，预定应用已打开，预定容器已存在。

检测过程：

——正常情况检测

本检测项作为 8.2.5.3 和 8.2.5.4 的一部分进行检测。

——异常情况检测

使用非法参数调用本接口，应返回错误码。

通过标准：

正常情况检测和异常情况检测均得到预期结果。

8.2.5.7 导入数字证书

检测目的：

检测是否能正确向预定容器内导入数字证书。

检测条件：

设备已连接，预定应用已打开，预定容器已打开。

检测过程：

——正常情况检测

调用 SKF_ImportCertificate 接口，导入数字证书到预定容器。

——异常情况检测

使用非法参数调用本接口，应返回错误码。

通过标准：

正常情况检测步骤 1 后插拔设备，再导出容器内的数字证书应与步骤 1 中导入的数

字证书一致。

异常情况检测得到预期结果。

8.2.5.8 导出数字证书

检测目的：

检测是否能正确从当前容器内导出数字证书。

检测条件：

设备已连接，预定应用已打开，预定容器中已导入数字证书。

检测过程：

——正常情况检测

调用 SKF_ExportCertificate 接口，导出预定容器中的数字证书。

——异常情况检测

- 使用非法参数调用本接口，应返回错误码。
- 容器内不存在对应类型的数字证书，应返回错误码。

通过标准：

正常情况检测导出的数字证书应与预定容器中的数字证书一致，并且在步骤 1 中设置 pbCert=NULL 时，应能通过 pulCertLen 返回数据所需要缓冲区的长度。

异常情况检测得到预期结果。

8.2.6 密码服务

8.2.6.1 生成随机数

检测目的：

检测是否能正确生成预定长度的随机数。

检测条件：

设备已连接。

检测过程：

——正常情况检测

调用 SKF_GenRandom 接口，获取随机数。

——异常情况检测

使用非法参数调用本接口，应返回错误码。

通过标准：

正常情况检测可获取到预定长度的随机数，且每次获取的数据不相同。

异常情况检测得到预期结果。

8.2.6.2 生成 RSA 签名密钥对

检测目的：

检测是否能在预定容器中正确生成合法有效的 RSA 签名密钥对，并输出签名公钥。

检测条件：

设备已连接，预定容器已打开。

检测过程：

——正常情况检测

步骤1. 调用 SKF_VerifyPIN 接口，验证用户 PIN;

步骤2. 调用 SKF_GenRSAKeyPair 接口,生成签名密钥对，返回签名公钥;

步骤3. 调用 SKF_RSASignData 接口,用步骤 2 生成的密钥对被签名数据进行签

名，返回签名结果。

——异常情况检测

- 使用非法参数调用本接口，应返回错误码。
- 安全状态不满足，应返回错误码。

通过标准：

正常情况检测用步骤 2 返回的签名公钥对步骤 3 返回的签名结果验签应成功。

异常情况检测得到预期结果。

8.2.6.3 导入 RSA 加密密钥对

检测目的：

检测是否能在预定容器中正确导入 RSA 加密密钥对。

检测条件：

设备已连接，含有 RSA 签名密钥对的预定容器已打开。

检测过程：

——正常情况检测

步骤1. 调用 SKF_VerifyPIN 接口，验证用户 PIN；

步骤2. 调用 SKF_ImportRSAKeyPair 接口，导入 RSA 加密密钥对；

步骤3. 调用 SKF_ImportSessionKey 接口，导入预定会话密钥；

步骤4. 使用步骤 3 导入的会话密钥句柄，调用 SKF_EncryptInit 接口，完成加密初始化；

步骤5. 调用 SKF_Encrypt 接口，对预定原文加密。

——异常情况检测

- 使用非法参数调用本接口，应返回错误码。
- 安全状态不满足，应返回错误码。
- RSA 签名密钥对不存在，应返回错误码。

通过标准：

正常情况检测在步骤 5 中得到的加密结果与预定密文相符。

异常情况检测可得到预期结果。

8.2.6.4 RSA 签名

检测目的：

检测是否能正确使用预定签名私钥，对预定数据进行数字签名,并输出签名结果。

检测条件：

设备已连接，含有 RSA 签名密钥对的预定容器已打开。

检测过程：

——正常情况检测

步骤1. 调用 SKF_VerifyPIN 接口，验证用户 PIN；调用 SKF_RSASignData 接口，对预定数据进行签名，返回签名结果；

步骤2. 调用 SKF_ExportPublicKey 接口，导出 RSA 签名公钥。

——异常情况检测

- 使用非法参数调用本接口，应返回错误码。
- 安全状态不满足，应返回错误码。
- RSA 签名密钥对不存在，应返回错误码。

通过标准：

正常情况检测用步骤 3 返回的签名公钥对步骤 2 返回的签名结果验签应成功。

异常情况检测得到预期结果。

8.2.6.5 RSA 验签

检测目的：

检测是否能正确使用从外部输入的 RSA 公钥对数据进行签名验证。

检测条件：

设备已连接。

检测过程：

——正常情况检测

使用待验证的 RSA 公钥和原文、签名值，调用 SKF_ RSAVerify 接口。

——异常情况检测

使用非法参数调用本接口，应返回错误码。

通过标准：

正常情况检测待验证的 RSA 公钥和原文、签名值匹配应验签成功，不匹配应验签失败。

异常情况检测可得到预期结果。

8.2.6.6 RSA 生成并导出会话密钥

检测目的：

检测是否能正确生成会话密钥，并能用外部公钥加密导出。

检测条件：

设备已连接，预定容器已打开。

检测过程：

——正常情况检测

步骤1. 使用预定公钥，设置正确的会话密钥算法标识，调用 SKF_RSAExportSessionKey 接口，导出会话密钥；

步骤2. 使用步骤 1 导出的会话密钥句柄，调用 SKF_EncryptInit 接口，完成加密初始化；

步骤3. 调用 SKF_Encrypt 接口，对预定原文加密。

——异常情况检测

- 使用非法参数调用本接口，应返回错误码。
- 步骤 2 之前调用 SKF_CloseHandle 接口，关闭步骤 1 导出的会话密钥句柄，步骤 2 应返回错误码。

通过标准：

正常情况检测对所有支持的会话密钥算法标识，使用预定私钥解密得到会话密钥，并使用会话密钥解密步骤 3 中得到的加密密文，解密后得到的数据与预定原文相同。

异常情况检测得到预期结果。

8.2.6.7 生成 ECC 签名密钥对

检测目的：

检测是否能在预定应用的当前容器中正确生成 ECC 签名密钥对，并输出签名公钥。

检测条件：

设备已连接，预定容器已打开。

检测过程：

——正常情况检测

步骤1. 调用 SKF_VerifyPIN 接口，验证用户 PIN。

步骤2. 调用 SKF_GenECCKeyPair 接口，生成 ECC 签名密钥对，返回签名公钥；

步骤3. 调用 SKF_ECCSignData 接口，用步骤 2 生成的密钥对被签名数据进行签名，返回签名结果。

——异常情况检测

- 使用非法参数调用本接口，应返回错误码。
- 安全状态不满足，应返回错误码。

通过标准：

正常情况检测用步骤 2 返回的签名公钥对步骤 3 返回的签名结果验签应成功。

异常情况检测得到预期结果。

8.2.6.8 导入 ECC 加密密钥对

检测目的：

检测是否能正确向预定应用的预定容器中导入 ECC 加密密钥对。

检测条件：

设备已连接，含有 ECC 签名密钥对的预定容器已打开。

检测过程：

——正常情况检测

步骤1. 调用 SKF_VerifyPIN 接口，验证用户 PIN；

步骤2. 调用 SKF_ImportECCKeyPair 接口，导入 ECC 加密密钥对；

步骤3. 调用 SKF_ImportSessionKey 接口，导入预定的会话密钥；

步骤4. 使用步骤 3 导入的会话密钥句柄，调用 SKF_EncryptInit 接口，完成加密初始化；

步骤5. 调用 SKF_Encrypt 接口，对预定原文加密。

——异常情况检测

- 使用非法参数调用本接口，应返回错误码。
- 安全状态不满足，应返回错误码。
- SM2 签名密钥对不存在，应返回错误码。

通过标准：

正常情况检测使用步骤 3 导入的预定会话密钥对步骤 5 得到的密文解密后，得到的结果与预定原文相同。

异常情况检测得到预期结果。

8.2.6.9 ECC 签名

检测目的：

检测是否能正确采用预定容器中签名私钥对输入数据进行签名，并输出签名结果。

检测条件：

设备已连接，含有 ECC 签名密钥对的预定容器已打开。

输入的待签名数据为待签名数据杂凑值，当使用 SM2 算法时，该输入数据为原始数据经过 SM2 签名预处理的结果，预处理过程遵循 GB/T 35276。

检测过程：

——正常情况检测

步骤1. 调用 SKF_VerifyPIN 接口,验证用户 PIN;调用 SKF_ECCSignData 接口，返回签名结果;

步骤2. 调用 SKF_ExportPublicKey 接口，导出 ECC 签名公钥。

——异常情况检测

- 使用非法参数调用本接口，应返回错误码。
- 安全状态不满足，应返回错误码。
- SM2 签名密钥对不存在，应返回错误码。

通过标准:

正常情况检测用步骤 3 返回的签名公钥对步骤 2 返回的签名结果验签应成功。

异常情况检测得到预期结果。

8.2.6.10 ECC 验签

检测目的:

检测是否能正确用从外部输入的 ECC 公钥对数据进行签名验证。

检测条件:

设备已连接。

检测过程:

——正常情况检测

使用待验证的 SM2 公钥和待签数据经过 SM2 签名预处理的结果、签名值，调用 SKF_ECCVerify 接口。

——异常情况检测

使用非法参数调用本接口，应返回错误码。

通过标准:

正常情况检测待验证的 SM2 公钥和待签数据经过 SM2 签名预处理的结果、签名值匹配应验签成功，不匹配应验签失败。

异常情况检测得到预期结果。

8.2.6.11 生成会话密钥并加密导出会话密钥密文

检测目的:

检测是否能正确在预定的容器中生成会话密钥并用外部公钥加密导出。

检测条件:

设备已连接，预定容器已打开。

检测过程:

——正常情况检测

步骤1. 使用预定公钥，设置正确的会话密钥算法标识，调用 SKF_ECCExportSessionKey 接口，导出会话密钥;

步骤2. 使用步骤 1 导出的会话密钥句柄，调用 SKF_EncryptInit 接口，完成加密初始化;

步骤3. 调用 SKF_Encrypt 接口，对预定原文加密。

——异常情况检测

- 使用非法参数调用本接口，应返回错误码。
- 步骤 2 之前调用 SKF_CloseHandle 接口,关闭步骤 1 导出的会话密钥句柄，步骤 2 应返回错误码。

通过标准:

正常情况检测对所有支持的会话密钥算法标识，使用预定私钥解密得到会话密钥，并使用会话密钥解密步骤 3 中得到的加密密文，解密后得到的数据与预定原文相同。

异常情况检测得到预期结果。

8.2.6.12 ECC 外来公钥加密

检测目的：

检测是否能正确使用外部传入的 ECC 公钥对输入数据做加密运算，并输出加密结果。

检测条件：

设备已连接。

检测过程：

——正常情况检测

使用预定 ECC 密钥对，调用 SKF_ExtECCEncrypt 接口，对预定原文进行加密。

——异常情况检测

使用非法参数调用本接口，应返回错误码。

通过标准：

正常情况检测使用预定私钥对加密密文进行解密，得到的结果与预定原文相同。

异常情况检测得到预期结果。

8.2.6.13 ECC 生成密钥协商参数并输出

检测目的：

检测是否能正确使用 ECC 密钥协商算法，为计算会话密钥而产生协商参数，并输出临时 ECC 密钥对的公钥及密钥协商句柄。

检测条件：

设备已连接，预定容器已打开。

检测过程：

——正常情况检测

步骤1. 调用待测样品的 SKF_GenerateAgreementDataWithECC 接口，使用预定发起方 ID，返回临时 ECC 密钥对的公钥及密钥协商句柄；

步骤2. 辅助设备利用步骤 1 返回的临时 ECC 密钥对的公钥、步骤 1 使用的发起方 ID 以及预定响应方 ID，产生协商参数并计算会话密钥，输出临时 ECC 密钥对的公钥；

步骤3. 调用待测样品的 SKF_GenerateKeyWithECC 接口，使用步骤 1 返回的密钥协商句柄、步骤 2 使用的发起方 ID 与响应方 ID 以及步骤 2 输出的临时 ECC 密钥对的公钥，计算会话密钥，返回会话密钥句柄；

步骤4. 辅助设备使用步骤 2 计算的会话密钥对预定原文加密，返回数据密文；

步骤5. 待测样品使用步骤3返回的会话密钥句柄对步骤4返回的数据密文解密，返回解密结果。

——异常情况检测

- 使用非法参数调用本接口，应返回错误码。
- 步骤 1 之后调用 SKF_CloseHandle 接口，关闭步骤 1 所返回的密钥协商句柄，步骤 3 应返回错误码。

通过标准：

正常情况检测步骤 5 的解密结果与预定原文一致。

异常情况检测得到预期结果。

8.2.6.14 ECC 产生协商数据并计算会话密钥

检测目的：

检测是否能正确使用 ECC 密钥协商算法，产生协商参数并计算会话密钥，并输出临时 ECC 密钥对的公钥及产生的会话密钥 ID。

检测条件：

设备已连接，预定容器已打开。

检测过程：

——正常情况检测

步骤1. 调用待测样品的 SKF_GenerateAgreementDataAndKeyWithECC 接口，利用预定 ECC 密钥对的公钥、预定发起方 ID 以及预定响应方 ID，产生协商参数并计算会话密钥，返回临时 ECC 密钥对的公钥及会话密钥句柄；

步骤2. 辅助设备利用预定 ECC 密钥对的私钥、步骤 1 使用的发起方 ID 与响应方 ID 以及步骤 1 输出的 ECC 密钥对的公钥，计算会话密钥，对预定原文加密，返回数据密文；

步骤3. 待测样品使用步骤 1 返回的会话密钥句柄对步骤 2 返回的数据密文解密，返回解密结果。

——异常情况检测

- 使用非法参数调用本接口，应返回错误码。
- 步骤 3 之前调用 SKF_CloseHandle 接口，关闭步骤 1 所返回的会话密钥句柄，步骤 3 应返回错误码。

通过标准：

正常情况检测步骤 3 的解密结果与预定原文一致。

异常情况检测得到预期结果。

8.2.6.15 ECC 计算会话密钥

检测目的：

检测是否能正确使用 ECC 密钥协商算法，使用自身密钥协商句柄和响应方的协商参数计算会话密钥，同时返回会话密钥句柄。

检测条件：

设备已连接，预定容器已打开。

检测过程：

——正常情况检测

本项目作为 8.2.6.13 的一部分进行检测。

——异常情况检测

使用非法参数调用本接口，应返回错误码。

通过标准：

正常情况检测和异常情况检测均得到预期结果。

8.2.6.16 导出公钥

检测目的：

检测是否能正确地从预定容器中导出签名公钥或加密公钥。

检测条件：

设备已连接，含有密钥对的预定容器已打开。

检测过程：

——正常情况检测

步骤1. 调用 SKF_ExportPublicKey 接口导出公钥；

步骤2. 对于签名密钥对，调用 SKF_ECCSignData 接口或 SKF_RSASignData 接口对预定数据计算签名（当使用 SM2 算法时，调用 SKF_ECCSignData 接口的输入数据为预定数据经过 SM2 签名预处理的结果），使用步骤 1 导出的公钥对返回的签名结果验签；对于加密密钥对，调用 SKF_Decrypt 接口对预定密文数据解密得到明文数据。

——异常情况检测

- 使用非法参数调用本接口，应返回错误码。
- 密钥对不存在，应返回错误码。

通过标准：

正常情况检测应验签通过或者明文数据与预定数据一致，并且步骤 1 中设置 pbBlob=NULL 时，应能通过 pulBlobLen 返回 pbBlob 缓冲区的 长度。

异常情况检测得到预期结果。

8.2.6.17 导入会话密钥

检测目的：

检测是否能正确导入会话密钥。

检测条件：

设备已连接，含有加密密钥对的预定容器已打开。

检测过程：

——正常情况检测

步骤1. 调用 SKF_VerifyPIN 接口，验证用户 PIN；

步骤2. 调用 SKF_ImportSessionKey 接口，将预定会话密钥密文导入，得到会话密钥句柄。预定会话密钥密文是预定会话密钥用预定公钥加密的结果，预定公钥及相应的密钥对在容器中；

步骤3. 使用步骤 2 导入的会话密钥句柄，调用 SKF_EncryptInit 接口，完成加密初始化；

步骤4. 调用 SKF_Encrypt 接口，使用步骤 2 返回的会话密钥句柄对预定原文加密得到密文。

——异常情况检测

- 使用非法参数调用本接口，应返回错误码。
- 安全状态不满足，应返回错误码。
- 加密密钥对不存在，应返回错误码。
- 步骤 3 之前调用 SKF_CloseHandle 接口，关闭步骤 2 导入的会话密钥句柄，步骤 3 应返回错误码。

通过标准：

正常情况检测步骤 4 的密文与预定密文一致，预定密文是预定会话密钥对预定原文加密后的结果。

异常情况检测得到预期结果。

8.2.6.18 加密初始化

检测目的：

检测是否能正确进行加密操作前的参数设置。

检测条件：

设备已连接，加密密钥已存在。

检测过程：

——正常情况检测

步骤1. 调用 SKF_EncryptInit 接口，设置加密密钥句柄和算法参数；

步骤2. 调用 SKF_Encrypt 接口，对预定原文进行加密，返回密文数据。

——异常情况检测

- 使用非法参数调用本接口，应返回错误码。
- 加密密钥不存在，应返回错误码。
- 不执行步骤 1，步骤 2 应返回错误码。
- 步骤 2 之前调用 SKF_CloseHandle 接口，关闭步骤 1 所使用的密钥句柄，步骤 2 应返回错误码。

通过标准：

正常情况检测对所支持的算法参数步骤 2 返回的密文数据与预定密文一致。

异常情况检测得到预期结果。

8.2.6.19 单组数据加密

检测目的：

检测是否能正确进行单一分组数据的加密操作。

检测条件：

设备已连接，加密密钥已存在。

检测过程：

——正常情况检测

本检测项作为 8.2.6.18 的一部分进行检测。

——异常情况检测

- 使用非法参数调用本接口，应返回错误码。
- 加密密钥不存在，应返回错误码。

通过标准：

正常情况检测在 8.2.6.18 的步骤 2 中设置 pbEncryptedData=NULL 时，应能通过 pulEncryptedLen 返回加密后数据长度。

异常情况检测得到预期结果。

8.2.6.20 多组数据加密

检测目的：

检测是否能正确进行多个分组数据的加密操作。

检测条件：

设备已连接，加密密钥已存在。

检测过程：

——正常情况检测

步骤1. 调用 SKF_EncryptInit 接口，设置密钥句柄和算法参数；

步骤2. 调用 SKF_EncryptUpdate 接口，对多个预定明文数据分组进行加密，返回密文数据；

步骤3. 调用 SKF_EncryptFinal 接口结束加密。

——异常情况检测

- 使用非法参数调用本接口，应返回错误码。
- 加密密钥不存在，应返回错误码。
- 不执行步骤 1，直接执行步骤 2 或步骤 3 应返回错误码。
- 步骤 2 之前调用 SKF_CloseHandle 接口，关闭步骤 1 所使用的密钥句柄，步骤 2 应返回错误码。

通过标准：

正常情况检测步骤 2 和步骤 3 返回的加密密文与预定密文一致。

异常情况检测得到预期结果。

8.2.6.21 结束加密

检测目的：

检测是否能正确结束多个分组数据的加密操作并输出剩余加密结果。

检测条件：

设备已连接，加密密钥已存在。

检测过程：

——正常情况检测

本检测项作为 8.2.6.20 的一部分进行检测。

——异常情况检测

- 使用非法参数调用本接口，应返回错误码。
- 加密密钥不存在，应返回错误码。

通过标准：

正常情况检测和异常情况检测均得到预期结果。

8.2.6.22 解密初始化

检测目的：

检测是否能正确进行解密操作前的参数设置。

检测条件：

设备已连接，解密密钥已存在。

检测过程：

——正常情况检测

步骤1. 调用 SKF_DecryptInit 接口，设置解密密钥句柄和算法参数；

步骤2. 调用 SKF_Decrypt 接口，对预定密文进行解密，返回明文数据。

——异常情况检测

- 使用非法参数调用本接口，应返回错误码。
- 解密密钥不存在，应返回错误码。
- 不执行步骤 1，步骤 2 应返回错误码。
- 步骤 2 之前调用 SKF_CloseHandle 接口，关闭步骤 1 所使用的密钥句柄，步骤 2 应返回错误码。

通过标准：

正常情况检测对所支持的算法参数步骤 2 返回的明文数据与预定原文一致。

异常情况检测得到预期结果。

8.2.6.23 单组数据解密

检测目的：

验证是否能正确进行单组数据解密运算。

检测条件：

设备已连接，解密密钥已存在。

检测过程：

——正常情况检测

本检测项作为 8.2.6.22 的一部分进行检测。

——异常情况检测

- 使用非法参数调用本接口，应返回错误码。
- 解密密钥不存在，应返回错误码。

通过标准：

正常情况检测在 8.2.6.22 的步骤 2 中设置 pbData=NULL 时，应能通过 pulDataLen 返回解密后数据长度。

异常情况检测得到预期结果。

8.2.6.24 多组数据解密

检测目的：

验证是否能正确进行多组数据解密运算。

检测条件：

设备已连接，解密密钥已存在。

检测过程：

——正常情况检测

步骤1. 调用 SKF_DecryptInit 接口，设置密钥句柄和算法参数；

步骤2. 调用 SKF_DecryptUpdate 接口，对多个预定密文数据分组进行解密，返回明文数据；

步骤3. 调用 SKF_DecryptFinal 接口结束解密。

——异常情况检测

- 使用非法参数调用本接口，应返回错误码。
- 解密密钥不存在，应返回错误码。
- 不执行步骤 1，直接执行步骤 2 或步骤 3 应返回错误码。
- 步骤 2 之前调用 SKF_CloseHandle 接口，关闭步骤 1 所使用的密钥句柄，步骤 2 应返回错误码。

通过标准：

正常情况检测步骤 2 和步骤 3 返回的明文数据与预定原文一致。

异常情况检测得到预期结果。

8.2.6.25 结束解密

检测目的：

检测是否能正确结束多个分组数据的解密操作，并输出剩余解密结果。

检测条件：

设备已连接，解密密钥已存在。

检测过程：

——正常情况检测

本检测项作为 8.2.6.24 的一部分进行检测。

——异常情况检测

- 使用非法参数调用本接口，应返回错误码。
- 解密密钥不存在，应返回错误码。

通过标准：

正常情况检测和异常情况检测均得到预期结果。

8.2.6.26 密码杂凑初始化

检测目的：

检测是否能正确进行初始化密码杂凑计算操作。

检测条件：

设备已连接。

检测过程：

——正常情况检测

本检测项作为 8.2.6.27、8.2.6.28 的一部分进行检测。

——异常情况检测

使用非法参数调用本接口，应返回错误码。

通过标准：

正常情况检测和异常情况检测均得到预期结果。

8.2.6.27 单组数据密码杂凑

检测目的：

检测是否能正确对单组数据进行密码杂凑运算。

检测条件：

设备已连接。

检测过程：

——正常情况检测

步骤1. 调用 SKF_DigestInit 接口，获取密码杂凑对象句柄；

步骤2. 调用 SKF_Digest 接口，对预定原文计算杂凑值。

——异常情况检测

- 使用非法参数调用本接口，应返回错误码。
- 步骤 2 之前调用 SKF_CloseHandle 接口，关闭步骤 1 所获取的密码杂凑对象句柄，步骤 2 应返回错误码。

通过标准：

正常情况检测步骤 2 中得到的杂凑值应与预定结果一致，步骤 2 中设置 pbHashData=NULL 时，应能通过 pulHashLen 返回结果数据实际长度。

异常情况检测得到预期结果。

注：在步骤 1 中，当 ulAlgID 为 SGD_SM3 且 ulIDLen 不为 0 时，预定签名者公钥和签名者 ID 有效，执行 SM2 算法签名预处理 1 操作。

8.2.6.28 多组数据密码杂凑

检测目的：

检测是否能正确对多组数据进行密码杂凑运算。

检测条件：

设备已连接。

检测过程：

——正常情况检测

步骤1. 调用 SKF_DigestInit 接口，获取密码杂凑对象句柄；

步骤2. 调用 SKF_DigestUpdate 接口，对多个预定数据进行杂凑计算；

步骤3. 调用 SKF_DigestFinal 接口，结束杂凑计算，得到杂凑结果。

——异常情况检测

- 使用非法参数调用本接口，应返回错误码。
- 步骤 2 之前调用 SKF_CloseHandle 接口，关闭步骤 1 所获取的密码杂凑对象句柄，步骤 2 应返回错误码。

通过标准：

正常情况检测步骤 3 中得到的杂凑结果应与预定杂凑值一致。

异常情况检测得到预期结果。

8.2.6.29 结束密码杂凑

检测目的：

验证是否能正确结束多组数据的密码杂凑计算操作，并输出密码杂凑结果。

检测条件：

设备已连接。

检测过程：

——正常情况检测

本检测项作为 8.2.6.28 的一部分进行检测。

——异常情况检测

使用非法参数调用本接口，应返回错误码。

通过标准：

正常情况检测 8.2.6.28 的步骤 3 中设置 pHashData=NULL 时，应能通过 pulHashLen 返回密码杂凑结果的长度。

异常情况检测得到预期结果。

8.2.6.30 消息鉴别码运算初始化

检测目的：

检测是否能正确进行初始化消息鉴别码计算操作。

检测条件：

设备已连接，密钥已存在。

检测过程：

——正常情况检测

本检测项作为 8.2.6.31、8.2.6.32 的一部分进行检测。

——异常情况检测

- 使用非法参数调用本接口，应返回错误码。
- 密钥不存在，应返回错误码。

通过标准：

正常情况检测和异常情况检测均得到预期结果。

8.2.6.31 单组数据消息鉴别码运算

检测目的：

检测是否能正确进行单组数据消息鉴别码运算。

检测条件：

设备已连接，密钥已存在。

检测过程：

——正常情况检测

步骤1. 调用 SKF_MacInit 接口，获得消息鉴别码对象句柄；

步骤2. 调用 SKF_Mac 接口，使用预定数据，计算消息鉴别码，返回结果。

——异常情况检测

- 使用非法参数调用本接口，应返回错误码。
- 密钥不存在，应返回错误码。
- 步骤 2 之前调用 SKF_CloseHandle 接口，关闭步骤 1 所获取的消息鉴别码对象句柄，步骤 2 应返回错误码。

通过标准：

正常情况检测步骤 2 中得到的结果应与预定鉴别码一致，步骤 2 中设置 pbMacData=NULL 时，应能通过 pulMacLen 返回 Mac 结果的长度。

异常情况检测得到预期结果。

8.2.6.32 多组数据消息鉴别码运算

检测目的：

检测是否能正确进行多组数据消息鉴别码运算。

检测条件：

设备已连接，密钥已存在。

检测过程：

——正常情况检测

步骤1. 调用 SKF_MacInit 接口，获得消息鉴别码对象句柄；

步骤2. 调用 SKF_MacUpdate 接口，对多个预定数据进行消息鉴别码计算；

步骤3. 调用 SKF_MacFinal 接口，结束消息鉴别码计算，得到结果。

——异常情况检测

- 使用非法参数调用本接口，应返回错误码。
- 密钥不存在，应返回错误码。
- 步骤 2 之前调用 SKF_CloseHandle 接口，关闭步骤 1 所获取的消息鉴别码对象句柄，步骤 2 应返回错误码。

通过标准：

正常情况检测步骤 3 中得到的结果应与预定鉴别码一致。

异常情况检测得到预期结果。

8.2.6.33 结束消息鉴别码运算

检测目的：

检测是否能正确结束多组数据的消息鉴别码运算。

检测条件：

设备已连接，密钥已存在。

检测过程：

——正常情况检测

本检测项作为 8.2.6.32 的一部分进行检测。

——异常情况检测

- 使用非法参数调用本接口，应返回错误码。
- 密钥不存在，应返回错误码。

通过标准：

正常情况检测8.2.6.32的步骤3中设置pbMacData=NULL时，应能通过pulMacDataLen返回消息鉴别码的长度。

异常情况检测得到预期结果。

8.2.6.34 关闭密码对象句柄

检测目的：

检验是否能正确关闭会话密钥、密码杂凑对象、消息鉴别码对象、ECC密钥协商等句柄。

检测条件：

设备已连接。

检测过程：

——正常情况检测

本检测项作为8.2.6.6、8.2.6.11、8.2.6.13、8.2.6.14、8.2.6.17、8.2.6.18、8.2.6.20、8.2.6.22、8.2.6.24、8.2.6.27、8.2.6.28、8.2.6.31和8.2.6.32的一部分进行检测。

——异常情况检测

使用非法参数调用本接口，应返回错误码。

通过标准：

正常情况检测和异常情况检测均得到预期结果。

8.3 安全性检测

8.3.1 权限分类

检测目的：

检测设备权限、用户权限、管理员权限的权限分类情况。

检测条件：

设备已连接。

检测过程：

a) 设备权限

步骤1. 调用SKF_EnumApplication接口，获得设备中的应用名称列表；

步骤2. 调用SKF_CreateApplication接口，创建与步骤1的列表中名称不同的应用，应不成功；

步骤3. 按照GB/T 35291-2017的8.2.3组织数据，调用SKF_DevAuth接口完成设备认证；

步骤4. 调用SKF_CreateApplication接口，创建与步骤1的列表中名称不同的应用；

步骤5. 调用SKF_EnumApplication接口，获得设备中的应用名称列表，此列表应包括步骤1和步骤4中的应用名称；

步骤6. 拔插设备，重新连接设备；

步骤7. 调用SKF_DeleteApplication接口，删除步骤4中创建的应用，应不成功；

步骤8. 按照GB/T 35291-2017的8.2.3组织数据，调用SKF_DevAuth接口完成

设备认证；

步骤9. 调用 SKF_DeleteApplication 接口，删除步骤 4 中创建的应用；

步骤10. 调用 SKF_EnumApplication 接口，获得设备中的应用名称列表,此列表应不包括步骤 4 中的应用名称；

步骤11. 修改设备认证密钥作为 8.2.2.1 的一部分进行检测。

b) 用户权限

步骤1. 调用 SKF_OpenApplication 接口，打开预定应用，获得应用句柄；

步骤2. 在步骤 1 的应用，使用错误的用户 PIN 重复调用 SKF_VerifyPIN 接口，直到用户 PIN 被锁；

步骤3. 使用步骤 1 的应用，按 8.1.1 进行证书申请检测，应不通过；

步骤4. 调用 SKF_OpenApplication 接口,打开另一个预定应用,获得应用句柄；

步骤5. 使用步骤 4 的应用句柄和步骤 4 中应用的正确管理员 PIN，调用 SKF_UnblockPIN 接口，解锁用户 PIN；

步骤6. 使用步骤 4 的应用句柄和步骤 4 中应用的正确用户 PIN，调用 SKF_VerifyPIN 接口；

步骤7. 使用步骤 1 的应用，按 8.1.1 进行证书申请检测，应不成功；

步骤8. 使用步骤 1 的应用句柄和步骤 1 中应用的正确管理员 PIN，调用 SKF_UnblockPIN 接口，解锁用户 PIN；

步骤9. 使用步骤 1 的应用，按 8.1.1 进行证书申请检测，应成功。

c) 管理员权限

步骤1. 调用 SKF_OpenApplication 接口，打开预定应用，获得应用句柄；

步骤2. 在步骤 1 的应用，使用错误的用户 PIN 重复调用 SKF_VerifyPIN 接口，直到用户 PIN 被锁；

步骤3. 使用步骤 1 的应用句柄，按 8.2.5.1 进行创建容器检测，应不成功；

步骤4. 调用 SKF_OpenApplication 接口,打开另一个预定应用,获得应用句柄；

步骤5. 使用步骤 4 的应用句柄和步骤 4 中应用的正确管理员 PIN，调用 SKF_UnblockPIN 接口，解锁用户 PIN；

步骤6. 使用步骤 1 的应用句柄，按 8.2.5.1 进行创建容器检测，应不成功；

步骤7. 使用步骤 1 的应用句柄和步骤 1 中应用的正确管理员 PIN，调用 SKF_UnblockPIN 接口，解锁用户 PIN；

步骤8. 使用步骤 1 的应用句柄，按 8.2.5.1 进行创建容器检测，应成功。

通过标准：

检测得到预期结果。

8.3.2 权限使用

检测目的：

检测设备权限、用户权限、管理员权限、应用权限、容器权限和文件权限的使用控制情况。

检测条件：

设备已连接。

检测过程：

a) 设备权限

步骤1. 创建应用需使用到设备权限,本检测项作为 8.2.3.1 的一部分进行检测；

步骤2. 删除应用需使用到设备权限,本检测项作为 8.2.3.3 的一部分进行检测；

步骤3. 修改设备认证密钥需使用到设备权限,本检测项作为 8.2.2.1 的一部分进行检测。

b) 用户权限

步骤1. 用户 PIN 码具有最大重试次数，调用 SKF_VerifyPIN、SKF_ChangePIN 校验或修改用户 PIN 码时,用户 PIN 码错误验证次数达到最大重试次数后，用户 PIN 码即被锁死，本检测项作为 8.2.5.1 的一部分进行检测;

步骤2. 调用 SKF_CreateApplication 指定在应用下建立文件、建立容器需要用户权限，在调用 SKF_CreateFile 、SKF_DeleteFile 、SKF_CreateContainer、SKF_DeleteContainer 需使用到用户权限，本检测项作为 8.2.3.1、8.2.4.1、8.2.4.2、8.2.5.1 和 8.2.5.2 的一部分进行检测;

步骤3. 调用 SKF_CreateFile 指定本文件的读取、写入需要用户权限，调用 SKF_ReadFile、SKF_WriteFile 需使用到用户权限，本检测项做为 8.2.4.1、8.2.4.5、8.2.4.6 的一部分进行检测;

步骤4. 调用 SKF_GenRSAKeyPair 接口生成签名密钥对，调用 SKF_RSASignData 接口对预定数据进行签名时时需使用到用户权限，本检测项作为 8.2.6.2 和 8.2.6.4 的一部分进行检测;

步骤5. 调用SKF_ImportRSAKeyPair接口和SKF_ImportECCKeyPair接口导入加密密钥对，调用 SKF_ImportSessionKey 接口导入会话密钥时需使用到用户权限，本检测项作为 8.2.6.3、8.2.6.8 和 8.2.6.17 一部分进行检测;

步骤6. 调用SKF_GenECCKeyPair接口生成签名密钥对 ,调用SKF_ECCSignData 接口返回签名结果时需使用到用户权限，本检测项作为 8.2.6.7 和 8.2.6.9 的一部分进行检测。

c) 管理员权限

步骤1. 管理员 PIN 码具有最大重试次数，调用 SKF_VerifyPIN、SKF_ChangePIN 校验或修改管理员 PIN 码，调用 SKF_UnblockPIN 解锁用户 PIN 码时，管理员 PIN 码验证次数达到最大重试次数后，管理员 PIN 码即被锁死，本检测项作为 8.2.2.3、8.2.2.4 和 8.2.2.6 的一部分进行检测;

步骤2. 修改管理员 PIN 码需使用到管理员权限,本检测项作为 8.2.2.3 的一部分进行检测;

步骤3. 解锁用户 PIN 码需使用到管理员权限,本检测项作为 8.2.2.6 的一部分进行检测;

步骤4. 调用 SKF_CreateApplication 指定在应用下建立文件、建立容器需要管理员权限，调用 SKF_CreateFile、SKF_DeleteFile、SKF_CreateContainer、SKF_DeleteContainer 需使用到管理员权限，本检测项作为 8.2.3.1、8.2.4.1、8.2.4.2、8.2.5.1 和 8.2.5.2 的一部分进行检测;

步骤5. 调用 SKF_CreateFile 指定本文件的读取、写入需要管理员权限，调用 SKF_ReadFile、SKF_WriteFile 需使用到管理员权限，本检测项做为 8.2.4.1、8.2.4.5、8.2.4.6 的一部分进行检测。

d) 应用权限

步骤1. 设置该应用下建立容器/文件的权限为 SECURE_NEVER_ACCOUNT，调用 SKF_CreateApplication 接口建立应用;

步骤2. 调用 SKF_VerifyPIN 接口验证用户权限和管理员权限；
步骤3. 调用 SKF_CreateContainer 接口创建容器，调用 SKF_CreateFile 接口创建文件，应不成功；
步骤4. 设置该应用下建立容器/文件的权限为 SECURE_EVERYONE_ACCOUNT，调用 SKF_CreateApplication 接口建立应用；
步骤5. SKF_ClearSecureState 接口清除所有权限；
步骤6. 调用 SKF_CreateContainer 接口创建、调用 SKF_DeleteContainer 接口删除容器，调用 SKF_CreateFile 接口创建、调用 SKF_DeleteFile 接口删除文件，应成功；
步骤7. 按照 GB/T 35291-2017 的 6.4.12 权限类型，设置该应用下建立容器/文件的权限不为 SECURE_NEVER_ACCOUNT 和 SECURE_EVERYONE_ACCOUNT，调用 SKF_CreateApplication 接口建立应用；
步骤8. 调用 SKF_VerifyPIN 接口验证符合步骤 7 权限类型的权限；
步骤9. 调用 SKF_CreateContainer 接口创建、调用 SKF_DeleteContainer 接口删除容器，调用 SKF_CreateFile 接口创建、调用 SKF_DeleteFile 接口删除文件，应成功；
步骤10. 调用 SKF_ClearSecureState 接口，再执行步骤 9，应不成功；
步骤11. 调用 SKF_VerifyPIN 接口验证不符合步骤 7 权限类型的权限，再执行步骤 9，应不成功。

e) 容器权限

本检测项作为 8.2.6.2、8.2.6.3、8.2.6.4、8.2.6.7、8.2.6.8、8.2.6.9 和 8.2.6.17 一部分进行检测。

f) 文件权限

步骤1. 设置该应用下读和写文件的权限为 SECURE_NEVER_ACCOUNT，调用 SKF_CreateFile 接口建立文件；
步骤2. 调用 SKF_VerifyPIN 接口验证用户权限和管理员权限；
步骤3. 调用 SKF_ReadFile 接口读取文件，SKF_WriteFile 接口写入数据，应不成功；
步骤4. 设置该应用下读和写文件的权限为 SECURE_EVERYONE_ACCOUNT，调用 SKF_CreateFile 接口建立文件；
步骤5. SKF_ClearSecureState 接口清除所有权限；
步骤6. 调用 SKF_ReadFile 接口读取文件，SKF_WriteFile 接口写入数据，应成功；
步骤7. 按照 GB/T 35291-2017 的 6.4.12 权限类型，设置该应用下读和写文件的权限不为 SECURE_NEVER_ACCOUNT 和 SECURE_EVERYONE_ACCOUNT，调用 SKF_CreateFile 接口建立文件；
步骤8. 调用 SKF_VerifyPIN 接口验证符合步骤 7 权限类型的权限；
步骤9. 调用 SKF_ReadFile 接口读取文件，SKF_WriteFile 接口写入数据，应成功；
步骤10. 调用 SKF_ClearSecureState 接口，再执行步骤 9，应不成功；
步骤11. 调用 SKF_VerifyPIN 接口验证不符合步骤 7 权限类型的权限，再执行步骤 9，应不成功。

通过标准：

检测得到预期结果。

8.3.3 设备认证

本检测项作为8.2.2.2的一部分进行检测。

8.3.4 PIN码安全要求

检测目的：

检测PIN码长度合法性是否满足要求。

检测条件：

检测样品存在至少一个用户PIN未锁定、创建文件需要验证用户PIN的应用，应用名称和用户PIN已知；

设备已连接。

检测过程：

——PIN码长度不小于6个字节。

调用SKF_ChangePIN接口，输入小于6位的新PIN码，进行修改密码操作，应返回“密码长度错误”。

——PIN码在设备和本接口之间的传输过程中应采取保护措施，防止PIN码泄露。

需要提供安全设计说明。

——PIN码在设备中应安全存储，不可从设备中导出。

需要提供安全设计说明。

通过标准：

检测得到预期结果。

8.3.5 密钥安全要求

对智能密码钥匙密钥的安全性检测和评估，应遵循GM/T 0027。

8.3.6 随机数安全要求

调用SKF_GenRandom接口获取随机数，应遵循GB/T 32915。

8.4 兼容性检测

8.4.1 系统兼容性

检测目的：

检测在不同系统下接口库是否能正常使用。

检测条件：

无。

检测过程：

步骤1. 在主流操作系统下调用接口，接口应能正常调用，系统包括但不限于Windows操作系统；

步骤2. 在以上系统对应的32位与64位平台下调用接口，接口应能正常调用。

通过标准：

检测得到预期结果。

8.4.2 交错兼容性

检测目的：

检测待测接口库接口与其他接口的兼容性。

检测条件：

具备已配置其他接口的环境，且其他接口已通过检测。

检测过程：

——正常情况检测

在上述检测条件下，接口可以正常使用，互不影响。

——异常情况检测

将其他接口配套的密码钥匙插入电脑，调用待测接口库接口，接口应返回找不到设备。

通过标准：

正常情况检测和异常情况检测均得到预期结果。

8.5 互操作性检测

检测目的：

检测待测接口库的互操作性。

检测条件：

需具备经互操作认可的接口库，待测接口库与认可接口库均存在 SM2 签名密钥对和加密密钥对。

检测过程：

a) 签名验签互操作

步骤1. 使用预定数据调用待测接口库签名接口 SKF_ECCSignData 与导出公钥接口 SKF_ExportPublicKey；

步骤2. 使用预定数据与步骤 1 的结果调用认可接口库验签接口 SKF_ECCVerify，测试验签是否通过；

步骤3. 使用预定数据调用认可接口库签名接口 SKF_ECCSignData 与导出公钥接口 SKF_ExportPublicKey；

步骤4. 使用预定数据与步骤 3 的结果调用待测接口库验签接口 SKF_ECCVerify，测试验签是否通过。

b) 密钥协商互操作

步骤1. 调用待测接口库 SKF_GenerateAgreementDataWithECC 接口；

步骤2. 调用认可接口库 SKF_GenerateAgreementDataAndKeyWithECC 接口，得到会话密钥句柄 A；

步骤3. 调用待测接口库 SKF_GenerateKeyWithECC 接口，得到会话密钥句柄 B；

步骤4. 使用会话密钥句柄 A 和 B 分别对相同的预定数据做加密操作，均应得到与预定密文相同的密文数据；

步骤5. 调用认可接口库 SKF_GenerateAgreementDataWithECC 接口；

步骤6. 调用待测接口库 SKF_GenerateAgreementDataAndKeyWithECC 接口，得到会话密钥句柄 C；

步骤7. 调用认可接口库 SKF_GenerateKeyWithECC 接口，得到会话密钥句柄 D；

步骤8. 使用会话密钥句柄 C 和 D 分别对相同的预定数据做加密操作，均应得到与预定密文相同的密文数据。

c) 数字信封互操作

步骤1. 使用待测接口库接口，执行 8.1.6 的过程生成数字信封；

步骤2. 使用认可接口库，执行 8.1.7 的过程进行数字信封解封，数字信封应该解封成功；

步骤3. 使用认可接口库接口，执行 8.1.6 的过程生成数字信封；

步骤4. 使用待测接口库，执行 8.1.7 的过程进行数字信封解封，数字信封应该解封成功。

通过标准：

检测得到预期结果。