ICS 35.040
CCS L 80

GM

中华人民共和国密码行业标准

GM/T 0102—2020

密码设备应用接口符合性检测规范

Cryptographic device application interface test specification

2020-12-28 发布　　2021-07-01 实施

国家密码管理局　　发布

目　　次

前　言

本文件按照GB/T 1.1—2020《标准化工作导则　第1部分：标准化文件的结构和起草规则》的规定起草。

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专利的责任。

本文件由密码行业标准化技术委员会提出并归口。

本文件起草单位：卫士通信息产业股份有限公司、四川大学、深圳文鼎创数据科技有限公司、山东大学、格尔软件股份有限公司、国家密码管理局商用密码检测中心、山东得安信息技术有限公司。

本文件主要起草人：罗俊、龚勋、胡显荃、刘伟丰、孔凡玉、郑强、罗鹏、马洪富。

密码设备应用接口符合性检测规范

1 范围

本文件规定了GB/T 36322—2018的符合性检测要求和检测方法。

本文件适用于按照GB/T 36322—2018实现的密码设备应用接口的检测，也可用于指导基于该接口规范的密码设备、模块、固件和软件产品的研制和应用开发。

2 规范性引用文件

下列文件中的内容通过文中的规范性引用而构成本文件必不可少的条款。其中，注日期的引用文件，仅该日期对应的版本适用于本文件；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 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 32918.5—2017 信息安全技术 SM2椭圆曲线公钥密码算法 第5部分：参数定义

GB/T 33560 信息安全技术 密码应用标识规范

GB/T 35276 信息安全技术 SM2密码算法使用规范

GB/T 36322—2018 信息安全技术 密码设备应用接口规范

GB/T 36968 信息安全技术 IPSec VPN技术规范

GM/T 0024 SSL VPN技术规范

GM/Z 4001—2013 密码术语

3 术语和定义

GM/Z 4001—2013和GB/T 36322—2018界定的术语和定义适用于本文件。

4 缩略语

下列缩略语适用于本文件。

API 应用程序接口(Application Program Interface)

ECB (分组密码的)电子密本(工作方式)(Electronic Codebook)

ECC 椭圆曲线算法(Elliptic Curve Cryptography)

EPK 外部加密公钥(External Public Key)

IPK 内部加密公钥(Internal Public Key)

ISK 内部加密私钥(Internal PrⅣate Key)

KEK 密钥加密密钥(Key Encrypt Key)

注：本文件中的ECC专指SM2算法。

5 检测环境要求

5.1 网络部署拓扑

检测平台由检测控制台和运行测试程序的服务器组成，密码设备应用接口以 API 库文件的形式在服务器上进行安装。检测控制台向服务器上运行的测试程序发送测试指令，测试程序根据测试指令调用相应的 API 接口。当通过网络方式连接密码设备时，参考的网络拓扑结构如图 1 所示。

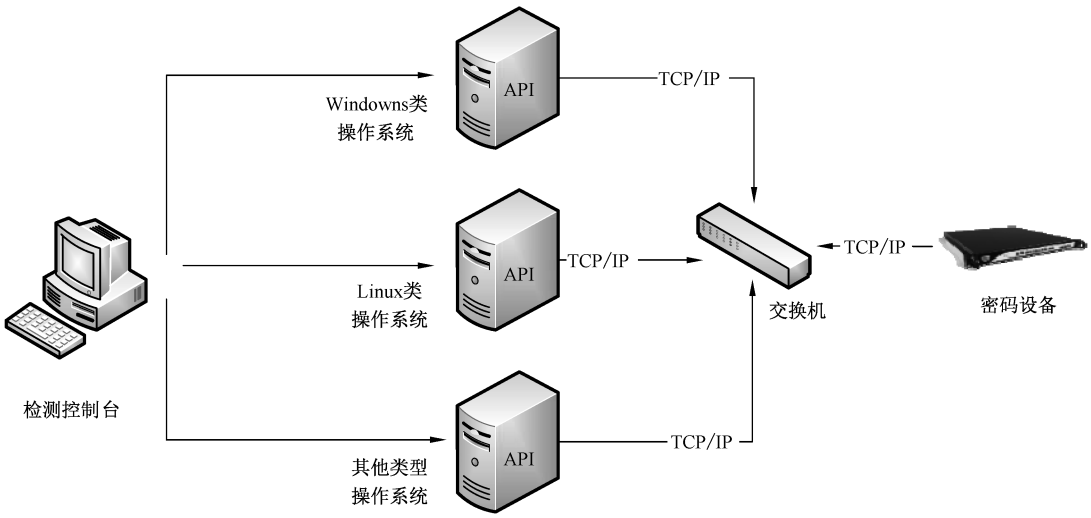

图 1 检测环境网络拓扑图

5.2 API 提供方式

密码设备应用接口应以 API 库文件的形式提供给检测平台，并在检测平台的服务器上进行安装。当以动态链接库形式提供时，应提供至少一种操作系统环境下的动态库文件，操作系统类型和版本以检测平台为准。检测平台的服务器在所选定的操作系统环境下运行测试程序，测试程序启动后加载密码设备应用接口的动态库。

当密码设备应用接口以静态链接库形式提供时，应提供至少一种操作系统环境下的静态库文件和头文件，操作系统类型和版本以检测平台为准。检测平台的服务器在所选定的操作系统环境下运行编译程序，加载密码设备应用接口的静态库，形成可执行的测试程序。

API 应符合 GB/T 36322—2018 定义的函数接口和参数格式。API 库文件内部和具体密码设备、密码模块、密码固件或软件产品的接口方式、数据格式、传输协议等由各厂商自己实现，不做统一规定。

5.3 关于检测环境的说明

送检方可不限于以上 API 提供形式和网络拓扑，但需详细说明检测环境的构建方式并提供可行的检测手段。

6 测试内容

6.1 测试项目及说明

本标准相关检测内容依据 GB/T 36322—2018 中第 6 章，密码设备应用接口测试的主要内容包括：

a) API 初始化测试；
b) 设备管理类接口测试；
c) 密钥管理类接口测试；
d) 非对称算法运算类接口测试；
e) 对称算法运算类接口测试；
f) 杂凑运算类接口测试；
g) 用户文件操作类接口测试；
h) 接口稳定性测试；
i) 边界和异常条件测试；
j) 接口安全性测试；
k) 接口库卸载测试。

本标准中涉及的函数原型(包括函数名称、接口和参数格式等)以及相关数据结构和宏定义应遵循 GB/T 36322—2018。本标准中除 6.10 以外所有测试项的输入参数均为有效参数，RSA 相关函数的测试均要求模长至少为 2 048 位。

6.2 API 初始化测试

当厂商以动态链接库形式提供 API 时，应提供至少一种操作系统环境下的动态库文件，操作系统类型和版本以检测平台为准。检测平台在所选定的操作系统环境下运行测试程序。测试程序应遵循 GB/T 36322—2018 定义的函数原型(包括函数名称、接口和参数格式等)定义设备管理、密钥管理、非对称算法运算、对称算法运算、杂凑运算、用户文件操作六大类函数指针，测试程序启动后根据厂商提供的文件名和路径加载密码设备应用接口的动态库，并依次按照函数名称获取六大类函数指针，加载和获取操作均成功则初始化完成。

当厂商以静态链接库形式提供 API 时，应提供至少一种操作系统环境下的静态库文件和头文件，头文件应遵循 GB/T 36322—2018 定义的函数原型(包括函数名称、接口和参数格式等)定义设备管理、密钥管理、非对称算法运算、对称算法运算、杂凑运算、用户文件操作六大类函数。操作系统类型和版本以检测平台为准。检测平台在所选定的操作系统环境下运行编译程序，将测试程序的目标代码和静态库文件一起编译生成可执行程序。测试程序启动成功则初始化完成。

厂商至少应以配置文件、命令行参数、管理界面等其中之一的形式提供客户端 API 访问密码设备的 IP 地址和传输层端口的设置手段。

6.3 设备管理类接口测试

6.3.1 测试项目

需测试的设备管理类函数接口如表 1 所示。

表 1 设备管理类函数

函数名称	功能
SDF_OpenDevice	打开设备
SDF_CloseDevice	关闭设备
SDF_OpenSession	创建会话
SDF_CloseSession	关闭会话
SDF_GetDeviceInfo	获取设备信息

表 1（续）

函数名称	功能
SDF_GenerateRandom	产生随机数
SDF_GetPrⅣateKeyAccessRight	获取私钥使用权限
SDF_ReleasePrⅣateKeyAccessRigh	释放私钥使用权限

6.3.2 打开设备

原型： LONG SDF_OpenDevice(HANDLE * phDeviceHandle)

描述： 打开密码设备。

参数： phDeviceHandle[in] 返回的设备句柄

测试步骤： a） 创建(定义)设备句柄并初始化为零值；

b） 采用传地址的方式将设备句柄作为参数调用设备打开函数；

c） 函数返回值为零而且设备句柄值不为零，则函数调用成功。

注：本测试应和 6.3.3 结合进行。

6.3.3 关闭设备

原型： LONG SDF_CloseDevice(HANDLE hDeviceHandle)

描述： 关闭密码设备，并释放相关资源。

参数： hDeviceHandle[in] 已打开的设备句柄

测试步骤： a） 确认 6.3.2 中已打开的设备句柄值不为零；

b） 将已打开并确认的设备句柄作为参数调用设备关闭函数；

c） 函数返回值为零而且设备句柄值变为零，则函数调用成功。

注：本测试应在 6.3.2 的基础上进行。

6.3.4 创建会话

原型： LONG SDF_OpenSession(HANDLE hDeviceHandle, HANDLE * phSessionHandle)

描述： 创建与密码设备的会话。

参数： hDeviceHandle[in] 已打开的设备句柄

phSessionHandle[out] 返回与密码设备建立的新会话句柄

测试步骤： a） 创建(定义)会话句柄和设备句柄并初始化为零值；

b） 调用打开设备函数，获取设备句柄；

c） 将会话句柄的地址和设备句柄作为参数调用会话创建函数；

d） 函数返回值为零而且会话句柄值不为零，则函数调用成功。

注：本测试应先调用 6.3.2 后再测试，并和 6.3.5 结合进行。

6.3.5 关闭会话

原型： LONG SDF_CloseSession(HANDLE hSessionHandle)

描述： 关闭与密码设备已建立的会话，并释放相关资源。

参数： hSessionHandle [in] 与密码设备已建立的会话句柄

测试步骤： a） 确认 6.3.4 中已打开的设备句柄值和会话句柄值不为零；

b） 将已打开并确认的会话句柄作为参数调用会话关闭函数；

c) 确认函数返回值为零而且会话句柄值变为零，否则函数调用失败；

d) 调用关闭设备函数，确认函数返回值为零而且设备句柄值变为零，则函数调用成功。

注：本测试应在 6.3.4 的基础上进行。

6.3.6 获取设备信息

原型：
```
LONG SDF_GetDeviceInfo (
        HANDLE hSessionHandle,
        DEVICEINFO * pstDeviceInfo)
```

描述：获取密码设备能力描述。

参数：hSessionHandle[in] 与设备建立的会话句柄

pstDeviceInfo[out] 设备能力描述信息，内容及格式见设备信息定义

测试步骤：

a) 创建(定义)会话句柄和设备句柄并初始化为零值；

b) 创建(定义)设备能力描述信息的数据结构并初始化结构内各值为零；

c) 调用打开设备函数，获取设备句柄；

d) 调用创建会话函数，获取会话句柄；

e) 以会话句柄和设备能力描述信息结构的地址为参数调用获取设备信息函数；

f) 函数返回值为零，则以可见字符形式打印设备能力描述信息，包括设备生产厂商名称、设备型号、设备编号(日期-批次号-流水号)、设备软件版本号、接口规范版本号、非对称算法标识、对称算法标识、杂凑算法标识、最大文件存储空间等，并与厂商提供信息进行对照；

g) 调用关闭会话函数，确认函数返回值为零而且会话句柄值变为零；

h) 调用关闭设备函数，确认函数返回值为零而且设备句柄值变为零，则调用成功。

6.3.7 产生随机数

原型：
```
LONG SDF_GenerateRandom (
        HANDLE hSessionHandle,
        ULONG uiLength,
        BYTE * pucRandom)
```

描述：获取指定长度的随机数。

参数：hSessionHandle[in] 与设备建立的会话句柄

uiLength[in] 欲获取的随机数长度

pucRandom[out]缓冲区指针，用于存放获取的随机数

测试步骤：

a) 创建(定义)会话句柄和设备句柄并初始化为零值；

b) 创建(定义)不同大小的的数据缓冲区(32、64、128、256、512、1 024、2 048 字节)并用全零填充；

c) 以读写方式创建 1 000 个二进制流文件；

d) 调用打开设备函数，获取设备句柄；

e) 调用创建会话函数，获取会话句柄；

f) 分别以不同长度的数据缓冲区地址为参数调用产生随机数函数，每次调用返回值应为 0；

g) 以 1 024 字节长度数据缓冲区地址为参数循环调用产生随机数函数 128 次，每次调用成功后将数据缓冲区的数据写入同一二进制文件并对缓冲区填零，最终生成 128 K 字节大小的随机数文件；

h) 将步骤 g 循环执行 1 000 次，每次采用不同的二进制文件，最终生成 1 000 个 128 K字节大小的随机数文件；

i) 调用关闭会话函数，确认函数返回值为零而且会话句柄值变为零；

j) 调用关闭设备函数，确认函数返回值为零而且设备句柄值变为零；

k) 将步骤 h)生成的 1 000 个 128 K 字节大小的随机数文件导入到随机性检测平台进行随机性检测，通过检测则产生随机数函数调用成功。

6.3.8 获取私钥使用权限

原型： LONG SDF_GetPrivateKeyAccessRight (
HANDLE hSessionHandle,
ULONG uiKeyIndex,
LPSTR pucPassword,
ULONG uiPwdLength)

描述： 获取密码设备内部存储的指定索引私钥的使用权。

参数： hSessionHandle[in] 与设备建立的会话句柄
uiKeyIndex[in] 密码设备存储私钥的索引值
pucPassword[in] 使用私钥权限的标识码
uiPwdLength[in] 私钥访问控制码长度，不少于 8 字节

测试步骤：

a) 创建(定义)会话句柄和设备句柄并初始化为零值；

b) 调用打开设备函数，获取设备句柄；

c) 调用创建会话函数，获取会话句柄；

d) 用厂家提供的标识码和从 1 到 n(厂家提供最大索引值)的索引值依次调用获取私钥使用权限函数，返回值均应为零；

e) 用非厂家提供的标识码调用获取私钥使用权限函数，返回值均应不为零；

f) 用厂家提供的标识码和 1 到 n 之外的任意索引值调用获取私钥使用权限函数，返回值均应不为零。

注：本测试应在密码设备提供商通过管理工具初始化内部密钥的基础上进行，并和 6.3.9 结合进行。

6.3.9 释放私钥使用权限

原型： LONG SDF_ReleasePrivateKeyAccessRight (
HANDLE hSessionHandle,
ULONG uiKeyIndex)

描述： 释放密码设备存储的指定索引私钥的使用授权。

参数： hSessionHandle[in] 与设备建立的会话句柄
uiKeyIndex[in] 密码设备存储私钥索引值

测试步骤：

a) 使用 6.3.8 中获取的会话句柄，分别用从 1 到 n (厂家提供最大索引值)索引值依次调用释放私钥使用权限函数，返回值均应为零。

b) 调用关闭会话函数，确认函数返回值为零而且会话句柄值变为零；

c) 调用关闭设备函数,确认函数返回值为零而且设备句柄值变为零。

注:本测试应在 6.3.8 的基础上进行。

6.4 密钥管理类接口测试

6.4.1 测试项目

需测试的密钥管理类函数接口如表 2 所示。

表 2 密钥管理类函数

函数名称	功能
SDF_ExportSignPublicKey_RSA	导出 RSA 签名公钥
SDF_ExportEncPublicKey_RSA	导出 RSA 加密公钥
SDF_GenerateKeyPair_RSA	产生 RSA 非对称密钥对并输出
SDF_GenerateKeyWithIPK_RSA	生成会话密钥并用内部 RSA 公钥加密输出
SDF_GenerateKeyWithEPK_RSA	生成会话密钥并用外部 RSA 公钥加密输出
SDF_ImportKeyWithISK_RSA	导入会话密钥并用内部 RSA 私钥解密
SDF_ExchangeDigitEnvelopeBaseOnRSA	基于 RSA 算法的数字信封转换
SDF_ExportSignPublicKey_ECC	导出 ECC 签名公钥
SDF_ExportEncPublicKey_ECC	导出 ECC 加密公钥
SDF_GenerateKeyPair_ECC	产生 ECC 非对称密钥对并输出
SDF_GenerateKeyWithIPK_ECC	生成会话密钥并用内部 ECC 公钥加密输出
SDF_GenerateKeyWithEPK_ECC	生成会话密钥并用外部 ECC 公钥加密输出
SDF_ImportKeyWithISK_ECC	导入会话密钥并用内部 ECC 私钥解密
SDF_GenerateAgreementDataWithECC	生成密钥协商参数并输出
SDF_GenerateKeyWithECC	计算会话密钥
SDF_GenerateAgreementDataAndKeyWithECC	产生协商数据并计算会话密钥
SDF_ExchangeDigitEnvelopeBaseOnECC	基于 ECC 算法的数字信封转换
SDF_GenerateKeyWithKEK	生成会话密钥并用密钥加密密钥加密输出
SDF_ImportKeyWithKEK	导入会话密钥并用密钥加密密钥解密
SDF_GenerateKeywithIKE	计算 IKE 工作密钥
SDF_GenerateKeywithEPK_IKE	计算 IKE 工作密钥并用外部 ECC 公钥加密输出
SDF_GenerateKeywithIPSEC	计算 IPSEC 会话密钥
SDF_GenerateKeywithEPK_IPSEC	计算 IPSEC 会话密钥并用外部 ECC 公钥加密输出
SDF_GenerateKeywithSSL	计算 SSL 工作密钥
SDF_GenerateKeywithEPK_SSL	计算 SSL 工作密钥并用外部 ECC 公钥加密输出
SDF_GenerateKeywithECDHE_SSL	计算 SSL 工作密钥(ECDHE)
SDF_GenerateKeywithEPK_ECDHE_SSL	计算 SSL 工作密钥并用外部 ECC 公钥加密输出(ECDHE)
SDF_DestroyKey	销毁会话密钥

6.4.2 导出 RSA 签名公钥

原型：　LONG SDF_ExportSignPublicKey_RSA(
　　HANDLE hSessionHandle,
　　ULONG uiKeyIndex,
　　RSArefPublicKey * pucPublicKey)

描述：　导出密码设备内部存储的指定索引位置的签名公钥。

参数：

	hSessionHandle[in]	与设备建立的会话句柄
	uiKeyIndex[in]	密码设备存储的 RSA 密钥对索引值
	pucPublicKey[out]	RSA 公钥结构

返回值：

0	成功
非 0	失败，返回错误代码

测试步骤：

a) 使用被测试密码设备的管理工具产生 1 到 n（密钥对索引值，n 为厂家提供最大索引值）对 2 048 位模长 RSA 签名密钥对并在设备内部存储；
b) 创建（定义）会话句柄和设备句柄并初始化为零值；
c) 调用打开设备函数，获取设备句柄；
d) 调用创建会话函数，获取会话句柄；
e) 创建公钥输出缓冲区（大小为 RSArefPublicKey 结构体）；
f) 分别用从 1 到 n（厂家提供最大索引值）索引值依次调用本函数，调用返回值应为 0；
g) 调用关闭会话函数，确认函数返回值为零而且会话句柄值变为零；
h) 调用关闭设备函数，确认函数返回值为零而且设备句柄值变为零。

注：本测试应在密码设备提供商通过管理工具初始化内部密钥的基础上进行。

6.4.3 导出 RSA 加密公钥

原型：　LONG SDF_ExportEncPublicKey_RSA(
　　HANDLE hSessionHandle,
　　ULONG uiKeyIndex,
　　RSArefPublicKey * pucPublicKey)

描述：　导出密码设备内部存储的指定索引位置的加密公钥。

参数：

	hSessionHandle[in]	与设备建立的会话句柄
	uiKeyIndex[in]	密码设备存储的 RSA 密钥对索引值
	pucPublicKey[out]	RSA 公钥结构

返回值：

0	成功
非 0	失败，返回错误代码

测试步骤：

a) 使用被测试密码设备的管理工具产生 1 到 n（密钥对索引值，n 为厂家提供最大索引值）对 2 048 位模长 RSA 加密密钥对并在设备内部存储；
b) 创建（定义）会话句柄和设备句柄并初始化为零值；
c) 调用打开设备函数，获取设备句柄；
d) 调用创建会话函数，获取会话句柄；
e) 创建公钥输出缓冲区（大小为 RSArefPublicKey 结构体）；
f) 分别用从 1 到 n（厂家提供最大索引值）索引值依次调用本函数，调用返回值应为 0；

g) 调用关闭会话函数，确认函数返回值为零而且会话句柄值变为零；

h) 调用关闭设备函数，确认函数返回值为零而且设备句柄值变为零。

注：本测试应在密码设备提供商通过管理工具初始化内部密钥的基础上进行。

6.4.4 产生 RSA 密钥对并输出

原型：LONG SDF_GenerateKeyPair_RSA(
HANDLE hSessionHandle,
ULONG uiKeyBits,
RSArefPublicKey * pucPublicKey,
RSArefPrivateKey * pucPrivateKey)

描述：请求密码设备产生指定模长的 RSA 密钥对。

参数：

hSessionHandle[in]	与设备建立的会话句柄
uiKeyBits[in]	指定密钥模长
pucPublicKey[out]	RSA 公钥结构
pucPrivateKey[out]	RSA 私钥结构

返回值：

0	成功
非 0	失败，返回错误代码

测试步骤：

a) 创建(定义)会话句柄和设备句柄并初始化为零值；

b) 调用打开设备函数，获取设备句柄；

c) 调用创建会话函数，获取会话句柄；

d) 输入不同的 RSA 密钥模长(模长不小于 2 048 位)；

e) 创建公钥输出缓冲区(大小为 RSArefPublicKey 结构体)；

f) 创建私钥输出缓冲区(大小为 RSArefPrivateKey 结构体)；

g) 以缓冲区地址为参数调用本函数，调用返回值应为 0；

h) 调用关闭会话函数，确认函数返回值为零而且会话句柄值变为零；

i) 调用关闭设备函数，确认函数返回值为零而且设备句柄值变为零。

注：本函数输出的 2 048 模长 RSA 密钥对保存至 6.4.6 和 6.4.8 使用。

6.4.5 生成会话密钥并用内部 RSA 公钥加密输出

原型：LONG SDF_GenerateKeyWithIPK_RSA (
HANDLE hSessionHandle,
ULONG uiIPKIndex,
ULONG uiKeyBits,
BYTE * pucKey,
ULONG * puiKeyLength,
HANDLE * phKeyHandle)

描述：生成会话密钥并用指定索引的内部加密公钥加密输出，同时返回密钥句柄。

参数：

hSessionHandle[in]	与设备建立的会话句柄
uiIPKIndex[in]	密码设备内部存储公钥的索引值
uiKeyBits[in]	指定产生的会话密钥长度
pucKey[out]	缓冲区指针，用于存放返回的密钥密文
puiKeyLength[out]	返回的密钥密文长度
phKeyHandle[out]	返回的密钥句柄

返回值：　0　　成功

非 0　　失败，返回错误代码

测试步骤：
a) 创建(定义)会话句柄和设备句柄并初始化为零值；
b) 调用打开设备函数，获取设备句柄；
c) 调用创建会话函数，获取会话句柄；
d) 分别用 1 到 n(厂家提供最大索引值)索引值并创建相应密钥输出缓冲区、密钥长度变量和密钥句柄指针，依次调用本函数，调用返回值应为 0；
e) 调用关闭会话函数，确认函数返回值为零而且会话句柄值变为零；
f) 调用关闭设备函数，确认函数返回值为零而且设备句柄值变为零。

注：公钥加密数据时填充方式与 PKCS#1 v1.5 相同。本函数输出的 n 段密钥密文(分别对应公钥的索引值 1 到 n)保存至 6.4.7 和 6.4.8 使用。

6.4.6 生成会话密钥并用外部 RSA 公钥加密输出

原型：
```
LONG SDF_GenerateKeyWithEPK_RSA (
        HANDLE hSessionHandle,
        ULONG uiKeyBits,
        RSArefPublicKey * pucPublicKey,
        BYTE * pucKey,
        ULONG * puiKeyLength,
        HANDLE * phKeyHandle)
```

描述：生成会话密钥并用外部公钥加密输出，同时返回密钥句柄。

参数：

hSessionHandle[in]	与设备建立的会话句柄
uiKeyBits[in]	指定产生的会话密钥长度
pucPublicKey[in]	输入的外部 RSA 公钥结构
pucKey[out]	缓冲区指针，用于存放返回的密钥密文
puiKeyLength[out]	返回的密钥密文长度
phKeyHandle[out]	返回的密钥句柄

返回值：　0　　成功

非 0　　失败，返回错误代码

测试步骤：
a) 创建(定义)会话句柄和设备句柄并初始化为零值；
b) 调用打开设备函数，获取设备句柄；
c) 调用创建会话函数，获取会话句柄；
d) 创建密钥输出缓冲区；
e) 创建密钥长度变量；
f) 创建密钥句柄指针；
g) 任取一对 6.4.4 中产生的 2 048 位模长 RSA 密钥对公钥结构为参数调用本函数，调用返回值应为 0；
h) 调用关闭会话函数，确认函数返回值为零而且会话句柄值变为零；
i) 调用关闭设备函数，确认函数返回值为零而且设备句柄值变为零。

注：公钥加密数据时填充方式与 PKCS#1 v1.5 相同。

6.4.7 导入会话密钥并用内部 RSA 私钥解密

原型：
```
LONG SDF_ImportKeyWithISK_RSA (
```

HANDLE hSessionHandle,
ULONG uiISKIndex,
BYTE * pucKey,
ULONG puiKeyLength,
HANDLE * phKeyHandle)

描述：　导入会话密钥并用内部私钥解密，同时返回密钥句柄。

参数：

hSessionHandle[in]	与设备建立的会话句柄
uiISKIndex[in]	密码设备内部存储加密私钥的索引值，对应于加密时的公钥
pucKey[in]	缓冲区指针，用于存放输入的密钥密文
puiKeyLength[in]	输入的密钥密文长度
phKeyHandle[out]	返回的密钥句柄

返回值：

0	成功
非 0	失败，返回错误代码

测试步骤：

a) 创建(定义)会话句柄和设备句柄并初始化为零值；
b) 调用打开设备函数，获取设备句柄；
c) 调用创建会话函数，获取会话句柄；
d) 分别用 1 到 n(厂家提供最大索引值)索引值并创建相应密钥句柄指针，相对应地将 6.4.5 中输出的 1 到 n 段密钥密文、密钥密文长度为参数依次调用本函数，调用返回值应为 0；
e) 调用关闭会话函数，确认函数返回值为零而且会话句柄值变为零；
f) 调用关闭设备函数，确认函数返回值为零而且设备句柄值变为零。

注：填充方式与公钥加密时相同。

6.4.8 基于 RSA 算法的数字信封转换

原型：　LONG SDF_ExchangeDigitEnvelopeBaseOnRSA(
HANDLE hSessionHandle,
ULONG uiKeyIndex,
RSArefPublicKey * pucPublicKey,
BYTE * pucDEInput,
ULONG uiDELength,
BYTE * pucDEOutput,
ULONG * puiDELength)

描述：　将由内部加密公钥加密的会话密钥转换为由外部指定的公钥加密，可用于数字信封转换。

参数：

hSessionHandle[in]	与设备建立的会话句柄
uiKeyIndex[in]	密码设备存储的内部 RSA 密钥对索引值
pucPublicKey [in]	外部 RSA 公钥结构
pucDEInput [in]	缓冲区指针，用于存放输入的会话密钥密文
uiDELength[in]	输入的会话密钥密文长度
pucDEOutput[out]	缓冲区指针，用于存放输出的会话密钥密文
puiDELength[out]	输出的会话密钥密文长度

返回值：

0	成功
非 0	失败，返回错误代码

测试步骤：

a) 创建(定义)会话句柄和设备句柄并初始化为零值；

b) 调用打开设备函数，获取设备句柄；
c) 调用创建会话函数，获取会话句柄；
d) 分别用 1 到 n（厂家提供最大索引值）索引值并创建相应输出密钥密文缓冲区和输出密钥密文长度变量，相对应地将 6.4.5 中输出的 1 到 n 段密钥密文及密钥长度，以及任取一对 6.4.4 中产生的 2 048 位模长 RSA 密钥对为参数，依次调用本函数，调用返回值应为 0；
e) 调用关闭会话函数，确认函数返回值为零而且会话句柄值变为零；
f) 调用关闭设备函数，确认函数返回值为零而且设备句柄值变为零。

6.4.9 导出 ECC 签名公钥

原型：LONG SDF_ExportSignPublicKey_ECC(
HANDLE hSessionHandle,
ULONG uiKeyIndex,
ECCrefPublicKey * pucPublicKey)

描述：导出密码设备内部存储的指定索引位置的签名公钥。

参数：

	hSessionHandle[in]	与设备建立的会话句柄
	uiKeyIndex[in]	密码设备存储的 ECC 密钥对索引值
	pucPublicKey[out]	ECC 公钥结构

返回值：

0	成功
非 0	失败，返回错误代码

测试步骤：
a) 使用被测试密码设备的管理工具产生 1 到 n（密钥对索引值，n 为厂家提供最大索引值）对 ECC 签名密钥对并在设备内部存储；
b) 创建（定义）会话句柄和设备句柄并初始化为零值；
c) 调用打开设备函数，获取设备句柄；
d) 调用创建会话函数，获取会话句柄；
e) 分别用 1 到 n（厂家提供最大索引值）索引值并创建相应导出公钥缓冲区（大小为 ECCrefPublicKey 结构体），依次调用本函数，调用返回值应为 0；
f) 调用关闭会话函数，确认函数返回值为零而且会话句柄值变为零；
g) 调用关闭设备函数，确认函数返回值为零而且设备句柄值变为零。

注：本测试应在密码设备提供商通过管理工具初始化内部密钥的基础上进行。

6.4.10 导出 ECC 加密公钥

原型：LONG SDF_ExportEncPublicKey_ECC(
HANDLE hSessionHandle,
ULONG uiKeyIndex,
ECCrefPublicKey * pucPublicKey)

描述：导出密码设备内部存储的指定索引位置的加密公钥。

参数：

	hSessionHandle[in]	与设备建立的会话句柄
	uiKeyIndex[in]	密码设备存储的 ECC 密钥对索引值
	pucPublicKey[out]	ECC 公钥结构

返回值：

0	成功
非 0	失败，返回错误代码

测试步骤：
a) 使用被测试密码设备的管理工具产生 1 到 n（密钥对索引值，n 为厂家提供最大

索引值)对 ECC 加密密钥对并在设备内部存储;

b) 创建(定义)会话句柄和设备句柄并初始化为零值;

c) 调用打开设备函数,获取设备句柄;

d) 调用创建会话函数,获取会话句柄;

e) 分别用 1 到 n(厂家提供最大索引值)索引值并创建导出公钥缓冲区(大小为 ECCrefPublicKey 结构体),依次调用本函数,调用返回值应为 0;

f) 调用关闭会话函数,确认函数返回值为零而且会话句柄值变为零;

g) 调用关闭设备函数,确认函数返回值为零而且设备句柄值变为零。

注:本测试应在密码设备提供商通过管理工具初始化内部密钥的基础上进行。

6.4.11 产生 ECC 密钥对并输出

原型:
```
LONG SDF_GenerateKeyPair_ECC(
        HANDLE hSessionHandle,
        ULONG uiAlgID,
        ULONG uiKeyBits,
        ECCrefPublicKey * pucPublicKey,
        ECCrefPrIVateKey * pucPrIVateKey)
```

描述:请求密码设备产生指定类型和模长的 ECC 密钥对。

参数:

hSessionHandle[in]	与设备建立的会话句柄
uiAlgID[in]	指定算法标识
uiKeyBits [in]	指定密钥长度
pucPublicKey[out]	ECC 公钥结构
pucPrIVateKey[out]	ECC 私钥结构

返回值:

0	成功
非 0	失败,返回错误代码

测试步骤:

a) 创建(定义)会话句柄和设备句柄并初始化为零值;

b) 调用打开设备函数,获取设备句柄;

c) 调用创建会话函数,获取会话句柄;

d) 指定算法 ID 为 SGD_SM2;

e) 指定密钥模长;

f) 创建公钥输出缓冲区(大小为 ECCrefPublicKey 结构体);

g) 创建私钥输出缓冲区(大小为 ECCrefPrIVateKey 结构体);

h) 将缓冲区地址为参数调用本函数,调用返回值应为 0;

i) 调用关闭会话函数,确认函数返回值为零而且会话句柄值变为零;

j) 调用关闭设备函数,确认函数返回值为零而且设备句柄值变为零。

注:本函数输出的 ECC 密钥对保存至 6.4.13 和 6.4.18 使用。

6.4.12 生成会话密钥并用内部 ECC 公钥加密输出

原型:
```
LONG SDF_GenerateKeyWithIPK_ECC (
        HANDLE hSessionHandle,
        ULONG uiIPKIndex,
        ULONG uiKeyBits,
        ECCCipher * pucKey,
```

HANDLE * phKeyHandle)

描述：　生成会话密钥并用指定索引的内部 ECC 加密公钥加密输出，同时返回密钥句柄。

参数：

	hSessionHandle[in]	与设备建立的会话句柄
	uiIPKIndex[in]	密码设备内部存储公钥的索引值
	uiKeyBits[in]	指定产生的会话密钥长度
	pucKey[out]	缓冲区指针，用于存放返回的密钥密文
	phKeyHandle[out]	返回的密钥句柄
返回值：	0	成功
	非 0	失败，返回错误代码

测试步骤：

a) 创建(定义)会话句柄和设备句柄并初始化为零值；
b) 调用打开设备函数，获取设备句柄；
c) 调用创建会话函数，获取会话句柄；
d) 指定会话密钥长度；
e) 分别用 1 到 n(厂家提供最大索引值)索引值并创建相应密钥输出缓冲区和密钥句柄指针，依次调用本函数，调用返回值应为 0；
f) 调用关闭会话函数，确认函数返回值为零而且会话句柄值变为零；
g) 调用关闭设备函数，确认函数返回值为零而且设备句柄值变为零。

注：本函数输出的 n(分别对应公钥的索引值 1 到 n)个 ECC 密钥密文结构保存至 6.4.14 和 6.4.18 使用。

6.4.13 生成会话密钥并用外部 ECC 公钥加密输出

原型：

```
LONG SDF_GenerateKeyWithEPK_ECC (
        HANDLE hSessionHandle,
        ULONG uiKeyBits,
        ULONG uiAlgID,
        ECCrefPublicKey * pucPublicKey,
        ECCCipher * pucKey,
        HANDLE * phKeyHandle)
```

描述：　生成会话密钥并用外部 ECC 公钥加密输出，同时返回密钥句柄。

参数：

	hSessionHandle[in]	与设备建立的会话句柄
	uiKeyBits[in]	指定产生的会话密钥长度
	uiAlgID[in]	外部 ECC 公钥的算法标识
	pucPublicKey[in]	输入的外部 ECC 公钥结构
	pucKey[out]	缓冲区指针，用于存放返回的密钥密文
	phKeyHandle[out]	返回的密钥句柄
返回值：	0	成功
	非 0	失败，返回错误代码

测试步骤：

a) 创建(定义)会话句柄和设备句柄并初始化为零值；
b) 调用打开设备函数，获取设备句柄；
c) 调用创建会话函数，获取会话句柄；
d) 指定会话密钥长度；
e) 指定 ECC 公钥算法 ID 为 SGD_SM2；
f) 创建密钥输出缓冲区；
g) 创建密钥句柄指针；

h) 将外部公钥地址，缓冲区地址，及密钥句柄指针为参数调用本函数，调用返回值应为 0；

i) 调用关闭会话函数，确认函数返回值为零而且会话句柄值变为零；

j) 调用关闭设备函数，确认函数返回值为零而且设备句柄值变为零。

6.4.14 导入会话密钥并用内部 ECC 私钥解密

原型：
```
LONG SDF_ImportKeyWithISK_ECC (
        HANDLE hSessionHandle,
        ULONG uiISKIndex,
        ECCCipher * pucKey,
        HANDLE * phKeyHandle)
```

描述：导入会话密钥并用内部 ECC 加密私钥解密，同时返回密钥句柄。

参数：

	hSessionHandle[in]	与设备建立的会话句柄
	uiISKIndex[in]	密码设备内部存储加密私钥的索引值，对应于加密时的公钥
	pucKey[in]	缓冲区指针，用于存放输入的密钥密文
	phKeyHandle[out]	返回的密钥句柄
返回值：	0	成功
	非 0	失败，返回错误代码

测试步骤：

a) 创建(定义)会话句柄和设备句柄并初始化为零值；

b) 调用打开设备函数，获取设备句柄；

c) 调用创建会话函数，获取会话句柄；

d) 分别用 1 到 n(厂家提供最大索引值)索引值并创建相应密钥句柄指针，相对应地将 6.4.12 中输出的 1 到 n 个 ECC 密钥密文结构为参数依次调用本函数，调用返回值应为 0；

e) 调用关闭会话函数，确认函数返回值为零而且会话句柄值变为零；

f) 调用关闭设备函数，确认函数返回值为零而且设备句柄值变为零。

6.4.15 生成密钥协商参数并输出

原型：
```
LONG SDF_GenerateAgreementDataWithECC (
        HANDLE hSessionHandle,
        ULONG uiISKIndex,
        ULONG uiKeyBits,
        BYTE * pucSponsorID,
        ULONG uiSponsorIDLength,
        ECCrefPublicKey * pucSponsorPublicKey,
        ECCrefPublicKey * pucSponsorTmpPublicKey,
        HANDLE * phAgreementHandle)
```

描述：使用 ECC 密钥协商算法，为计算会话密钥而产生协商参数，同时返回指定索引位置的 ECC 公钥、临时 ECC 密钥对的公钥及协商句柄。

参数：

	hSessionHandle[in]	与设备建立的会话句柄
	uiISKIndex[in]	密码设备内部存储加密私钥的索引值，该私钥用于参与密钥协商

	uiKeyBits[in]	要求协商的密钥长度
	pucSponsorID[in]	参与密钥协商的发起方 ID 值
	uiSponsorIDLength[in]	发起方 ID 长度
	pucSponsorPublicKey[out]	返回的发起方 ECC 公钥结构
	pucSponsorTmpPublicKey[out]	返回的发起方临时 ECC 公钥结构
	phAgreementHandle[out]	返回的协商句柄，用于计算协商密钥
返回值：	0	成功
	非 0	失败，返回错误代码

测试步骤：

a) 创建(定义)会话句柄和设备句柄并初始化为零值；

b) 调用打开设备函数，获取设备句柄；

c) 调用创建会话函数，获取会话句柄；

d) 采用 GB/T 35276 中的用户身份标识 ID 默认值作为发起方 ID，分别用 1 到 n(厂家提供最大索引值)索引值并指定协商密钥长度、创建相应发起方公钥输出缓冲区和发起方临时公钥输出缓冲区并创建发起方协商句柄指针，依次调用本函数，调用返回值应为 0。

注：为协商会话密钥，协商的发起方应首先调用本函数。本测试输出的 n 个发起方 ECC 公钥结构、发起方临时 ECC 公钥结构和协商句柄保存至 6.4.16 和 6.4.17 使用。当采用 GB/T 32918.5—2017 中附录 B 的 SM2 椭圆曲线密钥交换参数和输入数据时，相应的输出数据应该和 GB/T 32918.5—2017 中附录 B 的结果参考数据一致。

6.4.16 计算会话密钥

原型：

```
LONG SDF_GenerateKeyWithECC (
        HANDLE hSessionHandle,
        BYTE * pucResponseID,
        ULONG uiResponseIDLength,
        ECCrefPublicKey * pucResponsePublicKey,
        ECCrefPublicKey * pucResponseTmpPublicKey,
        HANDLE hAgreementHandle,
        HANDLE * phKeyHandle)
```

描述：使用 ECC 密钥协商算法，使用自身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。

参数：	hSessionHandle[in]	与设备建立的会话句柄
	pucResponseID[in]	外部输入的响应方 ID 值
	uiResponseIDLength[in]	外部输入的响应方 ID 长度
	pucResponsePublicKey[in]	外部输入的响应方 ECC 公钥结构
	pucResponseTmpPublicKey[in]	外部输入的响应方临时 ECC 公钥结构
	hAgreementHandle[in]	协商句柄，用于计算协商密钥
	phKeyHandle[out]	返回的密钥句柄
返回值：	0	成功
	非 0	失败，返回错误代码

测试步骤：

a) 沿用 6.4.15 中的设备句柄和会话句柄；

b) 采用 GB/T 35276 中的用户身份标识 ID 默认值作为响应方 ID，分别用 1 到 n(厂家提供最大索引值)索引值并分别相对应地将 6.4.15 中输出的 1 到 n 个协商句柄和 6.4.17 中输出的 1 到 n 个响应方 ECC 公钥结构和响应方临时 ECC 公钥结构，

相对应密钥长度，以及新创建密钥句柄指针作为参数，依次调用本函数，调用返回值应为 0；

c) 以 6.4.17 中产生的 1 到 n（厂家提供最大索引值）段明文数据和Ⅳ并分别以上一步骤中输出的 1 到 n 个会话密钥句柄为参数调用 6.6.2 对称加密函数，输出 1 到 n 段密文数据，和 6.4.17 中输出的 1 到 n 段密文数据应分别匹配相同；

d) 调用关闭会话函数，确认函数返回值为零而且会话句柄值变为零；

e) 调用关闭设备函数，确认函数返回值为零而且设备句柄值变为零。

注：本测试应在 6.4.15 和 6.4.17 的基础上进行，协商的发起方获得响应方的协商参数后调用本函数，计算会话密钥。使用 SM2 算法计算会话密钥的过程见 GB/T 35276。会话密钥计算完成后，应销毁协商句柄并释放为协商句柄分配的内存等资源。本测试的 c)步骤成功，则测试 6.4.15 到 6.4.17 均为成功；本测试的 c)步骤失败，则测试 6.4.15 到 6.4.17 均为失败。本步骤产生的会话密钥句柄保存至 6.4.27 和 6.4.28 使用。

6.4.17 产生协商数据并计算会话密钥

原型：
```
LONG SDF_GenerateAgreementDataAndKeyWithECC (
        HANDLE hSessionHandle,
        ULONG uiISKIndex,
        ULONG uiKeyBits,
        BYTE *pucResponseID,
        ULONG uiResponseIDLength,
        BYTE *pucSponsorID,
        ULONG uiSponsorIDLength,
        ECCrefPublicKey *pucSponsorPublicKey,
        ECCrefPublicKey *pucSponsorTmpPublicKey,
        ECCrefPublicKey *pucResponsePublicKey,
        ECCrefPublicKey *pucResponseTmpPublicKey,
        HANDLE *phKeyHandle)
```

描述：使用 ECC 密钥协商算法，产生协商参数并计算会话密钥，同时返回产生的协商参数和密钥句柄。

参数：

	hSessionHandle[in]	与设备建立的会话句柄
	uiISKIndex[in]	密码设备内部存储加密私钥的索引值，该私钥用于参与密钥协商
	uiKeyBits[in]	协商后要求输出的密钥长度
	pucResponseID[in]	响应方 ID 值
	uiResponseIDLength[in]	响应方 ID 长度
	pucSponsorID[in]	发起方 ID 值
	uiSponsorIDLength[in]	发起方 ID 长度
	pucSponsorPublicKey[in]	外部输入的发起方 ECC 公钥结构
	pucSponsorTmpPublicKey[in]	外部输入的发起方临时 ECC 公钥结构
	pucResponsePublicKey[out]	返回的响应方 ECC 公钥结构
	pucResponseTmpPublicKey[out]	返回的响应方临时 ECC 公钥结构
	phKeyHandle[out]	返回的密钥句柄
返回值：	0	成功
	非 0	失败，返回错误代码

测试步骤：a) 沿用 6.4.15 中的设备句柄和会话句柄；

b) 采用 GB/T 35276 中的用户身份标识 ID 默认值作为发起方和响应方 ID，分别用 6.4.15 中输出的 1 到 n 个发起方 ECC 公钥结构和发起方临时 ECC 公钥结构，密钥长度，对应本函数索引值（此处对应关系为 6.4.15 中索引值 i 对应本函数中的索引值为 $i+1$ 对 n 取模））以及创建相应响应方公钥输出缓冲区和响应方临时公钥输出缓冲区并创建响应方密钥句柄指针作为参数，依次调用本函数，每次调用返回值应为 0；

c) 随机产生 1 到 n（厂家提供最大索引值）段明文数据和Ⅳ并分别以上一步骤中输出的 1 到 n 个会话密钥句柄依次调用 6.6.2 对称加密函数，输出 1 到 n 段密文数据。

注：本测试应在 6.4.15 的基础上并先于 6.4.16 进行，协商的响应方获得发起方的协商参数后调用本函数，计算会话密钥。本次测试输出的 n 个响应方 ECC 公钥结构、响应方临时 ECC 公钥结构和 n 段明文数据、Ⅳ和密文数据保存至 6.4.16 使用。当采用 GB/T 32918.5—2017 中附录 B 的 SM2 椭圆曲线密钥交换参数和输入数据时，相应的输出数据应该和 GB/T 32918.5—2017 中附录 B 的结果参考数据一致。

6.4.18 基于 ECC 算法的数字信封转换

原型：
```
LONG SDF_ExchangeDigitEnvelopeBaseOnECC(
        HANDLE hSessionHandle,
        ULONG uiKeyIndex,
        ULONG uiAlgID,
        ECCrefPublicKey *pucPublicKey,
        ECCCipher *pucEncDataIn,
        ECCCipher *pucEncDataOut)
```

描述：将由内部加密公钥加密的会话密钥转换为由外部指定的公钥加密，可用于数字信封转换。

参数：

hSessionHandle[in]	与设备建立的会话句柄
uiKeyIndex[in]	密码设备存储的 ECC 密钥对索引值
uiAlgID[in]	外部 ECC 公钥的算法标识
pucPublicKey[in]	外部 ECC 公钥结构
pucEncDataIn[in]	缓冲区指针，用于存放输入的会话密钥密文
pucEncDataOut[out]	缓冲区指针，用于存放输出的会话密钥密文

返回值：

0	成功
非 0	失败，返回错误代码

测试步骤：a) 创建（定义）会话句柄和设备句柄并初始化为零值；

b) 调用打开设备函数，获取设备句柄；

c) 调用创建会话函数，获取会话句柄；

d) 指定算法 ID 为 SGD_SM2；

e) 分别用 1 到 n（厂家提供最大索引值）索引值并创建相应输出会话密钥密文缓冲区，相对应地将 6.4.12 中输出的 1 到 n 段会话密钥密文及密钥长度，以及任取一对 6.4.11 中产生的 ECC 密钥对公钥结构为参数，依次调用本函数，调用返回值应为 0；

f) 调用关闭会话函数，确认函数返回值为零而且会话句柄值变为零；

g) 调用关闭设备函数，确认函数返回值为零而且设备句柄值变为零。

6.4.19 生成会话密钥并用密钥加密密钥加密输出

原型：
```
LONG SDF_GenerateKeyWithKEK (
        HANDLE hSessionHandle,
        ULONG uiKeyBits,
        ULONG uiAlgID,
        ULONG uiKEKIndex,
        BYTE *pucKey,
        ULONG *puiKeyLength,
        HANDLE *phKeyHandle)
```

描述：生成会话密钥并用密钥加密密钥加密输出，同时返回密钥句柄。

参数：

	hSessionHandle[in]	与设备建立的会话句柄
	uiKeyBits[in]	指定产生的会话密钥长度
	uiAlgID[in]	算法标识，指定对称加密算法
	uiKEKIndex[in]	密码设备内部存储密钥加密密钥的索引值
	pucKey[out]	缓冲区指针，用于存放返回的密钥密文
	puiKeyLength[out]	返回的密钥密文长度
	phKeyHandle[out]	返回的密钥句柄
返回值：	0	成功
	非 0	失败，返回错误代码

测试步骤：

a) 创建(定义)会话句柄和设备句柄并初始化为零值；

b) 调用打开设备函数，获取设备句柄；

c) 调用创建会话函数，获取会话句柄；

d) 按照 GB/T 33560 的分组密码算法标识表和厂商申报支持算法逐一指定对称算法标识 ID；

e) 对应步骤 d)中指定的每种算法 ID，分别用 1 到 n(厂家提供最大索引值)索引值并创建密钥输出缓冲区、输出密钥长度变量和输出密钥句柄调用本函数，调用返回值应为 0；

f) 调用关闭会话函数，确认函数返回值为零而且会话句柄值变为零；

g) 调用关闭设备函数，确认函数返回值为零而且设备句柄值变为零。

注：加密模式使用 ECB 模式。本测试输出的对应每种对称算法 ID 的 1 到 n 个密钥密文及其长度保存至 6.4.20 使用。

6.4.20 导入会话密钥并用密钥加密密钥解密

原型：
```
LONG SDF_ImportKeyWithKEK (
        HANDLE hSessionHandle,
        ULONG uiAlgID,
        ULONG uiKEKIndex,
        BYTE *pucKey,
        ULONG puiKeyLength,
        HANDLE *phKeyHandle)
```

描述：导入会话密钥并用密钥加密密钥解密，同时返回会话密钥句柄。

参数：

	hSessionHandle[in]	与设备建立的会话句柄

	uiAlgID[in]	算法标识，指定对称加密算法
	uiKEKIndex[in]	密码设备内部存储密钥加密密钥的索引值
	pucKey[in]	缓冲区指针，用于存放输入的密钥密文
	puiKeyLength[in]	输入的密钥密文长度
	phKeyHandle[out]	返回的密钥句柄
返回值：	0	成功
	非 0	失败，返回错误代码

测试步骤：

a) 创建(定义)会话句柄和设备句柄并初始化为零值；

b) 调用打开设备函数，获取设备句柄；

c) 调用创建会话函数，获取会话句柄；

d) 按照 GB/T 33560 的分组密码算法标识表和厂商申报支持算法逐一指定对称算法标识 ID；

e) 对应步骤 d)中指定的每种算法 ID，分别用 1 到 n(厂家提供最大索引值)索引值及 6.4.19 中输出的相对应的 1 到 n 个密钥密文及其长度并创建密钥输出句柄调用本函数，调用返回值应为 0；

f) 调用关闭会话函数，确认函数返回值为零而且会话句柄值变为零；

g) 调用关闭设备函数，确认函数返回值为零而且设备句柄值变为零。

注：加密模式使用 ECB 模式。

6.4.21 计算 IKE 工作密钥

原型：

```
LONG SDF_GenerateKeywithIKE (
        HANDLE hSessionHandle,
        BYTE * pucSponsorNonce,
        ULONG uiSponsorNonceLength,
        BYTE * pucResponseNonce,
        ULONG uiResponseNonceLength,
        BYTE * pucSponsorCookie,
        ULONG uiSponsorCookieLength,
        BYTE * pucResponseCookie,
        ULONG uiResponseCookieLength,
        ULONG uiPrfAlgID,
        ULONG uiKeyBitsD,
        HANDLE * phKeyHandleD,
        ULONG uiKeyBitsA,
        HANDLE * phKeyHandleA,
        ULONG uiKeyBitsE,
        HANDLE * phKeyHandleE)
```

描述：使用 IKE 一阶段(主模式)交换得到的密钥计算参数计算 IKE 工作密钥，同时返回工作密钥句柄。

参数：	hSessionHandle[in]	与设备建立的会话句柄
	pucSponsorNonce[in]	发起方 nonce 载荷主体
	uiSponsorNonceLength[in]	发起方 nonce 载荷主体长度
	pucResponseNonce[in]	响应方 nonce 载荷主体

	uiResponseNonceLength[in]	响应方 nonce 载荷主体长度
	pucSponsorCookie[in]	发起方 cookie
	uiSponsorCookieLength[in]	发起方 cookie 长度
	pucResponseCookie[in]	响应方 cookie
	uiResponseCookieLength[in]	响应方 cookie 长度
	uiPrfAlgID[in]	PRF 算法标识
	uiKeyBitsD[in]	SKEYID_d 密钥长度
	phKeyHandleD[out]	返回的 SKEYID_d 密钥句柄
	uiKeyBitsA[in]	SKEYID_a 密钥长度
	phKeyHandleA[out]	返回的 SKEYID_a 密钥句柄
	uiKeyBitsE[in]	SKEYID_e 密钥长度
	phKeyHandleE[out]	返回的 SKEYID_e 密钥句柄
返回值：	0	成功
	非 0	失败，返回错误代码

测试步骤：

a) 创建(定义)会话句柄和设备句柄并初始化为零值；

b) 调用打开设备函数，获取设备句柄；

c) 调用创建会话函数，获取会话句柄；

d) 按照 GB/T 36968 的相关要求产生发起方 cookie、发起方 nonce 和响应方 cookie、响应方 nonce，用以上参数及其长度作为输入，PRF 算法标识采用 SM3 密码杂凑算法标识(GB/T 33560 定义)，分别创建 SKEYID_d、SKEYID_a 和 SKEYID_a 密钥句柄并按照 GB/T 36968 的相关要求设定各自的密钥长度，调用本函数，调用返回值应为 0；

e) 调用关闭会话函数，确认函数返回值为零而且会话句柄值变为零；

f) 调用关闭设备函数，确认函数返回值为零而且设备句柄值变为零。

注：本测试产生的发起方 nonce、响应方 nonce 和 SKEYID_d 密钥句柄保存至 6.4.23 号 6.4.24 使用。

6.4.22 计算 IKE 工作密钥并用外部 ECC 公钥加密输出

原型：

```
LONG SDF_GenerateKeywithEPK_IKE (
        HANDLE hSessionHandle,
        BYTE * pucSponsorNonce,
        ULONG uiSponsorNonceLength,
        BYTE * pucResponseNonce,
        ULONG uiResponseNonceLength,
        BYTE * pucSponsorCookie,
        ULONG uiSponsorCookieLength,
        BYTE * pucResponseCookie,
        ULONG uiResponseCookieLength,
        ULONG uiPrfAlgID,
        ULONG uiEccAlgID,
        ECCrefPublicKey * pucPublicKey,
        ULONG uiKeyBitsD,
        ECCCipher * pucKeyD,
        HANDLE * phKeyHandleD,
```

```
    ULONG uiKeyBitsA,
    ECCCipher *pucKeyA,
    HANDLE *phKeyHandleA,
    ULONG uiKeyBitsE,
    ECCCipher *pucKeyE,
    HANDLE *phKeyHandleE)
```

描述：　使用 IKE 一阶段(主模式)交换得到的密钥计算参数计算 IKE 工作密钥，并用外部 ECC 公钥加密输出，同时返回工作密钥句柄。

参数：

hSessionHandle[in]	与设备建立的会话句柄
pucSponsorNonce[in]	发起方 nonce 载荷主体
uiSponsorNonceLength[in]	发起方 nonce 载荷主体长度
pucResponseNonce[in]	响应方 nonce 载荷主体
uiResponseNonceLength[in]	响应方 nonce 载荷主体长度
pucSponsorCookie[in]	发起方 cookie
uiSponsorCookieLength[in]	发起方 cookie 长度
pucResponseCookie[in]	响应方 cookie
uiResponseCookieLength[in]	响应方 cookie 长度
uiPrfAlgID [in]	PRF 算法标识
uiEccAlgID[in]	外部 ECC 公钥的算法标识
pucPublicKey[in]	输入的外部 ECC 公钥结构
uiKeyBitsD [in]	SKEYID_d 密钥长度
pucKeyD[out]	缓冲区指针，用于存放返回的 SKEYID_d 密钥密文
phKeyHandleD [out]	返回的 SKEYID_d 密钥句柄
uiKeyBitsA [in]	SKEYID_a 密钥长度
pucKeyA[out]	缓冲区指针，用于存放返回的 SKEYID_A 密钥密文
phKeyHandleA [out]	返回的 SKEYID_a 密钥句柄
uiKeyBitsE [in]	SKEYID_e 密钥长度
pucKeyE[out]	缓冲区指针，用于存放返回的 SKEYID_E 密钥密文
phKeyHandleE [out]	返回的 SKEYID_e 密钥句柄

返回值：

0	成功
非 0	失败，返回错误代码

测试步骤：

a) 创建(定义)会话句柄和设备句柄并初始化为零值；

b) 调用打开设备函数，获取设备句柄；

c) 调用创建会话函数，获取会话句柄；

d) 按照 GB/T 36968 的相关要求产生发起方 cookie、发起方 nonce 和响应方 cookie、响应方 nonce，用以上参数及其长度作为输入，PRF 算法标识采用 SM3 密码杂凑算法标识(GB/T 33560 定义)，外部 ECC 公钥算法标识采用 SM2 椭圆曲线加密算法(GB/T 33560 定义)，分别创建 SKEYID_d、SKEYID_a 和 SKEYID_a 密钥句柄及相应的公钥加密结构缓冲区，并按照 GB/T 36968 的相关要求设定各自的密钥长度，调用本参数，调用返回值应为 0；

e) 调用关闭会话函数，确认函数返回值为零而且会话句柄值变为零；

f) 调用关闭设备函数，确认函数返回值为零而且设备句柄值变为零。

6.4.23 计算 IPSEC 会话密钥

原型：
```
LONG SDF_GenerateKeywithIPSEC (
        HANDLE hSessionHandle,
        BYTE *pucProtocolID,
        ULONG uiProtocolIDLength,
        BYTE *pucSpi,
        ULONG uiSpiLength,
        BYTE *pucSponsorNonce,
        ULONG uiSponsorNonceLength,
        BYTE *pucResponseNonce,
        ULONG uiResponseNonceLength,
        HANDLE phKeyHandle,
        ULONG uiPrfAlgID,
        ULONG uiKeyBitsEnc,
        HANDLE *phKeyHandleEnc,
        ULONG uiKeyBitsMac,
        HANDLE *phKeyHandleMac)
```

描述：使用 IKE 二阶段(快速模式)交换得到的密钥计算参数计算 IPSEC 会话密钥，同时返回会话密钥句柄。

参数：

hSessionHandle[in]	与设备建立的会话句柄
pucProtocolID[in]	协议 ID
uiProtocolIDLength[in]	协议 ID 长度
pucSpi [in]	安全参数索引
SPIuiSpiLength [in]	安全参数索引 SPI 长度
pucSponsorNonce[in]	发起方 nonce 载荷主体
uiSponsorNonceLength[in]	发起方 nonce 载荷主体长度
pucResponseNonce[in]	响应方 nonce 载荷主体
uiResponseNonceLength[in]	响应方 nonce 载荷主体长度
hKeyHandle[in]	输入的 SKEYID_d 密钥句柄
uiPrfAlgID [in]	PRF 算法标识
uiKeyBitsEnc [in]	加密密钥长度
phKeyHandleEnc[out]	返回的加密密钥句柄
uiKeyBitsMac [in]	杂凑密钥长度
phKeyHandleMac[out]	返回的杂凑密钥句柄

返回值：

0	成功
非 0	失败，返回错误代码

测试步骤：

a) 创建(定义)会话句柄和设备句柄并初始化为零值；

b) 调用打开设备函数，获取设备句柄；

c) 调用创建会话函数，获取会话句柄；

d) 采用 6.4.21 中产生的发起方 nonce 和响应方 nonce 并按照 GB/T 36968 的相关要求产生安全参数索引 SPI 及各参数长度作为输入，协议 ID 采用 ESP 协议(GB/T 36968 定义)，PRF 算法标识采用 SM3 密码杂凑算法标识(GB/T 33560

定义),6.4.21 中产生的 SKEYID_d 密钥输出句柄作为输入的 SKEYID_d 密钥句柄,分别创建加密密钥和杂凑密钥句柄并按照 GB/T 36968 的相关要求设定各自的密钥长度,调用本参数,调用返回值应为 0;

e) 调用关闭会话函数,确认函数返回值为零而且会话句柄值变为零;

f) 调用关闭设备函数,确认函数返回值为零而且设备句柄值变为零。

6.4.24 计算 IPSEC 会话密钥并用外部 ECC 公钥加密输出

原型:

```
LONG SDF_GenerateKeywithEPK_IPSEC (
        HANDLE hSessionHandle,
        BYTE * pucProtocolID,
        ULONG uiProtocolIDLength,
        BYTE * pucSpi,
        ULONG uiSpiLength,
        BYTE * pucSponsorNonce,
        ULONG uiSponsorNonceLength,
        BYTE * pucResponseNonce,
        ULONG uiResponseNonceLength,
        HANDLE phKeyHandle,
        ULONG uiPrfAlgID,
        ULONG uiEccAlgID,
        ECCrefPublicKey * pucPublicKey,
        ULONG uiKeyBitsEnc,
        ECCCipher * pucKeyEnc
        HANDLE * phKeyHandleEnc,
        ULONG uiKeyBitsMac,
        ECCCipher * pucKeyMac
        HANDLE * phKeyHandleMac)
```

描述: 使用 IKE 二阶段(快速模式)交换得到的密钥计算参数计算 IPSEC 会话密钥,并用外部 ECC 公钥加密输出,同时返回会话密钥句柄。

参数:

hSessionHandle[in]	与设备建立的会话句柄
pucProtocolID[in]	协议 ID
uiProtocolIDLength[in]	协议 ID 长度
pucSpi [in]	安全参数索引 SPI
uiSpiLength [in]	安全参数索引 SPI 长度
pucSponsorNonce[in]	发起方 nonce 载荷主体
uiSponsorNonceLength[in]	发起方 nonce 载荷主体长度
pucResponseNonce[in]	响应方 nonce 载荷主体
uiResponseNonceLength[in]	响应方 nonce 载荷主体长度
hKeyHandle[in]	输入的 SKEYID_d 密钥句柄
uiPrfAlgID [in]	PRF 算法标识
uiEccAlgID[in]	外部 ECC 公钥的算法标识
pucPublicKey[in]	输入的外部 ECC 公钥结构
uiKeyBitsEnc [in]	加密密钥长度

	pucKeyEnc [out]	缓冲区指针，用于存放返回的加密密钥密文
	phKeyHandleEnc[out]	返回的加密密钥句柄
	uiKeyBitsMac [in]	杂凑密钥长度
	pucKeyMac [out]	缓冲区指针，用于存放返回的杂凑密钥密文
	phKeyHandleMac[out]	返回的杂凑密钥句柄
返回值：	0	成功
	非 0	失败，返回错误代码

测试步骤：

a) 创建(定义)会话句柄和设备句柄并初始化为零值；

b) 调用打开设备函数，获取设备句柄；

c) 调用创建会话函数，获取会话句柄；

d) 采用 6.4.21 中产生的发起方 nonce 和响应方 nonce 并按照 GB/T 36968 的相关要求产生安全参数索引 SPI 及各参数长度作为输入，协议 ID 采用 ESP 协议(GB/T 36968 定义)，PRF 算法标识采用 SM3 密码杂凑算法标识(GB/T 33560 定义)，6.4.21 中产生的 SKEYID_d 密钥输出句柄作为输入的 SKEYID_d 密钥句柄，外部 ECC 公钥算法标识采用 SM2 椭圆曲线加密算法(GB/T 33560 定义)，分别创建加密密钥和杂凑密钥句柄及相应的公钥加密结构缓冲区，并按照 GB/T 36968的相关要求设定各自的密钥长度，调用本参数，调用返回值应为 0；

e) 调用关闭会话函数，确认函数返回值为零而且会话句柄值变为零；

f) 调用关闭设备函数，确认函数返回值为零而且设备句柄值变为零。

6.4.25 计算 SSL 工作密钥

原型：

```
LONG SDF_GenerateKeywithSSL (
        HANDLE hSessionHandle,
        BYTE * pucKeyPreMaster,
        ULONG uiKeyPreMasterLength,
        BYTE * pucClientRandom,
        ULONG uiClientRandomLength,
        BYTE * pucServerRandom,
        ULONG uiServerRandomLength,
        ULONG uiPrfAlgID,
        ULONG uiKeyBitsClientMac,
        HANDLE * phKeyHandleClientMac,
        ULONG uiKeyBitsServerMac,
        HANDLE * phKeyHandleServerMac,
        ULONG uiKeyBitsClientEnc,
        HANDLE * phKeyHandleClientEnc,
        ULONG uiKeyBitsServerEnc,
        HANDLE * phKeyHandleServerEnc)
```

描述：使用 SSL 握手协议得到的密钥计算参数计算 SSL 工作密钥，同时返回工作密钥句柄。

参数：	hSessionHandle[in]	与设备建立的会话句柄
	pucKeyPreMaster[in]	预主密钥 pre_master_secret
	uiKeyPreMasterLength[in]	预主密钥长度
	pucClientRandom[in]	客户端随机数

	uiClientRandomLength[in]	客户端随机数长度
	pucServerRandom[in]	服务端随机数
	uiServerRandomLength[in]	服务端随机数长度
	uiPrfAlgID [in]	PRF 算法标识
	uiKeyBitsClientMac[in]	客户端杂凑密钥长度
	phKeyHandleClientMac[out]	返回的客户端杂凑密钥句柄
	uiKeyBitsServerMac[in]	服务端杂凑密钥长度
	phKeyHandleServerMac[out]	返回的服务端杂凑密钥句柄
	uiKeyBitsClientEnc[in]	客户端加密密钥长度
	phKeyHandleClientEnc[out]	返回的客户端加密密钥句柄
	uiKeyBitsServerEnc[in]	服务端加密密钥长度
	phKeyHandleServerEnc[out]	返回的服务端加密密钥句柄
返回值：	0	成功
	非 0	失败，返回错误代码

测试步骤：

a) 创建(定义)会话句柄和设备句柄并初始化为零值；

b) 调用打开设备函数，获取设备句柄；

c) 调用创建会话函数，获取会话句柄；

d) 按照 GM/T 0024 的相关要求产生预主密钥、客户端随机数、服务端随机数，用以上参数及其长度作为输入，PRF 算法标识采用 SM3 密码杂凑算法标识(GB/T 33560定义)，分别创建客户端杂凑密钥、服务端杂凑密钥、客户端加密密钥、服务端加密密钥句柄并按照 GM/T 0024 的相关要求设定各自的密钥长度，调用本函数，调用返回值应为 0；

e) 调用关闭会话函数，确认函数返回值为零而且会话句柄值变为零；

f) 调用关闭设备函数，确认函数返回值为零而且设备句柄值变为零。

6.4.26 计算 SSL 工作密钥并用外部 ECC 公钥加密输出

原型：

```
LONG SDF_GenerateKeywithEPK_SSL (
        HANDLE hSessionHandle,
        BYTE * pucKeyPreMaster,
        ULONG uiKeyPreMasterLength,
        BYTE * pucClientRandom,
        ULONG uiClientRandomLength,
        BYTE * pucServerRandom,
        ULONG uiServerRandomLength,
        ULONG uiPrfAlgID,
        ULONG uiEccAlgID,
        ECCrefPublicKey * pucPublicKey,
        ULONG uiKeyBitsClientMac,
        ECCCipher * pucKeyClientMac,
        HANDLE * phKeyHandleClientMac,
        ULONG uiKeyBitsServerMac,
        ECCCipher * pucKeyServerMac,
        HANDLE * phKeyHandleServerMac,
```

ULONG uiKeyBitsClientEnc,
ECCCipher * pucKeyClientEnc,
HANDLE * phKeyHandleClientEnc,
ULONG uiKeyBitsServerEnc,
ECCCipher * pucKeyServerEnc,
HANDLE * phKeyHandleServerEnc)

描述：使用 SSL 握手协议得到的密钥计算参数计算 SSL 工作密钥，并用外部 ECC 公钥加密输出，同时返回工作密钥句柄。

参数：

hSessionHandle[in]	与设备建立的会话句柄
pucKeyPreMaster[in]	预主密钥 pre_master_secret
uiKeyPreMasterLength[in]	预主密钥长度
pucClientRandom[in]	客户端随机数
uiClientRandomLength[in]	客户端随机数长度
pucServerRandom[in]	服务端随机数
uiServerRandomLength[in]	服务端随机数长度
uiPrfAlgID [in]	PRF 算法标识
uiEccAlgID [in]	外部 ECC 公钥的算法标识
pucPublicKey[in]	输入的外部 ECC 公钥结构
uiKeyBitsClientMac[in]	客户端杂凑密钥长度
pucKeyClientMac[out]	缓冲区指针，用于存放返回的客户端杂凑密钥
phKeyHandleClientMac[out]	返回的客户端杂凑密钥句柄
uiKeyBitsServerMac[in]	服务端杂凑密钥长度
pucKeyServerMac[out]	缓冲区指针，用于存放返回的服务端杂凑密钥
phKeyHandleServerMac[out]	返回的服务端杂凑密钥句柄
uiKeyBitsClientEnc[in]	客户端加密密钥长度
pucKeyClientEnc[out]	缓冲区指针，用于存放返回的客户端加密密钥
phKeyHandleClientEnc[out]	返回的客户端加密密钥句柄
uiKeyBitsServerEnc[in]	服务端加密密钥长度
pucKeyServerEnc[out]	缓冲区指针，用于存放返回的服务端加密密钥
phKeyHandleServerEnc[out]	返回的服务端加密密钥句柄

返回值：

0	成功
非 0	失败，返回错误代码

测试步骤：

a) 创建(定义)会话句柄和设备句柄并初始化为零值；
b) 调用打开设备函数，获取设备句柄；
c) 调用创建会话函数，获取会话句柄；
d) 按照 GM/T 0024 的相关要求产生预主密钥、客户端随机数、服务端随机数，用以上参数及其长度作为输入，PRF 算法标识采用 SM3 密码杂凑算法标识(GB/T 33560定义)，外部 ECC 公钥算法标识采用 SM2 椭圆曲线加密算法(GB/T 33560定义)，分别创建客户端杂凑密钥、服务端杂凑密钥、客户端加密密钥、服务端加密密钥句柄及相应的公钥加密结构缓冲区，并按照 GM/T 0024 的相关要求设定各自的密钥长度，调用本函数，调用返回值应为 0；
e) 调用关闭会话函数，确认函数返回值为零而且会话句柄值变为零；
f) 调用关闭设备函数，确认函数返回值为零而且设备句柄值变为零。

6.4.27 计算 SSL 工作密钥(ECDHE)

原型：
```
LONG SDF_GenerateKeywithECDHE_SSL (
        HANDLE hSessionHandle,
        HANDLE phKeyHandlePreMaster,
        BYTE * pucClientRandom,
        ULONG uiClientRandomLength,
        BYTE * pucServerRandom,
        ULONG uiServerRandomLength,
        ULONG uiPrfAlgID,
        ULONG uiKeyBitsClientMac,
        HANDLE * phKeyHandleClientMac,
        ULONG uiKeyBitsServerMac,
        HANDLE * phKeyHandleServerMac,
        ULONG uiKeyBitsClientEnc,
        HANDLE * phKeyHandleClientEnc,
        ULONG uiKeyBitsServerEnc,
        HANDLE * phKeyHandleServerEnc)
```

描述：使用 SSL 握手协议得到的密钥计算参数计算 SSL 工作密钥,同时返回工作密钥句柄。

参数：

hSessionHandle[in]	与设备建立的会话句柄
phKeyHandlePreMaster[in]	预主密钥 pre_master_secret 密钥句柄
pucClientRandom[in]	客户端随机数
uiClientRandomLength[in]	客户端随机数长度
pucServerRandom[in]	服务端随机数
uiServerRandomLength[in]	服务端随机数长度
uiPrfAlgID [in]	PRF 算法标识
uiKeyBitsClientMac[in]	客户端杂凑密钥长度
phKeyHandleClientMac[out]	返回的客户端杂凑密钥句柄
uiKeyBitsServerMac[in]	服务端杂凑密钥长度
phKeyHandleServerMac[out]	返回的服务端杂凑密钥句柄
uiKeyBitsClientEnc[in]	客户端加密密钥长度
phKeyHandleClientEnc[out]	返回的客户端加密密钥句柄
uiKeyBitsServerEnc[in]	服务端加密密钥长度
phKeyHandleServerEnc[out]	返回的服务端加密密钥句柄

返回值：

0	成功
非 0	失败,返回错误代码

测试步骤：

a) 创建(定义)会话句柄和设备句柄并初始化为零值；

b) 调用打开设备函数,获取设备句柄；

c) 调用创建会话函数,获取会话句柄；

d) 使用 6.4.16 产生的任一会话密钥句柄作为预主密钥句柄,按照 GM/T 0024 的相关要求产生客户端随机数、服务端随机数及其长度作为输入,PRF 算法标识采用 SM3 密码杂凑算法标识(GB/T 33560 定义),分别创建客户端杂凑密钥、服务端杂凑密钥、客户端加密密钥、服务端加密密钥句柄并按照 GM/T 0024 的相关要

求设定各自的密钥长度，调用本函数，调用返回值应为 0；

e) 调用关闭会话函数，确认函数返回值为零而且会话句柄值变为零；

f) 调用关闭设备函数，确认函数返回值为零而且设备句柄值变为零。

6.4.28 计算 SSL 工作密钥并用外部 ECC 公钥加密输出(ECDHE)

原型：

```
LONG SDF_GenerateKeywithEPK_ECDHE_SSL (
        HANDLE hSessionHandle,
        HANDLE phKeyHandlePreMaster,
        BYTE * pucClientRandom,
        ULONG uiClientRandomLength,
        BYTE * pucServerRandom,
        ULONG uiServerRandomLength,
        ULONG uiPrfAlgID,
        ULONG uiEccAlgID,
        ECCrefPublicKey * pucPublicKey,
        ULONG uiKeyBitsClientMac,
        ECCCipher * pucKeyClientMac,
        HANDLE * phKeyHandleClientMac,
        ULONG uiKeyBitsServerMac,
        ECCCipher * pucKeyServerMac,
        HANDLE * phKeyHandleServerMac,
        ULONG uiKeyBitsClientEnc,
        ECCCipher * pucKeyClientEnc,
        HANDLE * phKeyHandleClientEnc,
        ULONG uiKeyBitsServerEnc,
        ECCCipher * pucKeyServerEnc,
        HANDLE * phKeyHandleServerEnc)
```

描述：使用 SSL 握手协议得到的密钥计算参数计算 SSL 工作密钥，并用外部 ECC 公钥加密输出，同时返回工作密钥句柄。

参数：

hSessionHandle[in]	与设备建立的会话句柄
phKeyHandlePreMaster[in]	预主密钥 pre_master_secret 密钥句柄
pucClientRandom[in]	客户端随机数
uiClientRandomLength[in]	客户端随机数长度
pucServerRandom[in]	服务端随机数
uiServerRandomLength[in]	服务端随机数长度
uiPrfAlgID [in]	PRF 算法标识
uiEccAlgID [in]	外部 ECC 公钥的算法标识
pucPublicKey[in]	输入的外部 ECC 公钥结构
uiKeyBitsClientMac[in]	客户端杂凑密钥长度
pucKeyClientMac[out]	缓冲区指针，用于存放返回的客户端杂凑密钥
phKeyHandleClientMac[out]	返回的客户端杂凑密钥句柄
uiKeyBitsServerMac[in]	服务端杂凑密钥长度
pucKeyServerMac[out]	缓冲区指针，用于存放返回的服务端杂凑密钥

	phKeyHandleServerMac[out]	返回的服务端杂凑密钥句柄
	uiKeyBitsClientEnc[in]	客户端加密密钥长度
	pucKeyClientEnc[out]	缓冲区指针，用于存放返回的客户端加密密钥
	phKeyHandleClientEnc[out]	返回的客户端加密密钥句柄
	uiKeyBitsServerEnc[in]	服务端加密密钥长度
	pucKeyServerEnc[out]	缓冲区指针，用于存放返回的服务端加密密钥
	phKeyHandleServerEnc[out]	返回的服务端加密密钥句柄
返回值：	0	成功
	非 0	失败，返回错误代码

测试步骤：

a) 创建(定义)会话句柄和设备句柄并初始化为零值；

b) 调用打开设备函数，获取设备句柄；

c) 调用创建会话函数，获取会话句柄；

d) 使用 6.4.16 产生的任一会话密钥句柄作为预主密钥句柄，按照 GM/T 0024 的相关要求产生客户端随机数、服务端随机数及其长度作为输入，PRF 算法标识采用 SM3 密码杂凑算法标识(GB/T 33560 定义)，外部 ECC 公钥算法标识采用 SM2 椭圆曲线加密算法(GB/T 33560 定义)，分别创建客户端杂凑密钥、服务端杂凑密钥、客户端加密密钥、服务端加密密钥句柄及相应的公钥加密结构缓冲区，并按照 GM/T 0024 的相关要求设定各自的密钥长度，调用本函数，调用返回值应为 0；

e) 调用关闭会话函数，确认函数返回值为零而且会话句柄值变为零；

f) 调用关闭设备函数，确认函数返回值为零而且设备句柄值变为零。

6.4.29 销毁会话密钥

原型：

```
LONG SDF_DestroyKey (
        HANDLE hSessionHandle,
        HANDLE hKeyHandle)
```

描述：销毁会话密钥，并释放为密钥句柄分配的内存等资源。

参数：	hSessionHandle[in]	与设备建立的会话句柄
	hKeyHandle[in]	输入的密钥句柄
返回值：	0	成功
	非 0	失败，返回错误代码

测试步骤：

a) 创建(定义)会话句柄和设备句柄并初始化为零值；

b) 调用打开设备函数，获取设备句柄；

c) 调用创建会话函数，获取会话句柄；

d) 分别用不同对称算法 ID 和 1 到 n(厂家提供最大索引值)索引值调用 6.4.19 生成会话密钥并用密钥加密密钥加密输出函数，获得多个密钥句柄；

e) 将会话密钥句柄为参数调用本函数，调用返回值应为 0；

f) 调用关闭会话函数，确认函数返回值为零而且会话句柄值变为零；

g) 调用关闭设备函数，确认函数返回值为零而且设备句柄值变为零。

注：在对称算法运算完成后，应调用本函数销毁会话密钥。

6.5 非对称算法运算类接口测试

6.5.1 测试项目

需要测试的非对称算法运算类函数接口如表3所示。

表3 非对称算法运算类函数

函数名称	功能
SDF_ExternalPublicKeyOperation_RSA	外部公钥RSA运算
SDF_InternalPublicKeyOperation_RSA	内部公钥RSA运算
SDF_InternalPrⅣateKeyOperation_RSA	内部私钥RSA运算
SDF_ExternalVerify_ECC	外部密钥ECC验证
SDF_InternalSign_ECC	内部密钥ECC签名
SDF_InternalVerify_ECC	内部密钥ECC验证
SDF_ExternalEncrypt_ECC	外部密钥ECC加密

6.5.2 外部公钥RSA运算

原型：
```
LONG SDF_ExternalPublicKeyOperation_RSA(
        HANDLE hSessionHandle,
        RSArefPublicKey * pucPublicKey,
        BYTE * pucDataInput,
        ULONG uiInputLength,
        BYTE * pucDataOutput,
        ULONG * puiOutputLength)
```

描述：指定使用外部公钥对数据进行运算。

参数：

	hSessionHandle[in]	与设备建立的会话句柄
	pucPublicKey [in]	外部RSA公钥结构
	pucDataInput [in]	缓冲区指针，用于存放输入的数据
	uiInputLength[in]	输入的数据长度
	pucDataOutput[out]	缓冲区指针，用于存放输出的数据
	puiOutputLength[out]	输出的数据长度

返回值：

0	成功
非0	失败，返回错误代码

测试步骤：

a) 创建(定义)会话句柄和设备句柄并初始化为零值；

b) 调用打开设备函数，获取设备句柄；

c) 调用创建会话函数，获取会话句柄；

d) 创建输出数据缓冲区；

e) 创建输出数据长度变量；

f) 分别用6.5.4中输出的1到n段RSA私钥运算数据以及相应的RSA公钥作为输入数据，依次调用本函数，调用返回值应为0；

g) 每次调用的输出数据和6.5.4中的输入数据进行比较，结果应相同；

h) 调用关闭会话函数，确认函数返回值为零而且会话句柄值变为零；

i) 调用关闭设备函数，确认函数返回值为零而且设备句柄值变为零。

注：数据格式由应用层封装。本测试在 6.5.4 基础上进行。

6.5.3 内部公钥 RSA 运算

原型：LONG SDF_InternalPublicKeyOperation_RSA(
HANDLE hSessionHandle,
ULONG uiKeyIndex,
BYTE *pucDataInput,
ULONG uiInputLength,
BYTE *pucDataOutput,
ULONG *puiOutputLength)

描述：使用内部指定索引的公钥对数据进行运算。

参数：

hSessionHandle[in]	与设备建立的会话句柄
uiKeyIndex[in]	密码设备内部存储公钥的索引值
pucDataInput[in]	缓冲区指针，用于存放外部输入的数据
uiInputLength[in]	输入的数据长度
pucDataOutput[out]	缓冲区指针，用于存放输出的数据
puiOutputLength[out]	输出的数据长度

返回值：

0	成功
非 0	失败，返回错误代码

测试步骤：

a) 创建(定义)会话句柄和设备句柄并初始化为零值；

b) 调用打开设备函数，获取设备句柄；

c) 调用创建会话函数，获取会话句柄；

d) 创建输出数据缓冲区；

e) 创建输出数据长度变量；

f) 随机产生指定长度的输入数据；

g) 分别用 6.5.4 中输出的 1 到 n 段 RSA 私钥运算数据以及相应的 RSA 内部公钥索引值(除特别注明，应和 RSA 私钥运算使用的内部私钥索引值相同)作为输入数据，依次调用本函数，调用返回值应为 0；

h) 每次调用的输出数据和 6.5.4 中的输入数据进行比较，结果应相同；

i) 调用关闭会话函数，确认函数返回值为零而且会话句柄值变为零；

j) 调用关闭设备函数，确认函数返回值为零而且设备句柄值变为零。

注：索引范围仅限于内部签名密钥对，数据格式由应用层封装。本测试在 6.5.4 基础上进行。

6.5.4 内部私钥 RSA 运算

原型：LONG SDF_InternalPrivateKeyOperation_RSA(
HANDLE hSessionHandle,
ULONG uiKeyIndex,
BYTE *pucDataInput,
ULONG uiInputLength,
BYTE *pucDataOutput,
ULONG *puiOutputLength)

描述：　使用内部指定索引的私钥对数据进行运算。

参数：

hSessionHandle[in]	与设备建立的会话句柄
uiKeyIndex[in]	密码设备内部存储私钥的索引值
pucDataInput[in]	缓冲区指针，用于存放外部输入的数据
uiInputLength[in]	输入的数据长度
pucDataOutput[out]	缓冲区指针，用于存放输出的数据
puiOutputLength[out]	输出的数据长度

返回值：

0	成功
非 0	失败，返回错误代码

测试步骤：

a) 使用被测试密码设备的管理工具预先在密码设备中产生或导入 n（n 为厂家提供最大索引值）对 RSA 密钥对；

b) 创建（定义）会话句柄和设备句柄并初始化为零值；

c) 调用打开设备函数，获取设备句柄；

d) 调用创建会话函数，获取会话句柄；

e) 指定输入数据长度；

f) 创建输入及输出数据缓冲区；

g) 创建输出数据长度变量；

h) 随机产生指定长度的输入数据；

i) 分别用 1 到 n（厂家提供最大索引值）索引值及输入数据缓冲区和长度、输出数据缓冲区及输出数据长度变量地址为参数依次调用本函数，调用返回值应为 0；

j) 调用关闭会话函数，确认函数返回值为零而且会话句柄值变为零；

k) 调用关闭设备函数，确认函数返回值为零而且设备句柄值变为零。

注：索引范围仅限于内部签名密钥对，数据格式由应用层封装。本测试先于 6.5.2 及 6.5.3，产生或导入的 RSA 公钥、使用的输入数据及输出的 n 段 RSA 私钥运算数据保存至 6.5.2 及 6.5.3 使用。

6.5.5 外部密钥 ECC 验证

原型：

```
LONG SDF_ExternalVerify_ECC(
        HANDLE hSessionHandle,
        ULONG uiAlgID,
        ECCrefPublicKey *pucPublicKey,
        BYTE *pucDataInput,
        ULONG uiInputLength,
        ECCSignature *pucSignature)
```

描述：　使用外部 ECC 公钥对 ECC 签名值进行验证运算。

参数：

hSessionHandle[in]	与设备建立的会话句柄
uiAlgID[in]	算法标识，指定使用的 ECC 算法
pucPublicKey[in]	外部 ECC 公钥结构
pucDataInput[in]	缓冲区指针，用于存放外部输入的数据
uiDataLength[in]	输入的数据长度
pucSignature[in]	缓冲区指针，用于存放输入的签名值数据

返回值：

0	成功
非 0	失败，返回错误代码

测试步骤：

a) 创建（定义）会话句柄和设备句柄并初始化为零值；

b) 调用打开设备函数，获取设备句柄；
c) 调用创建会话函数，获取会话句柄；
d) 指定 ECC 算法标识 ID，应按 GB/T 33560 非对称密码算法标识；
e) 分别将 6.5.6 中输出的 1 到 n 个 ECC 签名值（ECC 签名数据结构）、对应的输入数据及数据长度、对应签名私钥的 ECC 公钥结构作为参数依次调用本函数，调用返回值应为 0；
f) 调用关闭会话函数，确认函数返回值为零而且会话句柄值变为零；
g) 调用关闭设备函数，确认函数返回值为零而且设备句柄值变为零。

注：输入数据为待签数据的杂凑值。当使用 SM2 算法时，该输入数据为待签数据经过 SM2 签名预处理的结果，预处理过程见 GB/T 35276。本函数的测试应在 6.5.6 的基础上进行。

6.5.6 内部密钥 ECC 签名

原型：
```
LONG SDF_InternalSign_ECC(
        HANDLE hSessionHandle,
        ULONG uiISKIndex,
        BYTE * pucData,
        ULONG uiDataLength,
        ECCSignature * pucSignature)
```

描述：使用内部 ECC 私钥对数据进行签名运算。

参数：

	hSessionHandle[in]	与设备建立的会话句柄
	uiISKIndex [in]	密码设备内部存储的 ECC 签名私钥的索引值
	pucData[in]	缓冲区指针，用于存放外部输入的数据
	uiDataLength[in]	输入的数据长度
	pucSignature [out]	缓冲区指针，用于存放输出的签名值数据

返回值：

0	成功
非 0	失败，返回错误代码

测试步骤：
a) 预先在密码设备中产生或导入 n（n 为厂家提供最大索引值）对 ECC 密钥对；
b) 创建（定义）会话句柄和设备句柄并初始化为零值；
c) 调用打开设备函数，获取设备句柄；
d) 调用创建会话函数，获取会话句柄；
e) 指定预处理数据长度；
f) 创建预处理和输入数据缓冲区（固定为 32 字节）及输出签名值数据结构；
g) 随机产生指定长度的预处理数据，并对其进行 SM2 签名预处理，得到 32 字节长度的 SM3 杂凑值作为外部输入数据；
h) 分别用 1 到 n（厂家提供最大索引值）索引值并创建输出签名值数据结构，依次调用本函数，调用返回值应为 0；
i) 调用关闭会话函数，确认函数返回值为零而且会话句柄值变为零；
j) 调用关闭设备函数，确认函数返回值为零而且设备句柄值变为零。

注：输入数据为待签数据的 SM3 杂凑值，是待签数据经过 SM2 签名预处理的结果，预处理过程见 GB/T 35276。本测试先于 6.5.5，使用的 ECC 密钥对公钥结构、输入数据（SM3 杂凑值）和输出的签名值数据（ECC 签名数据结构）应保留用于 6.5.5 外部密钥 ECC 验证以及 6.5.7 内部密钥 ECC 验证测试。当采用 GB/T 32918.5 中附录 A 的 SM2 椭圆曲线数字签名相关参考数据作为本测试内置数据及输入参数时，相应的输出数据和 GB/T 32918.5 中附录 A 的数字签名结果参考数据一致。

6.5.7 内部密钥 ECC 验证

原型：
```
LONG SDF_InternalVerify_ECC(
        HANDLE hSessionHandle,
        ULONG uiISKIndex,
        BYTE * pucData,
        ULONG uiDataLength,
        ECCSignature * pucSignature)
```

描述：使用内部 ECC 公钥对 ECC 签名值进行验证运算。

参数：

hSessionHandle[in]	与设备建立的会话句柄
uiISKIndex [in]	密码设备内部存储的 ECC 签名公钥的索引值
pucData[in]	缓冲区指针，用于存放外部输入的数据
uiDataLength[in]	输入的数据长度
pucSignature[in]	缓冲区指针，用于存放输入的签名值数据

返回值：

0	成功
非 0	失败，返回错误代码

测试步骤：

a) 创建(定义)会话句柄和设备句柄并初始化为零值；

b) 调用打开设备函数，获取设备句柄；

c) 调用创建会话函数，获取会话句柄；

d) 分别将 6.5.6 中输出的 1 到 n 个 ECC 签名值(ECC 签名数据结构)、对应的输入数据(SM3 杂凑值)及数据长度、对应签名私钥的 ECC 签名公钥索引(除特别注明，应和内部密钥 ECC 签名使用的内部私钥索引值相同)作为参数依次调用本函数，调用返回值应为 0；

e) 调用关闭会话函数，确认函数返回值为零而且会话句柄值变为零；

f) 调用关闭设备函数，确认函数返回值为零而且设备句柄值变为零。

注：输入数据为待签数据的杂凑值。当使用 SM2 算法时，该输入数据为待签数据经过 SM2 签名预处理的结果，预处理过程见 GB/T 35276。

6.5.8 外部密钥 ECC 加密

原型：
```
LONG SDF_ExternalEncrypt_ECC(
        HANDLE hSessionHandle,
        ULONG uiAlgID,
        ECCrefPublicKey * pucPublicKey,
        BYTE * pucData,
        ULONG uiDataLength,
        ECCCipher * pucEncData)
```

描述：使用外部 ECC 公钥对数据进行加密运算。

参数：

hSessionHandle[in]	与设备建立的会话句柄
uiAlgID[in]	算法标识，指定使用的 ECC 算法
pucPublicKey[in]	外部 ECC 公钥结构
pucData[in]	缓冲区指针，用于存放外部输入的数据
uiDataLength[in]	输入的数据长度
pucEncData[out]	缓冲区指针，用于存放输出的数据密文

返回值：　0　　成功

非 0　　失败，返回错误代码

测试步骤：a）　创建（定义）会话句柄和设备句柄并初始化为零值；

b）　调用打开设备函数，获取设备句柄；

c）　调用创建会话函数，获取会话句柄；

d）　指定算法 ID 为 SGD_SM2；

e）　指定外部 ECC 公钥；

f）　指定运算数据；

g）　指定数据长度；

h）　创建 ECC 加密数据结构作为输出数据缓冲区；

i）　将算法 ID，外部 ECC 公钥，运算数据和长度，输出数据缓冲区为参数调用本函数，调用返回值应为 0，输出的 ECC 加密数据结构应正确；

j）　调用关闭会话函数，确认函数返回值为零而且会话句柄值变为零；

k）　调用关闭设备函数，确认函数返回值为零而且设备句柄值变为零。

注：当采用 GB/T 32918.5 中附录 C 的 SM2 椭圆曲线加解密相关参考数据作为本测试输入参数时，相应的输出数据和 GB/T 32918.5 中附录 A 的加密结果参考数据一致。

6.6　对称算法运算类接口测试

6.6.1　测试项目

需测试的对称算法运算类函数接口如表 4 所示。

表 4　对称算法运算类函数

函数名称	功能
SDF_Encrypt	对称加密
SDF_Decrypt	对称解密
SDF_CalculateMAC	计算 MAC

6.6.2　对称加密

原型：
```
LONG SDF_Encrypt(
        HANDLE hSessionHandle,
        HANDLE hKeyHandle,
        ULONG uiAlgID,
        BYTE *pucIV,
        BYTE *pucData,
        ULONG uiDataLength,
        BYTE *pucEncData,
        ULONG *puiEncDataLength)
```

描述：使用指定的密钥句柄和 IV 对数据进行对称加密运算。

参数：hSessionHandle[in]　　与设备建立的会话句柄

hKeyHandle[in]　　指定的密钥句柄

uiAlgID[in]　　算法标识，指定对称加密算法

	pucⅣ[in|out]	缓冲区指针，用于存放输入和返回的Ⅳ数据
	pucData[in]	缓冲区指针，用于存放输入的数据明文
	uiDataLength[in]	输入的数据明文长度
	pucEncData[out]	缓冲区指针，用于存放输出的数据密文
	puiEncDataLength[out]	输出的数据密文长度
返回值：	0	成功
	非 0	失败，返回错误代码

测试步骤：

a) 以任一有效的内部密钥加密密钥索引值调用 6.4.19 生成会话密钥并用密钥加密密钥加密输出函数，保存该索引值以及输出的会话密钥密文；

b) 创建(定义)会话句柄和设备句柄并初始化为零值；

c) 调用打开设备函数，获取设备句柄；

d) 调用创建会话函数，获取会话句柄；

e) 以 a)中保存的密钥加密密钥索引值以及输出的会话密钥密文调用 6.4.20 导入会话密钥并用密钥加密密钥解密函数，获取会话密钥句柄；

f) 按照 GB/T 33560 的分组密码算法标识表和厂商申报支持算法逐一指定对称算法标识 ID；

g) 对应步骤 f)中指定的每种算法 ID，按照该算法规范随机产生规定长度的Ⅳ，并随机产生 $n(2<n<256)$块 1 个分组长度以上的输入数据，用步骤 e)中返回的会话密钥句柄并创建输出密文缓冲区，按照多包加密依次调用本函数 n 次(每次返回的Ⅳ作为下次输入的Ⅳ)，调用返回值应为 0；

h) 调用销毁会话密钥函数，销毁会话密钥；

i) 调用关闭会话函数，确认函数返回值为零而且会话句柄值变为零；

j) 调用关闭设备函数，确认函数返回值为零而且设备句柄值变为零。

注：此函数不对数据进行填充处理，输入的数据宜是指定算法分组长度的整数倍。返回的Ⅳ数据用于多包数据对称加密运算。本测试使用的密钥加密密钥索引值以及输出的会话密钥密文、Ⅳ向量(首包加密输入的Ⅳ)和输入的明文数据、输出的密文数据保留到 6.6.3 使用。当采用 GB/T 32907—2016 中附录 A 的示例相关参考数据作为本测试内置数据及输入参数时，相应的输出数据和 GB/T 32907—2016 中附录 A 的加密结果参考数据一致。

6.6.3 对称解密

原型：

```
LONG SDF_Decrypt (
        HANDLE hSessionHandle,
        HANDLE hKeyHandle,
        ULONG uiAlgID,
        BYTE *pucⅣ,
        BYTE *pucEncData,
        ULONG uiEncDataLength,
        BYTE *pucData,
        ULONG *puiDataLength)
```

描述：	使用指定的密钥句柄和Ⅳ对数据进行对称解密运算。	
参数：	hSessionHandle[in]	与设备建立的会话句柄
	hKeyHandle[in]	指定的密钥句柄
	uiAlgID[in]	算法标识，指定对称加密算法

	puc IV[in|out]	缓冲区指针，用于存放输入和返回的IV数据
	pucEncData[in]	缓冲区指针，用于存放输入的数据密文
	uiEncDataLength[in]	输入的数据密文长度
	pucData[out]	缓冲区指针，用于存放输出的数据明文
	puiDataLength[out]	输出的数据明文长度
返回值：	0	成功
	非0	失败，返回错误代码

测试步骤：

a) 创建(定义)会话句柄和设备句柄并初始化为零值；

b) 调用打开设备函数，获取设备句柄；

c) 调用创建会话函数，获取会话句柄；

d) 以6.6.2中保存的密钥加密密钥索引值以及输出的会话密钥密文调用6.4.20导入会话密钥并用密钥加密密钥解密函数，获取会话密钥句柄；

e) 分别以6.6.2中使用的各算法标识ID和保存的相应IV向量以及多包密文数据并创建对应明文数据缓冲区为参数，按照多包解密依次调用本函数 n 次(n 为6.6.2中对应每个算法ID的多包加密块数，每次返回的IV作为下次输入的IV)，返回值应为0，每次调用输出的明文数据和6.6.2中相应调用输入的明文数据进行比对，比对结果应相同；

f) 调用销毁会话密钥函数，销毁会话密钥；

g) 调用关闭会话函数，确认函数返回值为零而且会话句柄值变为零；

h) 调用关闭设备函数，确认函数返回值为零而且设备句柄值变为零。

注：此函数不对数据进行填充处理，输入的数据宜是指定算法分组长度的整数倍。返回的IV数据用于多包数据对称解密运算。

6.6.4 计算MAC

```
原型：  LONG SDF_CalculateMAC(
            HANDLE hSessionHandle,
            HANDLE hKeyHandle,
            ULONG uiAlgID,
            BYTE *pucIV,
            BYTE *pucData,
            ULONG uiDataLength,
            BYTE *pucMAC,
            ULONG *puiMACLength)
```

描述：	使用指定的密钥句柄和IV对数据进行MAC运算。	
参数：	hSessionHandle[in]	与设备建立的会话句柄
	hKeyHandle[in]	指定的密钥句柄
	uiAlgID[in]	算法标识，指定MAC加密算法
	pucIV[in|out]	缓冲区指针，用于存放输入和返回的IV数据
	pucData[in]	缓冲区指针，用于存放输入的数据明文
	uiDataLength[in]	输入的数据明文长度
	pucMAC[out]	缓冲区指针，用于存放输出的MAC值
	puiMACLength[out]	输出的MAC值长度
返回值：	0	成功

非 0　　失败，返回错误代码

测试步骤：
a) 以任一有效的内部密钥加密密钥索引值调用 6.4.19 生成会话密钥并用密钥加密密钥加密输出函数，保存该索引值以及输出的会话密钥密文；
b) 创建(定义)会话句柄和设备句柄并初始化为零值；
c) 调用打开设备函数，获取设备句柄；
d) 调用创建会话函数，获取会话句柄；
e) 以 a)中保存的密钥加密密钥索引值以及输出的会话密钥密文调用 6.4.20 导入会话密钥并用密钥加密密钥解密函数，获取会话密钥句柄；
f) 按照 GB/T 33560 的分组密码算法标识表和厂商申报支持算法逐一指定对称算法标识 ID；
g) 对应步骤 f)中指定的每种算法 ID，按照该算法规范随机产生规定长度的 IV，并随机产生 $n(2<n<256)$ 块 1 个分组长度以上的输入数据，用步骤 e)中返回的会话密钥句柄并创建输出 MAC 缓冲区，按照多包计算依次调用本函数 n 次(每次返回的 IV 作为下次输入的 IV)，调用返回值应为 0；
h) 调用销毁会话密钥函数，销毁会话密钥；
i) 调用关闭会话函数，确认函数返回值为零而且会话句柄值变为零；
j) 调用关闭设备函数，确认函数返回值为零而且设备句柄值变为零。

注：此函数不对数据进行分包处理，多包数据 MAC 运算由 IV 控制最后的 MAC 值。

6.7 杂凑运算类接口测试

6.7.1 测试项目

需测试的杂凑运算类函数接口如表 5 所示。

表 5 杂凑运算类函数

函数名称	功能
SDF_HashInit	杂凑运算初始化
SDF_HashUpdate	多包杂凑运算
SDF_HashFinal	杂凑运算结束

6.7.2 杂凑运算初始化

原型：
```
LONG SDF_HashInit(
        HANDLE hSessionHandle,
        ULONG uiAlgID
        ECCrefPublicKey * pucPublicKey,
        BYTE * pucID,
        ULONG uiIDLength)
```

描述：三步式数据杂凑运算第一步。

参数：

hSessionHandle[in]	与设备建立的会话句柄
uiAlgID[in]	指定杂凑算法标识
pucPublicKey[in]	签名者公钥。当 uiAlgID 为 SGD_SM3 时有效。
pucID[in]	签名者的 ID 值，当 uiAlgID 为 SGD_SM3 时有效。

	uiIDLength[in]	签名者 ID 的长度，当 uiAlgID 为 SGD_SM3 时有效。
返回值：	0	成功
	非 0	失败，返回错误代码

测试步骤：

a) 创建(定义)会话句柄和设备句柄并初始化为零值；
b) 调用打开设备函数，获取设备句柄；
c) 调用创建会话函数，获取会话句柄；
d) 按照 GB/T 33560 的杂凑算法标识表和厂商申报支持算法，指定杂凑算法标识 ID；
e) 当算法标识 ID 为 SGD_SM3 时，指定签名者公钥，当算法标识 ID 为其他值时，签名者公钥设为空并在计算时忽略；
f) 当算法标识 ID 为 SGD_SM3 时，指定签名者 ID，当算法标识 ID 为其他值时，签名者 ID 设为空并在计算时忽略；
g) 指定签名值 ID 长度(当算法标识 ID 为 SGD_SM3 时)；
h) 将算法标识 ID，签名者公钥，签名者 ID 及长度为参数调用本函数，调用返回值应为 0。

注：uiIDLength 非零且 uiAlgID 为 SGD_SM3 时，函数执行 SM2 的预处理 1 操作。计算过程见 GB/T 0009。本测试打开的会话句柄保存至 6.7.3 和 6.7.4 使用。针对每种杂凑算法 ID，需 6.7.2～6.7.4 三个测试连续完成，再从头进行其他算法 ID 的测试。

6.7.3 多包杂凑运算

```
原型：  LONG SDF_HashUpdate(
            HANDLE hSessionHandle,
            BYTE *pucData,
            ULONG uiDataLength)
```

描述：	三步式数据杂凑运算第二步，对输入的明文进行杂凑运算。	
参数：	hSessionHandle[in]	与设备建立的会话句柄
	pucData[in]	缓冲区指针，用于存放输入的数据明文
	uiDataLength[in]	输入的数据明文长度
返回值：	0	成功
	非 0	失败，返回错误代码

测试步骤：

a) 沿用 6.7.2 测试中打开的设备句柄和会话句柄；
b) 对应 6.7.2 步骤 d)中指定的算法 ID，随机产生 $n(2<n<256)$块 1 个分组长度以上的输入数据，按照多包计算依次调用本函数 n 次，调用返回值应为 0。

注：本测试在 6.7.2 的基础上完成。

6.7.4 杂凑运算结束

```
原型：  LONG SDF_HashFinal(
            HANDLE hSessionHandle,
            BYTE *pucHash,
            ULONG *puiHashLength)
```

描述：	三步式数据杂凑运算第三步，杂凑运算结束返回杂凑数据并清除中间数据。	
参数：	hSessionHandle[in]	与设备建立的会话句柄
	pucHash[out]	缓冲区指针，用于存放输出的杂凑数据

puiHashLength[out]　　返回的杂凑数据长度

返回值：　0　　成功

非 0　　失败，返回错误代码

测试步骤：a）　沿用 6.7.2 测试中打开的设备句柄和会话句柄；

b）　创建输出数据缓冲区；

c）　创建数据数据长度变量；

d）　将输出数据缓冲区及数据长度变量地址为参数调用本函数，调用返回值应为 0；

e）　调用关闭会话函数，确认函数返回值为零而且会话句柄值变为零；

f）　调用关闭设备函数，确认函数返回值为零而且设备句柄值变为零。

注：本测试在 6.7.2 和 6.7.3 的基础上完成。当采用 GB/T 32905—2016 中附录 A 示例 2 的消息作为 6.7.3 测试输入数据时，本测试的输出杂凑值和 GB/T 32905—2016 中附录 A 示例 2 的杂凑值参考数据一致。

6.8　用户文件操作类接口测试

6.8.1　测试项目

需测试的用户文件操作类函数接口如表 6 所示。

表 6　用户文件操作类函数

函数名称	功能
SDF_CreateFile	创建文件
SDF_ReadFile	读取文件
SDF_WriteFile	写文件
SDF_DeleteFile	删除文件

6.8.2　创建文件

原型：　LONG SDF_CreateFile(

HANDLE hSessionHandle，

LPSTR pucFileName，

ULONG uiNameLen，

ULONG uiFileSize)

描述：　在密码设备内部创建用于存储用户数据的文件。

参数：　hSessionHandle[in]　　与设备建立的会话句柄

pucFileName[in]　　缓冲区指针，用于存放输入的文件名，最大长度 128 字节

uiNameLen[in]　　文件名长度

uiFileSize[in]　　文件所占存储空间的长度

返回值：　0　　成功

非 0　　失败，返回错误代码

测试步骤：a）　创建(定义)会话句柄和设备句柄并初始化为零值；

b）　调用打开设备函数，获取设备句柄；

c）　调用创建会话函数，获取会话句柄；

d）　以最大长度 128 字节指定文件名；

e) 以设备提供商支持的最大文件长度指定文件所占存储空间大小；
f) 将文件名，文件名长度及占用存储空间大小为参数调用本函数，调用返回值应为 0；
g) 调用关闭会话函数，确认函数返回值为零而且会话句柄值变为零；
h) 调用关闭设备函数，确认函数返回值为零而且设备句柄值变为零。

6.8.3 读取文件

原型：
```
LONG SDF_ReadFile(
        HANDLE hSessionHandle,
        LPSTR pucFileName,
        ULONG uiNameLen,
        ULONG uiOffset,
        ULONG * puiFileLength,
        BYTE * pucBuffer)
```

描述：读取在密码设备内部存储用户数据的文件的内容。

参数：

hSessionHandle[in]	与设备建立的会话句柄
pucFileName[in]	缓冲区指针，用于存放输入的文件名，最大长度 128 字节
uiNameLen[in]	文件名长度
uiOffset[in]	指定读取文件时的偏移值
puiFileLength[in|out]	入参时指定读取文件内容的长度；出参时返回实际读取文件内容的长度
pucBuffer[out]	缓冲区指针，用于存放读取的文件数据

返回值：

0	成功
非 0	失败，返回错误代码

测试步骤：
a) 创建(定义)会话句柄和设备句柄并初始化为零值；
b) 调用打开设备函数，获取设备句柄；
c) 调用创建会话函数，获取会话句柄；
d) 以 6.8.2 中创建的文件名、该文件所占存储空间长度范围内任一随机数为偏移值、文件所占存储空间长度减去偏移值所得差值范围内任一随机数为读取长度并创建相应长度的缓冲区，调用本函数，调用返回值应为 0；
e) 调用关闭会话函数，确认函数返回值为零而且会话句柄值变为零；
f) 调用关闭设备函数，确认函数返回值为零而且设备句柄值变为零。

6.8.4 写文件

原型：
```
LONG SDF_WriteFile(
        HANDLE hSessionHandle,
        LPSTR pucFileName,
        ULONG uiNameLen,
        ULONG uiOffset,
        ULONG uiFileLength,
        BYTE * pucBuffer)
```

描述：向密码设备内部存储用户数据的文件中写入内容。

参数：

hSessionHandle[in]	与设备建立的会话句柄
pucFileName[in]	缓冲区指针，用于存放输入的文件名，最大长度 128 字节
uiNameLen[in]	文件名长度
uiOffset[in]	指定写入文件时的偏移值
uiFileLength[in]	指定写入文件内容的长度
pucBuffer[in]	缓冲区指针，用于存放输入的写文件数据

返回值：

0	成功
非 0	失败，返回错误代码

测试步骤：

a) 创建(定义)会话句柄和设备句柄并初始化为零值；

b) 调用打开设备函数，获取设备句柄；

c) 调用创建会话函数，获取会话句柄；

d) 以 6.8.2 中创建的文件名、该文件所占存储空间长度范围内任一随机数为偏移值、文件所占存储空间长度减去偏移值所得差值范围内任一随机数为写入长度并创建相应长度的缓冲区且用随机数填充该缓冲区，调用本函数，调用返回值应为 0；

e) 调用关闭会话函数，确认函数返回值为零而且会话句柄值变为零；

f) 调用关闭设备函数，确认函数返回值为零而且设备句柄值变为零。

6.8.5 删除文件

原型：

```
LONG SDF_DeleteFile(
        HANDLE hSessionHandle,
        LPSTR pucFileName,
        ULONG uiNameLen)
```

描述：删除指定文件名的密码设备内部存储用户数据的文件。

参数：

hSessionHandle[in]	与设备建立的会话句柄
pucFileName[in]	缓冲区指针，用于存放输入的文件名，最大长度128 字节
uiNameLen[in]	文件名长度

返回值：

0	成功
非 0	失败，返回错误代码

测试步骤：

a) 创建(定义)会话句柄和设备句柄并初始化为零值；

b) 调用打开设备函数，获取设备句柄；

c) 调用创建会话函数，获取会话句柄；

d) 以 6.8.2 中创建的文件名调用本函数，调用返回值应为 0；

e) 调用关闭会话函数，确认函数返回值为零而且会话句柄值变为零；

f) 调用关闭设备函数，确认函数返回值为零而且设备句柄值变为零。

6.9 接口稳定性测试

6.9.1 设备管理类接口测试

测试方法：

a) 重复执行打开设备和关闭设备函数调用组合若干次(建议为 1 000 次以上)，成功率 100%；

b) 在成功打开设备的前提下，重复执行创建会话和关闭会话函数调用组合若干次（建议为 1 000 次以上），成功率 100%，执行完毕后能够成功关闭设备；

c) 在成功打开设备并创建会话的前提下，重复执行获取设备信息函数调用组合若干次（建议为 1 000 次以上），成功率 100%，执行完毕后能够成功关闭会话和关闭设备；

d) 在成功打开设备并创建会话的前提下，以不同大小的数据缓冲区（32、64、128、256、512、1 024、2 048 字节）为参数调用产生随机数函数，重复执行（建议为 1 000 次以上），成功率 100%，执行完毕后能够成功关闭会话和关闭设备；

e) 在成功打开设备并创建会话的前提下，用厂家提供的标识码和从 1 到 n（厂家提供最大索引值）索引值依次调用获取私钥使用权限函数和释放私钥使用权限函数调用组合，重复执行若干次（建议为 1 000 次以上），成功率 100%，执行完毕后能够成功关闭会话和关闭设备。

6.9.2 密钥管理类接口测试

测试方法：

a) 在成功打开设备并创建会话的前提下，从 1 到 n（RSA 签名密钥对索引值，n 为厂家提供最大索引值）依次调用导出 RSA 签名公钥函数，重复执行若干次（建议为 1 000 次以上），成功率 100%，执行完毕后能够成功关闭会话和关闭设备。

b) 在成功打开设备并创建会话的前提下，从 1 到 n（RSA 加密密钥对索引值，n 为厂家提供最大索引值）依次调用导出 RSA 加密公钥函数，重复执行若干次（建议为 1 000 次以上），成功率 100%，执行完毕后能够成功关闭会话和关闭设备。

c) 在成功打开设备并创建会话的前提下，以 2 048 位为模长调用产生 RSA 密钥对并输出函数，重复执行若干次（建议为 1 000 次以上），成功率 100%，执行完毕后能够成功关闭会话和关闭设备。

d) 在成功打开设备并创建会话的前提下，分别以不同长度会话密钥（32、64、128、256、512、1 024、2 048 字节）从 1 到 n（内部 RSA 加密公钥索引值，n 为厂家提供最大索引值）依次调用生成会话密钥并用内部 RSA 公钥加密输出函数，重复执行若干次（建议为 1 000 次以上），成功率 100%，执行完毕后能够成功关闭会话和关闭设备。

e) 在成功打开设备并创建会话的前提下，分别以不同长度会话密钥（32、64、128、256、512、1 024、2 048 字节）和若干（建议 100 个以上）不同的外部 RSA 加密公钥调用生成会话密钥并用外部 RSA 公钥加密输出函数，重复执行若干次（建议为 1 000 次以上），成功率 100%，执行完毕后能够成功关闭会话和关闭设备。

f) 在成功打开设备、创建会话并获取私钥使用权限的前提下，分别以不同长度会话密钥（32、64、128、256、512、1 024、2 048 字节）从 1 到 n（内部 RSA 加密私钥索引值，n 为厂家提供最大索引值）依次调用导入会话密钥并用内部 RSA 私钥解密函数，重复执行若干次（建议为 1 000 次以上），成功率 100%，执行完毕后能够成功关闭会话和关闭设备。

g) 在成功打开设备并创建会话的前提下，分别以不同长度会话密钥（32、64、128、256、512、1 024、2 048 字节）和若干（建议 100 个以上）不同的外部 RSA 公钥从 1 到 n（内部 RSA 密钥对索引值，n 为厂家提供最大索引值）依次调用基于 RSA 算法的数字信封转换函数，重复执行若干次（建议为 1 000 次以上），成功率 100%，执行完毕后能够成功关闭会话和关闭设备。

h) 在成功打开设备并创建会话的前提下，从 1 到 n（ECC 签名密钥对索引值，n 为厂家提供最大索引值）依次调用导出 ECC 签名公钥函数，重复执行若干次（建议为 1 000 次以上），成功率 100%，执行完毕后能够成功关闭会话和关闭设备。

i) 在成功打开设备并创建会话的前提下，从 1 到 n（ECC 加密密钥对索引值，n 为厂家提供最大索引值）依次调用导出 ECC 加密公钥函数，重复执行若干次（建议为 1 000 次以上），成功率

100%，执行完毕后能够成功关闭会话和关闭设备。

j) 在成功打开设备并创建会话的前提下，调用产生 ECC 密钥对并输出函数，重复执行若干次（建议为 1 000 次以上），成功率 100%，执行完毕后能够成功关闭会话和关闭设备。

k) 在成功打开设备并创建会话的前提下，分别以不同长度会话密钥（32、64、128、256、512、1 024、2 048 字节）从 1 到 n（内部 ECC 加密公钥索引值，n 为厂家提供最大索引值）依次调用生成会话密钥并用内部 ECC 公钥加密输出函数，重复执行若干次（建议为 1 000 次以上），成功率 100%，执行完毕后能够成功关闭会话和关闭设备。

l) 在成功打开设备并创建会话的前提下，分别以不同长度会话密钥（32、64、128、256、512、1 024、2 048 字节）、不同算法标识（见 GB/T 33560）和若干（建议 100 个以上）不同的外部 ECC 公钥调用生成会话密钥并用外部 ECC 公钥加密输出函数，重复执行若干次（建议为 1 000 次以上），成功率 100%，执行完毕后能够成功关闭会话和关闭设备。

m) 在成功打开设备、创建会话并获取私钥使用权限的前提下，分别以不同长度会话密钥（32、64、128、256、512、1 024、2 048 字节）生成 ECC 加密数据结构并从 1 到 n（内部 ECC 加密私钥索引值，n 为厂家提供最大索引值）依次调用导入会话密钥并用内部 ECC 私钥解密函数，重复执行若干次（建议为 1 000 次以上），成功率 100%，执行完毕后能够成功关闭会话和关闭设备。

n) 在成功打开设备、创建会话并获取私钥使用权限的前提下，生成若干对随机数（建议 100 以上）分别作为发起方 ID 和响应方 ID，对每对随机数执行以下调用，成功率 100%，执行完毕后能够成功关闭会话和关闭设备：

   1) 用发起方 ID 分别以不同长度会话密钥（32、64、128、256、512、1 024、2 048 字节）从 1 到 n（内部 ECC 加密私钥索引值，n 为厂家提供最大索引值）依次调用生成密钥协商参数并输出函数，得到相应的发起方协商句柄、发起方公钥（和加密私钥索引值对应）和发起方临时公钥。

   2) 分别用步骤 1）中生成的每对发起方公钥（和加密私钥索引值对应）和发起方临时公钥、发起方 ID、响应方 ID、指定内部 ECC 加密私钥索引值（步骤 1 中索引值+1 后对 n 取模）调用产生协商数据并计算会话密钥函数得到相应的响应方密钥句柄、响应方公钥（和加密私钥索引值对应）和响应方临时公钥。

   3) 用步骤 2）中生成的每对响应方公钥（和加密私钥索引值对应）和响应方临时公钥、响应方 ID 以及所对应步骤 1）中的发起方协商句柄调用计算会话密钥函数，得到发起方密钥句柄。

6.9.3 非对称算法运算类接口测试

测试方法：

a) 在成功打开设备并创建会话的前提下，按照 6.5.4 的测试步骤调用内部私钥 RSA 运算函数，重复执行若干次（建议为 1 000 次以上），每次随机使用不同的数据缓冲区长度，成功率 100%，执行完毕后能够成功关闭会话和关闭设备。

b) 在成功打开设备并创建会话的前提下，对照步骤 a）测试中输出的 RSA 私钥运算数据以及相应的 RSA 内部公钥索引值，按照 6.5.3 的测试步骤调用内部公钥 RSA 运算函数，重复执行若干次（建议为 1 000 次以上），成功率 100%，执行完毕后能够成功关闭会话和关闭设备。

c) 在成功打开设备并创建会话的前提下，对照步骤 a）测试中输出的 RSA 私钥运算数据以及相应的 RSA 公钥，按照 6.5.2 的测试步骤调用外部公钥 RSA 运算函数，重复执行若干次（建议为 1 000 次以上），成功率 100%，执行完毕后能够成功关闭会话和关闭设备。

d) 在成功打开设备并创建会话的前提下，按照 6.5.6 的测试步骤调用内部密钥 ECC 签名运算函数，重复执行若干次（建议为 1 000 次以上），每次随机使用不同的数据缓冲区长度，成功率

100%，执行完毕后能够成功关闭会话和关闭设备。

e) 在成功打开设备并创建会话的前提下，对照步骤 d)测试中输出的签名数据(签名值和杂凑值)和相应的内部 ECC 密钥索引，按照 6.5.7 的测试步骤调用内部密钥 ECC 验证运算函数，重复执行若干次(建议为 1 000 次以上)，成功率 100%，执行完毕后能够成功关闭会话和关闭设备。

f) 在成功打开设备并创建会话的前提下，对照步骤 d)测试中输出的签名数据(签名值和杂凑值)和相应的 ECC 公钥结构，按照 6.5.5 的测试步骤调用外部密钥 ECC 验证运算函数，重复执行若干次(建议为 1 000 次以上)，成功率 100%，执行完毕后能够成功关闭会话和关闭设备。

g) 在成功打开设备并创建会话的前提下，按照 6.5.8 的测试步骤调用外部密钥 ECC 加密运算函数，重复执行若干次(建议为 1 000 次以上)，每次随机使用不同的数据缓冲区长度，成功率 100%，执行完毕后能够成功关闭会话和关闭设备。

6.9.4 对称算法运算类接口测试

测试方法：

a) 在成功打开设备并创建会话的前提下，按照 6.6.2 的测试步骤调用对称加密运算函数，重复执行若干次(建议为 1 000 次以上)，成功率 100%，执行完毕后能够成功关闭会话和关闭设备。

b) 在成功打开设备并创建会话的前提下，对照步骤 a)每次测试中输出的密文数据和相应的会话密钥，按照 6.6.3 的测试步骤调用对称解密运算函数，成功率 100%，执行完毕后能够成功关闭会话和关闭设备。

c) 在成功打开设备并创建会话的前提下，按照 6.6.4 的测试步骤调用计算 MAC 运算函数，重复执行若干次(建议为 1 000 次以上)，成功率 100%，执行完毕后能够成功关闭会话和关闭设备。

6.9.5 杂凑运算类接口测试

测试方法：在成功打开设备并创建会话的前提下，对应于每种杂凑算法 ID，按照 6.7.2～6.7.4 的测试步骤连续调用三步式数据杂凑运算的完整三个步骤，全程重复执行若干次(建议为 1 000 次以上)，当算法标识 ID 为 SGD_SM3 时，每次使用不同的外部 ECC 公钥，成功率 100%，执行完毕后能够成功关闭会话和关闭设备。

6.9.6 用户文件操作类接口测试

测试方法：

a) 在成功打开设备并创建会话的前提下，按照 6.8.2 的测试步骤调用创建文件函数，重复执行若干次(建议为 1 000 次以上或设备提供商支持的最大文件数)，每次使用 128 字节以内随机产生的不同文件名，以及最大文件长度范围内随机长度的文件存储空间，成功率 100%，执行完毕后能够成功关闭会话和关闭设备。

b) 在成功打开设备并创建会话的前提下，对于步骤 a)中创建的每个文件，按照 6.8.3 的测试步骤调用读取文件函数，重复执行若干次(建议为 1 000 次以上)，成功率 100%，执行完毕后能够成功关闭会话和关闭设备。

c) 在成功打开设备并创建会话的前提下，对于步骤 a)中创建的每个文件，按照 6.8.4 的测试步骤调用写文件函数，重复执行若干次(建议为 1 000 次以上)，成功率 100%，执行完毕后能够成功关闭会话和关闭设备。

d) 在成功打开设备并创建会话的前提下，对于步骤 a)中创建的每个文件，按照 6.8.5 的测试步骤调用删除文件函数，成功率 100%，执行完毕后能够成功关闭会话和关闭设备。

6.10 边界和异常条件测试

测试方法：

构造以下边界和异常条件，并进行相应接口函数的调用，应按照 GB/T 36322—2018 附录 A 的要求返回对应的错误代码：

a) 执行 GB/T 36322—2018 所定义的函数接口和设备提供商声明之外的函数调用，应返回“不支持的接口调用”错误代码；

b) 断开检测服务器与密码设备之间的通信信道并执行任意函数调用，应返回“与设备通信失败”错误代码；

c) 使密码设备内部的密码运算部件处于不可用状态，执行任意密码运算函数调用，应返回“运算模块无响应”错误代码；

d) 使密码设备处于不可用状态，执行“打开设备”调用，应返回“打开设备失败”错误代码；

e) 执行“打开设备”调用成功之后，使密码设备处于不可用状态，执行“创建会话”调用，应返回“创建会话失败”错误代码；

f) 在不执行“获取私钥使用权限”调用的情况下，调用签名或内部私钥运算等私钥相关函数调用，应返回“无私钥使用权限”错误代码；

g) 使用没有创建或已经释放的密钥索引或密钥句柄调用密钥相关函数，应返回“不支持的密钥调用”错误代码；

h) 使用 GB/T 33560 所定义范围之外的且未经厂家申明的算法标识调用密码运算相关函数，应返回“不支持的算法调用”错误代码；

i) 使用 GB/T 33560 所定义范围之外且未经厂家申明的的算法模式调用密码运算相关函数，应返回“不支持的算法模式调用”错误代码；

j) 使密码设备公钥运算功能处于不可用状态，调用内部/外部公钥 RSA 运算和外部密钥 ECC 加密等公钥运算函数，应返回“公钥运算失败”错误代码；

k) 使密码设备私钥运算功能处于不可用状态，调用内部私钥 RSA 运算等私钥运算函数，应返回“私钥运算失败”错误代码；

l) 使密码设备签名运算功能处于不可用状态，调用内部密钥 ECC 签名等签名运算函数，应返回“签名运算失败”错误代码；

m) 使密码设备验证签名功能处于不可用状态，调用内部密钥 ECC 验证等验证签名运算函数，应返回“验证签名失败”错误代码；

n) 使密码设备对称算法运算功能处于不可用状态，调用对称加密和对称解密等对称算法运算函数，应返回“对称算法运算失败”错误代码；

o) 按照错误的步骤调用多包对称加解密、多包计算 MAC、多包杂凑等多包密码运算函数，应返回“多步运算步骤错误” 代码；

p) 以超出设备提供商支持的最大文件长度作为长度参数调用用户文件操作类函数，应返回“文件长度超出限制”错误代码；

q) 以不存在的文件路径或文件名调用读写和删除文件函数，应返回“指定的文件不存在”错误代码；

r) 以错误的文件偏移位置作为参数调用读写和删除文件函数，应返回“文件起始位置错误”代码；

s) 以 ECC 密钥索引号为参数调用内部 RSA 公私钥运算或者以 RSA 密钥索引号为参数调用内部 ECC 公私钥运算，应返回“密钥类型错误”代码；

t) 以不符合随机性要求的密钥（如全“0”或全“1”等），用内部 ECC 或 RSA 私钥对应的公钥加密后作为参数执行导入会话密钥并用内部 ECC 或 RSA 私钥解密函数，应返回“密钥错误”代码；

u) 按 256 位长（ECC）或 2 048 位长（RSA）产生随机数填写 ECC 或 RSA 公钥结构并调用外部 ECC 或 RSA 公钥相关函数，应返回“密钥错误”代码；

v) 按 256 位长（ECC）产生随机数作为密钥，按 2 048 位长产生随机数作为密文数据，按 256 位长

产生随机数作为杂凑值，填写 ECC 加密数据结构并调用导入会话密钥并用 ECC 内部私钥解密函数，应返回“ECC 加密数据错误”代码；

w) 使密码设备产生随机数功能处于不可用状态，调用产生 RSA/ECC 密钥对或生成会话密钥相关函数，应返回“随机数产生失败”错误代码；

x) 使密码设备计算 MAC 功能处于不可用状态，调用计算 MAC 函数，应返回“MAC 运算失败”错误代码；

y) 以已经存在的文件路径调用创建文件函数，应返回“指定文件已存在”错误代码；

z) 以超出文件长度的偏移位置作为参数调用写文件函数，应返回“文件写入失败”错误代码；

aa) 以超出设备提供商支持的最大存储空间长度作为文件长度调用创建文件函数，应返回“存储空间不足”错误代码；

bb) 以 NULL 值作为输入缓冲区指针参数或相应输入的数据结构指针参数调用密钥导入导出类、对称算法运算类、非对称算法运算类函数，应返回“输入参数错误”代码；

cc) 以 NULL 值作为输出缓冲区指针参数或相应输出的数据结构指针参数调用密钥导入导出类、对称算法运算类、非对称算法运算类函数，应返回“输出参数错误”代码。

6.11 接口安全性测试

测试方法：

通过工具软件对接口程序源代码进行静态检查并对源代码编译生成的库文件进行运行时的动态检查和实时监控，确认该接口库不存在以下安全问题：

a) 缓冲区溢出和明显的内存泄露；

b) 对系统目录、系统库文件和系统配置文件的非授权访问；

c) 对特定权限系统函数的非授权调用；

d) 对用户和用户组权限以及用户当前路径的非授权改变；

e) 存在未经许可和未声明的在所调用密码设备之外的网络流量；

f) 存在未经许可和未声明的在所调用密码模块之外的总线流量；

g) 存在未经许可和未申明的对文件和设备的访问；

h) 存在 GB/T 36322—2018 所允许范围之外的明文密钥；

i) 存在 GB/T 36322—2018 接口函数之外的数据获取途径；

j) 存在未经许可和未申明的在 GB/T 36322—2018 之外的接口函数；

k) 所使用过的存储空间(包括持久性和易失性存储空间)留存有敏感信息；

l) 未关闭或限制程序崩溃时内存数据到磁盘数据的自动保存(core 文件)。

6.12 接口库卸载测试

测试方法：

通过工具软件对接口库文件运行时的实时监控，以及库文件运行前后系统资源的变化情况，确认该接口库不会对运行环境造成大的影响：

a) 接口库卸载之后，所占用的内存和 CPU 资源能够在设定时间内释放；

b) 接口库卸载之后，相关网络流量清零，已建立网络连接能够在设定时间内释放；

c) 接口库卸载之后，所占用的文件描述符、网络套接字、进程/线程描述符能够在设定时间内释放；

d) 接口库卸载之后，所创建临时文件能够在设定时间内清零并完全删除；

e) 接口库卸载之后，在所使用的内存空间、存储介质无代码和数据残留。

7 送检文档技术要求

送检单位按照国家密码管理主管部门检测要求提交相关文档资料，作为密码设备应用接口的检测依据。文档资料应包含但不限于以下内容：

a) 应用编程接口的结构框图、流程图和基本功能的源代码；
b) 应用编程接口的工作原理说明；
c) 自测程序的工作原理说明；
d) 敏感数据信息的存储和使用说明；
e) 应用编程接口库使用说明。

8 合格判定

6.3～6.11 中除 6.4.21～6.4.28 以外的任意一项检测不通过，判定为不符合 GB/T 36322—2018。

参 考 文 献

[1] GB/T 17901.1—1999 信息技术 安全技术 密钥管理 第1部分:框架

[2] GB/T 17903.2—2008 信息技术 安全技术 抗抵赖 第2部分:采用对称技术的机制

[3] GB/T 17903.3—2008 信息技术 安全技术 抗抵赖 第3部分:采用非对称技术的机制

[4] GB/T 17964—2008 信息安全技术 分组密码算法的工作模式

[5] GB/T 18238.1—2000 信息技术 安全技术 散列函数 第1部分:概述

[6] GB/T 18238.2—2002 信息技术 安全技术 散列函数 第2部分:采用 n 位块密码的散列函数

[7] GB/T 18238.3—2002 信息技术 安全技术 散列函数 第3部分:专用散列函数

[8] PKCS #1: RSA Cryptography Standard RSA 密码标准

中华人民共和国密码
行业标准
密码设备应用接口符合性检测规范
GM/T 0102—2020
*
中国标准出版社出版发行
北京市朝阳区和平里西街甲2号(100029)
北京市西城区三里河北街16号(100045)
网址 www.spc.net.cn
总编室:(010)68533533 发行中心:(010)51780238
读者服务部:(010)68523946
中国标准出版社秦皇岛印刷厂印刷
各地新华书店经销
*
开本 880×1230 1/16 印张 3.5 字数 104 千字
2021年5月第一版 2021年5月第一次印刷
*
书号:155066·2-35845 定价 58.00 元

GM/T 0102-2020

码上扫一扫 正版服务到