ICS 35.040
L 80
备案号：

GM

中华人民共和国密码行业标准

GM/T 0018—2012

密码设备应用接口规范

Interface specifications of cryptography device application

2012-11-22 发布　　2012-11-22 实施

国家密码管理局　　发 布

中华人民共和国密码
行业标准
密码设备应用接口规范
GM/T 0018—2012
*
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(100013)
北京市西城区三里河北街16号(100045)
网址 www.spc.net.cn
总编室:(010)64275323 发行中心:(010)51780235
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 0.00 字数 00 千字
2013年 月第一版 2013年 月第一次印刷
*
书号:155066·2-0019 定价 00.00 元

GM/T 0018-2012

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。

本标准中的附录 A 为规范性附录。

本标准由国家密码管理局提出并归口。

本标准起草单位：卫士通信息产业股份有限公司、无锡江南信息安全工程技术中心、上海格尔软件股份有限公司、北京数字认证股份有限公司、兴唐通信科技股份有限公司、山东得安计算机技术有限公司、海泰方圆科技有限公司。

本标准主要起草人：刘平、李元正、徐强、谭武征、李述胜、李玉峰、高志权、柳增寿。

引　言

本标准的目标是为公钥密码基础设施应用体系框架下的服务类密码设备制定统一的应用接口标准，通过该接口调用密码设备，向上层提供基础密码服务。为该类密码设备的开发、使用及检测提供标准依据和指导，有利于提高该类密码设备的产品化、标准化和系列化水平。

本标准凡涉及密码算法相关内容，按国家有关法规实施。

密码设备应用接口规范

1 范围

本标准规定了公钥密码基础设施应用技术体系下服务类密码设备的应用接口标准。

本标准适用于服务类密码设备的研制、使用，以及基于该类密码设备的应用开发，也可用于指导该类密码设备的检测。

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。

GM/T 0006 密码应用标识规范

GM/T AAAA SM2 密码算法使用规范

3 术语和定义

以下术语和定义适用于本标准。

3.1

算法标识 algorithm identifier

用于对密码算法进行唯一标识的符号。

3.2

非对称密码算法/公钥密码算法 asymmetric cryptographic algorithm/public key cryptographic algorithm

加解密使用不同密钥的密码算法。

3.3

解密 decipherment/decryption

加密过程对应的逆过程。

3.4

设备密钥 device key pair

存储在设备内部的用于设备管理的非对称密钥对，包含签名密钥对和加密密钥对。

3.5

加密 encipherment/encryption

对数据进行密码变换以产生密文的过程。

3.6

密钥加密密钥 key encryption key；KEK

对密钥进行加密保护的密钥。

3.7

公钥基础设施 public key infrastructure；PKI

用公钥密码技术建立的普遍适用的基础设施，为用户提供证书管理和密钥管理等安全服务。

3.8

私钥访问控制码 private key access password

用于验证私钥使用权限的口令字。

3.9

对称密码技术/对称密码体制　symmetric cryptographic technique

原发者和接收者均采用同一秘密密钥进行变换的密码技术(体制)。其中,加密密钥与解密密钥相同,或者一个密钥可以从另一个密钥导出的密码体制。

3.10

会话密钥　session key

处于层次化密钥结构中的最低层,仅在一次会话中使用的密钥。

3.11

用户密钥　user key

存储在设备内部的用于应用密码运算的非对称密钥,包含签名密钥对和加密密钥对。

4　符号和缩略语

下列缩略语适用于本部分:

ECC　椭圆曲线算法(Elliptic Curve Cryptography)

IPK　内部加密公钥(Internal Public Key)

ISK　内部加密私钥(Internal Private Key)

EPK　外部加密公钥(External Public Key)

KEK　密钥加密密钥(Key Encrypt Key)

5　算法标识和数据结构

5.1　算法标识定义

本标准中所使用的算法其算法标识见 GM/T 0006。

5.2　设备信息定义

字段名称	数据长度(字节)	含　义
IssuerName	40	设备生产厂商名称
DeviceName	16	设备型号
DeviceSerial	16	设备编号,包含:日期(8 字符)、批次号(3 字符)、流水号(5 字符)
DeviceVersion	4	密码设备内部软件的版本号
StandardVersion	4	密码设备支持的接口规范版本号
AsymAlgAbility	8	前 4 字节表示支持的算法,表示方法为非对称算法标识按位或的结果;后 4 字节表示算法的最大模长,表示方法为支持的模长按位或的结果
SymAlgAbility	4	所有支持的对称算法,表示方法为对称算法标识按位或运算结果
HashAlgAbility	4	所有支持的杂凑算法,表示方法为杂凑算法标识按位或运算结果
BufferSize	4	支持的最大文件存储空间(单位字节)

实际数据结构定义：

```
typedef struct DeviceInfo_st{
    unsigned char IssuerName[40];
    unsigned char DeviceName[16];
    unsigned char DeviceSerial[16]
    unsigned int DeviceVersion;
    unsigned int StandardVersion;
    unsigned int AsymAlgAbility[2];
    unsigned int SymAlgAbility;
    unsigned int HashAlgAbility;
    unsigned int BufferSize;
}DEVICEINFO;
```

5.3 密钥分类及存储定义

5.3.1 设备密钥与用户密钥

设备密钥只能在设备初始化时生成或安装，用户密钥通过密码设备管理工具生成或安装。

设备密钥和用户密钥存放于密钥存储区，索引号从 0 开始检索，每个索引号对应一个签名密钥对和一个加密密钥对。其中，索引号为 0 表示设备密钥。索引号 1 开始表示用户密钥。

密钥对索引号	公钥	私　钥
0x00	设备签名公钥	设备签名私钥
	设备加密公钥	设备加密私钥
0x01	用户签名公钥	用户签名私钥
	用户加密公钥	用户加密私钥
……	……	……
	……	……

5.3.2 密钥加密密钥

密钥加密密钥通过密码设备管理工具生成或安装，密钥长度为 128 位，存放于密钥存储区，使用索引号从 1 开始。

密钥索引号	密钥加密密钥
0x01	密钥加密密钥 001
……	……

5.3.3 会话密钥

会话密钥使用设备接口函数生成或导入，会话密钥使用句柄检索。

5.4 RSA 密钥数据结构定义

RSA 密钥结构存储时顺序为从高到低，即密钥存放时从密钥结构数组的最高位开始，最高字节填在最高位，不足位填充数据 0。

公钥数据结构定义		
字段名称	数据长度(字节)	含义
bits	4	模长
M	256	模 N
E	256	公钥指数
私钥数据结构定义		
字段名称	数据长度	含义
bits	4	模长
M	256	模 N
E	256	公钥指数
D	256	私钥指数
prime[2]	128 * 2	素数 p 和 q
pexp[2]	128 * 2	Dp 和 Dq
coef	128	系数 i

实际数据结构定义：

```
#define RSAref_MAX_BITS      2048
#define RSAref_MAX_LEN       ((RSAref_MAX_BITS + 7) / 8)
#define RSAref_MAX_PBITS     ((RSAref_MAX_BITS + 1) / 2)
#define RSAref_MAX_PLEN      ((RSAref_MAX_PBITS + 7)/ 8)
typedef struct RSArefPublicKey_st
{
    unsigned int bits;
    unsigned char m[RSAref_MAX_LEN];
    unsigned char e[RSAref_MAX_LEN];
} RSArefPublicKey;

typedef struct RSArefPrivateKey_st
{
    unsigned int bits;
    unsigned char m[RSAref_MAX_LEN];
    unsigned char e[RSAref_MAX_LEN];
    unsigned char d[RSAref_MAX_LEN];
    unsigned char prime[2][RSAref_MAX_PLEN];
    unsigned char pexp[2][RSAref_MAX_PLEN];
    unsigned char coef[RSAref_MAX_PLEN];
} RSArefPrivateKey;
```

5.5 ECC 密钥数据结构定义

公钥数据结构定义		
字段名称	数据长度(字节)	含义
bits	4	密钥位长
x	ECCref_MAX_LEN	公钥 x 坐标
y	ECCref_MAX_LEN	公钥 y 坐标
私钥数据结构定义		
字段名称	数据长度(字节)	含义
bits	4	密钥位长
	ECCref_MAX_LEN	私钥

实际数据结构定义：

```
#define ECCref_MAX_BITS          512
#define ECCref_MAX_LEN           ((ECCref_MAX_BITS+7) / 8)
typedef struct ECCrefPublicKey_st
{
    unsigned int bits;
    unsigned char x[ECCref_MAX_LEN];
    unsigned char y[ECCref_MAX_LEN];
} ECCrefPublicKey;

typedef struct ECCrefPrivateKey_st
{
    unsigned int bits;
    unsigned char K[ECCref_MAX_LEN];
} ECCrefPrivateKey;
```

5.6 ECC 加密数据结构定义

加密数据结构定义		
字段名称	数据长度(字节)	含义
x	ECCref_MAX_LENP	X 分量
y	ECCref_MAX_LEN	Y 分量
M	32	明文的 SM3 杂凑值
L	4	密文数据长度
C	L	密文数据

实际数据结构定义：

```
typedef struct ECCCipher_st
{
    unsigned char x[ECCref_MAX_LEN];
```

```
    unsigned char y[ECCref_MAX_LEN];
    unsigned char M[32];
    unsigned int L;
    unsigned char C[1];
} ECCCipher;
```

5.7 ECC 签名数据结构定义

签名数据结构定义		
字段名称	数据长度(字节)	含义
r	ECCref_MAX_LEN	签名的 r 部分
s	ECCref_MAX_LEN	签名的 s 部分

实际数据结构定义：

```
typedef struct ECCSignature_st
{
    unsigned char r[ECCref_MAX_LEN];
    unsigned char s[ECCref_MAX_LEN];
} ECCSignature;
```

5.8 ECC 加密密钥对保护结构

密钥管理系统下发到设备中的 ECC 加密密钥对使用本结构表示。

(1) 类型定义

```
typedef struct SDF_ENVELOPEDKEYBLOB{
    unsigned long ulAsymmAlgID;
    unsigned long ulSymmAlgID;
    ECCCIPHERBLOB ECCCipherBlob;
    ECCPUBLICKEYBLOB PubKey;
    unsigned char cbEncryptedPriKey[64];
}ENVELOPEDKEYBLOB, * PENVELOPEDKEYBLOB;
```

(2) 数据项描述：

数据项	类型	意义	备注
ulAsymmAlgID	unsigned long	保护对称密钥的非对称算法标识	
ulSymmAlgID	unsigned long	对称算法标识	必须为 ECB 模式
ECCCipherBlob	ECCCIPHERBLOB	对称密钥密文	
PubKey	ECCPUBLICKEYBLOB	加密密钥对的公钥	
cbEncryptedPrivKey	无符号数组	加密密钥对的私钥密文,其有效长度为原文的(ulBits + 7)/8	私钥原文为 ECCPRIVATEKEYBLOB 结构中的 PrivateKey

6 设备接口描述

6.1 密码设备应用接口在公钥密码基础设施应用技术体系框架中的位置

在公钥密码基础设施应用技术体系框架中，密码设备服务层由密码机、密码卡、智能密码终端等设备组成，通过本标准规定的密码设备应用接口向通用密码服务层提供基础密码服务。如图1所示。

基础密码服务包括密钥生成、单一的密码运算、文件管理等的服务。

本标准采用C语言描述接口函数。如无特别说明，函数中参数的长度单位均为字节数。

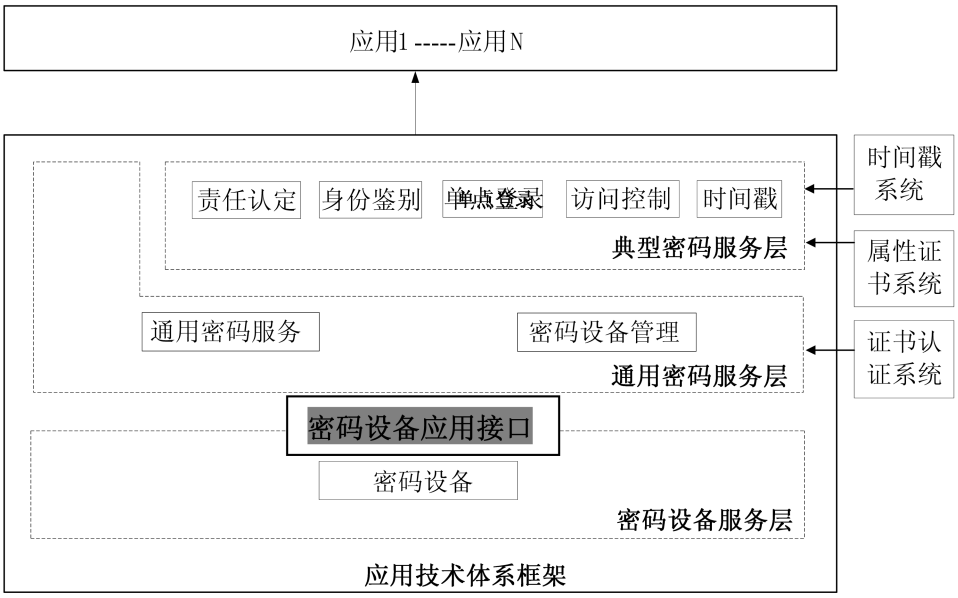

图1 密码设备应用接口在公钥密码基础设施应用技术体系结构中的位置

6.2 设备管理类函数

设备管理类函数包括以下具体函数，各函数返回值见附录A函数返回代码定义：

A. 打开设备：SDF_OpenDevice
B. 关闭设备：SDF_CloseDevice
C. 创建会话：SDF_OpenSession
D. 关闭会话：SDF_CloseSession
E. 获取设备信息：SDF_GetDeviceInfo
F. 产生随机数：SDF_GenerateRandom
G. 获取私钥使用权限：SDF_GetPrivateKeyAccessRight
H. 释放私钥使用权限：SDF_ReleasePrivateKeyAccessRight

6.2.1 打开设备

原型：	int SDF_OpenDevice(void * * phDeviceHandle);	
描述：	打开密码设备	
参数：	phDeviceHandle[out]	返回设备句柄
返回值：	0	成功
	非0	失败，返回错误代码

备注：　phDeviceHandle 由函数初始化并填写内容

6.2.2 关闭设备

原型：　int SDF_CloseDevice(void * 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
返回值：　0　成功
　　非 0　失败，返回错误代码

6.2.3 创建会话

原型：　int SDF_OpenSession(void * hDeviceHandle,void * * 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
返回值：　0　成功
　　非 0　失败，返回错误代码

6.2.4 关闭会话

原型：　int SDF_CloseSession(void * 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
返回值：　0　成功
　　非 0　失败，返回错误代码

6.2.5 获取设备信息

原型：　int SDF_GetDeviceInfo
　　(void * 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
返回值：　0　成功
　　非 0　失败，返回错误代码

6.2.6 产生随机数

原型：　int SDF_GenerateRandom (
　　void * hSessionHandle,
　　unsigned int uiLength,
　　unsigned char * 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

返回值： 0 成功
非 0 失败，返回错误代码

6.2.7 获取私钥使用权限

原型： int SDF_GetPrivateKeyAccessRight (
void * hSessionHandle,
unsigned int uiKeyIndex,
unsigned char * pucPassword,
unsigned int 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
返回值： 0 成功
非 0 失败，返回错误代码
备注： 本标准涉及密码设备存储的密钥对索引值的的起始索引值为 1，最大为 n，密码设备的实际存储容量决定 n 值，

6.2.8 释放私钥使用权限

原型： int SDF_ReleasePrivateKeyAccessRight (
void * hSessionHandle,
unsigned int 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
返回值： 0 成功
非 0 失败，返回错误代码

6.3 密钥管理类函数

密钥管理类函数包括以下具体函数，各函数返回值见附录 A 函数返回代码定义：

A. 导出 RSA 签名公钥：SDF_ExportSignPublicKey_RSA
B. 导出 RSA 加密公钥：SDF_ExportEncPublicKey_RSA
C. 产生 RSA 非对称密钥对并输出：SDF_GenerateKeyPair_RSA
D. 生成会话密钥并用内部 RSA 公钥加密输出：SDF_GenerateKeyWithIPK_RSA
E. 生成会话密钥并用外部 RSA 公钥加密输出：SDF_GenerateKeyWithEPK_RSA
F. 导入会话密钥并用内部 RSA 私钥解密：SDF_ImportKeyWithISK_RSA
G. 基于 RSA 算法的数字信封转换：SDF_ExchangeDigitEnvelopeBaseOnRSA
H. 导出 ECC 签名公钥：SDF_ExportSignPublicKey_ECC
I. 导出 ECC 加密公钥：SDF_ExportEncPublicKey_ECC
J. 产生 ECC 非对称密钥对并输出：SDF_GenerateKeyPair_ECC
K. 生成会话密钥并用内部 ECC 公钥加密输出：SDF_GenerateKeyWithIPK_ECC
L. 生成会话密钥并用外部 ECC 公钥加密输出：SDF_GenerateKeyWithEPK_ECC

M. 导入会话密钥并用内部 ECC 私钥解密:SDF_ImportKeyWithISK_ECC
N. 生成密钥协商参数并输出:SDF_GenerateAgreementDataWithECC
O. 计算会话密钥:SDF_GenerateKeyWithECC
P. 产生协商数据并计算会话密钥:SDF_GenerateAgreementDataAndKeyWithECC
Q. 基于 ECC 算法的数字信封转换:SDF_ExchangeDigitEnvelopeBaseOnECC
R. 生成会话密钥并用密钥加密密钥加密输出:SDF_GenerateKeyWithKEK
S. 导入会话密钥并用密钥加密密钥解密:SDF_ImportKeyWithKEK
T. 导入明文会话密钥:SDF_ImportKey
U. 销毁会话密钥:SDF_DestroyKey

6.3.1 导出 RSA 签名公钥

原型: int SDF_ExportSignPublicKey_RSA(
void *hSessionHandle,
unsigned int 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

6.3.2 导出 RSA 加密公钥

原型: int SDF_ExportEncPublicKey_RSA(
void *hSessionHandle,
unsigned int 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

6.3.3 产生 RSA 密钥对并输出

原型: int SDF_GenerateKeyPair_RSA(
void *hSessionHandle,
unsigned int 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

6.3.4 生成会话密钥并用内部 RSA 公钥加密输出

```
原型：  int SDF_GenerateKeyWithIPK_RSA(
            void *hSessionHandle,
            unsigned int uiIPKIndex,
            unsigned int uiKeyBits,
            unsigned char *pucKey,
            unsigned int *puiKeyLength,
            void **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

备注：公钥加密数据时填充方式按照 PKCS#1 v1.5 的要求进行。

6.3.5 生成会话密钥并用外部 RSA 公钥加密输出

```
原型：  int SDF_GenerateKeyWithEPK_RSA(
            void *hSessionHandle,
            unsigned int uiKeyBits,
            RSArefPublicKey *pucPublicKey,
            unsigned char *pucKey,
            unsigned int *puiKeyLength,
            void **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

备注：公钥加密数据时填充方式按照 PKCS#1 v1.5 的要求进行。

6.3.6 导入会话密钥并用内部 RSA 私钥解密

原型：
```
int SDF_ImportKeyWithISK_RSA (
        void * hSessionHandle,
        unsigned int uiISKIndex,
        unsigned char * pucKey,
        unsigned int puiKeyLength,
        void * * 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

备注：填充方式与公钥加密时相同。

6.3.7 基于 RSA 算法的数字信封转换

```
int SDF_ExchangeDigitEnvelopeBaseOnRSA(
        void * hSessionHandle,
        unsigned int uiKeyIndex,
        RSArefPublicKey * pucPublicKey,
        unsigned char * pucDEInput,
        unsigned int uiDELength,
        unsigned char * pucDEOutput,
        unsigned int * 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

6.3.8 导出 ECC 签名公钥

```
int SDF_ExportSignPublicKey_ECC(
        void * hSessionHandle,
        unsigned int 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

6.3.9 导出 ECC 加密公钥

```
原型：  int SDF_ExportEncPublicKey_ECC(
            void * hSessionHandle,
            unsigned int 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

6.3.10 产生 ECC 密钥对并输出

```
原型：  int SDF_GenerateKeyPair_ECC(
            void * hSessionHandle,
            unsigned int uiAlgID,
            unsigned int 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

6.3.11 生成会话密钥并用内部 ECC 公钥加密输出

原型：
```
int SDF_GenerateKeyWithIPK_ECC(
        void *hSessionHandle,
        unsigned int uiIPKIndex,
        unsigned int uiKeyBits,
        ECCCipher *pucKey,
        void **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

6.3.12 生成会话密钥并用外部 ECC 公钥加密输出

原型：
```
int SDF_GenerateKeyWithEPK_ECC(
        void *hSessionHandle,
        unsigned int uiKeyBits,
        unsigned int uiAlgID,
        ECCrefPublicKey *pucPublicKey,
        ECCCipher *pucKey,
        void **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

6.3.13 导入会话密钥并用内部 ECC 私钥解密

原型：
```
int SDF_ImportKeyWithISK_ECC(
        void *hSessionHandle,
        unsigned int uiISKIndex,
        ECCCipher *pucKey,
        void **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

6.3.14 生成密钥协商参数并输出

原型：
```
int SDF_GenerateAgreementDataWithECC (
        void * hSessionHandle,
        unsigned int uiISKIndex,
        unsigned int uiKeyBits,
        unsigned char * pucSponsorID,
        unsigned int uiSponsorIDLength,
        ECCrefPublicKey * pucSponsorPublicKey,
        ECCrefPublicKey * pucSponsorTmpPublicKey,
        void * * 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
	pucSelfPublicKey[out]	返回的发起方 ECC 公钥结构
	pucSelfTmpPublicKey[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

备注：为协商会话密钥，协商的发起方应首先调用本函数。

6.3.15 计算会话密钥

原型：
```
int SDF_GenerateKeyWithECC (
        void * hSessionHandle,
        unsigned char * pucResponseID,
        unsigned int uiResponseIDLength,
        ECCrefPublicKey * pucResponsePublicKey,
        ECCrefPublicKey * pucResponseTmpPublicKey,
        void * hAgreementHandle,
        void * * 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

备注：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 GM/T AAAA。

6.3.16 产生协商数据并计算会话密钥

原型：
```
int SDF_GenerateAgreementDataAndKeyWithECC(
        void *hSessionHandle,
        unsigned int uiISKIndex,
        unsigned int uiKeyBits,
        unsigned char *pucResponseID,
        unsigned int uiResponseIDLength,
        unsigned char *pucSponsorID,
        unsigned int uiSponsorIDLength,
        ECCrefPublicKey *pucSponsorPublicKey,
        ECCrefPublicKey *pucSponsorTmpPublicKey,
        ECCrefPublicKey *pucResponsePublicKey,
        ECCrefPublicKey *pucResponseTmpPublicKey,
        void **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

备注： 本函数由响应方调用。使用 SM2 算法计算会话密钥的过程见 GM/T AAAA。

6.3.17 基于 ECC 算法的数字信封转换

```
int SDF_ExchangeDigitEnvelopeBaseOnECC(
        void * hSessionHandle,
        unsigned int uiKeyIndex,
        unsigned int 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

6.3.18 生成会话密钥并用密钥加密密钥加密输出

原型：

```
int SDF_GenerateKeyWithKEK (
        void * hSessionHandle,
        unsigned int uiKeyBits,
        unsigned int uiAlgID,
        unsigned int uiKEKIndex,
        unsigned char * pucKey,
        unsigned int * puiKeyLength,
        void * * 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

备注：　加密模式使用 ECB 模式。

6.3.19　导入会话密钥并用密钥加密密钥解密

原型：
```
int SDF_ImportKeyWithKEK (
        void * hSessionHandle,
        unsigned int uiAlgID,
        unsigned int uiKEKIndex,
        unsigned char * pucKey,
        unsigned int puiKeyLength,
        void * * 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

备注：　加密模式使用 ECB 模式。

6.3.20　导入明文会话密钥

原型：
```
int SDF_ImportKey (
        void * hSessionHandle,
        unsigned char * pucKey,
        unsigned int uiKeyLength,
        void * * phKeyHandle);
```

描述：　导入明文会话密钥,同时返回密钥句柄

参数：		
	hSessionHandle[in]	与设备建立的会话句柄
	pucKey[in]	缓冲区指针,用于存放输入的密钥明文
	puiKeyLength[in]	输入的密钥明文长度
	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

6.3.21　销毁会话密钥

原型：
```
int SDF_DestroyKey (
        void * hSessionHandle,
        void * 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
备注：　在对称算法运算完成后，应调用本函数销毁会话密钥。

6.4　非对称算法运算类函数

非对称算法运算类函数包括以下具体函数，各函数返回值见附录 A 函数返回代码定义：

- 外部公钥 RSA 运算：SDF_ExternalPublicKeyOperation_RSA
- 外部私钥 RSA 运算：SDF_ExternalPrivateKeyOperation_RSA
- 内部公钥 RSA 运算：SDF_InternalPublicKeyOperation_RSA
- 内部私 RSA 运算：SDF_InternalPrivateKeyOperation_RSA
- 外部密钥 ECC 签名：SDF_ExternalSign_ECC
- 外部密钥 ECC 验证：SDF_ExternalVerify_ECC
- 内部密钥 ECC 签名：SDF_InternalSign_ECC
- 内部密钥 ECC 验证：SDF_InternalVerify_ECC
- 外部密钥 ECC 加密：SDF_ExternalEncrypt_ECC
- 外部密钥 ECC 解密：SDF_ExternalDecrypt_ECC

6.4.1　外部公钥 RSA 运算

```
原型：　int SDF_ExternalPublicKeyOperation_RSA(
            void * hSessionHandle,
            RSArefPublicKey * pucPublicKey,
            unsigned char * pucDataInput,
            unsigned int uiInputLength,
            unsigned char * pucDataOutput,
            unsigned int * 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

备注：　数据格式由应用层封装

6.4.2　外部私钥 RSA 运算

```
原型：　int SDF_ExternalPrivateKeyOperation_RSA(
            void * hSessionHandle,
            RSArefPrivateKey * pucPrivateKey,
            unsigned char * pucDataInput,
            unsigned int uiInputLength,
            unsigned char * pucDataOutput,
            unsigned int * puiOutputLength);
```

描述：　指定使用外部私钥对数据进行运算

参数：		
	hSessionHandle[in]	与设备建立的会话句柄
	pucPrivateKey [in]	外部 RSA 私钥结构
	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

备注：　数据格式由应用层封装

6.4.3　内部公钥 RSA 运算

```
原型：  int SDF_InternalPublicKeyOperation_RSA(
            void *hSessionHandle,
            unsigned int uiKeyIndex,
            unsigned char *pucDataInput,
            unsigned int uiInputLength,
            unsigned char *pucDataOutput,
            unsigned int *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

备注：　索引范围仅限于内部签名密钥对，数据格式由应用层封装

6.4.4　内部私钥 RSA 运算

```
原型：  int SDF_InternalPrivateKeyOperation_RSA(
            void *hSessionHandle,
            unsigned int uiKeyIndex,
            unsigned char *pucDataInput,
            unsigned int uiInputLength,
            unsigned char *pucDataOutput,
            unsigned int *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

备注: 索引范围仅限于内部签名密钥对,数据格式由应用层封装

6.4.5 外部密钥 ECC 签名

原型:
```
int SDF_ExternalSign_ECC(
        void * hSessionHandle,
        unsigned int uiAlgID,
        ECCrefPrivateKey * pucPrivateKey,
        unsigned char * pucData,
        unsigned int uiDataLength,
        ECCSignature * pucSignature);
```

描述: 使用外部 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
	uiAlgID[in]	算法标识,指定使用的 ECC 算法
	pucPrivateKey[in]	外部 ECC 私钥结构
	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
	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

备注: 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 GM/T AAAA。

6.4.6 外部密钥 ECC 验证

原型:
```
int SDF_ExternalVerify_ECC(
        void * hSessionHandle,
        unsigned int uiAlgID,
        ECCrefPublicKey * pucPublicKey,
        unsigned char * pucDataInput,
        unsigned int 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
	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
备注：	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 GM/T AAAA。	

6.4.7 内部密钥 ECC 签名

```
原型：  int SDF_InternalSign_ECC(
            void * hSessionHandle,
            unsigned int uiISKIndex,
            unsigned char * pucData,
            unsigned int 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
备注：	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 GM/T AAAA。	

6.4.8 内部密钥 ECC 验证

```
原型：  int SDF_InternalVerify_ECC(
            void * hSessionHandle,
            unsigned int uiISKIndex,
            unsigned char * pucData,
            unsigned int 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
备注：	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 GM/T AAAA。	

6.4.9 外部密钥 ECC 公钥加密

原型：int SDF_ExternalEncrypt_ECC(
void *hSessionHandle,
unsigned int uiAlgID,
ECCrefPublicKey *pucPublicKey,
unsigned char *pucData,
unsigned int 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

6.4.10 外部密钥 ECC 私钥解密

原型：int SDF_ExternalDecrypt_ECC(
void *hSessionHandle,
unsigned int uiAlgID,
ECCrefPrivateKey *pucPrivateKey,
ECCCipher *pucEncData,
unsigned char *pucData,
unsigned int *puiDataLength);

描述：使用外部 ECC 私钥进行解密运算

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
	pucPrivateKey[in]	外部 ECC 私钥结构
	pucEncData[in]	缓冲区指针，用于存放输入的数据密文
	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

6.5 对称算法运算类函数

对称算法运算类函数包括以下具体函数，各函数返回值见附录 A 函数返回代码定义：

- 对称加密：SDF_Encrypt
- 对称解密：SDF_Decrypt
- 计算 MAC：SDF_CalculateMAC

6.5.1 对称加密

原型：
```
int SDF_Encrypt(
        void * hSessionHandle,
        void * hKeyHandle,
        unsigned int uiAlgID,
        unsigned char * pucIV,
        unsigned char * pucData,
        unsigned int uiDataLength,
        unsigned char * pucEncData,
        unsigned int * 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

备注：此函数不对数据进行填充处理，输入的数据必须是指定算法分组长度的整数倍。

6.5.2 对称解密

原型：
```
int SDF_Decrypt (
        void * hSessionHandle,
        void * hKeyHandle,
        unsigned int uiAlgID,
        unsigned char * pucIV,
        unsigned char * pucEncData,
        unsigned int uiEncDataLength,
        unsigned char * pucData,
        unsigned int * 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
	pucIV[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

备注： 此函数不对数据进行填充处理,输入的数据必须是指定算法分组长度的整数倍。

6.5.3 计算 MAC

```
原型：  int SDF_CalculateMAC(
            void * hSessionHandle,
            void * hKeyHandle,
            unsigned int uiAlgID,
            unsigned char * pucIV,
            unsigned char * pucData,
            unsigned int uiDataLength,
            unsigned char * pucMAC,
            unsigned int * 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,用于存放输出的数据明文
	uiDataLength[in]	输出的数据明文长度
	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

备注： 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。

6.6 杂凑运算类函数

杂凑运算类函数包括以下具体函数,各函数返回值见附录 A 函数返回代码定义：

- 杂凑运算初始化:SDF_HashInit
- 多包杂凑运算:SDF_HashUpdate
- 杂凑运算结束:SDF_HashFinal

6.6.1 杂凑运算初始化

```
原型：  int SDF_HashInit(
            void * hSessionHandle,
            unsigned int uiAlgID
            ECCrefPublicKey * pucPublicKey,
            unsigned char * pucID,
            unsigned int 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

备注：　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 GM/T AAAA。

6.6.2　多包杂凑运算

```
原型：  int SDF_HashUpdate(
            void *hSessionHandle,
            unsigned char *pucData,
            unsigned int 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

6.6.3　杂凑运算结束

```
原型：  int SDF_HashFinal(
            void *hSessionHandle,
            unsigned char *pucHash,
            unsigned int *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

6.7　用户文件操作类函数

用户文件操作类函数包括以下具体函数，各函数返回值见附录 A 函数返回代码定义：

A.　创建文件：SDF_CreateFile

B.　读取文件：SDF_ReadFile

C.　写文件：SDF_WriteFile

D.　删除文件：SDF_DeleteFile

6.7.1 创建文件

原型：int SDF_CreateFile(
void * hSessionHandle,
unsigned char * pucFileName,
unsigned int uiNameLen,
unsigned int 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

6.7.2 读取文件

原型：int SDF_ReadFile(
void * hSessionHandle,
unsigned char * pucFileName,
unsigned int uiNameLen,
unsigned int uiOffset,
unsigned int * puiFileLength,
unsigned char * 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

6.7.3 写文件

原型：int SDF_WriteFile(
void * hSessionHandle,
unsigned char * pucFileName,
unsigned int uiNameLen,
unsigned int uiOffset,
unsigned int uiFileLength,
unsigned char * 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

6.7.4 删除文件

原型：
```
int SDF_DeleteFile(
        void *hSessionHandle,
        unsigned char *pucFileName,
        unsigned int 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

7 安全要求

7.1 密钥管理要求

基于本标准设计、开发的密码设备在密钥管理方面，应满足以下要求：

1) 设备密钥的使用不对应用系统开放；
2) 密钥必须用安全的方法产生并存储；
3) 在任何时间、任何情况下，除公钥外的密钥均不能以明文形式出现在密码设备外；
4) 密码设备内部存储的密钥应具备有效的密钥保护机制，防止解剖、探测和非法读取；
5) 密码设备内部存储的密钥应具备权限控制机制，防止非法使用和导出。

7.2 密码服务要求

基于本标准设计、开发的密码设备在提供服务方面，应满足以下要求：

1) 配用的密码算法应得到国家密码主管部门的批准；
2) 使用国家密码主管部门认可的密码算法芯片；
3) 本标准所列的所有接口函数均应能被应用系统任意调用。

7.3 设备状态要求

基于本标准设计、开发的密码设备在设备状态方面，应满足以下要求：

1) 密码设备应具有初始和就绪两个状态；

2） 未安装设备密钥的密码设备应处于初始状态，已安装设备密钥的密码设备应处于就绪状态；
3） 在初始状态下，除可读取设备信息、设备密钥的生成或恢复操作外，不能执行任何操作，生成或恢复设备密钥后，密码设备处于就绪状态；
4） 在就绪状态下，除设备密钥的生成或恢复操作外，应能执行任何操作；
5） 在就绪状态下进行的密钥操作，设备操作员应经过密码设备的认证。

7.4 其他安全要求

密码设备内部存储的私钥的使用权限设置应由设备管理员完成，可采用口令字方式，口令字编码长度应不低于8字符，同时口令字内容应为字符与数字的混合体。

密码设备应有安全机制和措施，保证密钥在生成、安装、导入、存储、备份、恢复及销毁整个生存期间的安全，此安全机制可由设备厂商自行设计实现。

附 录 A
（规范性附录）
函数返回代码定义

错误代码标识		
宏描述	预定义值	说明
# define SDR_OK	0x0	操作成功
# define SDR_BASE	0x01000000	错误码基础值
# define SDR_UNKNOWERR	SDR_BASE + 0x00000001	未知错误
# define SDR_NOTSUPPORT	SDR_BASE + 0x00000002	不支持的接口调用
# define SDR_COMMFAIL	SDR_BASE + 0x00000003	与设备通信失败
# define SDR_HARDFAIL	SDR_BASE + 0x00000004	运算模块无响应
# define SDR_OPENDEVICE	SDR_BASE + 0x00000005	打开设备失败
# define SDR_OPENSESSION	SDR_BASE + 0x00000006	创建会话失败
# define SDR_PARDENY	SDR_BASE + 0x00000007	无私钥使用权限
# define SDR_KEYNOTEXIST	SDR_BASE + 0x00000008	不存在的密钥调用
# define SDR_ALGNOTSUPPORT	SDR_BASE + 0x00000009	不支持的算法调用
# define SDR_ALGMODNOTSUPPORT	SDR_BASE + 0x0000000A	不支持的算法模式调用
# define SDR_PKOPERR	SDR_BASE + 0x0000000B	公钥运算失败
# define SDR_SKOPERR	SDR_BASE + 0x0000000C	私钥运算失败
# define SDR_SIGNERR	SDR_BASE + 0x0000000D	签名运算失败
# define SDR_VERIFYERR	SDR_BASE + 0x0000000E	验证签名失败
# define SDR_SYMOPERR	SDR_BASE + 0x0000000F	对称算法运算失败
# define SDR_STEPERR	SDR_BASE + 0x00000010	多步运算步骤错误
# define SDR_FILESIZEERR	SDR_BASE + 0x00000011	文件长度超出限制
# define SDR_FILENOEXIST	SDR_BASE + 0x00000012	指定的文件不存在
# define SDR_FILEOFSERR	SDR_BASE + 0x00000013	文件起始位置错误
# define SDR_KEYTYPEERR	SDR_BASE + 0x00000014	密钥类型错误
# define SDR_KEYERR	SDR_BASE + 0x00000015	密钥错误
# define SDR_ENCDATAERR	SDR_BASE + 0x00000016	ECC 加密数据错误
# define SDR_RANDERR	SDR_BASE + 0x00000017	随机数产生失败
# define SDR_PRKRERR	SDR_BASE + 0x00000018	私钥使用权限获取失败
# define SDR_MACERR	SDR_BASE + 0x00000019	MAC 运算失败
# define SDR_FILEEXISTS	SDR_BASE + 0x0000001A	指定文件已存在
# define SDR_FILEWERR	SDR_BASE + 0x0000001B	文件写入失败
# define SDR_NOBUFFER	SDR_BASE + 0x0000001C	存储空间不足
# define SDR_INARGERR	SDR_BASE + 0x0000001D	输入参数错误
# define SDR_OUTARGERR	SDR_BASE + 0x0000001E	输出参数错误
……	SDR_BASE + 0x0000001F 至 SDR_BASE + 0x00FFFFFF	预留

参 考 文 献

[1] GB/T 17903.3—1999 信息技术 安全技术 密钥管理 第1部分:框架

[2] GB/T 17903.3—1999 信息技术 安全技术 密钥管理 第2部分:使用对称技术的机制

[3] GB/T 17903.3—1999 信息技术 安全技术 密钥管理 第3部分:使用非对称技术的机制

[4] GB/T 17964—2000 信息技术 安全技术 加密算法 第1部分:概述

[5] GB/T 17964—2000 信息技术 安全技术 加密算法 第2部分:非对称加密

[6] GB/T 17964—2000 信息技术 安全技术 加密算法 第3部分:对称加密

[7] GB/T 18336—2001 信息技术 安全技术 信息技术安全性评估准则

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

[9] 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

[10] 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