ICS 35.030 CCS L 80

**中** **华** **人** **民** **共** **和** **国** **密** **码** **行** **业** **标** **准**

GM/T 0018—2023 代 替GM/T 0018—2012



**密码设备应用接口规范**

**Cryptographic device application interface specification**

**2023-12-04发布** **2024-06-01实施**



**国家密码管理局** **发** **布**

**GM/T 0018—2023**

**目** **次**

[前言 Ⅲ](#bookmark2)

[引言 IV](#bookmark3)

[1 范围 1](#bookmark4)

[2 规范性引用文件 1](#bookmark5)

[3 术语和定义 1](#bookmark6)

[4 缩 略 语 1](#bookmark7)

[5 算法标识和数据结构 2](#bookmark8)

[5.1 算法标识定义 2](#bookmark1)

[5.2 基本数据类型定义 2](#bookmark9)

[5.3 设备信息定义 2](#bookmark1)

[5.4 密钥分类及存储定义 3](#bookmark10)

[5.5 RSA 密钥数据结构定义 4](#bookmark11)

[5.6 ECC 密钥数据结构定义 5](#bookmark12)

[5.7 ECC 加密数据结构定义 5](#bookmark13)

[5.8 ECC 签名数据结构定义 6](#bookmark14)

[5.9 ECC 密钥对保护结构定义 6](#bookmark15)

[6 设备接口描述 7](#bookmark16)

[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 7](#bookmark17)

[6.2 设备管理类函数 8](#bookmark18)

[6.3 密钥管理类函数 11](#bookmark19)

[6.4 非对称算法运算类函数 20](#bookmark20)

[6.5 对称算法运算类函数 25](#bookmark21)

[6.6 杂凑运算类函数 37](#bookmark22)

[6.7 用户文件操作类函数 40](#bookmark23)

[6.8 验证调试类函数 42](#bookmark24)

[附录 A(规范性) 函数返回代码定义 50](#bookmark25)

[附录B (规范性) SM9算法相关数据结构和接口函数 52](#bookmark26)

[附录C (规范性) VPN 设备相关接口函数 69](#bookmark27)

[参考文献 79](#bookmark28)

**GM/T 0018—2023**

**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。

本文件代替GM/T 0018—2012《密码设备应用接口规范》,与GM/T 0018—2012 相比，除结构调整

和编辑性改动外，主要技术变化如下：

a) 更改了ECC 密钥对保护结构(见5.9,2012年版的5.8);

b) 删除了基于 RSA 算法的数字信封转换和基于 ECC算法的数字信封转换两个接口函数(见 2012年版的6.3.7、6.3.17);

c) 增加了多包对称加解密、多包MAC 计算和单包可鉴别加解密、多包可鉴别加解密相关的17 个接口函数(见6.5,2012年版的6.5);

d) 增加了带密钥的杂凑运算相关的3个接口函数(见6.6);

e) 增加了验证调试类函数(见6.8);

f) 删除了第7章“安全要求”(见2012年版的第7章);

g) 增加了用户标识不匹配时的错误码(见附录A);

h) 增加了SM9 标识密码算法相关数据结构以及 SM9 标识密码算法相关的15个接口函数的定 义(见附录B);

i) 增加了计算IKE 工作密钥、计算 IPSEC 会话密钥、计算SSL 工作密钥共6个接口函数的定义 (见附录C)。

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：中电科网络安全科技股份有限公司、无锡江南信息安全工程技术中心、四川大学、 北京国脉信安科技有限公司、格尔软件股份有限公司、北京数字认证股份有限公司、兴唐通信科技有限 公司、山东得安信息技术有限公司、三未信安科技股份有限公司、北京海泰方圆科技股份有限公司、山东 大学。

本文件主要起草人：刘平、罗俊、龚勋、李川、袁峰、李元正、徐强、郑强、李述胜、李玉峰、孔凡玉、 马洪富、高志权、徐明翼、柳增寿、蒋红宇。

**GM/T 0018—2023**

**引**

**言**

本文件的目标是为公钥密码基础设施应用体系框架下的服务端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。

**GM/T 0018—2023**

**密码设备应用接口规范**

**1** **范围**

本文件规定了公钥密码基础设施应用技术体系下服务端密码设备应用接口的算法标识、数据结构 和接口函数。

本文件适用于服务端密码设备的研制、使用，以及基于该类密码设备的应用开发。

**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 15852.2 信息技术 安全技术 消息鉴别码 第2部分：采用专用杂凑函数的机制 GB/T 35276 信息安全技术 SM2 密码算法使用规范

GB/T 36624 信息技术安全技术 可鉴别的加密机制

GB/T 38635(所有部分)信息安全技术 SM9 标识密码算法 GM/T 0006 密码应用标识规范

GM/T 0022 IPSec VPN技术规范 GM/T 0024 SSL VPN技术规范 GM/Z 4001 密码术语

**3** **术语和定义**

**3.1**

**3.2**

GM/Z 4001界定的以及下列术语和定义适用于本文件。

**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。

**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** **缩略语**

下列缩略语适用于本文件。

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)

**GM/T 0018—2023**

**5** **算法标识和数据结构**

**5.1** **算法标识定义**

本文件中所使用的算法其算法标识见GM/T 0006。分组密码算法的算法标识包含其工作模式。

**5.2** **基本数据类型定义**

本文件中的基本数据类型均为高位字节在前(Big-Endian) 方式存储和交换。表1定义了基本数据 类型。

**表** **1** **基本数据类型**

|  |  |  |
| --- | --- | --- |
| 类型名称 | 描述 | 定义 |
| BYTE | 字节类型 ，无符号8位字符 | ypedef unsigned char BYTE |
| CHAR | 字得类型，无符号8位字符 | typede(unsighed char CHAR |
| LONG | 长整数，有符号32位整数 | typedef nt LONG |
| ULONG | 长整数学无符号32位整数 | ypcdcf unsigned int ULONG |
| FLAGS | 标志类亚，无号32位整数 | ypedel unsigned in FLAGS |
| LPSTR | 8 位 字雀 脂 学 符 串按照7F8桥 政 存 储 及 交 质 | vpcdeECHAR\*LPSTR |
| HANDLE | 句柄，指面数据对象的起始地址 | 1ypcdg yoid \*HANDLE |

**5.3** **设备信息定义**

**表2定义了设备信息。**

表 2 设 备 信 息 描 述

|  |  |  |
| --- | --- | --- |
| 字段名称 | 数据长度  字节) | 含义 |
| IssuerName |  | 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，本版本为2 |
| 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运算的结果； 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运算的结果 |
| 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的结果 |
| 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的结果 |
| 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{

**GM/T 0018—2023**

CHAR IssucrName[40];

CHAR DeviceName[16];

CHAR DcviceScrial[16];

ULONG DeviceVersion;

ULONG StandardVersion;

ULONG AsymAlgAbility[2];

ULONG SymAlgAbility;

ULONG HashAlgAbility;

ULONG BufferSize; }DEVICEINFO;

**5.4** **密钥分类及存储定义**

**5.4.1** **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。表3描述了设备密 钥和用户密钥存储方式。

**表** **3** **设备密钥和用户密钥存储描述**

|  |  |  |
| --- | --- | --- |
| 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 | 签名公钥 | 签名私钥 |
| 加密公钥 | 加密私钥 |
|  | … | …… |
| …… | … |

**5.4.2** **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，存放于密钥存储区，使用索引号从1开始。表4 描述了密钥加密密钥存储方式。

**表** **4** **密钥加密密钥存储描述**

|  |  |
| --- | --- |
| 密钥索引号 | 密钥加密密钥 |
| 0x01 | 密钥加密密钥001 |
| . | .. |

**5.4.3** **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。

**GM/T 0018—2023**

**5.5 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，不足位从高位向低位填充数据0。表5定义了RSA 密钥数据结构。

**表** **5** **RSA** **密钥数据结构**

|  |  |  |
| --- | --- | --- |
| 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 | RSAref\_MAX\_LEN | 模N |
| e | RSAref\_MAX\_LEN | 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 | RSAref=MAX IEN | 模N |
|  |  | 公钥指数 |
| d | RSAref MAX\_LEN | 私钥指数 |
| ppime[2 | RSA AX\_ \*2 | 素数p和 |
| pexp[27 | RS MA X\_FS 2 | Dp和Dq |
| coef | ef\_MAXPBN | 系数i |

实际数据结构定义：

#defime RSAref\_ MAXB 自S #define RSAref\_ MAX IGN #define RSAref\_MAX\_PBITS #define RSAref \_MAX\_PLEN

2078

(ORSAicl MAX BS 17/8)

((RSAref\_MAX\_BITS+1)/2) ((RSAref\_MAX\_PBITS+7)/8)

typedef struci RSAreIRublicKey\_st {

ULONG bits;

BYTE m [ RSAref\_ MAX LEN ] ;

BYTE e[RSAref\_MAX\_LEN]; }RSArefPublicKey;

typedef struct RSArefPrivateKey\_st {

ULONG bits;

BYTE m[RSAref\_MAX\_LEN]; BYTE e[RSAref\_MAX\_LEN]; BYTE d[RSAref\_MAX\_LEN];

BYTE prime[2][RSAref\_MAX\_PLEN];

BYTE pexp[2][RSAref\_MAX\_PLEN];

BYTE coef[RSAref\_MAX\_PLEN];

**GM/T 0018—2023**

}RSArefPrivatcKcy;

注：RSAref\_MAX\_BITS的数值能根据实际支持的最大模长进行定义。

**5.6 ECC密钥数据结构定义**

表6定义了ECC 密钥数据结构。

**表** **6** **ECC密钥数据结构**

|  |  |  |
| --- | --- | --- |
| 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 |
| K | ECCref\_MAX\_LEN | 私钥 |

实际数据结构定义：

井define ECCref\_MAX\_BITS #define ECCref\_MAX\_LEN

typedef struct ECCrcfPublicKey\_st {

512

((ECCref\_MAX\_BITS+7)/8)

ULONG bits;

BYTE x[ECCref\_MAX\_LEN];

BYTE y[ECCref\_MAX\_LEN]; }ECCrefPublicKey;

typedef struct ECCrefPrivateKey\_st {

ULONG bits;

BYTE K[ECCref\_MAX\_LEN]; }ECCrefPrivateKey;

**5.7 ECC加密数据结构定义**

表7定义了ECC 加密数据结构。

**表** **7** **ECC加密数据结构**

|  |  |  |
| --- | --- | --- |
| 加密数据结构定义 | | |
| 字段名称 | 数据长度(字节) | 含义 |
| X | ECCref\_MAX\_LEN | X分量 |
| y | ECCref\_MAX\_LEN | Y分量 |

**GM/T 0018—2023**

**表7** **ECC加密数据结构(续**)

|  |  |  |
| --- | --- | --- |
| 加密数据结构定义 | | |
| 字段名称 | 数据长度(字节) | 含义 |
| M | 32 | 明文的杂凑值 |
| 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

{

BYTE x[ECCref\_MAX\_LEN]; BYTE y[ECCref\_MAX\_LEN]; BYTE M[32];

ULONG L

BYTE C[]; }ECCCipher;

**5.8** **ECC签名数据结构定义**

表8定义了ECC 签名数据结构。

**表** **8** **ECC签名数据结构**

|  |  |  |
| --- | --- | --- |
| 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 {

BYTE r[ECCref\_MAX\_LEN];

BYTE s[ECCref\_MAX\_LEN]; }ECCSignature;

**5.9** **ECC密钥对保护结构定义**

密钥管理系统下发到设备中的 ECC加密密钥对使用本结构保护，表9定义了ECC 密钥对保护 结构。

**表9** **ECC加密密钥对保护结构参数**

|  |  |  |
| --- | --- | --- |
| 字段名称 | 数据长度(字节) | 含义 |
| Version | 4 | 版本号，本版本为1 |

**GM/T 0018—2023**

**表** **9** **ECC加密密钥对保护结构参数(续**)

|  |  |  |
| --- | --- | --- |
| 字段名称 | 数据长度(字节) | 含义 |
| ulSymmAlgID | 4 | 对称算法标识(必须为ECB模式) |
| ulBits | 4 | ECC密钥对的密钥位长 |
| cbEncryptedPrivKey | ECCref\_MAX\_LEN | 对称算法加密的ECC私钥，ECC私钥的原文  为ECCrefPrivateKcy,结构中的K,其有效长度 为原文的(ulBits +7)/8 |
| PubKey | ECCref\_MAX\_LEN\*2+4 | ECC密钥对的公钥 |
| ECCCipherBlob | ECCref\_MAX\_LEN \*2+32+  4+L | 用保护公钥加密过的对称密钥密文结构 |

实际数据结构定义：

typedef struct EnvelopedECCKey\_st{ ULONG Version;

ULONG ulSymmAlgID; ULONG ulBits;

BYTE cbEncryptedPriKey[ECCref\_MAX\_LEN]; ECCrefPublicKey PubKey;

ECCCipher ECCCipherBlob; }EnvelopedECCKey;

私钥密文结构中有效密文分组从0偏移量字节开始，顺序解密密文分组后将明文连接得到ECC 密 钥对的私钥的明文。

**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**

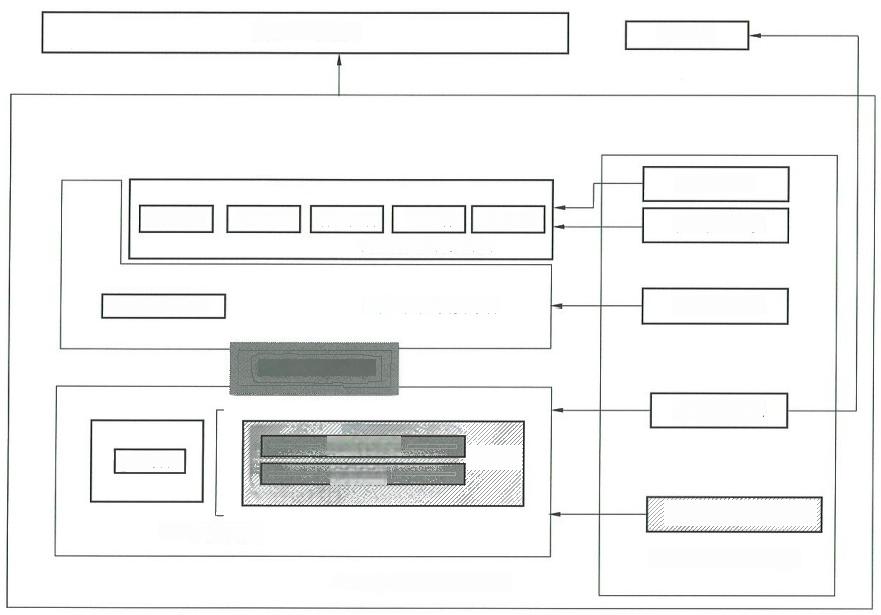
在公钥密码基础设施应用技术体系框架中，密码设备服务层由密码机、密码卡、智能密码终端设备 组成，通过本文件规定的密码设备应用接口向通用密码服务层提供基础密码服务，如图1所示。

基础密码服务包括密钥生成、单一的密码运算、文件管理服务。

本文件采用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。符合本文 件的密码设备应支持本文件中关于ECC算法(本文件中ECC特指SM2 算法)、对称算法以及杂凑运算 的函数接口。木文件所列的所有接口函数均应能被应用系统任意调用。附录 A 定义了函数返回代 码，附录B定义了SM9 算法相关数据结构和接口函数，附录C 定义了VPN 设备相关接口函数。

产品支持SM9 算法时，SM9算法相关数据结构和接口函数应符合附录 B的定义。产品支持VPN 时，VPN 设备相关接口函数应符合附录C 的定义。

**GM/T 0018—2023**



管理应用

时间戳系统

授权管理系统

证书认证系统

密码设备应用接口

密码设备管理

密码资源池

密码设备

密码设备

密码资源管理(云环境

密码设备服务层

基础设施安全支撑平台

公钥密码应用技术体系框架

身份鉴别 访问控制 时间戳 电子签章

典型密码应用支撑层

通用密码应用支撑层

应用1——应用N

通用密码服务

*云环境*

电子证据

**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框架中的位置**

**6.2** **设备管理类函数**

**6.2.1** **概述**

设备管理类函数提供设备打开与关闭、会话创建与关闭、设备信息获取、随机数产生、私钥权限获取 与释放功能，表10列出了设备管理类函数，附录A 定义了函数返回代码。

**表10** **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 CloseSessior | 关闭会话 |
| SDF\_GetDeviceInfo | 获取设备信息 |
| SDF\_GcncratcRandom | 产生随机数 |
| SDF\_GetPrivateKeyAccessRight | 获取私钥使用权限 |
| SDF\_ReleasePrivateKeyAccessRight | 释放私钥使用权限 |

**GM/T 0018—2023**

**6.2.2** **打开设备**

原型： LONG SDF\_OpenDevicc(HANDLE \*phDeviceHandle);

描述： 打开密码设备。phDeviceHandle由函数初始化并填写内容。

|  |  |  |
| --- | --- | --- |
| 参数： | phDeviceHandle[out] | 返回的设备句柄 |
| 返回值： | 0  非 0 | 成功  失败，返回错误代码 |

**6.2.3** **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 |
| 返回值： | 0  非 0 | 成功  失败，返回错误代码 |

**6.2.4** **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： | hDeviccHandle[in]  phSessionHandle[out] | 已打开的设备句柄  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

**6.2.5** **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： hSessionHandlc [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

**6.2.6** **获取设备信息**

原型： LONG SDF\_GetDeviceInfo(

HANDLE hSessionHandle,

**GM/T 0018—2023**

DEVICEINFO\*pstDeviceInfo);

|  |  |  |
| --- | --- | --- |
| 描述： | 获取密码设备能力描述。 |  |
| 参数： | hSessionHandle[in] pstDeviceInfo [out] | 与设备建立的会话句柄  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

**6.2.7** **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：

hScssionHandlel uilength [ in ]

pucRandom [ o

与段备建的会所命脑

获取的随机数长度

缓 指 针 ， 存 放 获取的随机数

返回值： 0 成

非0 失败 返回错误代码

**6.2.8** **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位置私钥的使用权。本文件涉及密码设备存储的密钥

对索引值的起始索引值为1,最大为n, 密码设备的实际存储容量决定n 值。私钥访问控 制码不少于8字节。

参数：

hSessionHandle[in] uiKeyIndex[in]

pucPassword[in] uiPwdLength[in]

与设备建立的会话句柄

密码设备存储的密钥对的索引值 使用私钥权限的访问控制码

私钥访问控制码长度

**GM/T 0018—2023**

返回值： 0 成功

非 0 失败，返回错误代码

**6.2.9** **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位置私钥的使用授权。

参数： hSessionHandle[in] 与设备建立的会话句柄

uiKeyIndex[in] 密码设备存储的密钥对的索引值

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**

**6.3.1** **概述**

密钥管理类函数提供密钥的生成、销毁和导入导出功能，包括签名公钥和加密公钥的导出、会话密 钥的生成并输出、会话密钥的导入、密钥协商参数的生成并输出、会话密钥的计算以及会话密钥的销毁 函数，表11列出了密钥管理类函数，附录A 定义了函数返回代码。

**表11** **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\_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\_ExportSignPublicKey\_ECC | 导出ECC签名公钥 |
| SDF\_ExportEncPublicKey\_ECC | 导出ECC加密公钥 |
| SDF\_GenerateKeyWithIPK\_ECC | 生成会话密钥并用内部ECC公钥加密输出 |
| SDF\_GeneratcKeyWithEPK\_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\_GencratcKcyWithECC | 计算会话密钥 |
| SDF\_GenerateAgreementDataAndKeyWithECC | 产生协商数据并计算会话密钥 |
| SDF\_GenerateKeyWithKEK | 生成会话密钥并用密钥加密密钥加密输出 |

**GM/T 0018—2023**

**表** **1** **1** **密钥管理类函数(续)**

|  |  |
| --- | --- |
| 函数名称 | 功能 |
| SDF\_ImportKeyWithKEK | 导入会话密钥并用密钥加密密钥解密 |
| SDF DestroyKey | 销毁会话密钥 |

**6.3.2 导** **出** **RSA签名公钥**

原型： LONG SDF\_ExportSignPublicKey RSA(

HANDLE hsessionHandle, UIQNG uiKeylndex

RSArelPublicKey \*pucPublicKey);

描述： 导排密码设备内部存储的指定索引位置的RSA 签名公钥。

参数： HSessionHandle [ in] 与设备建立的会话句柄

|  |  |
| --- | --- |
| uiKeyIndex [ | 密码设备存储的密钥对的素引值 |
| puePublicKeylo | RSA 公钥结构 |
| 返回值： 0 | 成 |
| 非 0 | 人 返回错读 |
| **6.3.3** **导** **出rSA** **加密公钥** |  |

原型： lo g SDF \_IspordocPnbheK 的BSA

HANDLEhSessionHLandle

ULONG uiKeyIndex

RSArefPublicKey\*pucPublicKey);

描述： 导出密码设备内部存储的指定索引位置的RSA 加密公钥

参数：

hSessionHandle [ tm] uiKeyIndex[in]

pucPublicKcy[out]

与设备建立的会话句柄

密码设备存储的密钥对的索引值 RSA 公钥结构

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： LONG SDF\_GencratcKeyWithIPK\_RSA(

HANDLE hSessionHandle,

**GM/T 0018—2023**

ULONG uiIPKIndex, ULONG uiKeyBits, BYTE \*pucKey,

ULONG \*puiKeyLength,

HANDLE \*phKeyHandle);

生成会话密钥并用指定索引的内部RSA 加密公钥加密输出，同时返回密钥句柄。公钥加 密数据时填充方式与PKCS≠1 v1.5 相同。

描述： 参数：

hScssionHandle[in]

uiIPKIndex[in]

uiKeyBits[in]

pucKcy[out]

puiKeyLength[out] phKeyHandle[out]

与设备建立的会话句柄

密码设备存储的密钥对的索引值 指定产生的会话密钥长度

缓冲区指针，用于存放返回的密钥密文 返回的密钥密文长度

返回的密钥句柄

返回值： 0 成功

非 0 失败，返回错误代码

**6.3.5** **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RSA公钥加密输出。公钥加密数据时填充方式与PKCS=1vl.5相同。

参数：

hSessionHandle[in]

uiKeyBits[in]

pucPublicKey[in] pucKey[out]

puiKeyLength[out phKeyHandle[out]

与设备建立的会话句柄

指定产生的会话密钥长度 输入的外部RSA 公钥结构

缓冲区指针，用于存放返回的密钥密文 返回的密钥密文长度

返回的密钥句柄

返回值： 0 成功

非 0 失败，返回错误代码

**GM/T 0018—2023**

**6.3.6** **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RSA 私钥解密，同时返回密钥句柄。

参数：

hSessionHandle[in] uiISKIndex[in]

pucKey[in]

puiKeyLength[in] phKeyHandle[out]

与设备建立的会话句柄

密码设备存储的密钥对的索引值

缓冲区指针，用于存放输入的密钥密文 输入的密钥密文长度

返回的密钥句柄

返回值： 0 成功

非 0 失败，返回错误代码

**6.3.7 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ECC签名公钥。

参数：

hSessionHandle[in] uiKeyIndex[in]

pucPublicKey[out]

与设备建立的会话句柄

密码设备存储的密钥对的索引值 ECC公钥结构

返回值： 0 成功

非 0 失败，返回错误代码

**6.3.8 导出ECC加密公钥**

原型： LONG SDF\_ExportEncPublicKey\_ECC(

HANDLE hSessionHandle, ULONG uiKeyIndex,

**GM/T 0018—2023**

ECCrefPublicKey \*pucPublicKey);

描述： 导出密码设备内部存储的指定索引位置的ECC加密公钥。

参数：

hSessionHandlc[in] uiKeyIndex[in]

pucPublicKey[out]

与设备建立的会话句柄

密码设备存储的密钥对的索引值 ECC公钥结构

返回值： 0 成功

非 0 失败，返回错误代码

**6.3.9** **生成会话密钥并用内部ECC** **公钥加密输出**

原型： LONG SDF\_GenerateKeyWithIPK\_ECC(

HANDLE hScssionHandle,

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：

hScssionHandle[in] uiIPKIndex[in]

uiKeyBits[in] pucKey[out]

phKeyHandle[out]

与设备建立的会话句柄

密码设备存储的密钥对的索引值 指定产生的会话密钥长度

缓冲区指针，用于存放返回的密钥密文 返回的密钥句柄

返回值： 0 成功

非 0 失败，返回错误代码

**6.3.10** **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。

**GM/T 0018—2023**

参数： hSessionHandle[in]

uiKeyBits[in]

uiAlgID[in]

pucPublicKey[in] pucKey[out]

phKeyHandle[out]

返回值： 0

非 0

与设备建立的会话句柄

指定产生的会话密钥长度 外部 ECC 公钥的算法标识 输入的外部ECC 公钥结构

缓冲区指针，用于存放返回的密钥密文 返回的密钥句柄

成功

失败，返回错误代码

**6.3.11** **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：

hScssionHandle[in] uiISKIndex[in]

pucKey[in]

phKeyHandle[out]

与设备建立的会话句柄

密码设备存储的密钥对的索引值

缓冲区指针，用于存放输入的密钥密文 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：

参数：

**GM/T 0018—2023**

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。协商会话密钥时，本函数首先由协商的

|  |  |
| --- | --- |
| 发起方调用。  hSessionHandle[in] uiISKIndex[in]  uiKeyBits[in]  pucSponsorID[in]  uiSponsorIDLength[in]  pucSponsorPublicKey[out]  pucSponsorTmpPublicKey[out] phAgreementHandle[out] | 与设备建立的会话句柄  密码设备存储的密钥对的索引值 要求协商的密钥长度  参与密钥协商的发起方ID 值 发起方ID 长度  返回的发起方ECC公钥结构  返回的发起方临时ECC公钥结构  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

**6.3.13** **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。使用SM2 算法计算会话密钥的过程应符合GB/T 35276。本函数由协 商的发起方在获得响应方的协商参数后调用。会话密钥计算完成后，协商句柄被销毁，为 协商句柄分配的内存资源也被释放。

参数： hSessionHandle[in]

pucResponscID[in]

uiResponseIDLength[in]

pucResponsePublicKey[in]

与设备建立的会话句柄

外部输入的响应方ID 值

外部输入的响应方ID 长度

外部输入的响应方ECC公钥结构

**GM/T 0018—2023**

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

phKeyHandlc[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** **产生协商数据并计算会话密钥**

原型： LONG SDF \_GenerateAgneementDataAndKeyWithECC(

HANDLE hSessionHandle,

UEONG uiISKIndex, UIONG uiKeyBits,

BYTE\*pucResponseID,

uiONG uResonscIDrcng BYTFSo

ULONGoSponsorIDL , ,

ECConauhekey ponso icKcy,

ECCrdP Key ponso Publickey

ECCrc cK \*pspoPublicker

ECCr Ke  pacRe ponscImpEublicKey

HANDLB xpbKeyHlandlo),

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。使用SM2 算法计算会话密钥的过程应符合GB/T 35276 。本函数由响应方 调用。

参数： hSessionHandle[in]

uiISKIndex[in] uiKeyBits[in]

pucResponseID[in]

uiResponseIDLength[in] pucSponsorID[in]

uiSponsorIDLength[in]

pucSponsorPublicKey[in]

pucSponsorTmpPublicKey[in] pucResponsePublicKey[out]

与设备建立的会话句柄

密码设备存储的密钥对的索引值 协商后要求输出的密钥长度

响应方ID 值

响应方ID 长度 发起方ID 值

发起方ID 长度

外部输入的发起方ECC 公钥结构

外部输入的发起方临时ECC 公钥结构 返回的响应方ECC 公钥结构

**GM/T 0018—2023**

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

**6.3.15** **生成会话密钥并用密钥加密密钥加密输出**

原型： LONG SDF\_GenerateKcyWithKEK(

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。加密模式为ECB模式。

参数：

hSessionHandle[in]

uiKeyBits[in]

uiAlgID[in]

uiKEKIndex[in]

pucKey[out]

puiKeyLength[out phKeyHandle[out]

与设备建立的会话句柄

指定产生的会话密钥长度

算法标识，指定分组密码算法和工作模式 密码设备存储的密钥加密密钥的索引值 缓冲区指针，用于存放返回的密钥密文

返回的密钥密文长度

返回的密钥句柄

返回值： 0 成功

非 0 失败，返回错误代码

**6.3.16** **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 uiKeyLength,

HANDLE\*phKeyHandle);

**GM/T 0018—2023**

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。加密模式为ECB 模式。

参数： hSessionHandle[in] 与设备建立的会话句柄

uiAlgID[in]

uiKEKIndex[in] pucKey[in]

uiKeyLength[in]

phKeyHandle[out]

返回值： 0

非 0

算法标识，指定分组密码算法和工作模式 密码设备存储的密钥加密密钥的索引值 缓冲区指针，用于存放输入的密钥密文

输入的密钥密文长度

返回的密钥句柄 成功

失败，返回错误代码

**6.3.17** **销毁会话密钥**

原型：

描述： 参数：

**I ONG SDF\_DestroyKey(**

HANDLEhSessionlladl

HANOLThKaItand

销毁会话密钥 开释放密钥句极分配的对存资源本面数在对称算法运算完成后调用。

hSessionHandle 设 备 建射 会 话 句栖

hkeyHandle 输入 的 密朗 耐柄

返回值： 0 

非 0 实题，返河错误代两

**6.4** **非对称算法运算类函数**

**6.4.1** **概述**

非对称算法运算类函数提供 RSA 公私钥运算、ECC 签名验证和加密功能，表12列出了非对称算 法运算类函数，附录 A 定义了函数返国代码。

**表12** **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 | 外部公钥ECC验证 |
| SDF\_InternalSign\_ECC | 内部私钥ECC签名 |
| SDF\_InternalVerify\_ECC | 内部公钥ECC验证 |
| SDF\_ExternalEncrypt\_ECC | 外部公钥ECC加密 |

**GM/T 0018—2023**

**6.4.2 外部公钥** **RSA 运算**

原型： LONG SDF\_ExternalPublicKeyOperation\_RSA(

HANDLE hSessionHandle,

RSArcfPublicKey \*pucPublicKey,

BYTE \*pucDataInput,

ULONG uilnputLength, BYTE \*pucDataOutput,

ULONG \*puiOutputLength);

描述： 指定使用外部公钥对数据进行RSA 运算。数据格式由应用层封装。

参数：

hSessionHandle[in]

pucPublicKey [in]

pucDataInput [in]

uiInputLength[in]

pucDataOutput[out] puiOutputLength[out]

与设备建立的会话句柄 外部RSA公钥结构

缓冲区指针，用于存放输入的数据 输入的数据长度

缓冲区指针，用于存放输出的数据 输出的数据长度

返回值： 0 成功

非0 失败，返回错误代码

**6.4.3 内部公钥RSA 运算**

原型： LONG SDF\_InternalPublicKeyOperation\_RSA(

HANDLE hSessionHandle, ULONG uiKeyIndex,

BYTE \*pucDataInput,

ULONG uiInputLength, BYTE\*pucDataOutput,

ULONG \*puiOutputLcngth);

描述： 使用内部指定索引的公钥对数据进行RSA 运算。索引范围仅限于内部签名密钥对，数据

格式由应用层封装。

参数： hSessionHandle[in] 与设备建立的会话句柄

uiKeyIndex[in] 密码设备存储的密钥对的索引值

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

**GM/T 0018—2023**

pucDataOutput[out] puiOutputLength[out]

返回值： 0

非 0

缓冲区指针，用于存放输出的数据 输出的数据长度

成功

失败，返回错误代码

**6.4.4 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RSA 运算。索引范围仅限于内部签名密钥对，数据

格式由应用层封装。

|  |  |  |
| --- | --- | --- |
| 参数： | hSessionHandle[in]  uiKeyIndex[in]  pucDataInput[in]  uiInputLength[in]  pucDataOutput[out] puiOutputLength[out] | 与设备建立的会话句柄  密码设备存储的密钥对的索引值  缓冲区指针，用于存放外部输入的数据 输入的数据长度  缓冲区指针，用于存放输出的数据 输出的数据长度 |
| 返回值： | 0  非 0 | 成功  失败，返回错误代码 |

**6.4.5 外部公钥ECC验证**

原型： LONG SDF\_ExternalVerify\_ECC(

HANDLE hSessionHandle, ULONG uiAlgID,

ECCrefPublicKey \*pucPublicKey, BYTE \*pucDataInput

ULONG uiInputLength,

ECCSignature \*pucSignature);

**GM/T 0018—2023**

描述： 使用外部公钥对ECC 签名值进行验证运算。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，SM2 算法预处理过程 应符合GB/T 35276。

参数：

hSessionHandle[in]

uiAlgID[in]

pucPublicKey[in] pucData[in]

uiDataLcngth[in] pucSignature[in]

与设备建立的会话句柄

算法标识，指定使用的ECC 算法 外部ECC 公钥结构

缓冲区指针，用于存放外部输入的数据 输入的数据长度

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

**6.4.6 内部私钥** **ECC签名**

原型： LONG SDF\_InternalSign\_ECC(

HANDLE hSessionHandle, ULONG uiISKIndex,

BYTE\*pucData,

ULONG uiDataLength,

ECCSignature \*pucSignature);

描述： 使用内部指定索引的私钥对数据进行 ECC 签名运算。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，SM2 算法预 处理过程应符合GB/T 35276。

参数：

hSessionHandle[in] uiISKIndex [in]

pucData[in]

uiDataLength[in]

pucSignature [out]

与设备建立的会话句柄

密码设备存储的密钥对的索引值

缓冲区指针，用于存放外部输入的数据 输入的数据长度

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

**6.4.7 内部公钥** **ECC验证**

原型： LONG SDF\_InternalVerify\_ECC(

HANDLE hSessionHandle,

**GM/T 0018—2023**

ULONG uiISKIndex, BYTE \*pucData,

ULONG uiDataLength,

ECCSignature \*pucSignature);

描述： 使用内部指定索引的公钥对ECC 签名值进行验证运算。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，SM2 算法预 处理过程应符合GB/T 35276。

|  |  |  |
| --- | --- | --- |
| 参数： | hSessionHandle[in]  uiISKIndex [in] pucData[in]  uiDataLength[in] pucSignature[in] | 与设备建立的会话句柄  密码设备存储的密钥对的索引值  缓冲区指针，用于存放外部输入的数据 输入的数据长度  缓冲区指针，用于存放输入的签名值数据 |
| 返回值： | 0  非 0 | 成功  失败，返回错误代码 |

**6.4.8 外部公钥ECC加密**

原型： LONG SDF\_ExternalEncrypt\_ECC(

HANDLE hSessionHandlc, ULONG uiAlgID,

ECCrefPublicKey \*pucPublicKey,

BYTE \*pucData,

ULONG uiDataLength,

ECCCipher \*pucEncData);

描述： 使用外部公钥对数据进行ECC 加密运算。

参数：

hSessionHandle[in]

uiAlgID[in]

pucPublicKey[in] pucData[in]

uiDataLength[in] pucEncData[out]

与设备建立的会话句柄

算法标识，指定使用的ECC 算法 外部ECC公钥结构

缓冲区指针，用于存放外部输入的数据 输入的数据长度

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

**GM/T 0018—2023**

**6.5** **对称算法运算类函数**

**6.5.1** **概述**

对称算法运算类函数提供对称加解密和MAC 计算功能，表13列出了对称算法运算类函数，附录 A定义了函数返回代码。

产品支持可鉴别加解密时，可鉴别加解密相关接口函数应符合6.5.5、6.5.6 及6.5.16～6.5.21的 定义。

**表13** **对称算法运算类函数**

|  |  |
| --- | --- |
| 函数名称 | 功能 |
| SDF\_Encrypt | 单包对称加密 |
| SDF\_Decrypt | 单包对称解密 |
| SDF\_CalculateMAC | 计算单包MAC |
| SDF\_AuthEnc | 单包可鉴别加密 |
| SDF\_AuthDec | 单包可鉴别解密 |
| SDF\_EncryptInit | 多包对称加密初始化 |
| SDF\_EncryptUpdate | 多包对称加密 |
| SDF\_EncryptFinal | 多包对称加密结束 |
| SDF\_DecryptInit | 多包对称解密初始化 |
| SDF\_DecryptUpdate | 多包对称解密 |
| SDF\_DecryptFinal | 多包对称解密结束 |
| SDF\_CalculateMACInit | 多包MAC初始化 |
| SDF\_CalculateMACUpdate | 多包MAC计算 |
| SDF\_CalculateMACFinal | 多 包 M A C 结 束 |
| SDF\_AuthEncInit | 多包可鉴别加密初始化 |
| SDF\_AuthEncUpdate | 多包可鉴别加密 |
| SDF\_AuthEncFinal | 多包可鉴别加密结束 |
| SDF\_AuthDecInit | 多包可鉴别解密初始化 |
| SDF\_AuthDecUpdate | 多包可鉴别解密 |
| SDF\_AuthDccFinal | 多包可鉴别解密结束 |

**6.5.2** **单包对称加密**

原型： LONG SDF\_Encrypt(

HANDLE hSessionHandle, HANDLE hKeyHandle,

ULONG uiAlgID,

BYTE \*pucIV,

**GM/T 0018—2023**

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。此函数不对数据进行填充处理。

此函数的IV 数据长度与算法分组长度相同。

参数：

hSessionHandle[in]

hKeyHandle[in] uiAlgID[in]

pucIV [ in

pucData [in

uiDataL ength[in] pucEhcData out

puiEncDataLei 山 ou

与设备建立的会话句柄 指定的密钥句柄

法 ，指定分组密码算法和工作模式 缓冲区指针，用于存放输入的IV 数 据

缓冲区指针，用于存放输入的数据明文 输入的数据明文长度

缓冲区指钆，用于存放输出的数据密文 输画的数据密文长度

返回值： 0 

非 0 败 ，返保 代码

**6.5.3** **单包对称解密**

原型： LOG SDF\_DeepC

HANDLE bSessionHandle

NANDLE hKeyHandle, ULONG uiAlgID,

BXTE\*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。此函数的IV 数据长度与算法分

组长度相同。

参数：

hSessionHandle[in] hKeyHandle[in]

uiAlgID[in]

pucIV[in]

与设备建立的会话句柄 指定的密钥句柄

算法标识，指定分组密码算法和工作模式 缓冲区指针，用于存放输入的IV 数据

**GM/T 0018—2023**

pucEncData[in]

uiEncDataLength[in] pucData[out]

puiDataLength[out]

返回值： 0

非 0

缓冲区指针，用于存放输入的数据密文 输入的数据密文长度

缓冲区指针，用于存放输出的数据明文

输出的数据明文长度 成功

失败，返回错误代码

**6.5.4** **计算单包** **MAC**

原型：

描述：

LONG SDF\_CalculateMAC(

HANDLE hSessionHandle, HANDLE hKeyHandle,

ULONG uiAlgID,

BYTE \*pucIV, BYTE\*pucData,

ULONG uiDataLength, BYTE \*pucMac,

ULONG \*puiMacLength);

使用指定的密钥句柄和IV 对数据进行对称加密MAC 运算。此函数不对数据进行填充 处理。MAC 算法标识和工作模式的约定同分组密码算法。

|  |  |  |
| --- | --- | --- |
| 参数： | hSessionHandle[in]  hKeyHandle[in] uiAlgID[in]  pucIV[in]  pucData[in]  uiDataLength[in] pucMac[out]  puiMacLength[out | 与设备建立的会话句柄 指定的密钥句柄  算法标识，指定MAC 加密算法和工作模式  缓冲区指针，用于存放输入的IV 数据 缓冲区指针，用于存放输入的数据明文 输入的数据明文长度  缓冲区指针，用于存放输出的MAC 值 输出的MAC 值长度 |
| 返回值： | 0  非 0 | 成功  失败，返回错误代码 |

**6.5.5** **单包可鉴别加密**

原型： LONG SDF\_AuthEnc(

HANDLE hSessionHandle,

**GM/T 0018—2023**

HANDLE hKeyHandle, ULONG uiAlgID

BYTE\*pucStartVar,

ULONG uiStartVarLength,

BYTE \*pucAad,

ULONG uiAadLength BYTE \*pucData,

ULONG uiDtaLength,

BYTE \*pucEneData,

ULoNG \*puiEncDataLength BXTE \*pucAuthData

ULONG \*puiAuthDataLength);

使用指定的密钥句栖对数据进行可鉴加密之算。此函数应用于可鉴别加密的CCM 和 GOM 模式，输大输出数据应根据GM/T0006 定义的算法标识并按照GB/T36624 的规 定进行确定。

描述：

参数：

|  |  |
| --- | --- |
| hSessionHandle | 设 备建会话句硒 |
| hKcyHandle | 脂定的密钥 柄 |
| uiAlgID[ in ] | 法 标 指定分组密码算法和二作模式 |
| pucStartVar[i 逼 | 缓 迪 区指 年 下 放 输 大 的 开始变量s |
| LiStahVarlengthliol | 输入的开始变量长度 |
| pueAadn] | 缓冲区指针，用于存放输入的额外的可鉴别数据 |
| uiAadLengih[in ] | 输入的额外的可鉴别数据长度 |
| pucData | 缓冲区指针，用于存放输入的数据明文 |
| uiDataLength [ in | 输人的数据明文长度 |
| pucEncData[out] | 缓冲区指钟，用于存放输出的数据密文 |
| puiEncDataLength[out] | 输出的数据密文长度 |
| pucAuthData[out] | 缓冲区指针，用于存放输出的鉴别数据 |
| puiAuthDataLength[out] | 输出的鉴别数据长度 |

返回值： 0

非 0

**6.5.6** **单包可鉴别解密**

原型： LONG SDF\_AuthDec(

成功

失败，返回错误代码

**GM/T 0018—2023**

HANDLE hSessionHandle, HANDLE hKeyHandle,

ULONG uiAlgID,

BYTE \*pucStartVar,

ULONG uiStartVarLength, BYTE \*pucAad,

ULONG uiAadLength, BYTE \*pucAuthData,

ULONG \*puiAuthDataLength, BYTE\*pucEncData,

ULONG uiEncDataLength, BYTE \*pucData,

ULONG\*puiDataLength);

描述： 使用指定的密钥句柄对数据进行可鉴别解密运算。此函数应用于可鉴别解密的CCM 和

GCM模式，输入输出数据应按照GM/T 0006定义的算法标识和 GB/T 36624的规定进 行确定。

参数：

hSessionHandle[in] hKeyHandle[in]

uiAlgID[in]

pucStartVar[in]

uiStartVarLength[in] pucAad[in]

uiAadLength[in]

pucAuthData[in]

puiAuthDataLength[in] pucEncData[in]

puiEncDataLength[in] pucData[out]

puiDataLength[out]

与设备建立的会话句柄 指定的密钥句柄

算法标识，指定分组密码算法和工作模式 缓冲区指针，用于存放输入的开始变量S 输入的开始变量长度

缓冲区指针，用于存放输入的额外的可鉴别数据 输入的额外的可鉴别数据长度

缓冲区指针，用于存放输入的鉴别数据 输入的鉴别数据长度

缓冲区指针，用于存放输入的数据密文 输入的数据密文长度

缓冲区指针，用于存放输出的数据明文 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

**GM/T 0018—2023**

**6.5.7** **多包对称加密初始化**

原型： LONG SDF\_EncryptInit(

HANDLE hSessionHandle,

HANDLE hKeyHandle, ULONG uiAlgID

BYTE \*pucIV,

ULONG uiIVLength);

描述： 多包数据加密初始化，设置加密密钥句柄、IV 和算法标识。

参数：

hSessionHandle[in]

hKeyHandle[in] uiAlgID[in]

pucIV[in]

uiIVLength[in]

与设备建立的会话句柄 指定的密钥句柄

算法标识，指定分组密码算法和工作模式 缓冲区指针，用于存放输入的IV 数据

输入的IV 数据长度

返回值： 0 成功

非 0 失败，返回错误代码

**6.5.8** **多包对称加密**

原型： LONG SDF\_EncryptUpdate(

HANDLE hSessionHandle,

BYTE \*pucData,

ULONG uiDataLength BYTE \*pucEncData

ULONG \*puiEncDataLength);

描述： 对多包数据进行对称加密运算。此函数不对数据进行填充处理。

参数：

hSessionHandle[in] pucData[in]

uiDataLength[in]

pucEncData[out]

puiEncDataLength[out]

与设备建立的会话句柄

缓冲区指针，用于存放输入的数据明文 输入的数据明文长度

缓冲区指针，用于存放输出的数据密文 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

**GM/T 0018—2023**

**6.5.9** **多包对称加密结束**

原型： LONG SDF\_EncryptFinal(

HANDLE hSessionHandle, BYTE \*pucLastEncData,

ULONG \*puiLastEncDataLength);

描述： 多包对称加密结束，释放相关资源。输出的最后部分密文可以为空。

参数：

hScssionHandle[in] pucLastEncData[out

puiLastEncDataLength[out]

与设备建立的会话句柄

缓冲区指针，用于存放输出的最后部分密文 输出的最后部分密文长度

返回值： 0 成功

非 0 失败，返回错误代码

**6.5.10** **多包对称解密初始化**

原型： LONG SDF\_DecryptInit(

HANDLE hSessionHandle, HANDLE hKeyHandle,

ULONG uiAlgID,

BYTE \*pucIV,

ULONG uiIVLength);

描述： 多包数据解密初始化，设置解密密钥句柄、IV 和算法标识。

参数：

hSessionHandle[in]

hKeyHandle[in] uiAlgID[in]

pucIV[in]

uiIVLength[in]

与设备建立的会话句柄 指定的密钥句柄

算法标识，指定分组密码算法和工作模式 缓冲区指针，用于存放输入的IV 数据

输入的IV 数据长度

返回值： 0 成功

非 0 失败，返回错误代码

**6.5.11** **多包对称解密**

原型： LONG SDF\_DecryptUpdate(

HANDLE hSessionHandle, BYTE \*pucEncData,

**GM/T 0018—2023**

ULONG uiEncDataLength, BYTE \*pucData,

ULONG \*puiDataLength);

描述： 对多包数据进行对称解密运算。

参数：

hSessionHandle[in] pucEncData[in]

uiEncDataLength[in] pucData[out]

puiDataLength[out

与设备建立的会话句柄

缓冲区指针，用于存放输入的数据密文 输入的数据密文长度

缓冲区指针，用于存放输出的数据明文 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

**6.5.12** **多包对称解密结束**

原型： LONG SDF\_DecryptFinal(

HANDLE hSessionHandle, BYTE \*pucLastData,

ULONG \*puiLastDataLength);

描述： 多包对称解密结束，释放相关资源。输出的最后部分明文可以为空。

参数：

hSessionHandle[in]

pucLastData[out]

puiLastDataLength[out]

与设备建立的会话句柄

缓冲区指针，用于存放输出的最后部分明文 输出的最后部分明文长度

返回值： 0 成功

非 0 失败，返回错误代码

**6.5.13** **多包MAC** **初始化**

原型： LONG SDF\_CalculateMACInit(

HANDLE hSessionHandle, HANDLE hKeyHandle,

ULONG uiAlgID,

BYTE \*pucIV

ULONG uiIVLength);

描述： 多包对称 MAC 初始化，设置MAC 密钥句柄、IV 和算法标识。对称MAC算法标识的约

定同分组密码算法。

|  |  |
| --- | --- |
|  | **GM/T 0018—2023** |
| 参数： hSessionHandle[in]  hKeyHandle[in] uiAlgID[in]  pucIV[in  uilVLength[in]  返回值： 0  非 0 | 与设备建立的会话句柄 指定的密钥句柄  算法标识，指定分组密码算法  缓冲区指针，用于存放输入的IV 数据 输入的IV 数据长度  成功  失败，返回错误代码 |

**6.5.14 多包MAC 计算**

原型： LONG SDF\_CalculateMACUpdate(

HANDLE hSessionHandle, BYTE \*pucData,

ULONG uiDataLength);

描述： 对多包数据进行对称MAC 计算。此函数不对数据进行填充处理。

参数：

hSessionHandle[in] pucData[in]

uiDataLength[in]

与设备建立的会话句柄

缓冲区指针，用于存放输入的数据 输入的数据长度

返回值： 0 成功

非0 失败，返回错误代码

**6.5.15 多包MAC 结束**

原型： LONG SDF\_CalculateMACFinal(

HANDLE hSessionHandlc, BYTE \*pucMac,

ULONG \*puiMacLength);

描述： 多包对称MAC结束，返回MAC 计算结果，释放相关资源。

参数：

hSessionHandle[in] pucMac[out]

puiMacLcngth[out]

与设备建立的会话句柄

缓冲区指针，用于存放输出的MAC 值 输出的MAC 值长度

返回值： 0 成功

非 0 失败，返回错误代码

**GM/T 0018—2023**

**6.5.16**

原型：

描述：

参数：

返回值：

**多包可鉴别加密初始化**

LONG SDF\_AuthEncInit(

HANDLE hSessionHandle, HANDLE hKeyHandle,

ULONG uiAlgID

BYTE \*pucStartVar,

ULONG uiStartVarLength

BYTE ×pucAad,

UAONG yiAadLength

ULONG uiDataLength);

多包数据可鉴别加密初始化，设置加密密钥句柄、算法标识、开始变量、额外的可鉴别数 据。此函数应用于可鉴别加密的 CCM 和GCM 模式，输入输出数据应接照GM/T 0006 定义的算法标识和GB/ T3662 1的 规 定进行 确 定 。GCM 模 式下的数据明文总长度参数

|  |  |
| --- | --- |
| 可为空。  hSessionHand n  hReyHandle 备 u AlgID[in事  pucStartVar [ i  uiSartVarLengih 信  pucAad[ in  uAadLength [in ] uiDataLength [in ]  0  非 0 | 没 备建立的会话的 定的密柄  法 标 识 分组密偶算法和工作模 级中 区指 ，用于存放输人的开始变量s 输入的开始变长度  缓 冲区指针，用于存放输入的额外的可鉴别数据 输入的额外的可鉴别数据长度  输入的数据明文总长度 成功  失败，返回错误代码 |

**6.5.17** **多包可鉴别加密**

原型： LONG SDF\_AuthEncUpdate(

HANDLE hSessionHandle, BYTE\*pucData,

ULONG uiDataLength, BYTE\*pucEncData,

ULONG\*puiEncDataLength);

描述： 对多包数据进行可鉴别加密运算。此函数的输出不包括鉴别数据。

**GM/T 0018—2023**

|  |  |  |
| --- | --- | --- |
| 参数： | hSessionHandle[in] pucData[in  uiDataLength[in]  pucEncData[out]  puiEncDataLength[out] | 与设备建立的会话句柄  缓冲区指针，用于存放输入的数据明文 输入的数据明文长度  缓冲区指针，用于存放输出的数据密文 输出的数据密文长度 |
| 返回值： | 0  非 0 | 成功  失败，返回错误代码 |

**6.5.18** **多包可鉴别加密结束**

原型： LONG SDF\_AuthEncFinal(

HANDLE hSessionHandle, BYTE \*pucLastEncData,

ULONG \*puiLastEncDataLength, BYTE \*pucAuthData,

ULONG \*puiAuthDataLength);

多包可鉴别加密结束，释放相关资源。此函数的输出数据应按照GM/T 0006定义的算 法标识和GB/T 36624 的规定进行确定，输出的最后部分密文可为空。

描述： 参数：

hSessionHandle[in]

pucLastEncData[out]

puiLastEncDataLength[out] pucAuthData[out]

puiAuthDataLength[out]

与设备建立的会话句柄

缓冲区指针，用于存放输出的最后部分密文 输出的最后部分密文长度

缓冲区指针，用于存放输出的鉴别数据 输出的鉴别数据长度

返回值： 0 成功

非 0 失败，返回错误代码

**6.5.19** **多包可鉴别解密初始化**

原型： LONG SDF\_AuthDecInit(

HANDLE hSessionHandle, HANDLE hKeyHandle,

ULONG uiAlgID,

BYTE \*pucStartVar,

ULONG uiStartVarLength, BYTE \*pucAad,

**GM/T 0018—2023**

ULONG uiAadLength,

BYTE \*pucAuthData,

ULONG uiAuthDataLcngth, ULONG uiDataLength);

描述： 多包数据可鉴别解密初始化，设置解密密钥句柄、算法标识、开始变量、额外的可鉴别数

据。此函数应用于可鉴别解密的CCM 和GCM 模式，输入输出数据应按照GM/T 0006

参数：

定义的算法标识和GB/T 可为空。

hSessionHandle[in] hKeyHandle[n]

uiAlgIDjin ]

pucStartVarl in

yiStartVarLength[in ] pucAad[ in]

uiAadLength [ i pucAuthData[

36624 的规定进行确定。GCM 模式下的数据明文总长度参数

与设备建立的会语向柄 指定的密钥句柄

算法标识，指定分组密码算法和工作模式 缓冲区指针，用于存放输入的开始变量S 输入的开始变量长度

级冲区指钍，用于存放输入的额外的面鉴别数据 输入的额外的再鉴数居长度

中 区 指 用 于 存放输天的鉴别数据

uiAuthDatafength 帘 临入 的 鉴房 据长度

uiDataLength 入的数据 文总长愈

返回值： 0 成功

非 0 失贩，反回错误代码

**6.5.20** **多包可鉴别解密**

原型： LONG SDF\_AuthDecUpdate(

HANDLE hSessionHandle, BYTE\*pucEncData

ULONG uiEncDataLcngth, BYTE \*pucData,

ULONG \*puiDataLength);

描述： 对多包数据进行可鉴别解密运算。

参数：

hSessionHandle[in] pucEncData[in]

uiEncDataLength[in] pucData[out]

与设备建立的会话句柄

缓冲区指针，用于存放输入的数据密文 输入的数据密文长度

缓冲区指针，用于存放输出的数据明文

|  |  |
| --- | --- |
|  | **GM/T 0018—2023** |
| puiDataLength[out]  返回值： 0  非 0 | 输出的数据明文长度 成功  失败，返回错误代码 |

**6.5.21** **多包可鉴别解密结束**

原型： LONG SDF\_AuthDecFinal(

HANDLE hScssionHandlc, BYTE \*pucLastData,

ULONG \*puiLastDataLength);

描述： 多包可鉴别解密结束，释放相关资源。输出的最后部分明文可为空。

参数：

hSessionHandle[in]

pucLastData[out]

puLastDataLcngth[out]

与设备建立的会话句柄

缓冲区指针，用于存放输出的最后部分明文 输出的最后部分明文

返回值： 0 成功

非 0 失败，返回错误代码

**6.6** **杂凑运算类函数**

**6.6.1** **概述**

杂凑运算类函数提供杂凑运算功能，表14列出了杂凑运算类函数，附录A 定义了函数返回代码。

**表14** **杂凑运算类函数**

|  |  |
| --- | --- |
| 函数名称 | 功能 |
| SDF\_HMACInit | 带密钥的杂凑运算初始化 |
| SDF\_HMACUpdate | 带密钥的多包杂凑运算 |
| SDF\_HMACFinal | 带密钥的杂凑运算结束 |
| SDF\_Hashlnit | 杂凑运算初始化 |
| SDF\_HashUpdatc | 多包杂凑运算 |
| SDF\_HashFinal | 杂凑运算结束 |

**6.6.2** **带密钥的杂凑运算初始化**

原型： LONG SDF\_HMACInit(

HANDLE hSessionHandle, HANDLE hKeyHandle,

ULONG uiAlgID);

**GM/T 0018—2023**

描述： 三步式带密钥的数据杂凑运算第一步。本函数执行带密钥的杂凑运算过程应符

合GB/T 15852.2。

参数： hSessionHandle[in]

hKeyHandle[in] uiAlgID[in]

返回值： 0

非 0

**6.6.3** **带密钥的多包杂凑运算**

与设备建立的会话句柄

指定的密钥句柄 指定杂凑算法标识 成功

失败，返回错误代码

原型： LONG SDF\_HMACUpdate(

HANDLE hSessionHandle, BYTE \*pucData,

ULONG uiDataLength);

描述： 三步式带密钥的数据杂凑运算第二步，对输入的明文进行杂凑运算。本函数执行带密钥

的杂凑运算过程应符合GB/T 15852.2。

|  |  |  |
| --- | --- | --- |
| 参数： | hSessionHandle[in] pucData[in]  uiDataLength[in] | 与设备建立的会话句柄  缓冲区指针，用于存放输入的数据明文 输入的数据明文长度 |
| 返回值： | 0  非 0 | 成功  失败，返回错误代码 |

**6.6.4** **带密钥的杂凑运算结束**

原型： LONG SDF\_HMACFinal(

HANDLE hSessionHandle, BYTE \*pucHMac,

ULONG \*puiHMacLength);

描述： 三步式带密钥的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。本

函数执行带密钥的杂凑运算过程应符合GB/T 15852.2。

参数： hSessionHandle[in] 与设备建立的会话句柄

pucHMac[out] 缓冲区指针，用于存放输出的杂凑数据

puiHMacLength[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

**GM/T 0018—2023**

**6.6.5** **杂凑运算初始化**

原型： LONG SDF\_HashInit(

HANDLE hScssionHandlc, ULONG uiAlgID

ECCrefPublicKey \*pucPublicKey, BYTE \*pucID,

ULONG uiIDLength);

描述： 三步式数据杂凑运算第一步。uiIDLength 非零且uiAlgID 为 SGD\_SM3时，本函数执行

的是SM2 的预处理1操作，此时 pucPublicKcy 不能为空，SM2 算法预处理过程应符合 GB/T 35276。

|  |  |  |
| --- | --- | --- |
| 参数： | hSessionHandle[in]  uiAlgID[in]  pucPublicKey[in] pucID[in]  uilDLength[in] | 与设备建立的会话句柄 指定杂凑算法标识  签名者公钥，当uiAlgID为SGD\_SM3时有效  签名者的ID 值，当uiAlgID为 SGD\_SM3 时有效 签名者ID 的长度，当uiAlgID 为SGD\_SM3时有效 |
| 返回值： | 0  非 0 | 成功  失败，返回错误代码 |

**6.6.6** **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： hScssionHandle[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 |

**6.6.7** **杂凑运算结束**

原型： LONG SDF\_HashFinal(

HANDLE hSessionHandle,

**GM/T 0018—2023**

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] pucHash[out

puiHashLength[out]

与设备建立的会话句柄

缓冲区指针，用于存放输出的杂凑数据 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**

**6.7.1** **概述**

用户文件操作类函数提供文件的创建、读写和删除功能，表15列出了用户文件操作类函数，附录A 定义了函数返回代码。

**表15** **用户文件操作类函数**

|  |  |
| --- | --- |
| 函数名称 | 功能 |
| SDF CrcatcFile | 创建文件 |
| SDF\_ReadFile | 读取文件 |
| SDF\_WritcFilc | 写文件 |
| SDF\_DeleteFile | 删除文件 |

**6.7.2** **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

**GM/T 0018—2023**

**6.7.3** **读取文件**

原型： LONG SDF\_ReadFile(

HANDLE hSessionHandle, LPSTR pucFileName,

ULONG uiNameIen

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] pucEueName in

uiNameLen[ i

hiOffset[ in ]

puiFileLength [inlou pucBuffer [ou

与设备建立的会话句柄

缓冲区指针，用于存放输入的文件名，最大长度128字节 文件名长度

指定读取文性时的偏移值

入参谢指啶读取文作内容的长度出参时返回实际读取 及内 容 的长度

区指钋于 存放 读 取的 文件 数 据

返回值： 0 

非 0 失败返圆错误代码

**6.7.4** **写文件**

原型： LONG SDF WriteFile(

HANDLE hSessionHandlc,

LPSTR pucFheName ULONG NameLen 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] pucFileName[in]

与设备建立的会话句柄

缓冲区指针，用于存放输入的文件名，最大长度128字节

**GM/T 0018—2023**

uiNameLen[in] uiOffset[in]

uiFileLength[in] pucBuffer[in]

返回值： 0

非0

文件名长度

指定写入文件时的偏移值 指定写入文件内容的长度

缓冲区指针，用于存放输入的写文件数据 成功

失败，返回错误代码

**6.7.5** **删除文件**

原型： LONG SDF\_DeleteFile(

HANDLE hSessionHandle, LPSTR pucFileName,

描述： 参数：

ULONG uiNameLen);

删除指定文件名的密码设备内部存储用户数据的文件。 hSessionHandle[in] 与设备建立的会话句柄

pucFileName[in] uiNamcLen[in]

返回值： 0

非 0

缓冲区指针，用于存放输入的文件名，最大长度128字节 文件名长度

成功

失败，返回错误代码

**6.8** **验证调试类函数**

**6.8.1** **概述**

验证调试类函数仅用于在调试、测试、检测场景下对产品的算法和功能进行验证，不用于实际密码 服务。表16列出了验证调试类函数，附录 A 定义了函数返回代码。

**表16** **验证调试类函数**

|  |  |
| --- | --- |
| 函数名称 | 功能 |
| SDF\_GenerateKeyPair\_RSA | 产生RSA非对称密钥对并输出 |
| SDF\_GenerateKeyPair\_ECC | 产生ECC非对称密钥对并输出 |
| SDF\_ExternalPrivateKeyOpcration\_RSA | 外部私钥RSA运算 |
| SDF\_ExternalSign ECC | 外部私钥ECC签名 |
| SDF\_ExternalDecrypt\_ECC | 外部私钥ECC解密 |
| SDF\_ExternalSign SM9 | 外部私钥SM9签名 |
| SDF\_ExternalDecrypt\_SM9 | 外部私钥SM9解密 |
| SDF\_ExternalKeyEncrypt | 外部密钥单包对称加密 |

**GM/T 0018—2023**

**表16** **验证调试类函数(续)**

|  |  |
| --- | --- |
| 函数名称 | 功能 |
| SDF ExternalKeyDecrypt | 外部密钥单包对称解密 |
| SDF\_ExternalKeyEncryptInit | 外部密钥多包对称加密初始化 |
| SDF\_ExiernalKeyDecryptInit | 外部密钥多包对称解密初始化 |
| SDF\_ExternalKcyHMACInit | 带外部密钥的杂凑运算初始化 |

**6.8.2 产生** **RSA非对称密钥对并输出**

原型： LONG SDF\_ GomerateKeyPai\_RSA (

ULONG KeyBits

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：

uiKcyBits [in ]

pucPublicKey [ o pucPrivateKcylout

指定密钥模长 R 公 钥 缆

私 钥结

返回值： 0 

非 0 失败 返风错误代码

**6.8 .3 产生ECC非对称密钥对并输出**

原型： LONG SDF \_GenerateKcyPair\_ECC(

ULONG uiAlgID,

UBONG mKeyBits,

ECCrefPublicKey \* pucPublicKey

ECCrefPrivateKey \*picPrivateKey )

描述： 请求密码设备产生指定类型和模长的 ECC 密钥对。

参数：

uiAlgID[in]

uiKeyBits [in]

pucPublicKey[out] pucPrivateKey[out

指定算法标识 指定密钥长度 ECC公钥结构 ECC 私钥结构

返回值： 0 成功

非 0 失败，返回错误代码

**GM/T 0018—2023**

**6.8.4 外部私钥RSA运算**

原型： LONG SDF\_ExternalPrivateKeyOperation\_RSA(

RSArefPrivateKey \*pucPrivateKey,

BYTE \*pucDataInput,

ULONG uilnputLength, BYTE \*pucDataOutput,

ULONG \*puiOutputLength);

描述： 指定使用外部RSA 私钥对数据进行运算。数据格式由应用层封装。

参数：

pucPrivateKey [in]

pucDataInput [in]

uiInputLength[in]

pucDataOutput[out] puiOutputLength[out]

外部RSA 私钥结构

缓冲区指针，用于存放输入的数据 输入的数据长度

缓冲区指针，用于存放输出的数据 输出的数据长度

返回值： 0 成功

非 0 失败，返回错误代码

**6.8.5 外部私钥ECC签名**

原型： LONG SDF\_ExternalSign\_ECC(

ULONG uiAlgID,

ECCrcfPrivateKey \*pucPrivateKey, BYTE\*pucDataInput,

ULONG uiInputLength,

ECCSignature \*pucSignature);

描述： 指定使用外部ECC 私钥对数据进行签名运算。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，SM2算法预处理过程应符合GB/T 35276。

参数： uiAlgID[in] 算法标识，指定使用的ECC算法

pucPrivateKey [in] 外部ECC 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

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

**GM/T 0018—2023**

**6.8.6 外部私钥ECC解密**

原型： LONG SDF\_ExternalDecrypt\_ECC(

ULONG uiAlgID

ECCrefPrivateKey \*pucPrivatcKey, ECCCipher \*pucEncData,

BYTE \*pucData,

ULONG \*uiDataLength);

描述： 指定使用外部 ECC私钥对数据进行解密运算。

参数：

uiAlgID[in]

pucPrivateKey [in] pucEncData[in]

pucData[out]

uiDataLength[out]

算法标识，指定使用的 ECC算法 外部ECC私钥结构

缓冲区指针，用于存放输入的数据密文 缓冲区指针，用于存放输出的数据明文 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

**6.8.7 外部私钥SM9签名**

原型： int SDF\_ExternalSign\_SM9(

SM9SignMasterPublicKey \*pSignMastPublicKey, SM9SignUserPrivateKey \*pSignUscrPrivatcKcy, BYTE pucData,

ULONG uiDataLength

SM9Signature \*pSignature);

描述： 使用外部签名主公钥和外部SM9 用户签名私钥对数据进行签名运算。输入数据为待签

数据的SM3 杂凑值。

参数： pSignMastPublicKey[in]

pSignUserPrivateKey[in]

pucData[in]

uiDataLength[in] pSignature[out]

返回值： 0

非0

外部输入的签名主公钥 外部用户签名私钥

缓冲区指针，用于存放外部输入的待签名数据 输入的数据长度

缓冲区指针，用于存放输出的签名值数据 成功

失败，返回错误代码

**GM/T 0018—2023**

**6.8.8 外部私钥SM9 解密**

原型：

int SDF\_ExternalDecrypt\_SM9(

SM9EncUserPrivateKey BYTE \*pucUserID, ULONG uiUserIDLen BYTE \*pucIV,

\*pEncUserPrivateKey,

BYTE \*pucData,

ULONG uiDataLength, SM9Cipher \*pEncData);

描述： 使用用户加密密钥对(包括用户私钥和用户标识)对数据进行解密运算

参数：

pEncUserPrivateKey[in]

pucUserID[in]

uiUserIDLen[in] ulAlgID[in]

pucIV [in]

pucData[in]

uiDataLength[in] pEncData [out]

外部输入的用户加密私钥 用户标识

用户标识的长度

加密明文的算法标识 输入的IV 数据

缓冲区指针，用于存放外部输入的密文数据 输入的数据长度

缓冲区指针，用于存放输出的明文

返回值： 0 成功

非0 失败，返回错误代码

**6.8.9** **外部密钥单包对称加密**

原型： LONG SDF\_ExternalKeyEncrypt(

ULONG uiAlgID,

BYTE \*pucKey,

ULONG uiKeyLength, BYTE \*pucIV,

ULONG uiIVLength, BYTE \*pucData,

ULONG uiDataLength,

**GM/T 0018—2023**

BYTE \*pucEncData,

ULONG \*puiEncDataLength);

描述： 使用外部密钥和IV 对数据进行对称加密运算。此函数不对数据进行填充处理。

参数：

uiAlgID[in]

pucKcy[in]

uiKeyLength[in] pucIV[in]

uiIVLength[in] pucData[in]

uiDataLength[in]

pucEncData[out]

puiEncDataLength[out]

算法标识，指定分组密码算法和工作模式 缓冲区指针，用于存放输入的密钥

输入的密钥长度

缓冲区指针，用于存放输入的IV 数据 输入的IV 长度

缓冲区指钋，用于存放输入的数据明文 输入的数据明文长度

缓冲区指针，用于存放输出的数据密文 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

**6.8.10** **外部密钥单包对称解密**

原型： LONG SDF\_ExternalKeyDecrypt(

ULONG uiAlgID

BYTE \*pucKey

ULONG uiKeyLength, BYTE \*pucIV,

ULONG uilVLength,

BYTE \*pucEncData,

ULONG uiEncDataLength, BYTE \*pucData,

ULONG \*puiDataLength);

描述： 使用外部密钥和IV 对数据进行对称解密运算。

参数：

uiAlgID[in]

pucKey[in]

uiKeyLcngth[in]

算法标识，指定分组密码算法和工作模式 缓冲区指针，用于存放输入的密钥

输入的密钥长度

**GM/T 0018—2023**

pucIV[in|out

uiIVLength[in]

pucEncData[in]

uiEncDataLength[in] pucData[out

puiDataLength[out]

返回值： 0

非0

缓冲区指针，用于存放输入的IV 数据 输入的IV 长度

缓冲区指针，用于存放输入的数据密文 输入的数据密文长度

缓冲区指针，用于存放输出的数据明文

输出的数据明文长度 成功

失败，返回错误代码

**6.8.11** **外部密钥多包对称加密初始化**

原型： LONG SDF\_ExternalKeyEncryptInit(

HANDLE hSessionHandle,

ULONG uiAlgID,

BYTE \*pucKey,

ULONG uiKeyLength, BYTE \*pucIV,

ULONG uiIVLength);

描述： 使用外部密钥的多包数据加密初始化，设置加密密钥、IV 和算法标识。

参数：

hSessionHandle[in]

uiAlgID[in]

pucKey[in]

uiKeyLength[in] pucIV[in]

uiIVLength[in]

与设备建立的会话句柄

算法标识，指定分组密码算法和工作模式 缓冲区指针，用于存放输入的密钥

输入的密钥长度

缓冲区指针，用于存放输入的IV 数据 输入的IV 数据长度

返回值： 0 成功

非0 失败，返回错误代码

**6.8.12** **外部密钥多包对称解密初始化**

原型： LONG SDF\_ExternalKeyDecryptInit(

HANDLE hSessionHandle, ULONG uiAlgID,

BYTE \*pucKey,

ULONG uiKeyLength,

**GM/T 0018—2023**

BYTE \*pucIV,

ULONG uiIVLength);

描述： 使用外部密钥的多包数据解密初始化，设置解密密钥、IV 和算法标识。

参数：

hSessionHandle[in]

uiAlgID[in]

pucKcy[in]

uiKeyLength[in] pucIV[in]

uiIVLength[in

与设备建立的会话句柄

算法标识，指定分组密码算法和工作模式 缓冲区指针，用于存放输入的密钥

输入的密钥长度

缓冲区指针，用于存放输入的IV 数据 输入的IV 数据长度

返回值： 0 成功

非 0 失败，返回错误代码

**6.8.13** **带外部密钥的杂凑运算初始化**

原型：

Lonc

SDF \_ExionalkeyUMAC

HANDIE 下SesstonHandl

ULONC iD BYTE eke

ULONG gth)

描述： 主步式带密钥的数据集法运算第一二 步。本函数执行形密研丽杂凑运算过程应符

否GB/15852.2

参数： hScssionHandlc [ in ] 与设备建立的会话句柄

uiAlgID[in 指定杂凑算法标识

pucKey[ in 缓冲区指针，用于存放输入的密钥

uiKeyLcngth [ in 输入的密钥长度

返回值： 0 成功

非 0 失败，返回错误代码

**GM/T 0018—2023**

**附** **录** **A**

**(规范性)**

**函数返回代码定义**

**A.1** **函数返回代码定义**

表A.1 定义了本文件的函数返回代码。

**表** **A.1** **函数返回代码定义**

|  |  |  |
| --- | --- | --- |
| 错误代码标识 | | |
| 宏描述 | 预定义值 | 说明 |
| #definc 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十0x00000003 | 与设备通信失败 |
| 井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 |
| 共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十0x00000009 | 不支持的算法调用 |
| 井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 |
| 井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十0x00000013 | 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 |

**GM/T 0018—2023**

**表** **A.1** **函数返回代码定义(续**)

|  |  |  |
| --- | --- | --- |
| 错误代码标识 | | |
| 宏描述 | 预定义值 | 说明 |
| =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士0x0000001E | 输出参数错误 |
| =define SDR\_USERIDERR | SDR\_BASE十0x0000001F | 用户标识错误 |
|  | SDR\_BASE +0x00000020 至  SDR\_BASE十0x00FFFFFF | 预留 |

**GM/T 0018—2023**

**附** **录** **B**

**(规范性)**

**SM9算法相关数据结构和接口函数**

**B.1 SM9 算法相关数据结构**

**B.1.1 SM9主私钥数据结构定义**

表 B.1定义了SM9 主私钥数据结构，用于SM9 签名主私钥和SM9 加密主私钥。

**表B.1 SM9 主私钥数据结构**

|  |  |  |
| --- | --- | --- |
| 字段名称 | 数据长度(字节) | 含义 |
| bits | 4 | 模数的实际位长度 |
| s | SM9ref\_MAX\_LEN | 主私钥 |

实际数据结构定义：

#define SM9ref\_MAX\_BITS 256

#define SM9ref\_MAX\_LEN ((SM9ref\_MAX\_BITS+7)/8)

typedef struct SM9refMasterPrivateKey\_st {

ULONG bits;

BYTE s[SM9ref\_MAX\_LEN]; }SM9MasterPrivateKey;

**B.1.2 SM9 签名主公钥数据结构定义**

表B.2 定义了SM9 签名主公钥数据结构。

**表B.2 SM9 签名主公钥数据结构**

|  |  |  |
| --- | --- | --- |
| 字段名称 | 数据长度(字节) | 含义 |
| bits | 4 | 模数的实际位长度 |
| xa | SM9ref\_MAX\_LEN | 曲线上点X坐标高维 |
| xb | SM9ref\_MAX\_LEN | 曲线上点X坐标低维 |
| ya | SM9ref\_MAX\_LEN | 曲线上点Y坐标高维 |
| yb | SM9ref\_MAX\_LEN | 曲线上点Y坐标低维 |

实际数据结构定义：

typedef struct SM9refSignMasterPublicKey\_st {

ULONG bits;

BYTE xa[SM9ref\_MAX\_LEN]; BYTE xb[SM9ref\_MAX\_LEN];

**GM/T 0018—2023**

BYTE ya[SM9rcf\_MAX\_LEN];

BYTE yb[SM9ref\_MAX\_LEN]; }SM9SignMastcrPublicKcy;

B.1.3 SM9 加密主公钥数据结构定义

表 B.3 定义了SM9 加密主公钥数据结构。

**表B.3 SM9 加密主公钥数据结构**

|  |  |  |
| --- | --- | --- |
| 字段名称 | 数据长度(字节) | 含义 |
| bits | 4 | 模数的实际位长度 |
| X | SM9ref\_MAX\_LEN | 曲线上点X坐标 |
| y | SM9ref\_MAX\_LEN | 曲线上点Y坐标 |

实际数据结构定义：

typedef struct SM9refEncMasterPublicKey\_st {

ULONG bits;

BYTE x[SM9ref\_MAX\_LEN];

BYTE y[SM9ref\_MAX\_LEN]; }SM9EncMasterPublicKey;

**B.1.4 SM9 用户签名私钥数据结构定义**

表 B.4 定义了SM9 用户签名私钥数据结构。

**表B.4** **SM9用户签名私钥数据结构**

|  |  |  |
| --- | --- | --- |
| 字段名称 | 数据长度(字节) | 含义 |
| bits | 4 | 模数的实际位长度 |
| X | SM9ref\_MAX\_LEN | 曲线上点X坐标 |
| y | SM9ref\_MAX\_LEN | 曲线上点Y坐标 |

实际数据结构定义：

typcdcf struct SM9refSignUserPrivateKey\_st {

ULONG bits;

BYTE x[SM9ref\_MAX\_LEN];

BYTE y[SM9ref\_MAX\_LEN]; }SM9SignUscrPrivateKey;

**B.1.5 SM9用户加密私钥数据结构定义**

表B.5 定义了SM9 用户加密私钥数据结构。

**GM/T 0018—2023**

**表B.5 SM9 用户加密私钥数据结构**

|  |  |  |
| --- | --- | --- |
| 字段名称 | 数据长度(字节) | 含义 |
| bits | 4 | 模数的实际位长度 |
| xa | SM9ref\_MAX\_LEN | 曲线上点X坐标高维 |
| xb | SM9ref\_MAX\_LEN | 曲线上点X坐标低维 |
| ya | SM9ref\_MAX\_LEN | 曲线上点Y坐标高维 |
| yb | SM9ref\_MAX\_LEN | 曲线上点Y坐标低维 |

实际数据结构定义：

typedef struct SM9refEncUserPrivateKey\_st {

ULONG bits;

BYTE xa[SM9ref\_MAX\_LEN]; BYTE xb[SM9ref\_MAX\_LEN]; BYTE ya[SM9ref\_MAX\_LEN];

BYTE yb[SM9ref\_MAX\_LEN]; }SM9EncUserPrivateKey;

**B.1.6 SM9 加密数据结构定义**

表B.6 定义了SM9 加密数据结构。

**表** **B.6 SM9 加密数据结构**

|  |  |  |
| --- | --- | --- |
| 字段名称 | 数据长度(字节) | 含义 |
| EncType | 4 | SM9加密算法机制 |
| X | SM9ref\_MAX\_LEN | 封装密文的X坐标 |
| y | SM9ref\_MAX\_LEN | 封装密文的Y坐标 |
| h | 32 | M A C 值 |
| L | 4 | 密文数据长度 |
| C | L | 密文数据 |

实际数据结构定义：

typedcf struct SM9refCipher\_st {

ULONG EncType;

BYTE x[SM9ref\_MAX\_LEN]; BYTE y[SM9ref\_MAX\_LEN];

BYTE h[32]; ULONG L;

BYTE C[ }SM9Cipher;

**GM/T 0018—2023**

**B.1.7 SM9 签名数据结构定义**

表B.7 定义了SM9 签名数据结构。

**表B.7 SM9签名数据结构**

|  |  |  |
| --- | --- | --- |
| 字段名称 | 数据长度(字节) | 含义 |
| h | SM9ref\_MAX\_LEN | 签名结果的H部分 |
| X | SM9ref\_MAX\_LEN | 签名结果S部分的X坐标 |
| y | SM9ref\_MAX\_LEN | 签名结果S部分的Y坐标 |

实际数据结构定义：

typedef struct SM9refSignaturc\_st



BYTE BYTE BYTE

h[SM9ref\_MAX\_LEN]; x[SM9ref\_MAX\_LEN]; y[SM9ref\_MAX\_LEN];

}SM9Signaturc;

**B.1.8** **SM9** **密钥封装数据结构定义**

表 B.8 定义了SM9 密钥封装数据结构。

**表B.8 SM9密钥封装数据结构**

|  |  |  |
| --- | --- | --- |
| 字段名称 | 数据长度(字节) | 含义 |
| X | SM9ref\_MAX\_LEN | 封装的交换密文曲线上点X坐标 |
| y | SM9ref\_MAX\_LEN | 封装的交换密文曲线上点Y坐标 |

实际数据结构定义：

typcdef struct SM9rcfKcyPackage\_st



BYTE BYTE

x[SM9ref\_MAX\_LEN]; y[SM9ref\_MAX\_LEN];

}SM9KeyPackage;

**B.1.9** **SM9用户加密密钥对保护结构定义**

表 B.9 定义了SM9 用户加密密钥对保护结构。

**表B.9** **SM9用户加密密钥对保护结构**

|  |  |  |
| --- | --- | --- |
| 字段名称 | 数据长度(字节) | 含义 |
| version | 4 | 版本号 |
| ulSymmAlgID | 4 | 对称算法标识，限定ECB模式 |
| bits | 4 | 用户加密密钥对的密钥位长 |

**GM/T 0018—2023**

**表** **B.9** **SM9用户加密密钥对保护结构(续)**

|  |  |  |
| --- | --- | --- |
| 字段名称 | 数据长度(字节) | 含义 |
| encryptedPriKey | SM9ref\_MAX\_LEN\*4 | 对称算法加密的密文，加密的原文为： SM9EncUserPrivateKey结构体中除去长 度的剩余部分 |
| encMasterPubKey | SM9EncMastPublicKey类型 | KGC的加密主公钥 |
| tmpMasterPubKcy | SM9EncMastPublicKey类型 | 产生对称密钥的密钥封装数据的主公钥 |
| userIDLen | 4 | 用户标识长度 |
| uscrID | 1024 | 用户标识，实际有效长度为userIDLen |
| keyLen | 4 | 对称密钥比特位数 |
| keyPackage | SM9KeyPackage类型 | 加密用户加密私钥的对称密钥的密钥封 装数据 |

实际数据结构定义：

typedef struct SM9refEncEnvclopcdKey\_st {

ULONG version;

ULONG ulSymmAlgID;

ULONG bits;

BYTE encryptedPriKey[SM9ref\_MAX\_LEN\*4]; SM9EncMastPublicKey encMastPubKey;

SM9EncMastPublicKey tmpMastPubKey; ULONG uscrIDLen;

BYTE uscrID[256]; ULONG keyLen;

SM9KeyPackage keyPackage; }SM9EncEnvelopedKey;

**B.1.10 SM9用户签名密钥对保护结构定义**

表 B.10定义了SM9 用户签名密钥对保护结构。

**表** **B.10** **SM9用户签名密钥对保护结构实际数据结构定义**

|  |  |  |
| --- | --- | --- |
| 字段名称 | 数据长度(字节) | 含义 |
| version | 4 | 版本号 |
| ulSymmAlgID | 4 | 对称算法标识，限定ECB模式 |
| bits | 4 | 用户加密密钥对的密钥位长 |
| encryptedPriKcy | SM9ref\_MAX\_LEN\*2 | 对称算法加密的密文，加密的原文为：  SM9SignUserPrivateKey结构体中除去长 度的剩余部分 |

**GM/T 0018—2023**

**表B.10** **SM9用户签名密钥对保护结构实际数据结构定义(续**)

|  |  |  |
| --- | --- | --- |
| 字段名称 | 数据长度(字节) | 含义 |
| signMastcrPubKey | SM9SignMastPublicKey类型 | KGC的签名主公钥 |
| tmpMasterPubKey | SM9EncMastPublicKey类型 | 产生对称密钥的密钥封装数据的主公钥 |
| userIDLen | 4 | 用户标识长度 |
| userID | 256 | 用户标识，实际有效长度为userIDLcn |
| keyLen | 4 | 对称密钥比特位数 |
| keyPackage | SM9KeyPackagc类型 | 加密用户加密私钥的对称密钥的密钥封  装数据 |

typedef struct SM9refSignEnvelopedKey\_st {

ULONG version

ULONG ulSymmAlgID;

ULONG bits

*enc: sptedPrikoyISM⁹aI MAX E*

BYTE

SM9SighMasPublicKey SM9 EncMas PublicKey ULONG userIDLen;

BYTE userID[256]; ULONG keyLen;

sgnMnse市 Key mFasPubKey

SM9 KcPackage keyPadkngo } SM9SignEn yelopedKey;

**B.2** **SM9** **算法相关接口函数**

**B.2.1 概述**

SM9 算法相关接口函数包括密钥管理类和非对称算法运算类接口数，表B.11 列出了SM9 算法相 关接口函数，附录 A定义了函数返同代码。

**表B.11** **SM9** **算法相关接口函数**

|  |  |
| --- | --- |
| 函数名称 | 功能 |
| SDF\_ExportUserID\_SM9 | 导出SM9用户标识 |
| SDF\_GenerateEncMasterKeyPair\_SM9 | 产生SM9临时加密主密钥对 |
| SDF\_GenerateKcyWithEMK\_SM9 | 生成会话密钥并用SM9外部加密主公钥和用户标识进行密钥封 装并输出 |
| SDF\_GenerateKeyWithIMK\_SM9 | 生成会话密钥并用SM9内部加密主公钥和用户标识进行密钥封 装并输出 |
| SDF\_ImportKcyWithISK\_SM9 | 使用SM9密钥封装结构导入会话密钥并用内部SM9加密密钥对 解封装 |

**GM/T 0018—2023**

**表B.11 SM9 算法相关接口函数(续)**

|  |  |
| --- | --- |
| 函数名称 | 功能 |
| SDF\_GenerateAgreementDataWithSM9 | SM9生成密钥协商参数并输出 |
| SDF\_GenerateKcyWithSM9 | SM9计算会话密钥 |
| SDF\_GenerateAgreementDataAndKeyWithSM9 | SM9产生协商数据并计算会话密钥 |
| SDF\_VerifyWithMasterEPK\_SM9 | 外部主公钥SM9验证 |
| SDF\_VerifyWithMasterIPK\_SM9 | 内部主公钥SM9验证 |
| SDF\_InternalSignWithMasterEPK\_SM9 | 外部主公钥SM9签名 |
| SDF\_InternalSignWithMasterIPK\_SM9 | 内部主公钥SM9签名 |
| SDF\_EncryptWithMasterEPK\_SM9 | 外部主公钥SM9加密 |
| SDF\_EncryptWithMasterIPK\_SM9 | 内部主公钥SM9加密 |
| SDF\_DecryptWithISK\_SM9 | 用户密钥对SM9解密 |

**B.2.2** **导出SM9** **用户标识**

|  |  |  |
| --- | --- | --- |
| 原型： | int | SDF\_ExportUserID\_SM9(  HANDLE hSessionHandle, ULONG uiKeyIndex,  BYTE \*pucUserID,  ULONG \*puiUserIDLen |

);

描述： 导出密码设备内部存储的指定索引位置的用户标识。

参数： hSessionHandle[in] 与设备建立的会话句柄

uiKeyIndex [in] 用户密钥索引

输出的用户标识缓冲区，若该项为 NULL, 则 puiUse-

pucUserID[out]

rIDLen 输出所需缓冲区长度

输入为pucUserID 的缓冲区长度，输出为pucUserID 实

puiUserIDLen[in|out]

际长度

返回值： 0 成功

非 0 失败，返回错误代码

**B.2.3** **产** **生SM9** **临时加密主密钥对**

原型： int SDF\_GenerateEncMasterKeyPair\_SM9(

HANDLE hSessionHandle,

**GM/T 0018—2023**

ULONG uiAlgID,

ULONG \*uiMasterKeyIndex,

SM9EncMastcrPublicKcy \*pEncMasterPuclicKcy);

描述： 本函数由用户密码设备调用，生成临时加密主密钥，并用于KGC 生 成SM9 用户密钥对保

护结构。当输入的 SM9 主密钥索引值为0xFFFFFFFF 时，输出的主密钥索引值由密码 设备内部产生。

|  |  |  |
| --- | --- | --- |
| 参数： | hSessionHandle[in]  uiAlgID[in]  uiMasterKeyIndex [imlout]  pEncMasterPaclicKey fout] | 与设备建立的会话句柄 指定算法标识  SM9 主密钥索引  输出的SM9 主公钥数据结构 |
| 返回值： | 0  非 0 | 成功  失败，返回错误代码 |

**B.2.4** **生成会话密钥并用SM** **9** **外部加密主公钥和用户标进行密钥封装并输出**

原型： int SDF\_GeneratcEezWEMIk \_SM9(

HANDLF hSesfonfandld ULONG 位Keyten

SM9 EncMostoPalcKey cM .ublick

BYTE\* CaD

LONGmUso LDLce

SM9 KeyPackagc \* pKeyPackagc, HANDLE \*phKeyHandle);

描述： 生成会话密钥并用用户标识加密的SM9 密钥封装结构导出，加密主公钥由外部输入。密

钥封装的具体过程应符合GB /T 38635 。密钥封装和解封装函数输入的对称密钥长度值 相同。

参数： hSessionHandle[in]

uiKeyLen[in]

pEncMasterPublicKey [in]

pucUserID[in]

uiUserIDLen[in] pKeyPackage[out] phKcyHandle[out]

与设备建立的会话句柄 对称密钥长度

外部输入的加密主公钥 用户标识

用户标识长度

输出的SM9 密钥封装结构 返回的密钥句柄

**GM/T 0018—2023**

返回值： 0 成功

非0 失败，返回错误代码

**B.2.5** **生成会话密钥并用SM9** **内部加密主公钥和用户标识进行密钥封装并输出**

原型： int SDF\_GenerateKeyWithIMK\_SM9(

HANDLE hSessionHandle, ULONG uiKeyLen,

ULONG uiMasterKeyIndex,

BYTE pucUserID, ULONG uiUserIDLen,

SM9KeyPackage \*pKeyPackage, HANDLE \*phKeyHandle);

描述： 生成会话密钥并用用户标识加密的SM9 密钥封装结构导出，加密主公钥由内部索引。密

钥封装的具体过程应符合GB/T 38635。密钥封装和解封装函数输入的对称密钥长度值 相同。

参数：

hSessionHandle[in] uiKeyLen[in]

uiMasterKeyIndex [in]

pucUserID[in]

uiUserIDLen [in] pKeyPackage[out] phKeyHandle[out]

与设备建立的会话句柄 对称密钥长度

内部加密主公钥索引值 用户标识

用户标识长度

输出的SM9 密钥封装结构 返回的密钥句柄

返回值： 0 成功

非0 失败，返回错误代码

**B.2.6** **使用SM9** **密钥封装结构导入会话密钥并用内部SM9** **加密密钥对解封装**

原型： int SDF\_ImportKeyWithISK\_SM9(

HANDLE hSessionHandle, ULONG uiISKIndex,

ULONG uiKeyLen,

SM9KeyPackage \*pKeyPackage, HANDLE \*phKeyHandle);

**GM/T 0018—2023**

描述： 导入会话密钥并根据指定索引用用户标识和用户加密私钥解密。密钥解封装的具体过程

应符合GB/T 38635,密码设备根据密钥封装结构产生指定长度对称密钥。

参数： hScssionHandlc[in] 与设备建立的会话句柄

uiISKIndex[in] 密码设备内部存储用户加密私钥的索引值，对应于密钥

封装使用的用户标识

uiKeyLen[in]

被封装的会话密钥长度

pKeyPackage[in]

SM9 密钥封装结构

phKeyHandlc[out]

返回的密钥句柄

返回值： 0 成功

非 0 失败，返回错误代码

**B.2.7** **SM9** **生成密钥协商参数并输出**

原型： int SDF\_GenerateAgreementDataWithSM9(

HANDLE hSessionHandle,

ULONG uiMasterKeyIndex,

ULONG uiISKIndex, ULONG uiKeyBits, BYTE\*pResponseID,

ULONG ulResponseIDLen, BYTE\*pSponsorID,

ULONG ulSponsorIDLen,

SM9EncMastPublicKey \*pPublicKey,

SM9EncMastPublicKey \*pSponsorTempPublicKey, HANDLE\*phAgreementHandle);

描述： 使用SM9 算法协商会话密钥，协商的发起方首先调用本函数。

参数：

hSessionHandle[in]

uiMasterKeyIndex[in] uiISKIndex[in]

uiKcyBits[in]

pResponseID[in]

ulResponseIDLen[in] pSponsorID[in]

与设备建立的会话句柄 内部加密主公钥索引

发起方用户加密私钥索引 协商后要求输出的密钥长度 响应方id

响应方id 长度 发起方id

**GM/T 0018—2023**

ulSponsorIDLen[in] pPublicKey[out]

pSponsorTempPublicKey[out] phAgreementHandle[out]

返回值： 0

非 0

发起方id 长度

返回的发起方主公钥

返回的发起方临时公钥

返回的协商句柄，用于计算协商密钥 成功

失败，返回错误代码

**B.2.8** **SM9计算会话密钥**

原型：

描述：

int SDF\_GenerateKeyWithSM9(

HANDLE hSessionHandle,

HANDLE hAgreementHandle,

SM9EncMastPublicKey \*pTempPublicKey, HANDLE \*phKeyHandle);

使用SM9 算法协商会话密钥，发起方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。使用 SM9 算法计算会话密钥的过程应符合GB/T 38635。 本函数由协商的发起方在获得响应方的协商参数后调用。会话密钥计算完成后，协商句

柄被销毁，为协商句柄分配的内存资源也被释放。

参数： hSessionHandle[in]

hAgreementHandle[in] pTempPublicKey[in] phKeyHandle[out

返回值： 0

非 0

与设备建立的会话句柄 协商句柄

响应方临时公钥 返回的密钥句柄 成功

失败，返回错误代码

**B.2.9 SM9 产生协商数据并计算会话密钥**

原型： int SDF\_GenerateAgreementDataAndKeyWithSM9(

HANDLE hSessionHandle,

ULONG uiMasterKeyIndex, ULONG uiISKIndex,

ULONG uiKeyBits,

BYTE \*pResponseID,

ULONG ulResponseIDLen, BYTE \*pSponsorID,

**GM/T 0018—2023**

ULONG ulSponsorIDLen,

SM9EncMastPublicKey \*pPublicKey,

SM9EncMastPublicKcy \*pSponsorTcmpPublicKcy, SM9EncMastPublicKey \*pResponseTempPublicKey, HANDLE \*phKeyHandle);

描述： 使用SM9 算法协商会话密钥，响应方产生协商参数并计算会话密钥，同时返回协商参数

和会话密钥句柄。

参数：

hSessionHandle[in]

uiMasterKeyIndex[in] uiISKIndex[in]

uiKeyBits[in]

pResponseID[in]

ulResponseIDLen[in] pSponsorID[in]

ulSponsorIDLen[in]

与设备建立的会话句柄 内部加密主公钥索引

响应方用户加密私钥索引

协商后要求输出的密钥长度 响应方id

响应方id长度 发起方id

发起方id 长度

pPublicKey[in] 发起方主公钥

pSponsorTempPublicKey[in] 发起方的临时主公钥

pResponseTempPublicKey[out] 返回的响应方临时主公钥

phKeyHandlc[out] 返回的密钥句柄

返回值： 0 成功

非 0 失败，返回错误代码

**B.2.10** **外部主公钥SM9** **验证**

|  |  |  |
| --- | --- | --- |
| 原型： | int | SDF\_VerifyWithMasterEPK\_SM9( HANDLE hSessionHandle,  SM9SignMasterPublicKey \*pSignPublicKcy,  BYTE \*pucUserID  ULONG uiUserIDLen, BYTE \*pucData  ULONG uiDataLength,  SM9Signature \*pSignature); |

描述： 使用外部签名主公钥和用户标识对 SM9 签名进行验证运算。输入数据为待签数据的

SM3 杂凑值。

**GM/T 0018—2023**

|  |  |  |
| --- | --- | --- |
| 参数： | hSessionHandle[in]  pSignPublicKey [in]  pucUserID[in  uiUserIDLen[in] pucData[in  uiDataLength[in] pSignature [in] | 与设备建立的会话句柄 外部输入的签名主公钥 用户标识  用户标识的长度  待验证签名的数据 数据长度  待验证签名值 |
| 返回值： | 0  非0 | 成功  失败，返回错误代码 |

**B.2.11 内部主公钥** **SM9 验证**

原型： int SDF\_VerifyWithMasterIPK\_SM9(

HANDLE hSessionHandle,

ULONG uiSignMasterKeyIndex,

BYTE \*pucUserID,

ULONG uiUserIDLen, BYTE \*pucData,

ULONG uiDataLength,

SM9Signature \*pSignature);

描述： 使用内部签名主公钥和用户标识对 SM9 签名进行验证运算。输入数据为待签数据的

SM3 杂凑值。

|  |  |  |  |
| --- | --- | --- | --- |
| 参数： | hSessionHandle[in]  uiSignMasterKeyIndex  pucUserID[in  uiUserIDLen[in] pucData[in  uiDataLength[in] pSignature [in] | [in | 与设备建立的会话句柄 内部签名主公钥的索引值 用户标识  用户标识的长度  待验证签名的数据 数据长度  待验证签名值 |

返回值： 0 成功

非 0 失败，返回错误代码

**B.2.12** **外部主公钥SM9** **签名**

原型： int SDF\_InternalSignWithMasterEPK\_SM9(

**GM/T 0018—2023**

HANDLE hSessionHandle,

SM9SignMasterPublicKey \*pSignPublicKey,

ULONG uiISKIndcx, BYTE\*pucData,

ULONG uiDataLength,

SM9Signaturc \*pSignature);

描述： 使用外部签名主公钥和SM9 用户签名私钥对数据进行签名运算。输入数据为待签数据

的 SM3 杂凑值。

|  |  |  |
| --- | --- | --- |
| 参数： | hSessionHandle [ in pSignPublicbcy [in uiISKIndex [i  pucData [ in  uiDataLengthin pSignature [oui | 与设备建立的会话句柄  外部输入的签名主公钥  内部用户签名私钥的索引值  缓冲区指针，用于存放外部输入的待签名数据  输人的数据长度  缓河指键 用于存放输出的签径值数据 |
| 返回值： | 0  非 0 | 成  败，返回 代码 |

**B.2.13** **内部主公钥** **SM9** **签备**

原型： nt SDF \_InternalSigrVMasterIPK\_ SM9(

HANDLE hScssionHLande

OLONG uiMasterKeyIndex, ULONG uiISKIndex,

BYTE\*pucData,

ULONG uiDataLength,

SM9Signature \* pSignature );

描述： 使用内部签名主公钥和SM9 用户签名私钥对数据进行签名运算。输入数据为待签数据

的SM3 杂凑值。

参数：

hSessionHandle[in]

uiMasterKeyIndex [in] uiISKIndex [in]

pucData[in]

uiDataLength[in] pSignaturc [out]

与设备建立的会话句柄

内部签名主公钥的索引值

内部用户签名私钥的索引值

缓冲区指针，用于存放外部输入的待签名数据 输入的数据长度

缓冲区指针，用于存放输出的签名值数据

**GM/T 0018—2023**

|  |  |
| --- | --- |
| 返回值： 0 | 成功 |
| 非0 | 失败，返回错误代码 |

**B.2.14 外部主公钥SM9加密**

原型： int SDF\_EncryptWithMasterEPK\_SM9(

HANDLE hSessionHandle,

SM9EncMasterPublicKey\*pEncPublicKey,

BYTE \*pucUserID,

ULONG uiUserIDLen, ULONG ulAlgID,

BYTE \*pucIV,

BYTE\*pucData,

ULONG uiDataLength, SM9Cipher \*pEncData);

描述： 使用用户标识和外部加密主公钥对数据进行加密运算。加密的具体过程应符合GB/T 38635。

IV 数据长度为16个字节，仅当uiAlgID为 SGD\_SM9 8 CBC、SGD\_SM9 8 OFB、SGD\_ SM9 8 CFB时有效。

|  |  |  |
| --- | --- | --- |
| 参数： | hSessionHandle[in] pEncPublicKey [in]  pucUserID[in]  uiUserIDLen[in] ulAlgID[in]  pucIV [in]  pucData[in7  uiDataLength[in pEncData [out] | 与设备建立的会话句柄 外部输入的加密主公钥 用户标识  用户标识的长度  加密明文的算法标识 输入的IV 数据  缓冲区指针，用于存放外部输入的数据 输入的数据长度  缓冲区指针，用于存放输出的数据密文 |
| 返回值： | 0  非 0 | 成功  失败，返回错误代码 |

**B.2.15 内部主公钥SM9 加密**

原型： int SDF\_EncryptWithMasterIPK\_SM9(

HANDLE hSessionHandle,

ULONG uiMasterKeyIndex,

**GM/T 0018—2023**

BYTE \*pucUserID,

ULONG uiUserIDLen, ULONG ulAlgID,

BYTE pIV,

BYTE \*pucData

ULONG uiDataLength SM9Cipher \*pEncData);

描述： 使用用户标识和内部加密主公钥对数据进行加密运算。加密的具体过程应符合GB/T 38635。

IV 数据长度为16个字节，仅当miAlgHD为 SGD\_SM9 8 CBC 、SGD\_SM9 8 OFB 、SGD\_ SM9 8 CFB 时有效。

|  |  |  |
| --- | --- | --- |
| 参数： | hSessionHandle in ]  uiMasterKeyIndex [in ]  pugUserD[ in  biUscriDLen [ ing ulAlgID[in ]  pIV [ in ]  pucData[ in  uiDalaLength[ in pEncData [out ] | 与设备建立的会话句柄  内部加密主公钥的索引值  用户标识  相际识的长度  加密明文的算法标识 的 IV 数  通 区 指钋 于存放外部输人的数据 输人的数抗飞度  缓冲区指针、川工存放偷的数据密文 |
| 返回值： | 0  非 | 成 功  失败，返回错误代码 |

**B.2.16** **用户密钥对SM9** **解密**

原型： int SDF \_DecryptWithISK SM9(

HANDLE hSessionHandle, ULONG uiISKIndex,

BYTE pucUserID,

ULONG uiUscrIDLen, ULONG ulAlgID,

BYTE \*pucIV.

SM9Cipher \*pEncDate,

ULONG \*puiDataLength, BYTE \*pucData);

**GM/T 0018—2023**

描述： 使用用户加密密钥对(包括用户私钥和用户标识)对数据进行解密运算。解密的具体过程

应符合GB/T 38635 。IV数据长度为16个字节，仅当uiAlgID 为 SGD\_SM9 8 CBC、 SGD\_SM9 8 OFB 、SGD\_SM9 8 CFB时有效。输入的用户标识应和用户加密密钥对索 引值对应的用户标识一致。

参数：

hSessionHandle[in]

uiISKIndex[in]

pucUserID[in]

uiUserIDLen[in] ulAlgID[in]

pucIV [in]

pEncData[in7

puiDataLength[out] pucData [out]

与设备建立的会话句柄

用户加密密钥对索引值，该密钥对包括加密私钥和用户 标识

用户标识

用户标识的长度

加密明文的算法标识 输入的IV 数据

缓冲区指针，用于存放外部输入的密文数据 输出的数据长度

缓冲区指针，用于存放输出的明文

返回值： 0 成功

非 0 失败，返回错误代码

**GM/T 0018—2023**

**附** **录** **C**

**(规范性)**

**VPN** **设备相关接口函数**

**C.1 概述**

VPN 设备相关接口函数包括计算工作密钥和会话密钥的接口函数，提供对VPN 设备密钥计算的 加速，表C.1 列出了VPN 设备相关接口函数，附录A 定义了函数返回代码。

**表C.1** **VPN设备相关接口函数**

|  |  |
| --- | --- |
| 函数名称 | 功能 |
| SDF\_GencratcKeywithIKE | 计算IKE工作密钥 |
| SDF\_GcncratcKcywithEPK\_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\_SSI | 计算SSL工作密钥并用外部ECC公钥加密输出 |

**C.2 计算IKE 工作密钥**

原型： LONG SDF\_GenerateKeywithIKE(

HANDLE hSessionHandle,

BYTE \*pucSponsorNonce,

ULONG uiSponsorNonceLength, BYTE \*pucResponse.Nonce,

ULONG uiResponseNonceLength, BYTE \*pucSponsorCookie,

ULONG uiSponsorCookieLength, BYTE\*pucResponseCookie,

ULONG uiResponseCookieLength, ULONG uiPrfAlgID,

HANDLE \*phKeyHandleD, ULONG uiKeyBitsD,

HANDLE \*phKeyHandleA, ULONG uiKeyBitsA,

**GM/T 0018—2023**

HANDLE \*phKeyHandleE, ULONG uiKeyBitsE);

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。计算IKE 工作密钥的过程应符合GM/T 0022 。IKE一阶段(主模式)消息3和 消息4交互完成后，本函数由参与通信的双方各自调用，计算后续工作密钥 SKEYID\_ d,SKEYID\_a,SKEYID\_c 。输入的密钥参数按顺序为Ni\_b、Nr\_b 、CKY-I 、CKY-R,返回 的密钥句柄按顺序为SKEYID\_d、SKEYID\_a 、SKEYID\_c。

参数：

hSessionHandle[in] pucSponsorNonce[in

uiSponsorNonceLength[in]

pucResponseNonce[in]

uiResponseNonceLength[in] pucSponsorCookie[in]

uiSponsorCookieLength[in] pucResponseCookie[in]

uiResponscCookicLength[in] uiPrfAlgID[in]

phKeyHandleD[out] uiKeyBitsD[in]

phKeyHandleA[out] uiKeyBitsA[in]

phKeyHandleE[out] uiKeyBitsE[in

与设备建立的会话句柄

发起方nonce 载荷主体

发起方nonce 载荷主体长度 响应方nonce 载荷主体

响应方 nonce 载荷主体长度 发起方cookie

发起方 cookie 长度 响应方cookie

响应方cookie 长度 PRF 算法标识

返回的SKEYID\_d 密钥句柄 SKEYID\_d 密钥长度

返回的SKEYID\_a 密钥句柄 SKEYID\_a 密钥长度

返回的SKEYID\_c 密钥句柄 SKEYID\_e 密钥长度

返回值： 0 成功

非0 失败，返回错误代码

**C.3 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,

**GM/T 0018—2023**

BYTE \*pucSponsorCookie,

ULONG uiSponsorCookieLength, BYTE \*pucResponseCookie,

ULONG uiResponseCookieLength, ULONG uiPrfAlgID

ULONG uiEccAlgID

ECCrefPublicKey \*pucPublicKey,

ECCCipher \*pucKeyD, ULONG uiKcyBitsD

ECCCipher \*pucKeyA, ULONG uiKeyBitsA

ECCCipher \*pucKcyE, ULONG uiKeyBitsE);

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。计算IKE 工作密钥的过程应符合GM/T 0022 。IKE一阶段(主模式)消 息3和消息4交互完成后，本函数由参与通信的双方各白调用，计算后续工作密钥 SKEYID\_d,SKEYID\_a,SKEYID\_e 。 输入的密钥参数按顺序为 Ni\_b、Nr\_b 、CKY-I、 CKY-R, 返回的密钥密文按顺序为 SKEYID\_d、SKEYID\_a、SKEYID\_e。

参数：

hSessionHandle[in]

pucSponsorNonce[in]

uiSponsorNonceLength[in]

pucResponseNonce[in]

uiResponseNonceLength[in] pucSponsorCookie[in]

uiSponsorCookicLength[in] pucResponseCookie[in]

uiResponseCookieLength[in] uiPrfAlgID [in]

uiEccAlgID[in]

pucPublicKey[in] pucKeyD[out]

uiKeyBitsD [in] pucKeyA[out]

与设备建立的会话句柄

发起方nonce 载荷主体

发起方nonce 载荷主体长度 响应方nonce 载荷主体

响应方nonce 载荷主体长度 发起方cookie

发起方cookic 长度 响应方 cookie

响应方 cookie 长度 PRF 算法标识

外部 ECC公钥的算法标识 输入的外部ECC公钥结构

缓冲区指针，用于存放返回的SKEYID\_d 密钥密文 SKEYID\_d密钥长度

缓冲区指针，用于存放返回的SKEYID\_a密钥密文

|  |  |
| --- | --- |
| **GM/T 0018—2023** |  |
| 参数： uiKeyBitsA [in  pucKeyE[out]  uiKeyBitsE [in]  返回值： 0  非 0 | SKEYID\_a密钥长度  缓冲区指针，用于存放返回的SKEYID\_e 密钥密文  SKEYID\_e密钥长度 成功  失败，返回错误代码 |

**C.4 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 hKeyHandle,

ULONG uiPrfAlgID,

HANDLE ×phKeyHandleEnc, ULONG uiKeyBitsEnc,

HANDLE \*phKeyHandleMac,

ULONG uiKeyBitsMac, BYTE \*pucSalt,

ULONG uiSaltLength);

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。计算IPSEC 会话密钥的过程应符合GM/T 0022 。IKE 二阶段(快速模式) 消息交互完成后，本函数由参与通信的双方各自调用，计算IPSEC 会话密钥，包括用于加 密的会话密钥和用于完整性校验的会话密钥。输入的密钥参数按顺序为protocol 、SPI、 Ni\_b、Nr\_b,返回的密钥句柄按顺序为加密密钥和杂凑密钥(用于完整性校验)。采用 GCM加密模式时，返回的杂凑密钥句柄应为空。

参数： hSessionHandle[in]

pucProtocolID[in]

与设备建立的会话句柄 协议ID

参数：

uiProtocolIDLength[in] pucSpi [in]

uiSpiLength [in]

pucSponsorNonce[in]

uiSponsorNonceLength[in] pucResponseNoncc[in]

uiResponseNonceLength[in] hKeyHandle[in]

uiPrfAlgID [in

phKeyHaodleEnc out ] uiKeyBitsEno[ in]

phKeyHandleMac [ out ]

**GM/T 0018—2023**

协议ID 长度

安全参数索引 SPI

安全参数索引SPI 长度 发起方nonce 载荷主体

发起方nonce 载荷主体长度 响应方 noncc 载荷主体

响应方nonce 载荷主体长度 输入的SKEYID \_d 密钥句柄 PRF 算法标识

返回的加密密钥句柄 加密密钥长度

返回的杂凑密钥句柄

yiKeyBitsMac fn 杂淡密创长度

pucSali[ out ] 缓冲区指针，用于存放返间的CCN 模式的Sat

|  |  |
| --- | --- |
| uiSaltLength [ in | 模 式的Salt长度 |
| 返回值： 0 |  |
| 非 0 | 败，返回错误代码 |

**C.5 计算IPSEC会话密钥并用外部ECC 公** **钥加密输出**

原型：

IONG

SDF\_GcneratekeywibEPK IPSEe

HANDLE hSessionHandle, BYTE \* pucProtocolID,

ULONG uiProtocolIDLength, BYTE \* puLSpi

ULONG uiSpiLcngth,

BYTE \*pucSponsorNonce,

ULONG uiSponsorNonceLength, BYTE \*pucResponse.Noncc,

ULONG uiResponseNonceLength, HANDLE hKeyHandle,

ULONG uiPrfAlgID, ULONG uiEccAlgID

**GM/T 0018—2023**

ECCrefPublicKey \*pucPublicKey,

ECCCipher \*pucKeyEnc ULONG uiKeyBitsEnc,

ECCCipher \*pucKeyMac, ULONG uiKeyBitsMac,

BYTE \*pucSalt,

ULONG uiSaltLength);

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。计算IPSEC 会话密钥的过程应符合GM/T 0022 。IKE 二阶段(快 速模式)消息交互完成后，本函数由参与通信的双方各自调用，计算 IPSEC 会话密钥，包 括用于加密的会话密钥和用于完整性校验的会话密钥。输入的密钥参数按顺序为 proto- col、SPI、Ni\_b、Nr\_b,返回的密钥密文按顺序为加密密钥和杂凑密钥(用于完整性校验)。 采用GCM 加密模式时，返回的杂凑密钥应为空。

参数：

hSessionHandle[in]

pucProtocolID[in]

uiProtocolIDLength[in] pucSpi [in]

uiSpiLength [in]

pucSponsorNonce[in]

uiSponsorNonceLength[in] pucResponscNonce[in]

uiResponseNonceLength[in]

hKeyHandle[in] uiPrfAlgID [in] uiEccAlgID[in]

pucPublicKey[in] pucKcyEnc[out] uiKeyBitsEnc[in] pucKeyMac[out] uiKeyBitsMac[in pucSalt[out]

uiSaltLength[in]

与设备建立的会话句柄 协议ID

协议 ID 长度

安全参数索引SPI

安全参数索引SPI 长度

发起方nonce 载荷主体

发起方nonce 载荷主体长度 响应方nonce 载荷主体

响应方 nonce 载荷主体长度 输入的SKEYID\_d密钥句柄 PRF 算法标识

外部ECC公钥的算法标识 输入的外部ECC公钥结构

缓冲区指针，用于存放返回的加密密钥密文 加密密钥长度

缓冲区指针，用于存放返回的杂凑密钥密文 杂凑密钥长度

缓冲区指针，用于存放返回的GCM 模式的 Salt GCM模式的Salt 长度

**GM/T 0018—2023**

|  |  |
| --- | --- |
| 返回值： 0 | 成功 |
| 非 0 | 失败，返回错误代码 |

**C.6 计算SSL工作密钥**

原型： LONG SDF\_GenerateKeywithSSL(

HANDLE hSessionHandle,

HANDLE hKeyHandlcPrcMaster, BYTE \*pucClientRandom,

ULONG uClientRandomlength, BYTE\* pucServerRandom ,

ULOXG uiServerRandomLength, uLONG uiPrfAlgID,

HANDLE \*-phBeyHandleClicntMac

ULONCmKeyBisClienuMae ,

HANDLE CphKeyHandlsever ULOXG mKogBitsScry c ,

HANDLP pbKeyHaeeOientE ULONGaKoBrsClientBnc

HANDLE phKeyHandleSencc ULONG uiKeyBitsServerEnc,

BYTE \*pucClientIV,

ULONG uiClicntIVLcngth, BYTE \*pucServerIV ,

ULONG oServerIVLength);

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。

SSL 计算工作密钥的过程应符合GM/T 0024 。SSL握手协议消息交互完成后，本函数由 参与通信的双方各自调用，计算 SSL 记录层协议的工作密钥并返回密钥句柄：client\_ write\_MAC\_secret (客户端杂凑密钥),server\_write\_MAC\_secret (服务端杂凑密钥), client\_write\_key (客户端加密密钥),server\_write\_key (服务端加密密钥)。ECDHE 密钥 交换方式的预主密钥句柄调用ECC 密钥协商相关函数(客户端调用生成密钥协商参数并 输出和计算会话密钥函数，服务端调用产生协商数据并计算会话密钥函数)产生，RSA 密 钥交换方式的预主密钥句柄调用导入会话密钥并用内部RSA 私钥解密函数产生，ECC 密钥交换方式的预主密钥句柄调用导入会话密钥并用内部ECC 私钥解密函数产生。

**GM/T 0018—2023**

|  |  |  |
| --- | --- | --- |
| 参数： | hSessionHandle[in]  hKeyHandlePreMaster[in] pucClicntRandom[in]  uiClientRandomLength[in] pucServerRandom[in]  uiServerRandomLength[in] uiPr[AlgID [in]  phKeyHandleClientMac[out] uiKeyBitsClicntMac[in]  phKeyHandleServerMac[out] uiKeyBitsServerMac[in]  phKeyHandleClientEnc[out] uiKeyBitsClientEnc[in]  phKeyHandleServerEnc[out] uiKeyBitsServerEnc[in]  pucClientIV[out]  uiClientIVLength[in] pucServerIV[out]  uiServerIVLength[in] | 与设备建立的会话句柄  预主密钥pre\_master\_secret 密钥句柄 客户端随机数  客户端随机数长度 服务端随机数  服务端随机数长度 PRF 算法标识  返回的客户端杂凑密钥句柄 客户端杂凑密钥长度  返回的服务端杂凑密钥句柄 服务端杂凑密钥长度  返回的客户端加密密钥句柄 客户端加密密钥长度  返回的服务端加密密钥句柄 服务端加密密钥长度  缓冲区指针，用于存放返回的客户端IV 客户端IV 长度  缓冲区指针，用于存放返回的服务端IV 服务端IV 长度 |
| 返回值： | 0  非0 | 成功  失败，返回错误代码 |

**C.7 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 hSessionHandlc,

HANDLE hKeyHandlePreMaster, BYTE \*pucClientRandom,

ULONG uiClientRandomLength, BYTE \*pucServerRandom,

ULONG uiServerRandomLength, ULONG uiPrfAlgID,

ULONG uiEccAlgID,

**GM/T 0018—2023**

ECCrefPublicKey \*pucPublicKey, ECCCipher \*pucKeyClientMac,

ULONG uiKeyBitsClientMac,

ECCCipher \*pucKeyServerMac, ULONG uiKeyBitsServerMac,

ECCCipher \*pucKcyClientEnc, ULONG uiKeyBitsClientEnc,

ECCCipher \*pucKeyServerEnc, ULONG uiKeyBitsServerEnc,

BYTE\*pucClientIV,

ULONG uiClientIVLength, BYTE \*pucScrvcrIV,

ULONG uiServerIVLength);

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 。SSL 计算工作密钥的过程应符合GM/T 0024 。SSL握手协议消息交互完成后，本函 数由参与通信的双方各自调用，计算SSL 记录层协议的工作密钥并返回密钥密文：client write\_MAC\_secret (客户端杂凑密钥),server\_write\_MAC\_secret (服务端杂凑密钥), client\_write\_key (客户端加密密钥),server\_write\_key (服务端加密密钥)。ECDHE 密钥 交换方式的预主密钥句柄调用ECC 密钥协商相关函数(客户端调用生成密钥协商参数并 输出和计算会话密钥函数，服务端调用产生协商数据并计算会话密钥函数)产生，RSA 密 钥交换方式的预主密钥句柄调用导入会话密钥并用内部 RSA 私钥解密函数产生，ECC 密钥交换方式的预主密钥句柄调用导入会话密钥并用内部ECC 私钥解密函数产生。

描述：

参数：

|  |  |
| --- | --- |
| hSessionHandle[in] | 与设备建立的会话句柄 |
| hKeyHandlePreMaster[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 |
| uiScrverRandomLength[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 |
| pucPublicKcy[in] | 输入的外部ECC 公钥结构 |
| pucKeyClientMac[out] | 缓冲区指针，用于存放返回的客户端杂凑密钥 |
| uiKeyBitsClientMac[in] | 客户端杂凑密钥长度 |

**GM/T 0018—2023**

pucKeyServerMac[out] uiKeyBitsServerMac[in] pucKeyClientEnc[out]

uiKeyBitsClientEnc[in] pucKeyServerEnc[out] uiKeyBitsServerEnc[in] pucClientIV[out]

uiClientIVLength[in] pucServcrIV[out]

uiServerIVLength[in]

返回值： 0

非 0

缓冲区指针，用于存放返回的服务端杂凑密钥 服务端杂凑密钥长度

缓冲区指针，用于存放返回的客户端加密密钥 客户端加密密钥长度

缓冲区指针，用于存放返回的服务端加密密钥 服务端加密密钥长度

缓冲区指针，用于存放返回的客户端IV 客户端IV 长度

缓冲区指针，用于存放返回的服务端IV 服务端IV 长度

成功

失败，返回错误代码

**GM/T 0018—2023**

**参** **考** **文** **献**

[1]GB/T 17901.1—2020 信息技术安全技术密钥管理 第1部分：框架

[2]GB/T 17903.2—2021 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—2021 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 v1.5 RSA Cryptography Standard

[9]RFC2104 HMAC:Keycd-Hashing for Message Authentication