ICS 35.030 CCS L 80

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

GM/T 0016—2023 代替GM/T 0016—2012



**智能密码钥匙密码应用接口规范**

**Smart token cryptography application interface specification**

**2023-12-04发布** 2024-06-01实施



**国家密码管理局** **发** **布**

**GM/T 0016—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 2](#bookmark1)

[5 结构模型 2](#bookmark1)

[5.1 层次关系 2](#bookmark1)

[5.2 设备的应用结构 3](#bookmark7)

[6 数据类型定义 4](#bookmark8)

[6.1 算法标识 4](#bookmark9)

[6.2 基本数据类型 4](#bookmark10)

[6.3 常量定义 4](#bookmark11)

[6.4 复合数据类型 5](#bookmark12)

[7 接口函数 12](#bookmark13)

[7.1 设备管理 12](#bookmark14)

[7.2 访问控制 15](#bookmark15)

[7.3 应用管理 18](#bookmark16)

[7.4 文件管理 20](#bookmark17)

[7.5 容器管理 22](#bookmark18)

[7.6 密码服务 25](#bookmark19)

[7.7 验证调试 40](#bookmark20)

[8 接口使用要求 43](#bookmark21)

[8.1 设备使用阶段 43](#bookmark22)

[8.2 权限管理 44](#bookmark23)

[8.3 其他安全要求 44](#bookmark24)

[附录 A(规范性)错误代码定义 45](#bookmark25)

[附录 B(规范性) SM9应用接口 47](#bookmark26)

[附录 C(规范性) VPN 相关接口 62](#bookmark27)

[附录 D(资料性) SM9编程范例 71](#bookmark28)

[参考文献 75](#bookmark29)

**GM/T 0016—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 0016—2012《智能密码钥匙密码应用接口规范》,与GM/T 0016—2012相比， 除结构调整和编辑性改动外，主要技术变化如下：

a) 删除了“填充方式”(见表11,2012年版的表11);

b) 更改了“修改设备认证密钥”函数(见7.2.2,2012年版的7.2.2);

c) 更改了“获得容器类型”(见7.5.7,2012年版的7.5.7);

d) 更改了“导出公钥”(见7.6.18,2012年版的7.6.17);

e) 更改了“导入会话密钥”(见7.6.19,2012年版的7.6.18);

f) 更改了“安全要求”(见第8章，2012年版的第8章);

g) 增加了HMAC 相关接口(见7.6.36、7.6.37、7.6.38、7.6.39);

h) 增加了验证调试类接口(见7.7);

i) 增加了SM9 应用接口(见附录B);

j) 增加了VPN 相关接口(见附录C);

k) 增加了SM9 编程范例(见附录D)。

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：北京海泰方圆科技股份有限公司、北京握奇智能科技有限公司、格尔软件股份有 限公司、无锡江南信息安全工程技术中心、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、北京三未信安科技发展有限公司、山东大学、北京大明五洲科技有限公司、恒 宝股份有限公司、深圳市明华澳汉科技股份有限公司、武汉天喻信息产业股份有限公司、北京飞天诚信 科技股份有限公司、华翔腾数码科技有限公司、北京鼎九信息工程研究院有限公司、北京百旺信安科技 有限公司、中电科网络安全科技股份有限公司、北京国脉信安科技有限公司、北京小雷科技有限公司。

本文件主要起草人：刘平、蒋红宇、柳增寿、张立廷、罗俊、袁峰、封维端、靳京、张渊、陈国、李勃、 郑强、李述胜、孔凡玉、王妮娜、马洪富、高志权、徐明翼、李增欣、于学东、郭宝安、石玉平、胡俊义、 管延军、项莉、雷继业、胡鹏、赵再兴、段晓毅、刘玉峰、刘伟丰、陈吉、何永福、李高锋、黄东杰、王建承、 汪雪林、赵李明、王烨。

本文件及其所代替文件的历次版本发布情况为：

——2012年首次发布版为GM/T 0016—2012; ——本次为第一次修订。

**GM/T 0016—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智能密码钥匙设备制定统一的应用接口 标准。通过该接口调用智能密码钥匙，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 0016—2023**

**智能密码钥匙密码应用接口规范**

**1** **范围**

本文件规定了公钥密码体制下的智能密码钥匙应用接口标准、密码相关应用接口的函数、数据类 型、参数的定义和设备的安全要求。

本文件适用于智能密码钥匙产品的研制、使用和检测。

**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。

|  |  |  |
| --- | --- | --- |
| GM/T | 0006—2023 | 密码应用标识规范 |
| GM/T | 0017—2023 | 智能密码钥匙密码应用接口数据格式规范 |
| GM/T | 0027—2014 | 智能密码钥匙技术规范 |
| GM/T | 0028—2014 | 密码模块安全要求 |

GM/Z 4001 密码术语

PKCS#1 RSA密码规范版本2.1(RSA Cryptography specification version 2.1)

**3** **术语和定义**

|  |  |
| --- | --- |
|  | GM/Z 4001界定的以及下列术语和定义适用于本文件。 |
| **3.1** |  |
|  | **容器** **container**  密码设备中用于保存密钥所划分的唯一性存储空间。 |
| **3.2** |  |
|  | **终端设备** **terminal device**  智能密码钥匙的统称。 |
| **3.3** |  |
|  | **设备认证** **device authentication**  智能密码钥匙对应用程序的认证。 |
| 3.4 |  |
|  | **设备认证密钥** **device authentication key**  用于设备认证的密钥。 |
| 3.5 |  |
|  | **设备标签** **device label**  终端设备的别名，可由用户进行设定并存储于设备内部。 |

**GM/T 0016—2023**

**3.6**

**3.7**

**3.8**

**管理员PIN administrator PIN**

管理员的口令，为ASCII 字符串。

**用户PIN user PIN**

用户的个人口令，为ASCIⅡ 字符串。

**应** **用** **application**

包括容器和文件的一种结构，具备独立的管理权限。

**4** **缩略语**

下列缩略语适用子禾文件

API: 应用编程接口(Application Programming Interface)

ASCII: 美国信息交换标准码(American Standard Code for Information Interchange) MAC: 消息鉴别码(Message Authentication Code)

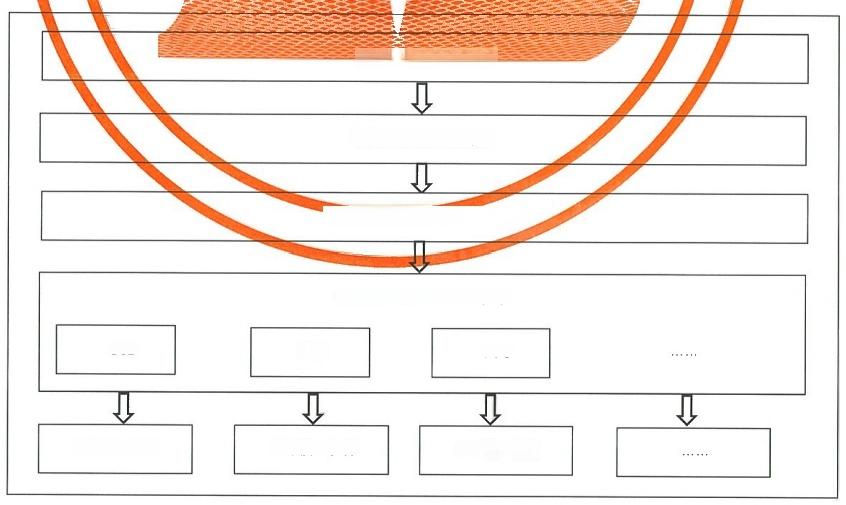
PKI: 公朗基研设施PaubicKey hntfastuctmc)

PIN: 个人身份识别码CPersonal Identheafonkanber)

**5** **结构模型**

**5.1** **层次关系**

智能密码钥匙应用接口 位 于 容能 密码 钥 是应 用与 智能 密 研 匙 应用接口数据格式之间，如 图1所示。智能密码钥匙硬件接日所符合GM/f 0027-20+E中6工的规定

智能密码钥匙应用程序

智能密码钥匙应用接口

智能密码钥匙应用接口数据格式

智能密码钥匙设备驱动程序

蓝牙 NFC

USB

蓝牙接口设备 NFC接口设备

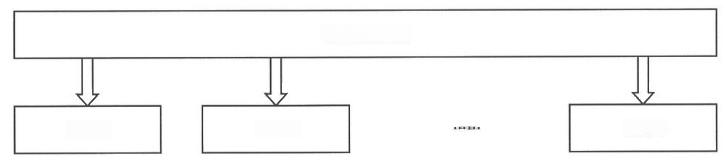
USB接口设备

**图** **1** **接口在应用层次关系中的位置**

**GM/T 0016—2023**

**5.2** **设备的应用结构**

终端设备应具有一个设备认证密钥。终端设备应支持一个或多个应用。应用之间应相互独立。 设备的逻辑结构如图2所示。

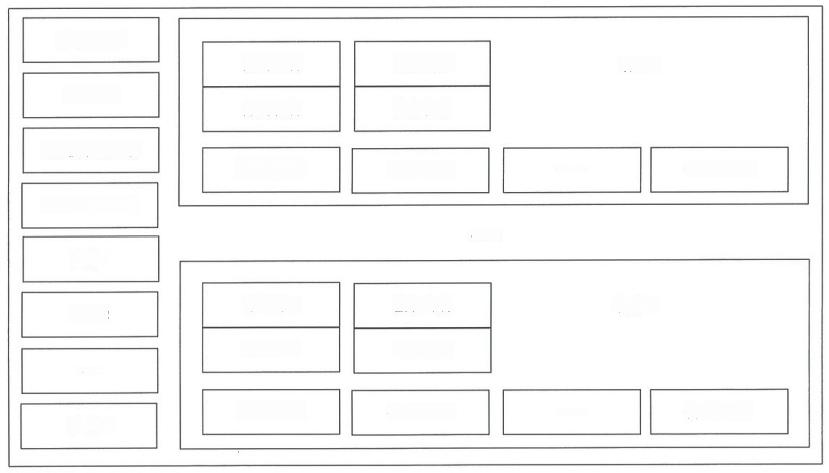
设备认证密钥

应用1 应用2 应用n

**图** **2** **设备逻辑结构**

应用由管理员PIN 、用户PIN、文件和容器组成，可存在多个文件和多个容器。应用应维护自身的 权限状态。管理员PIN 和用户PIN 的校验操作可改变应用权限状态。 一个应用的逻辑结构如图3 所示。

容器用于存放加密密钥对、签名密钥对和会话密钥。其中，加密密钥对用于保护会话密钥，签名密 钥对用于数字签名和验证，会话密钥用于数据加解密和 MAC 运算。容器中存放SM2 算法的密钥对 时，与其对应的数字证书也应存放在该容器中，且签名密钥对应由内部产生，加密密钥对应由外部安全 导入；容器中存放 SM9 算法的密钥对时，其公钥由标识表示；会话密钥可由内部产生，也可由外部产生 并安全导入。



管理员PIN

加密公钥

用户PIN

加密私钥

SM9签名主公钥

会话密钥1

SM9加密主公钥



文件1

文件2

加密私钥



会话密钥1

文件m

签名公钥

签名私钥

签名公钥

签名私钥

会话密钥

会话密钥2

会话密钥2

会话密钥k

加密公钥

容器1

容器n



**图** **3** **应用逻辑结构图**

**GM/T 0016—2023**

**6** **数据类型定义**

**6.1** **算法标识**

密码算法标识应符合GM/T 0006—2023中第6章的规定。

**6.2** **基本数据类型**

字节数组应使用高位字节在前(Big-Endian) 的方式存储和交换。基本数据类型定义见表1。 表十一基本数据类型

|  |  |  |
| --- | --- | --- |
| 类型名称 | 描述 | 备注 |
| INT8 | 有符号8位整数 | 例如typedefchar INT8 |
| INT16 | 有符号16位整数 | 例如，typedcf shortiNT16 |
| INT32 | 有符号32位整数 | 例如，typedeot INT32 |
| UINT8 | 无符号8位整数 | 例如，typedef unslgned char UINT8 |
| UINT16 | 无符号16位整数 | 例如，ypedefaunsigned shorUINT16 |
| UINT32 | 无符号132位整数 | 例姐江pedcausigned int UINT32 |
| BOOL | 布 尔 类 取 值 为 T R U E 或 L S E |  |
| BYTE | 字肯类型，无符号8位整数 | typedcf UINT8 BYTF |
| CHAR | 字行型九在号8位整 | typedeUNT8 CHAR |
| LONG | 长整数有的要位整数 | pede二N工32 LONQ |
| ULONG | 长整数，无符是32位班数 | y p e d e 图 2 U Ib N G |
| UINT | 无符32位脑数 | ypedeJINT32 UINT |
| WORD | 字类型，无符号16位整数 | typcdef-UINT16 wOR |
| DWORD | 双字类型，无符号32位整数 | typedef UINT32 DwORD |
| FLAGS | 标志类型，无符号32位整数 | typedef UINT32 FLAGS |
| LPSTR | 8位字符串指针，按照UTF8格式存储及交换 | typedef CHAR\*LPSTR |
| HANDLE | 句柄，指向任意数据对象的起始地址： | typedei void \*HANDLE |
| DEVHANDLE | 设备句柄 | lypedef HANDLE DEVHANDLE |
| HAPPLICATION | 应用句柄 | typedef HANDLE HAPPLICATION |
| HCONTAINER | 容器句柄 | typedef HANDLE HCONTAINER |

**6.3** **常量定义**

数据常量标识定义了在规范中用到的常量的取值。 数据常量标识的定义如表2所示。

**GM/T 0016—2023**

**表** **2** **常量定义**

|  |  |  |
| --- | --- | --- |
| 常量名 | 取值 | 描述 |
| TRUE | 0x00000001 | 布尔值为真 |
| FALSE | 0x00000000 | 布尔值为假 |
| DEVAPI | stdcall | \_stdcall函数调用方式 |
| ADMIN\_TYPE | 0 | 管理员PIN类型 |
| USER\_TYPE | 1 | 用户PIN类型 |

**6.4** **复合数据类型**

**6.4.1** **对齐方式**

复合数据类型应采用单字节对齐方式。

**6.4.2** **版本**

[**6.4.2.1**](6.4.2.1) **类型定义**

typedef struct Struct\_Version{ BYTE major;

BYTE minor }VERSION;

[**6.4.2.2**](6.4.2.2) **数据项**

数据项应符合表3的规定。

**表** **3** **版本定义**

|  |  |  |  |
| --- | --- | --- | --- |
| 数据项 | 类型 | 意义 | 备注 |
| major | BYTE | 主版本号 | 使用ASCII编码，取值范围为字符0至字符9 |
| minor | BYTE | 次版本号 | 使用ASCII编码，取值范围为字符0至字符9 |

**6.4.3** **设备信息** [**6.4.3.1**](6.4.3.1) **类型定义**

typedef struct VERSION CHAR

CHAR

CHAR

CHAR

VERSION VERSION

Struct\_DEVINFO{

Version;

Manufacturer[64]; Issucr[64];

Label[32];

SerialNumber[32]; HWVersion;

FirmwareVersion;

**GM/T 0016—2023**

ULONG AlgSymCap;

ULONG AlgAsymCap

ULONG AlgHashCap;

ULONG DevAuthAlgld

ULONG TotalSpace;

ULONG FreeSpace

ULONG MaxECCBufferSizc;

ULONG MaxBufferSize; BYTE Reserved[64];

}DEVINFO\_SKF,\*PDEVINFO\_SKF;

[**6.4.3.2**](6.4.3.2) **数据项**

数据项应符合表4的规定。

**表** **4** **设备信息描述**

|  |  |  |  |
| --- | --- | --- | --- |
| 数据项 | 类型 | 意义 | 备注 |
| Vcrsion | VERSION | 版本号 | 数据结构版本号，本结构的版本号为2.0 |
| Manufacturer | CHAR数组 | 设备厂商信息 | 以‘\0’为结束符的ASCII字符串 |
| Issuer | CHAR数组 | 发行厂商信息 | 以‘\0'为结束符的ASCII字符串 |
| Label | CHAR数组 | 设备标签 | 以‘\0'为结束符的ASCII字符串 |
| SerialNumber | CHAR数组 | 序列号 | 以‘\O'为结束符的ASCII字符串 |
| HWVersion | VERSION | 设备硬件版本 |  |
| FirmwareVersion | VERSION | 设备本身固件版本 |  |
| AlgSymCap | ULONG | 分组密码算法标识 |  |
| AlgAsymCap | ULONG | 非对称密码算法标识 |  |
| AlgHashCap | ULONG | 密码杂凑算法标识 | 一 |
| DevAuthAlgId | ULONG | 设备认证使用的分组密码算法标识 |  |
| TotalSpace | ULONG | 设备总空间大小 |  |
| FreeSpace | ULONG | 用户可用空间大小 |  |
| MaxECCBufferSizc | ULONG | 能够处理的ECC加密数据大小 |  |
| MaxBufferSize | ULONG | 能够处理的分组运算和杂凑运算的 数据大小 |  |
| Rescrved | BYTE | 保留扩展 | — |

**6.4.4 RSA公钥数据结构** [**6.4.4.1**](6.4.4.1) **类型定义**

typedef struct Struct\_RSAPUBLICKEYBLOB{ ULONG AlgID;

ULONG BitLen;

**GM/T 0016—2023**

BYTE Modulus[MAX\_RSA\_MODULUS\_LEN];

BYTE PublicExponent[MAX\_RSA\_EXPONENT\_LEN]; }RSAPUBLICKEYBLOB,\*PRSAPUBLICKEYBLOB;

MAX\_RSA\_MODULUS\_LEN 为算法模数的最大长度； MAX\_RSA\_EXPONENT\_LEN 为算法指数的最大长度。

[**6.4.4.2**](6.4.4.2) **数据项**

数据项应符合表5的规定。

**表5** **RSA 公钥数据结构**

|  |  |  |  |
| --- | --- | --- | --- |
| 数据项 | 类型 | 意义 | 备注 |
| AlgID | ULONG | 算法标识号 |  |
| BitLen | ULONG | 模数的实际位长度 | 应是8的倍数 |
| Modulus | BYTE数组 | 模数n=p\*q | 实际长度为BitLen/8字节  #define MAX\_RSA\_MODULUS\_LEN 512 #define MAX RSA EXPONENT\_LEN 4 |
| PublicExponent | BYTE数组 | 公开密钥c | 一般为00010001 |

**6.4.5 RSA私钥数据结构** [**6.4.5.1**](6.4.5.1) **类型定义**

typedef struct Struct\_RSAPRIVATEKEYBLOB{ ULONG AlgID;

ULONG BitLen;

BYTE Modulus[MAX\_RSA\_MODULUS\_LEN];

BYTE PublicExponent[MAX\_RSA\_EXPONENT\_LEN];

BYTE PrivateExponent[MAX\_RSA\_MODULUS\_LEN];

BYTE Primel[MAX\_RSA\_MODULUS\_LEN/2];

BYTE Prime2[MAX\_RSA\_MODULUS\_LEN/2];

BYTE PrimelExponent[MAX\_RSA\_MODULUS\_LEN/2];

BYTE Prime2Exponent[MAX\_RSA\_MODULUS\_LEN/2];

BYTE Coefficient[MAX\_RSA\_MODULUS\_LEN/2]; }RSAPRIVATEKEYBLOB,\*PRSAPRIVATEKEYBLOB;

MAX\_RSA\_MODULUS\_LEN 为 RSA 算法模数的最大长度。

[**6.4.5.2**](6.4.5.2) **数据项**

数据项应符合表6的规定。

**GM/T 0016—2023**

**表** **6** **RSA 私钥数据结构**

|  |  |  |  |
| --- | --- | --- | --- |
| 数据项 | 类型 | 意义 | 备注 |
| AlgID | ULONG | 算法标识号 |  |
| BitLen | ULONG | 模数的实际位长度 | 应是8的倍数 |
| Modulus | BYTE数组 | 模 数 n = p \* q | 实际长度为BitLen/8字节 |
| PublicExponent | BYTE数组 | 公开密钥e | 一般为00010001 |
| PrivatcExponent | BYTE数组 | 私有密钥d | 实际长度为BitLen/8字节 |
| Primel | BYTE数组 | 素数p | 实际长度为BitLen/16字节 |
| Prime2 | BYTE数组 | 素数q | 实际长度为BitLen/16字节 |
| PrimelExponent | BYTE数组 | d mod(p-1)的值 | 实际长度为BitLen/16字节 |
| Prime2Exponent | BYTE数组 | d mod(q-1)的值 | 实际长度为BitLen/16字节 |
| Coefficient | BYTE数组 | q模p的乘法逆元 | 实际长度为BitLen/16字节 |

**6.4.6 ECC公钥数据结构** [**6.4.6.1**](6.4.6.1) **类型定义**

typedef struct Struct\_ECCPUBLICKEYBLOB{ ULONG BitLen;

BYTE XCoordinate[ECC\_MAX\_XCOORDINATE\_BITS\_LEN/8];

BYTE YCoordinate[ECC\_MAX\_YCOORDINATE\_BITS\_LEN/8]; }ECCPUBLICKEYBLOB,\*PECCPUBLICKEYBLOB;

ECC\_MAX\_XCOORDINATE\_LEN 为 ECC 算法X 坐标的最大长度； ECC\_MAX\_YCOORDINATE\_LEN 为 ECC 算法 Y 坐标的最大长度。

[**6.4.6.2**](6.4.6.2) **数据项**

数据项描述见表7。

**表** **7** **ECC 公钥数据结构**

|  |  |  |  |
| --- | --- | --- | --- |
| 数据项 | 类型 | 意义 | 备注 |
| BitLen | ULONG | 模数的实际位长度 | 应是8的倍数 |
| XCoordinate | BYTE数组 | 曲线上点的X坐标 | 有限域上的整数  #define ECC\_MAX\_XCOORDINATE\_BITS\_LEN 512 |
| YCoordinate | BYTE数组 | 曲线上点的Y坐标 | 有限域上的整数  #define ECC\_MAX\_YCOORDINATE\_BITS\_LEN 512 |

**GM/T 0016—2023**

**6.4.7** **ECC私钥数据结构** [**6.4.7.1**](6.4.7.1) **类型定义**

typedef struct Struct\_ECCPRIVATEKEYBLOB{ ULONG BitLen;

BYTE PrivateKey[ECC\_MAX\_MODULUS\_BITS\_LEN/8];

}ECCPRIVATEKEYBLOB,\*PECCPRIVATEKEYBLOB;

ECC\_MAX\_MODULUS\_BITS\_LEN 为 ECC 算法模数的最大长度。

[**6.4.7.2**](6.4.7.2) **数据项**

数据项描述见表8。

**表8** **ECC 私钥数据结构**

|  |  |  |  |
| --- | --- | --- | --- |
| 数据项 | 类型 | 意义 | 备注 |
| BitLen | ULONG | 模数的实际位长度 | 应是8的倍数 |
| PrivateKey | BYTE数组 | 私有密钥 | 有限域上的整数  #define ECC\_MAX\_MODULUS\_BITS\_LEN 512 |

**6.4.8** **ECC密文数据结构** [**6.4.8.1**](6.4.8.1) **类型定义**

typedef struct Struct\_ECCCIPHERBLOB{

BYTE XCoordinate[ECC\_MAX\_XCOORDINATE\_BITS\_LEN/8];

BYTE YCoordinate[ECC\_MAX\_XCOORDINATE\_BITS\_LEN/8];

BYTE HASH[32];

ULONG CipherLen;

BYTE Cipher[1];

}ECCCIPHERBLOB,\*PECCCIPHERBLOB;

[**6.4.8.2**](6.4.8.2) **数据项**

数据项描述见表9。

**表** **9** **ECC 密文数据结构**

|  |  |  |  |
| --- | --- | --- | --- |
| 数据项 | 类型 | 意义 | 备注 |
| XCoordinate | BYTE数组 | 与y组成椭圆曲线上的点(x,y) |  |
| YCoordinate | BYTE数组 | 与x组成椭圆曲线上的点(x,y) |  |
| HASH | BYTE数组 | 明文的杂凑值 |  |
| CipherLen | ULONG | 密文数据长度 |  |
| Cipher | BYTE数组 | 密文数据 | 实际长度为CipherLen |

**GM/T 0016—2023**

**6.4.9 ECC签名数据结构** [**6.4.9.1**](6.4.9.1) **类型定义**

typedef struct Struct\_ECCSIGNATUREBLOB{

BYTE r[ECC\_MAX\_XCOORDINATE\_BITS\_LEN/8];

BYTE s[ECC\_MAX\_XCOORDINATE\_BITS\_LEN/8];

}ECCSIGNATUREBLOB,\*PECCSIGNATUREBLOB;

ECC\_MAX\_MODULUS\_BITS\_LEN 为 ECC 算法模数的最大长度。

[**6.4.9.2**](6.4.9.2) **数据项**

数据项描述见表10。

**表** **1** **0** **ECC 签名数据结构**

|  |  |  |  |
| --- | --- | --- | --- |
| 数据项 | 类型 | 意义 | 备注 |
| r | 8YTE数组 | 签名结果的r部分 |  |
| s BY T 数 组 | | 签各结果的s部分 |  |

**6.4.10** **分组密码参数** [**6.4.10.1**](6.4.10.1) **类型定义**

typedef struct Struci BEOCRCIPHEReARAM BYTE IV[MAX工EN

ULOnG IVLen;

ULONG FeedBi 通ens

}BLOCKCIPHERIARAM , \*PBLOCKCHPIERPARAM ,

[**6.4.10.2**](6.4.10.2) **数据项**

数据项描述见表11。

**表** **1** **1** **分组密码参数**

|  |  |  |  |
| --- | --- | --- | --- |
| 数据项 | 类型 | 意义 | 备注 |
| IV | BYTE数组 | 初始向量，MAX\_IV\_LEN为初始化向量的最 大长度#dcfinc MAX\_IV\_LEN 32 |  |
| IVLen | ULONG | 初始向量实际长度(按字节计算) |  |
| FeedBitLen | ULONG | 反馈值的位长度(按位计算) | 只针对OFB、CFB模式 |

**GM/T 0016—2023**

**6.4.11** **ECC加密密钥对保护结构** [**6.4.11.1**](6.4.11.1)**类型定义**

typedef struct Struct\_ENVELOPEDKEYBLOB

ULONG ULONG ULONG BYTE

Version;

ulSymmAlgID; ulBits;

cbEncryptedPriKey[64];

ECCPUBLICKEYBLOB PubKey;

ECCCIPHERBLOB ECCCipherBlob;

}ENVELOPEDKEYBLOB,\*PENVELOPEDKEYBLOB;

私钥密文数组中的有效密文分组应从0偏移量字节开始。顺序解密密文分组后将明文连接应得 到加密密钥对的私钥的明文。

[**6.4.11.2**](6.4.11.2) **数据项**

数据项描述见表12。

**表12** **加密密钥对保护结构参数**

|  |  |  |  |
| --- | --- | --- | --- |
| 数据项 | 类型 | 意义 | 备注 |
| Version | ULONG | 版本号，本版本为2 |  |
| ulSymmAlgID | ULONG | 对称算法标识 | 应为ECB模式 |
| ulBits | ULONG | 加密密钥对的密钥位长 |  |
| cbEncryptedPrivKey | BYTE数组 | 对称算法加密的加密私钥，加密私钥 的原文为ECCPRIVATEKEYBLOB 结构中的PrivateKcy | 其有效长度为原文的(ulBits 十7)/8 |
| PubKey | ECCPUBLICKEYB LOB | 加密密钥对的公钥 |  |
| ECCCipherBlob | ECCCIPHERBLOB | 用保护公钥加密过的对称密钥密文 | 一 |

**6.4.12** **文件属性**

[**6.4.12.1**](6.4.12.1) **类型定义**

typedef struct Struct\_FILEATTRIBUTE{

CHAR

ULONG ULONG

ULONG

FileName[32]; FileSize;

ReadRights; WriteRights;

}FILEATTRIBUTE,\*PFILEATTRIBUTE;

[**6.4.12.2**](6.4.12.2) **数据项**

数据项描述见表13。

**GM/T 0016—2023**

**表13** **文件属性**

|  |  |  |  |
| --- | --- | --- | --- |
| 数据项 | 类型 | 意义 | 备注 |
| FileName | C H A R 数 组 | 文件名 | 以‘\O'结束的ASCII字符串，最大长度为32 |
| FileSize | ULONG | 文件大小 | 创建文件时定义的文件大小 |
| RcadRights | ULONG | 读取权限 | 读取文件应已取得的权限 |
| WriteRights | ULONG | 写入权限 | 写入文件应已取得的权限 |

**6.4.13** **权限类型**

权限类型的定义见表1 4

**表14** **权限类型**

|  |  |  |
| --- | --- | --- |
| 权限类型 | 值 | 说明 |
| SECURE\_NEVER\_ACCOUNT | 0x00000000 | 不准许 |
| SECURE ADM\_ACC OUN | Qx0000000 | 理员权限 |
| SECUREUSERACCOUNT | 0x00000010 | 用主权限 |
| SECURE ANYONE\_ACCOuT | 07000000 | 低何人 |

**6.4.14** **设备状态**

设备状态的定义见表

**表设备状态**

|  |  |  |
| --- | --- | --- |
| 设备状态 | 值 | 说明 |
| DEV\_ABSENT\_STATR | 0x00000000 | 设备不存在 |
| DEV\_PRESENT\_STATE | Qx00000001 | 设备存在 |
| DEV\_UNKNOW\_STA TE | 0x00000002 | 设备状态未知 |

**6.4.15** **错误代码**

接口函数所返回的错误代码应符合附录 A 中的规定。

**7** **接口函数**

**7.1** **设备管理**

**7.1.1** **概述**

设备管理主要完成设备的插拔事件处理、枚举设备、连接设备、断开连接、获取设备状态、设置设备 标签、获取设备信息、锁定设备、解锁设备的操作。设备管理系列函数见表16。

**GM/T 0016—2023**

**表16** **设备管理系列函数**

|  |  |
| --- | --- |
| 函数名称 | 功能 |
| SKF\_WaitForDevEvent | 等待设备插拔事件 |
| SKF\_CancelWaitForDevEvent | 取消等待设备插拔事件 |
| SKF\_EnumDev | 枚举设备 |
| SKF\_ConnectDev | 连接设备 |
| SKF\_DisconnectDev | 断开连接 |
| SKF\_GetDevState | 获取设备状态 |
| SKF\_SetLabel | 设置设备标签 |
| SKF\_GetDevInfo | 获取设备信息 |
| SKF\_LockDev | 锁定设备 |
| SKF\_UnlockDev | 解锁设备 |

**7.1.2** **等待设备插拔事件**

原型 ULONG DEVAPI SKF\_WaitForDevEvent(LPSTR szDevName,ULONG \*pulDev-

NameLen,ULONG \*pulEvent)

功能描述 该函数等待设备插入或者拔除事件。szDevName 返回发生事件的设备名称。

参数 szDevName [OUT] 发生事件的设备名称。

pulDevNameLen

pulEvent

[IN,OUT] 输入/输出参数，当输入时表示缓冲区长度，输出时表 示设备名称的有效长度，长度包含字符串结束符。

[OUT] 事件类型。1表示插入，2表示拔出。

返回值 SAR\_OK: 成功。

其他： 错误码。 注：本函数为阻塞函数。

**7.1.3** **取消等待设备插拔事件**

原型 ULONG DEVAPI SKF\_CancelWaitForDevEvent()

功能描述 该函数取消等待设备插入或者拔除事件。

参数

返回值 SAR\_OK: 成功。

其他： 错误码。

注：使本进程正在执行的SKF\_WaitForDevEvent函数立即返回。

**7.1.4** **枚举设备**

原型 ULONG DEVAPI SKF\_EnumDev(BOOL bPresent,LPSTR szNameList,ULONG

\*pulSize)

功能描述 获得当前系统中的设备列表。

**GM/T 0016—2023**

参数

bPresent

szNameList

pulSize

[IN] 为TRUE 表示取当前设备状态为存在的设备列表。为FALSE 表示取当前驱动支持的设备列表。

[OUT] 设备名称列表。如果该参数为NULL, 将由pulSize 返回该列 表的字节长度。每个设备的名称以单个‘\0’结束，以双‘\O’ 表示列表 的结束。

[IN,OUT] 输入时表示设备名称列表的缓冲区长度，输出时表示 sz- NameList 所占用的空间大小。

返回值 SAR\_OK: 成功。

其他： 错误码。

**7.1.5** **连接设备**

原型 ULONG DEVAPI SKF\_ConnectDev(LPSTR szName,DEVHANDLE \*phDev)

功能描述 通过设备名称连接设备，返回设备的句柄。

参数 szName [IN] 设备名称。

phDev [OUT] 返回设备操作句柄。

返回值 SAR\_OK: 成功。

其他： 错误码。

**7.1.6** **断开连接**

原型 ULONG DEVAPI SKF\_DisConnectDev(DEVHANDLE hDev)

功能描述 断开一个已经连接的设备，并释放句柄。

参数 hDev [IN] 连接设备时返回的设备句柄。

返回值 SAR\_OK: 成功。

其他： 错误码。

注：如果该设备已被锁定，函数首先解锁该设备。断开连接操作并不影响设备的权限状态。

**7.1.7** **获取设备状态**

原型 ULONG DEVAPI SKF\_GetDevState(LPSTR szDevName,ULONG \*pulDevState)

功能描述 获取设备是否存在的状态。

参数 szDevName [IN] 设备名称。

pulDevState [OUT] 返回设备状态。

返回值 SAR\_OK: 成功。

其他： 错误码。

**7.1.8** **设置设备标签**

原型 ULONG DEVAPI SKF\_SetLabel(DEVHANDLE hDev,LPSTR szLabel)

功能描述 设置设备标签。

参数 hDev [IN] 连接设备时返回的设备句柄。

szLabel [IN] 设备标签字符串。该字符串应小于32字节。

**GM/T 0016—2023**

返回值 SAR\_OK: 成功。

其他： 错误码。

**7.1.9** **获取设备信息**

原型

功能描述

ULONG DEVAPI SKF\_GetDevInfo(DEVHANDLE hDev,DEVINFO\*pDevInfo) 获取设备的一些特征信息，包括设备标签、厂商信息和支持的算法。

参数 hDev [IN] 连接设备时返回的设备句柄。

pDevInfo [OUT] 返回设备信息。

返回值 SAR\_OK: 成功。

其他： 错误码。

**7.1.10** **锁定设备**

|  |  |
| --- | --- |
| 原型 | ULONG DEVAPI SKF\_LockDev(DEVHANDLE hDev,ULONG ulTimeOut) |
| 功能描述 | 获得设备的独占使用权。 |
| 参数 | hDev [IN] 连接设备时返回的设备句柄。  ulTimeOut [IN] 超时时间，单位为毫秒。如果为0xFFFFFFFF 表示无限等待。 |

返回值 SAR\_OK: 成功。

其他： 错误码。

**7.1.11** **解锁设备**

原型 ULONG DEVAPI SKF\_UnlockDev(DEVHANDLE hDev)

功能描述 释放对设备的独占使用权。

参数 hDev [IN] 连接设备时返回的设备句柄。

返回值 SAR\_OK: 成功。

其他： 错误码。

**7.2** **访问控制**

**7.2.1** **概述**

访问控制主要完成设备认证、PIN 码管理和安全状态管理操作。安全状态表示设备认证、校验 PIN 之后用户所取得的资源访问能力。访问控制系列函数见表17。

**表17** **访问控制系列函数**

|  |  |
| --- | --- |
| 函数名称 | 功能 |
| SKF\_ChangeDevAuthKey | 修改设备认证密钥 |
| SKF DevAuth | 设备认证 |
| SKF\_ChangePIN | 修改PIN |
| SKF GctPINInfo | 获得PIN码信息 |
| SKF\_VerifyPIN | 校验PIN |
| SKF UnblockPIN | 解锁PIN |
| SKF\_ClearSecueState | 清除应用安全状态 |

**GM/T 0016—2023**

**7.2.2** **修改设备认证密钥**

原型 ULONG DEVAPISKF\_ChangeDevAuthKey(DEVHANDLE hDev,BYTE \*pbKeyCi-

pher,ULONG ulKeyCipherLen,BYTE \*pbKeyMac,ULONG ulKeyMacLen)

功能描述 更改设备认证密钥。使用密文+MAC 方式进行密钥更新。MAC 用于保证只有原设备

认证密钥持有者才能完成本操作。

|  |  |  |
| --- | --- | --- |
| 参数 | hDev  pbKeyCipher | [IN] 连接时返回的设备句柄。  [IN] 新设备认证密钥的密文。该密文通过使用原设备认证密钥 对新设备认证密钥进行加密得到，计算过程应符合GM/T 0017— 2023中附录B的规定。 |
|  | ulKeyCipherLen pbKeyMac | [IN]pbKeyCipher 的长度。  [IN] 报文鉴别码，该报文鉴别码由原设备认证密钥对含有pbKey- Cipher的数据进行计算得到，计算过程应符合GM/T 0017—2023 中附录B的规定。 |
|  | ulKeyMacLen | [IN]pbKeyMac 的长度。 |

返回值

SAR\_OK: 其他：

成功。

错误码。

**7.2.3** **设备认证**

原型 ULONG DEVAPI SKF\_DevAuth(DEVHANDLE hDev,BYTE \*pbAuthData,

ULONG ulLen)

功能描述 设备认证是设备对应用程序的认证。认证过程应符合8.2.3中的规定。

参数 hDev [IN] 连接时返回的设备句柄。

pbAuthData [IN] 认证数据。

ulLen [IN] 认证数据的长度。

返回值 SAR\_OK: 成功。

其他： 错误码。

**7.2.4 修改PIN**

原型 ULONG DEVAPI SKF\_ChangePIN(HAPPLICATION hApplication,ULONG ulPIN-

Type,LPSTR szOldPin,LPSTR szNewPin,ULONG \*pulRetryCount)

功能描述 调用该函数可修改Administrator PIN和User PIN的值。

如果原PIN 码错误导致验证失败，该函数会返回相应PIN 码的剩余重试次数，当剩余次 数为0时，表示PIN已经被锁死。

参数 hApplication [IN] 应用句柄。

ulPINType [IN]PIN 类型，可为ADMIN\_TYPE 或USER\_TYPE。

szOldPin [IN] 原PIN 值。

szNewPin [IN] 新PIN 值。

pulRetryCount [OUT] 出错后重试次数。

**GM/T 0016—2023**

返回值 SAR\_OK: 成功。

其他： 错误码。

**7.2.5 获取** **PIN信息**

原型 ULONG DEVAPI SKF\_GetPINInfo(HAPPLICATION hApplication,ULONG ulPIN-

Type,ULONG \*pulMaxRetryCount,ULONG \*pulRemainRetryCount,BOOL\*pbDe- faultPin)

功能描述 获取PIN 码信息，包括最大重试次数、当前剩余重试次数，以及当前PIN 码是否为出厂默 认PIN 码。

参数 hApplication [IN] 应用句柄。

ulPINType [IN]PIN 类型。

pulMaxRetryCount [OUT] 最大重试次数。

pulRemainRetryCount [OUT] 当前剩余重试次数，当为0时表示已锁死。

pbDefaultPin [OUT] 是否为出厂默认PIN 码。

返回值 SAR\_OK: 成功。

其他： 错误码。

**7.2.6 校验PIN**

原型 ULONG DEVAPI SKF\_VerifyPIN(HAPPLICATION hApplication,ULONG ulPIN-

Type,LPSTR szPIN,ULONG \*pulRetryCount)

功能描述 校验PIN 码。校验成功后，会获得相应的权限，如果PIN 码错误，会返回PIN 码的重试次 数，当重试次数为0时表示PIN 码已经锁死。

参数 hApplication [IN] 应用句柄。

ulPINType [IN]PIN 类型。

szPIN [IN]PIN 值。

pulRetryCount [OUT] 出错后返回的重试次数。

返回值 SAR\_OK: 成功。

其他： 错误码。

**7.2.7 解锁PIN**

原型 ULONG DEVAPI SKF\_UnblockPIN(HAPPLICATION hApplication,LPSTR szAd-

minPIN,LPSTR szNewUserPIN,ULONG \*pulRetryCount) 功能描述 当用户的PIN 码锁死后，通过调用该函数来解锁用户PIN 码。

解锁后，用户PIN 码被设置成新值，用户PIN 码的重试次数也恢复到原值。

参数 hApplication [IN] 应用句柄。

szAdminPIN [IN] 管理员PIN 码。 szNewUserPIN [IN] 新的用户PIN 码。

pulRetryCount [OUT] 管理员PIN 码错误时，返回剩余重试次数。

返回值 SAR\_OK: 成功。

其他： 错误码。

**GM/T 0016—2023**

注：验证完管理员PIN才能够解锁用户PIN码，如果输入的Administrator PIN不正确或者已经锁死，会调用失 败，并返回Administrator PIN的重试次数。

**7.2.8** **清除应用安全状态**

原型 ULONG DEVAPI SKF\_ClearSecureState(HAPPLICATION hApplication)

功能描述 清除应用当前的安全状态。

参数 hApplication [IN] 应用句柄。

返回值 SAR\_OK: 成功。

其他： 错误码。

**7.3** **应用管理**

**7.3.1** **概述**

应用管理主要完成应用的创建、枚举、删除、打开、关闭操作。应用管理系列函数见表18。

**表18** **应用管理系列函数**

|  |  |
| --- | --- |
| 函数名称 | 功能 |
| SKF\_CreateApplcation | 创建应用 |
| SKF\_EnumApplication | 枚登 |
| SKF\_DeleteApplication | 删 除 |
| SKF\_OpehAppication | 打开应 |
| SKF CloseAppication | 关闭应朋 |

**7.3.2** **创建应用**

原型 ULoNG DEVAPISKF\_CreateApplieatron ( DEVHANDLEhDev, LPSTR szAppName,

PSTR szAdminPin,DWORD dwAdminPinRetryCount,LPSTR szUserPin,DWORD

dwOserPinBetryCount,DWORD dwCreateFileRights,HAPPLiCATiON\*phApplica -

tion)

功能描述 创建一个应用。

参数 hDev [IN] 连接设备时返回的设备何柄。

szAppName N ] 应用名称

szAdminPin [IN] 管理员PIN。

dwAdminPinRetryCount [IN] 管理员PIN 最大重试次数。

szUserPin [IN] 用 户PIN。

dwUserPinRetryCount [IN] 用 户PIN 最大重试次数。

dwCreateFileRights [IN] 在该应用下创建文件和容器的权限，权限类型应符合

6.4.12的规定。取值为各种权限的或值。

phApplication [OUT] 应用的句柄。

返回值 SAR\_OK: 成功。

其他： 错误码。 注：权限要求：已取得设备权限。

**GM/T 0016—2023**

**7.3.3** **枚举应用**

原型 ULONG DEVAPI SKF\_EnumApplication(DEVHANDLE hDev,LPSTR szAppName,

ULONG \*pulSize)

功能描述 枚举设备中存在的所有应用。

参数 hDev [IN] 连接设备时返回的设备句柄。

szAppName [OUT] 返回应用名称列表，如果该参数为空，将由pulSize返回

该列表的字节长度。每个应用的名称以单个‘\0’结束，以双‘\0’ 表示列表的结束。

pulSize [IN,OUT] 输入时表示应用名称的缓冲区长度，输出时返回

szAppName 所占用的空间大小。

返回值 SAR\_OK: 成功。

其他： 错误码。

**7.3.4** **删除应用**

原型 ULONG DEVAPI SKF\_DeleteApplication(DEVHANDLE hDev,LPSTR szAppName)

功能描述 删除指定的应用。

参数 hDev [IN] 连接设备时返回的设备句柄。

szAppName [IN] 应用名称。

返回值 SAR\_OK: 成功。

其他： 错误码。 注：权限要求：已取得设备权限。

**7.3.5** **打开应用**

原型 ULONG DEVAPI SKF\_OpenApplication(DEVHANDLE hDev,LPSTR szAppName,

HAPPLICATION \*phApplication)

功能描述 打开指定的应用。

参数 hDev [IN] 连接设备时返回的设备句柄。

szAppName [IN] 应用名称。

phApplication [OUT] 应用的句柄。

返回值 SAR\_OK: 成功。

其他： 错误码。

**7.3.6** **关闭应用**

原型 ULONG DEVAPI SKF\_CloseApplication(HAPPLICATION hApplication)

功能描述 关闭应用并释放应用句柄。

参数 hApplication [IN] 应用句柄。

返回值 SAR\_OK: 成功。

其他： 错误码。 注：此函数不影响应用安全状态。

**GM/T 0016—2023**

**7.4** **文件管理**

**7.4.1** **概述**

文件管理函数用以支持用户扩展开发，包括创建文件、删除文件、枚举文件、获取文件信息、文件读 操作、文件写操作。文件管理系列函数见表19。

**表** **1** **9** **文件管理系列函数**

|  |  |
| --- | --- |
| 函数名称 | 功能 |
| SKF\_CreateFile | 创建文件 |
| SKF\_DeleteFile | 删除文件 |
| SKF\_EnumFiles | 枚举文件 |
| SKF\_GetFileInfo | 获取文件信息 |
| SKF ReadFilc | 读文件 |
| SKF\_WriteFile | 写文件 |

**7.4.2** **创建文件**

原型 ULONG DEVAPISKECreato ( HAPLICATTON hegtion , BpSTR szFile-

Name, ULQNG a ieSize, G ulr ights, UONG uWriteRights)

功能描述 创建文件时要指定文件的名称大小，以及义件的读写权限。

参数 hApplication [IN] 句 柄

szFileName 文件名称。度不应人132个字

ulFleSize 文件大小

ulReadRights N 文件读权限，权限类型应符合64 .11的要求。取值可为各

种权限的或值。

ulWritekoghts [IN] 文件写权限，权限类型应符合6.41的要求。取值可为各

种权限的或值。

返回值 SAR\_OK 成 功

其他： 错误码。

注：权限要求：已取得应用指定的创建文件权限。

**7.4.3** **删除文件**

原型 ULONG DEVAPI SKF\_DeleteFile(HAPPLICATION hApplication,LPSTR szFile-

Name)

功能描述 删除指定文件：

文件删除后，文件中写入的所有信息将丢失。 文件在设备中的占用的空间将被释放。

参数 hApplication [IN] 要删除文件所在的应用句柄。

szFileName [IN] 要删除文件的名称。

**GM/T 0016—2023**

返回值 SAR\_OK: 成功。

其他： 错误码。

注：权限要求：已取得该文件的创建权限。

**7.4.4 枚举文件**

原型 ULONG DEVAPI SKF\_EnumFiles(HAPPLICATION hApplication,LPSTR szFileL-

ist,ULONG \*pulSize)

功能描述 枚举一个应用下存在的所有文件。

参数 hApplication [IN] 应用句柄。

szFileList [OUT] 返回文件名称列表，该参数为空，由pulSizc返回文件信息的

字节长度。每个文件名称以单个‘\0'结束，以双‘\0’表示列表的结束。

pulSize [IN,OUT] 输入时表示数据缓冲区的大小，输出时表示实际文件名

称列表的长度。

返回值 SAR\_OK: 成功。

其他： 错误码。

**7.4.5** **获取文件属性**

原型 ULONG DEVAPI SKF\_GetFileInfo(HAPPLICATION hApplication,LPSTR szFile-

Namc,FILEATTRIBUTE\*pFileInfo)

功能描述 获取文件信息：

获取应用文件的属性信息，例如，文件的大小、权限。

参数 hApplication [IN] 文件所在应用的句柄。

szFileName [IN] 文件名称。

pFileInfo [OUT] 文件信息，指向文件属性结构的指针。

返回值 SAR\_OK: 成功。

其他： 错误码。

**7.4.6** **读文件**

原型 ULONG DEVAPI SKF\_ReadFile(HAPPLICATION hApplication,LPSTR szFile-

Name,ULONG ulOffset,ULONG ulSize,BYTE\*pbOutData,ULONG \*pulOutLen)

功能描述 读取文件内容。

参数 hApplication [IN] 应用句柄。

|  |  |
| --- | --- |
| szFileName | [IN] 文件名。 |
| ulOffset | [IN] 文件读取偏移位置。 |
| ulSize | [IN] 要读取的长度。 |
| pbOutData | [OUT] 返回数据的缓冲区。 |
| pulOutLen | [IN,OUT] 输入时表示给出的缓冲区大小；输出时表示实际读取返回的 数据大小。 |

**GM/T 0016—2023**

返回值 SAR\_OK: 成功。

其他： 错误码。

注：权限要求：已取得该文件的读权限。

**7.4.7** **写文件**

原型 ULONG DEVAPI SKF\_WriteFile(HAPPLICATION hApplication,LPSTR szFile-

Name,ULONG ulOffset,BYTE \*pbData,ULONG ulSize)

功能描述 写数据到文件中。

参数 hApplication [IN] 应用句柄。

szFileName [IN] 文件名。

ulOffset [IN] 写入文件的偏移量。

pbData [IN] 写入数据缓冲区。

ulSize [IN] 写入数据的大小。

返回值 SAR\_OK: 成功。

其他： 错误码。

注：权限要求：已取得该文件的写权限。

**7.5** **容器管理**

**7.5.1** **概述**

容器管理提供容器的创建、删除、枚举、打开和关闭操作，也提供获取容器类型、导入数字证书和导 出数字证书操作。容器管理函数见表20。

**表20** **容器管理系列函数**

|  |  |
| --- | --- |
| 函数名称 | 功能 |
| SKF\_CreateContainer | 创建容器 |
| SKF\_DcleteContainer | 删除容器 |
| SKF\_EnumContainer | 枚举容器 |
| SKF\_OpenContainer | 打开容器 |
| SKF\_CloseContainer | 关闭容器 |
| SKF GetContaincrType | 获得容器类型 |
| SKF ImportCertificate | 导入数字证书 |
| SKF\_ExportCertificate | 导出数字证书 |

**7.5.2** **创建容器**

原型 ULONG DEVAPI SKF\_CreateContainer(HAPPLICATION hApplication,LPSTR sz-

ContainerName,HCONTAINER \*phContainer)

**GM/T 0016—2023**

功能描述 在应用下建立指定名称的容器并返回容器句柄。

参数 hApplication [IN] 应用句柄。

szContainerName [IN]ASCII字符串，表示所建立容器的名称，容器名称的最大长度

不能超过64字节。

phContainer [OUT] 返回所建立容器的容器句柄。

返回值 SAR\_OK: 成功。

其他： 错误码。 注：权限要求：已取得用户权限。

**7.5.3** **删除容器**

原型 ULONG DEVAPI SKF\_DeleteContainer(HAPPLICATION hApplication,LPSTR sz-

ContainerName)

功能描述 在应用下删除指定名称的容器并释放容器相关的资源。

参数 hApplication [IN] 应用句柄。

szContainerName [IN] 指向删除容器的名称。

返回值 SAR\_OK: 成功。

其他： 错误码。 注：权限要求：已取得用户权限。

**7.5.4** **打开容器**

原型 ULONG DEVAPI SKF\_OpenContainer(HAPPLICATION hApplication,LPSTR szCon-

tainerName,HCONTAINER \*phContainer)

功能描述 参数

获取容器句柄。

hApplication

szContainerName

phContainer

[IN] 应用句柄。

[IN] 容器的名称。

[OUT] 返回所打开容器的句柄。

返回值 SAR\_OK: 成功。

其他： 错误码。

**7.5.5** **关闭容器**

原型 ULONG DEVAPI SKF\_CloseContainer(HCONTAINER hContainer)

功能描述 关闭容器句柄，并释放容器句柄相关资源。

参数 hContainer [IN] 容器句柄。

返回值 SAR\_OK: 成功。

其他： 错误码。

**7.5.6** **枚举容器**

原型 ULONG DEVAPI SKF\_EnumContaincr(HAPPLICATION hApplication,LPSTR sz-

ContainerName,ULONG \*pulSize)

功能描述 枚举应用下所有容器并返回容器名称列表。

**GM/T 0016—2023**

参数 hApplication

szContainerName

pulSize

[IN] 应用句柄。

[OUT] 指向容器名称列表缓冲区。如果此参数为NULL 时， 由pulSize返回容器列表的字节长度，如果此参数不为NULL 时，返回容器名称列表，每个容器名以‘\O’为结束，列表以双‘\ 0’结束。

[IN,OUT] 输入时表示szContainerName缓存区的长度，输出 时表示容器名称列表的长度。

返回值 SAR\_OK: 成功。

其他： 错误码。

**7.5.7** **获得容器类型**

原型 ULONG DEVAPISKF\_GelContainerType(HCONTAINER hContainer,ULONG \*pul-

ContainerType)

功能描述获取容器的类型。

参数 hContainer

pulContainerType

[IN] 容器句柄。

[OUT] 获得的容器类型。指针指向的值为0表示未定、尚未 分配类型或者为空容器，为1表示为RSA容器，为2表示为

SM2容器，3表示SM9 容器。

返回值 SAR\_OK: 成功。

其他： 错误码。

**7.5.8** **导入数字证书**

原型 ULONG DEVAPI SKF\_ImportCertificate(HCONTAINER hContainer,BOOL bSign-

Flag,BYTE\*pbCert,ULONG ulCertLen)

功能描述 向容器内导入数字证书。

参数 hContainer [IN] 容器句柄。

bSignFlag [IN]TRUE 表示签名证书，FALSE 表示加密证书。

pbCert [IN] 指向证书内容缓冲区。

ulCertLen [IN] 证书长度。

返回值 SAR\_OK: 成功。

其他： 错误码。

**7.5.9** **导出数字证书**

原型 ULONG DEVAPISKF\_ExportCertificate(HCONTAINER hContainer,BOOL bSign-

Flag,BYTE\*pbCert,ULONG \*pulCertLen)

功能描述 从容器内导出数字证书。

参数 hContainer [IN] 容器句柄。

bSignFlag [IN]TRUE 表示签名证书，FALSE 表示加密证书。

pbCert [OUT] 指向证书内容缓冲区，如果此参数为NULL时 ，pul-

CertLen返回证书内容的字节长度；如果此参数不为NULL 时，返回数字证书内容。

**GM/T 0016—2023**

返回值

pulCertLen

SAR\_OK: 其他：

[IN,OUT] 输入时表示 pbCert缓冲区的长度，输出时表示证

书内容的长度。

成功。

错误码。

**7.6** **密码服务**

**7.6.1** **概述**

密码服务函数提供对称算法运算、非对称算法运算、密码杂凑运算、密钥管理和消息鉴别码计算 功能。

智能密码钥匙支持 SM9 密码算法的，其SM9 接口函数应符合附录 B 的规定。 智能密码钥匙支持 VPN 相关接口的，其VPN 接口函数应符合附录C 的规定。 密码服务系列函数见表21。

**表21** **密码服务系列函数**

|  |  |
| --- | --- |
| 函数名称 | 功能 |
| SKF\_GenRandom | 生成随机数 |
| SKF\_GenRSAKeyPair | 生成RSA签名密钥对 |
| SKF\_ImportRSAKeyPair | 导入RSA加密密钥对 |
| SKF\_RSASignData | RSA签名 |
| SKF\_RSAVerify | RSA验签 |
| SKF\_RSAExportSessionKey | RSA生成导出会话密钥 |
| SKF\_ExtRSAPubKeyOp | 外来RSA公钥运算 |
| SKF\_GenECCKeyPair | 生成ECC签名密钥对 |
| SKF\_ImportECCKeyPair | 导入ECC加密密钥对 |
| SKF\_ECCSignData | ECC签名 |
| SKF\_ECCVerify | ECC验签 |
| SKF\_ECCExportSessionKey | ECC生成并导出会话密钥 |
| SKF\_ExtECCEncrypt | ECC外来公钥加密 |
| SKF\_GenerateAgreementDataWithECC | ECC生成密钥协商参数并输出 |
| SKF\_GenerateKeyWithECC | ECC计算会话密钥 |
| SKF\_GenerateAgreementDataAndKeyWithECC | ECC产生协商数据并计算会话密钥 |
| SKF\_ExportPublicKey | 导出公钥 |
| SKF\_ImportSessionKey | 导入会话密钥 |
| SKF\_EncryptInit | 加密初始化 |
| SKF\_Encrypt | 单包数据加密 |
| SKF\_EncryptUpdate | 多包数据加密 |
| SKF\_EncryptFinal | 结束加密 |

**GM/T 0016—2023**

**表21** **密码服务系列函数** ( 续 )

|  |  |
| --- | --- |
| 函数名称 | 功能 |
| SKF\_DecryptInit | 解密初始化 |
| SKF\_Decrypt | 单包数据解密 |
| SKF DecryptUpdate | 多包数据解密 |
| SKF\_DecryptFinal | 结束解密 |
| SKF\_DigestInit | 密码杂凑初始化 |
| SKF\_Digest | 单包数据密码杂凑 |
| SKF\_DigestUpdate | 多包数据密码杂凑 |
| SKF\_DigestFinal | 结束密码杂凑 |
| SKF\_MacInit | 消息鉴别码运算初始化 |
| SKF\_Mac | 单包数据消息鉴别码运算 |
| SKF MacUpdgte | 多包数据消息鉴别码运算 |
| SKF\_MacFmal | 码 运 |
| SKF\_HMNCInit | 带密钥的杂凑运算初始化 |
| SKF\_HMAC | 带密铜的单包杂淡运算 |
| SKF\_HMACUpdate | 带密钥 包杂凑运算 |
| SKF\_HMACFinal | 带密钒的凑运算结策 |
| SKF\_CloscHandlc | 关闭密隔对象句柄 |

**7.6.2** **生成随机数**

原型 UONC DEVAPI SKF\_GenRandom(DEVHANDLE hDev,BXTE \*pbRandom,

ULONG unRandomLen)

功能描述 产生指定长度的随机数。

参数 hDev [IN] 设备句柄。

pbRandom [OUT ] 返回的随机数。

ulRandomLen [IN] 随机数长度

返回值 SAR\_OK: 成功。

其他： 错误码。

**7.6.3 生成RSA 签名密钥对**

原型 ULONG DEVAPI SKF\_GenRSAKeyPair(HCONTAINER hContainer,ULONG ulBit-

sLen,RSAPUBLICKEYBLOB\*pBlob)

功能描述 生 成RSA 签名密钥对并输出签名公钥。

参数 hContainer [IN] 容器句柄。

ulBitsLen [IN] 密钥模长。

**GM/T 0016—2023**

pBlob [OUT] 返回的RSA 公钥数据结构。

返回值 SAR\_OK: 成功。

其他： 错误码。 注：权限要求：已取得用户权限。

**7.6.4 导入RSA 加密密钥对**

原型 ULONG DEVAPI SKF\_ImportRSAKeyPair(

HCONTAINER hContainer,ULONG ulSymAlgId,

BYTE \*pbWrappedKey,ULONG ulWrappedKeyLen,

BYTE \*pbEncryptedData,ULONG ulEncryptedDataLen)

功能描述 导 入RSA 加密公私钥对。

参数 hContainer [IN] 容器句柄。

ulSymAlgId [IN] 对称算法密钥标识。

pbWrappedKey [IN] 使用该容器内签名公钥保护的对称算法密钥。

ulWrappedKeyLen [IN] 保护的对称算法密钥长度。

pbEncryptedData [IN] 对称算法密钥保护的RSA 加密私钥。私钥的格式遵循

PKCS#1 v2.1:RSA Cryptography Standard中的私钥格式定义。

ulEncryptedDataLcn [IN] 对称算法密钥保护的RSA 加密公私钥对长度。

返回值 SAR\_OK: 成功。

其他： 错误码。 注：权限要求：已取得用户权限。

**7.6.5 RSA签名**

原型 ULONG DEVAPI SKF\_RSASignData(HCONTAINER hContainer,BYTE\*pbData,

ULONG ulDataLen,BYTE \*pbSignature,ULONG \*pulSignLen)

功能描述

参数

返回值

使 用hContainer 指定容器的签名私钥，对指定数据pbData 进行数字签名。签名后的结果 存放到pbSignature 缓冲区，设置pulSignLen 为签名的长度。

[IN] 用来签名的私钥所在容器句柄。 [IN] 被签名的数据。

hContainer

pbData

ulDataLen

pbSignature

pulSignLen

SAR\_OK: 其他：

[IN] 签名数据长度，应不大于RSA 密钥模长-11。

[OUT] 存放签名结果的缓冲区指针，如果值为NULL, 用于取得签名 结果长度。

[IN,OUT] 输入时表示签名结果缓冲区大小，输出时表示签名结果长

度。

成功。

错误码。

注：权限要求：已取得用户权限。

**7.6.6 RSA验签**

原型 ULONG DEVAPI SKF\_RSAVerify(HCONTAINER hContaincr,BYTE \*pbData,

ULONG ulDataLen,BYTE \*pbSignature,ULONG ulSignLen)

**GM/T 0016—2023**

功能描述 使 用hContainer 指定容器的签名公钥，进行验证RSA 签名操作。

参数 hDev [IN] 设备句柄。

pRSAPubKeyBlob [IN]RSA公钥数据结构。

pbData [IN] 待验证签名的数据。

ulDataLen [IN] 数据长度，应不大于公钥模长-11。

pbSignature [IN] 待验证的签名值。

ulSignLen [IN] 签名值长度，应为公钥模长。

返回值 SAR\_OK: 成功。

其他： 错误码。

**7.6.7** **RSA生成导出会话密钥**

原型 ULONG DEVAPISKF\_RSAExportSessionKey(HCONTAINER hContainer,ULONG

ulAlgId,RSAPUBLICKEYBLOB\*pPubKey,BYTE \*pbData,UIONG \*pulDataLen,

HANDLE phSessionKey)

功能描述 牛成会话密钥并用外部RSA 公钥加密输出。

参数 hConzginer TN 容器句柄。

ulAigId NI 会话密班算法标。

pPubKey 加密会话密钥的 RSA 公 阴数 据结 构。

pbData [OUt 导 出密 会 话 密钥 密 文 接照 PKCS#11.5 要 求

封

pulDataLen [IN T ] 输 人 表示会话密钥密文数据缓冲区长度，输出

时 表 公话密钥 密文的实际长度

phSessionKey IQUT ] 导出的密钥年 病

返回值 SAR \_OK: *疲功。*

其 他 借误码。

**7.6.8 外来RSA 公钥运算**

原型 ULONG DEVAPI SKF\_ExtRSAPubKeyOperation(DEyHANDLE hDev,

RSAPUBLICKEYBLOB\*pRSAPubKeyBlob,BYTEpbInput,ULONG ulInputLen,

BYTE\*pbOutput,ULONG\* pulOutputLen )

功能描述 使用外部传入的RSA 公钥对输入数据做公钥运算并输出结果。

参数 hDev [IN] 设备句柄。

pRSAPubKeyBlob [IN]RSA 公钥数据结构。

pbInput [IN] 指向待运算的原始数据缓冲区。

ulInputLen [IN] 待运算原始数据的长度，应为公钥模长。

pbOutput [OUT] 指 向RSA 公钥运算结果缓冲区，如果该参数为NULL,

则由pulOutputLen 返回运算结果的实际长度。

pulOutputLen [IN,OUT] 输入时表示pbOutput 缓冲区的长度，输出时表示

RSA 公钥运算结果的实际长度。

返回值 SAR\_OK: 成功。

其他： 错误码。

**GM/T 0016—2023**

**7.6.9 生成ECC 签名密钥对**

原型

功能描述 参数

返回值

ULONG DEVAPI SKF\_GenECCKeyPair(HCONTAINER hContainer,ULONG ulAl-

gId,ECCPUBLICKEYBLOB\*pBlob)

生 成ECC 签名密钥对并输出签名公钥。

hContainer [IN] 密钥容器句柄。

ulAlgId [IN] 算法标识，只支持SGD\_SM2\_ 1 算法。

pBlob [OUT] 返回ECC 公钥数据结构。

SAR\_OK: 成功。

其他： 错误码。

注：权限要求：已取得用户权限。

**7.6.10 导入ECC加密密钥对**

原型 ULONG DEVAPI SKF\_ImportECCKeyPair(HCONTAINER hContainer,

PENVELOPEDKEYBLOB pEnvelopedKeyBlob)

功能描述 导入ECC公私钥对。

参数 hContainer [IN] 密钥容器句柄。

pEnvelopedKeyBlob [IN] 受保护的加密密钥对。

返回值 SAR\_OK: 成功。

其他： 错误码。 注：权限要求：已取得用户权限。

**7.6.11 ECC签名**

原型 ULONG DEVAPI SKF\_ECCSignData(HCONTAINER hContainer,BYTE \*pbData,

ULONG ulDataLen,PECCSIGNATUREBLOB pSignature)

功能描述 ECC 数字签名。采用ECC 算法和指定私钥hKey, 对指定数据pbData 进行数字签名。签

名后的结果存放到pSignature中 。

参数 hContainer [IN] 密钥容器句柄。

pbData [IN] 待签名的数据。

ulDataLcn [IN] 待签名数据长度，应小于密钥模长。

pSignature [OUT] 签名值。

返回值 SAR\_OK: 成功。

其他： 错误码。 注：权限要求：已取得用户权限。

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SM2 签名 预处理操作得到的结果，预处理过程符合GM/T 0009—2023中第8章的规定。

**GM/T 0016—2023**

**7.6.12** **ECC验签**

原型 ULONG DEVAPI SKF\_ECCVerify(DEVHANDLE hDev,ECCPUBLICKEYBLOB\*

pECCPubKeyBlob,BYTE \*pbData,ULONG ulDataLen,PECCSIGNATUREBLOB

pSignaturc)

功能描述 用 ECC 公钥对数据进行验签。

参数 hDev [IN] 设备句柄。

pECCPubKeyBlob [IN]ECC 公钥数据结构。

pbData [IN] 待验证签名的数据。

ulDataLen [IN] 数据长度。

pSignature [IN] 待验证签名值。

返回值 SAR\_OK: 成功。

其他： 错误码。

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， 预处理过程符合GM/T 0009—2023中第8章的规定。

**7.6.13** **ECC生成并导出会话密钥**

原型 ULONG DEVAPI SKF\_ECCExportSessionKey(HCONTAINER hContainer,ULONG

ulAlgId,ECCPUBLICKEYBLOB\*pPubKey,PECCCIPHERBLOB pData,HANDLE

\*phSessionKey)

功能描述 生成会话密钥并用外部公钥加密导出。

参数 hContainer [IN] 容器句柄。

ulAlgId [IN] 会话密钥算法标识。

pPubKey [IN] 外部输入的公钥结构。

pData [OUT] 会话密钥密文。

phSessionKey [OUT] 会话密钥句柄。

返回值 SAR\_OK: 成功。

其他： 错误码。

**7.6.14** **ECC外来公钥加密**

原型 ULONG DEVAPI SKF\_ExtECCEncrypt(DEVHANDLE hDev,

ECCPUBLICKEYBLOB\*pECCPubKeyBlob,BYTE\*pbPlainText,ULONG ulPlain-

TextLen,PECCCIPHERBLOB pCipherText)

功能描述 使用外部传入的ECC 公钥对输入数据做加密运算并输出结果。

参数 hDev [IN] 设备句柄。

pECCPubKeyBlob [IN]ECC 公钥数据结构。

pbPlainText [IN] 待加密的明文数据。

ulPlainTextLen [IN] 待加密明文数据的长度。

pCipherText [OUT] 密文数据。

返回值 SAR\_OK: 成功。

其他： 错误码。

**GM/T 0016—2023**

**7.6.15** **ECC生成密钥协商参数并输出**

原型 ULONG DEVAPI SKF\_GenerateAgreementDataWithECC(HCONTAINER hCon-

tainer,

ULONG ulAlgId,ECCPUBLICKEYBLOB\*pContaincrECCPubKeyBlob,

ECCPUBLICKEYBLOB\*pTempECCPubKeyBlob,BYTE\*pbID,ULONG ulIDLen,

HANDLE \*phAgreementHandle)

功能描述 使用ECC 密钥协商算法，为计算会话密钥而产生协商参数，返回容器中ECC 公钥、临时

ECC 密钥对的公钥及协商句柄。

参数 hContainer [IN] 容器句柄。

ulAlgId [IN] 会话密钥算法标识。

pTempECCPubKeyBlob [OUT] 发起方临时ECC 公钥。

pbID

ulIDLen

[IN] 发起方的ID。

[IN] 发起方ID 的长度，不大于32。

phAgreementHandle [OUT] 返回的密钥协商句柄。

返回值 SAR\_OK: 成功。

其他： 错误码。

注：为协商会话密钥，协商的发起方首先调用本函数。

**7.6.16** **ECC计算会话密钥**

原型： ULONG DEVAPI SKF\_GenerateKeyWithECC(HANDLE hAgreementHandle,

ECCPUBLICKEYBLOB\*pECCPubKeyBlob,

ECCPUBLICKEYBLOB\*pTempECCPubKeyBlob,

BYTE\*pbID,ULONG ulIDLen,HANDLE \*phKeyHandle)

功能描述： 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： hAgreementHandle [IN] 密钥协商句柄。

pECCPubKeyBlob [IN] 外部输入的响应方ECC 公钥。

pTempECCPubKeyBlob [IN] 外部输入的响应方临时ECC 公钥。

pbID [IN] 响应方的ID。

ulIDLen [IN] 响应方ID的长度，不大于32。

phKeyHandle [OUT] 返回的密钥句柄。

返回值 SAR\_OK: 成功。

其他： 错误码。

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。计算过程符合GM/T 0009—2023中 9.6的规定。

**GM/T 0016—2023**

**7.6.17 ECC产生协商数据并计算会话密钥**

原型： ULONG DEVAPISKF\_GenerateAgreementDataAndKeyWithECC(HANDLE hCon-

tainer,ULONG ulAlgId,ECCPUBLICKEYBLOB\*pSponsorECCPubKeyBlob, ECCPUBLICKEYBLOB\*pSponsorTempECCPubKeyBlob,

ECCPUBLICKEYBLOB\*pTempECCPubKeyBlob,

BYTE\*pbID,ULONG ulIDLen,BYTE \*pbSponsorID,ULONG ulSponsorIDLen,

HANDLE \*phKeyHandle)

功能描述： 使 用ECC 密钥协商算法，产生协商参数并计算会话密钥，输出临时ECC 密钥对公钥，并

返回产生的密钥句柄。

参数： hContainer [IN] 容器句柄。

ulAlgId [IN] 会话密钥算法标识。

pSponsorECCPubKeyBlob [IN] 发起方的ECC 公钥。

pSponsorTempECCPubKeyBlob [IN] 发起方的临时ECC 公钥。

pTempECCPubKeyBlob [OUT] 响应方的临时ECC 公钥。

pbID [IN] 响应方的ID。

ulIDLen [IN] 响应方ID的长度，不大于32。

pbSponsorID [IN] 发起方的ID。

ulSponsorIDLen [IN] 发起方ID 的长度，不大于32。

phKeyHandle [OUT] 返回的对称算法密钥句柄。

返回值 SAR\_OK: 成功。

其他： 错误码。 注：本函数由响应方调用。

**7.6.18** **导出公钥**

原型 ULONG DEVAPI SKF\_ExportPublicKey(HCONTAINER hContainer,BOOL bSign-

Flag,BYTE\*pbBlob,ULONG\*pulBlobLen)

功能描述 导出容器中的签名公钥或者加密公钥。

参数 hContainer [IN] 密钥容器句柄。

bSignFlag [IN]TRUE 表示导出签名公钥，FALSE 表示导出加密公钥。

pbBlob [OUT] 指向RSA 公钥结构(RSAPUBLICKEYBLOB), 或ECC 公钥

结构(ECCPUBLICKEYBLOB), 或SM9 用户标识(BYTE 数组),如果 此参数为NULL 时，由pulBlobLen返回pbBlob的长度。SM9 相关内容 见附录D。

pulBlobLen [IN,OUT] 输入时表示pbBlob缓冲区的长度，输出时表示导出公钥结

构的大小。

返回值 SAR\_OK: 成功。

其他： 错误码。

**GM/T 0016—2023**

**7.6.19** **导入会话密钥**

原型 ULONG DEVAPI SKF\_ImportSessionKey(HCONTAINER hContainer,ULONG ulAl-

gId,BYTE \*pbWrapedData,ULONG ulWrapedLen,HANDLE \*phKey)

功能描述 导入会话密钥密文，使用容器中的加密私钥解密得到会话密钥。

参数 hContainer [IN] 容器句柄。

ulAlgId [IN] 会话密钥算法标识。

pbWrapedData [IN] 要导入的会话密钥密文。当容器为ECC 类型时，此参数为

ECCCIPHERBLOB 密文数据，当容器为RSA 类型时，此参数为RSA 公 钥加密后的数据，当容器为SM9 类型时，此参数为SM9 密钥封装的封装 密文数据PAIRKEYPACKAGEBLOB。

ulWrapedLen [IN] 会话密钥密文长度。

phKey [OUT] 返回会话密钥句柄。

返回值 SAR\_OK: 成功。

其他： 错误码。 注：权限要求：已取得用户权限。

**7.6.20** **加密初始化**

原型 ULONG DEVAPI SKF\_EncryptInit(HANDLE hKey,BLOCKCIPHERPARAM En-

cryptParam)

功能描述 数据加密初始化。设置数据加密的算法相关参数。

参数 hKey [IN] 加密密钥句柄。

EncryptParam [IN] 分组密码算法相关参数：初始向量、初始向量长度、反馈值的位

长度。

返回值 SAR\_OK: 成功。

其他： 错误码。

**7.6.21** **单包数据加密**

原型

功能描述

参数

ULONG DEVAPI SKF\_Encrypt(HANDLE hKey,BYTE\*pbData,ULONG ulDat-

aLen,BYTE\*pbEncryptedData,ULONG \*pulEncryptedLen)

单包数据的加密操作。用指定加密密钥对指定数据进行加密，被加密的数据只包含一个 数据包，加密后的密文保存到指定的缓冲区中。SKF\_Encrypt 只对单包数据进行加密， 在调用SKF\_Encrypt 之前，应调用SKF\_EncryptInit 初始化加密操作。SKF\_Encypt 等价 于先调用SKF\_EncryptUpdate 再调用SKF\_EncryptFinal 。 单包数据可包含若干分组。

|  |  |
| --- | --- |
| hKey  pbData  ulDataLen  pbEncryptedData  pulEncryptedLen | [IN] 加密密钥句柄。  [IN] 待加密数据。  [IN] 待加密数据长度。  [OUT] 加密后的数据缓冲区指针，可为NULL, 用于获得加密后数 据长度。  [IN,OUT] 输入时表示结果数据缓冲区长度，输出时表示结果数 据实际长度。 |

**GM/T 0016—2023**

返回值 SAR\_OK: 成功。

其他： 错误码。

**7.6.22** **多包数据加密**

原型 ULONG DEVAPI SKF\_EncryptUpdate(HANDLE hKey,BYTE\*pbData,ULONG ul-

DataLen,BYTE \*pbEncryptedData,ULONG \*pulEncryptedLen)

功能描述 多个数据包的加密操作。用指定加密密钥对指定数据进行加密，被加密的数据包含多个

数据包，加密后的密文保存到指定的缓冲区中。 SKF\_EncryptUpdate 对数据进行加密， 在调用SKF\_EncryptUpdate 之 前，应调 用SKF\_EncryptInit 初始化加密操作；在调用

SKF \_EncryptUpdate 处理全部数据包之后，应调用SKF \_EncryptFinal 结束加密操作。

|  |  |  |
| --- | --- | --- |
| 参数 | hKey | [IN] 加密密钥句柄 |
|  | pbData | N] 待加密数据。 |
|  | ulDataLen | [IN] 待加密数据长度。 |

pbEncryptedData [OUT] 加密后的数据缓冲区指针。

pulEncryptedLen [OUT] 返回加密后的数据长度。

返回值 SAROK: 成功。

其 他： 错医码

**7.6.23** **结束加密**

LLonc DesKF E ial pLEh,BYTE\*pbErerypedData,

原型

ULONG enspedDatal

功能描述 结束多个分维数拥的加密，返，余加 一 果。先调用S KF Bncryptmit初始化加密操

作，再调用SkT ErptOpdate对多个分组数据进行加密取后调 用SkFEncryptFinal 结束多个分组 数 的 加 蓝

参数

返回值

hKey

pbEncyptedData

ulEncyptedDataLen SAR \_OK: 成 功。

EINI 加密密钥句柄。

[OUT] 加密结果的缓冲区。

[OUT] 加密结果的长度。

其他： 错 误 码。

**7.6.24** **解密初始化**

原型 ULONG DEVAPI SKF\_DecryptInit(HANDLE hKey,BLOCKCIPHERPARAM De-

cryptParam)

功能描述 数据解密初始化，设置解密密钥相关参数。调用SKF\_DecryptInit 之后，可调用SKF\_De-

crypt 对单个分组数据进行解密，也可多次调用SKF\_DecryptUpdate 之后再调用SKF\_De- cryptFinal 完成对多个分组数据的解密。

参数 hKey [IN] 解密密钥句柄。

DecryptParam [IN] 分组密码算法相关参数：初始向量、初始向量长度、反馈值

的位长度。

返回值 SAR\_OK: 成功。

其他： 错误码。

**GM/T 0016—2023**

**7.6.25** **单包数据解密**

原型 ULONG DEVAPI SKF\_Decrypt(HANDLE hKey,BYTE\*pbEncryptedData,ULONG

ulEncryptedLen,BYTE\*pbData,ULONG \*pulDataLen)

功能描述 单包数据的解密操作。用指定解密密钥对指定数据进行解密，被解密的数据只包含一个

数据包，解密后的明文保存到指定的缓冲区中。SKF\_Decrypt 只对单包数据进行解密， 在调用SKF\_Decrypt 之前，应调用SKF\_DecryptInit 初始化解密操作。SKF\_Decypt 等价 于先调用SKF\_DecryptUpdate 再调用SKF\_DecryptFinal 。 单包数据可包含若干分组。

参数 hKcy [IN] 解密密钥句柄。

pbEncryptedData [IN] 待解密数据。

ulEncryptedLen [IN] 待解密数据长度。

pbData [OUT] 指向解密后的数据缓冲区指针，当为NULL 时可获得解密

后的数据长度。

pulDataLen [IN,OUT] 输入时表示结果数据缓冲区长度，输出时表示结果数据

实际长度。

返回值 SAR\_OK: 成功。

其他： 错误码。

**7.6.26** **多包数据解密**

原型 ULONG DEVAPI SKF\_DecryptUpdate(HANDLE hKey,BYTE\*pbEncryptedData,

功能描述

参数

ULONG ulEncryptedLen,BYTE\*pbData,ULONG \*pulDataLen)

多个数据包的解密操作。用指定解密密钥对指定数据进行解密，被解密的数据包含多个 分组，解密后的明文保存到指定的缓冲区中。SKF\_DecryptUpdate 对数据进行解密，在

调用SKF\_DecryptUpdate 之前，应调用SKF\_DecryptInit 初始化解密操作；在调用

SKF\_DecryptUpdate 处理所有数据包之后，应调用SKF\_DecryptFinal 结束解密操作。

|  |  |
| --- | --- |
| hKey  pbEncryptedData ulEncryptedLen | [IN] 解密密钥句柄。  [IN] 待解密数据。  [IN] 待解密数据长度。 |

pbData [OUT] 指向解密后的数据缓冲区指针。

pulDataLen [IN,OUT] 输入时表示结果数据缓冲区长度，输出时表示结果数据

实际长度。

返回值 SAR\_OK: 成功。

其他： 错误码。

**7.6.27** **结束解密**

原型 ULONG DEVAPI SKF\_DecryptFinal(HANDLE hKey,BYTE \*pbDecryptedData,

ULONG \*pulDecrypledDataLen)

功能描述 结束多包数据的解密。先调用SKF\_DecryptInit 初始化解密操作，再调用SKF\_Decryp-

1Update对多包数据进行解密，最后调用SKF\_DecryptFinal 结束多包数据的解密。

参数 hKey [IN] 解密密钥句柄。

**GM/T 0016—2023**

|  |  |
| --- | --- |
| pbDecryptedData | [OUT] 指向解密结果的缓冲区，如果此参数为NULL 时，由  pulDecryptedDataLen 返回解密结果的长度。 |
| pulDecryptedDataLen | [IN,OUT] 输入时表示pbDecryptedData 缓冲区的长度，输出 时表示解密结果的长度。 |

返回值 SAR\_OK: 成功。

其他： 错误码。

**7.6.28** **密码杂凑初始化**

原型 ULONG DEVAPISKF\_ DigestInit(DEVHANDLE hDev,ULONG ulAlgID,

ECCPUBLICKEYBLOB \*pPubKey,BYTE \*pucHD,ULONG ulIDLen,HANDLE

\*phHash)

功能描述 初始化密码杂凑计算操作，指定计算密码杂凑的算法。

参数 hDev [IN] 连接设备时返回的设备句柄。

ulAlgID [IN] 密码杂凑算法标识。

pFubKey [IN] 签名者公钥。当ulAlgID 为 SGD \_SM3 时有效。

pucIDD TNI 签名 者的ID 值，当uAlgID 为SGD\_SM3 时有效。

ulIDLen TN 签 名者面 的 长 度 ， @AIgID 为 SGD \_SM3 时有效。

phHash 00T] 密码杂凑对象何柄。

返回值 SAR \_ OK: 我 功

其他： 储误明。

|  |  |
| --- | --- |
| 注：当ulAlgID为SGD SMOEDLen 不 | 清 况 下 ey 、pucID有效 行 SM2算法签名预处理1操作。 |
| 计算过程符合GM /F0000320 23中8.1 | 。 |

**7.6.29** **单包数据密码杂凑**

uloNG IEVAPISKE DgstIHANDEEhlasb,BYTIGspbData,uLoNG ulDat-

原型

aLen,BYTE \*pbHashData,ULONG \*pulHashLen)

功能描述 对单包的消息进行密码杂凑计算。调用SKF\_Digest 之前，应调用SKF\_DigestInit 初始化

密码杂凑计算操作。SKF\_Digest 等价于多次调用SKF DigestUpdate之后再调用SKF

DigestFinal 。

参数 hHash [IN] 密码杂凑对象何柄。

pbData N ] 指向消息数据的缓神区。

ulDataLen [IN ] 消息数据的长度。

pbHashData [OUT] 密码杂凑数据缓冲区指针，当此参数为NULL 时，由pul-

HashLen 返回密码杂凑结果的长度。

pulHashLen [IN,OUT] 输入时表示结果数据缓冲区长度，输出时表示结果数

据实际长度。

返回值 SAR\_OK: 成功。

其他： 错误码。

**GM/T 0016—2023**

**7.6.30** **多包数据密码杂凑**

原型 ULONG DEVAPI SKF\_DigestUpdate(HANDLE hHash,BYTE\*pbData,ULONG

ulDataLen)

功能描述 对多个分组的消息进行密码杂凑计算。调用SKF\_DigestUpdate 之前，应调用SKF\_Di-

gestInit 初始化密码杂凑计算操作；调用SKF\_DigestUpdate 之后，应调用SKF\_DigestFi- nal结束密码杂凑计算操作。

参数 hHash  [IN] 密码杂凑对象句柄。

pbData [IN] 指向消息数据的缓冲区。

ulDataLen [IN ] 消息数据的长度。

返回值 SAR\_OK: 成 功

其他： 错误码。

**7.6.31** **结束密码杂凑**

原型 ULONG bpEVAPI SKF\_DigestFinal(HANDLE hHash,BYTE\*pHashDala ,ULONG

\*pulHashLen

功能描述 结束多个分组消息的密码杂淡引算操作函密码杂淡结果保存到脂定的缓冲区。

参数 hHash [IN 密 面杂法 对 象句柄。

pHeshData [our 波 面 的 密 凑 结 冲 区指针学师乘比参数nuL 时 由pul

LtasHLcn 返回杂决 的 长 度

pulHashLen IN- OUT ] 输 入表 示 杂 凑生 果缓冲区的长度，输出时表示密码杂凑结

果的长度

返回值 SAR OK: 成 功

共 他 错误码。

注：SKF Digesthinal用于在调用着ESKD DigesuOptae之后完成密码杂凑操作。

**7.6.32** **消息鉴别码运算初始化**

原型 ULONG DEVARI SKF\_MacInit(HANDLE hKey,BLOCKCIPHERPARAM\*pMac-

Param,HANDLE\* phMac )

功能描述 初始化消息鉴别码计算操作，设置计算消息鉴别码的初始参数，并返回消息鉴别码句柄。

参数 hKey [IN] 计算消息鉴别码的密钥句柄。

pMacParam [IN ] 消息认证计算相关参数，包括初始向量、初始向量长度、填充方法。

phMac [OUT] 消息鉴别码对象句柄。

返回值 SAR\_OK: 成功。

其他： 错误码。

注：消息鉴别码计算采用分组加密算法的CBC模式，将加密结果的最后一块作为计算结果。待计算数据的长度 是分组加密算法块长的倍数，接口内部不作数据填充。

**7.6.33** **单包数据消息鉴别码运算**

原型 ULONG DEVAPI SKF\_Mac(HANDLE hMac,BYTE\*pbData,ULONG ulDataLen,

BYTE \*pbMacData,ULONG \*pulMacLen)

**GM/T 0016—2023**

|  |  |
| --- | --- |
| 功能描述 | SKF\_Mac 计算单包数据的消息鉴别码。 |
| 参数 | hMac [IN] 消息鉴别码句柄。  pbData [IN] 指向待计算数据的缓冲区。  ulDataLen [IN] 待计算数据的长度。  pbMacData [OUT] 指向计算后的Mac 结果，如果此参数为NULL 时，由 pulMacLen  返回计算后Mac 结果的长度。  pulMacLen [IN,OUT] 输入时表示pbMacData缓冲区的长度，输出时表示 Mac 结果  的长度。 |
| 返回值 | SAR\_OK: 成功。  其他： 错误码。 |

注：调用SKF\_Mac之前，调用SKF\_MacInit初始化消息鉴别码计算操作。SKF\_Mac等价于多次调用SKF\_Mac- Update之后再调用SKF\_MacFinal。

**7.6.34** **多包数据消息鉴别码运算**

原型 ULONG DEVAPI SKF\_MacUpdate(HANDLE hMac,BYTE\*pbData,ULONG ulDat-

aLen)

功能描述 计算多个分组数据的消息鉴别码。

参数 hMac [IN] 消息鉴别码句柄。

pbData [IN] 指向待计算数据的缓冲区。

plDataLen [IN] 待计算数据的长度。

返回值 SAR\_OK: 成功。

其他： 错误码。

注：调用SKF\_MacUpdate之前，调用SKF\_MacInit初始化消息鉴别码计算操作；调用SKF\_MacUpdate之后，调 用SKF\_MacFinal结束多个分组数据的消息鉴别码计算操作。

**7.6.35** **结束消息鉴别码运算**

原型 ULONG DEVAPI SKF\_MacFinal(HANDLE hMac,BYTE \*pbMacData,ULONG

\*pulMacDataLen)

功能描述 结束多个分组数据的消息鉴别码计算操作。

|  |  |  |
| --- | --- | --- |
| 参数 | hMac  pbMacData | [IN] 消息鉴别码句柄。  [OUT] 指向消息鉴别码的缓冲区，当此参数为NULL 时，由pulMac- DataLen 返回消息鉴别码返回的长度。 |
|  | pulMacDataLen | [OUT] 调用时表示消息鉴别码缓冲区的最大长度，返回消息鉴别码 的长度。 |
| 返回值 | SAR\_OK: 其他： | 成功。  错误码。 |

注：SKF\_MacFinal用于在SKF\_MacUpdate之后完成消息鉴别码运算操作。

**7.6.36** **带密钥的杂凑运算初始化**

原型 ULONG DEVAPI SKF\_HMACInit(HANDLE hKey,HANDLE \*phHMac)

**GM/T 0016—2023**

功能描述 三步式带密钥的数据杂凑运算的第一步。

参数 hKey [IN] 密钥句柄。

phHMac [OUT]HMAC 对象句柄。

返回值 SAR\_OK: 成功。

其他： 错误码。

**7.6.37** **带密钥的单包杂凑运算**

原型 ULONG DEVAPI SKF\_HMAC(HANDLE hHMac,BYTE \*pbData,

ULONG ulDataLength,BYTE\*pbHashData,ULONG \*pulHashLen)

功能描述 单包数据的带密钥的数据杂凑运算，对输入的明文进行杂凑运算。

参数 hHMac [IN] HMAC对象甸柄

pbData

ulDataLongth

[IN] [N]

缓冲区指针，指向明文数据。 明文数据的长度。

返回值

pbHashData

*puiHashLen*

SAR OK: 其他：

[ OUT] 密码杂凑数据缓冲区指针，当此参数为NULL时，由pulHash- Lcn 返回密码杂凑结果的长度。

fIN , OUT] 输人时表示结果数据缓冲区长度，输出时表示结果数据实

际长度。

成 功

循选店

**7.6.38** **带密钥的多包杂凑运算**

urONGDEVAPESK0 nMAc e( A oLEhifvacn \*pbDaa,tLonG

原型

ulDataLength )

功能描述 三步式带密钥的数据泵淡运算的郅步对输人的明文进行杂淡运算。

参数 HHMac DNI HMAC对象句柄。

pbData [IN] 缓冲区指针，指向明文数据。

ulDataLength [IN] 明文数据的长度。

返回值 SAR\_OK: 成功。

其他： 锆 误 码。

**7.6.39** **带密钥的杂凑运算结束**

原型 ULONG DEVAPISKF\_HMACFinal( HANDLE hHMac,BYTE \*pbHMac,ULONG

\*pulHMacLength)

功能描述 三步式带密钥的数据杂凑运算的第三步，杂凑运算结束返回结果。

参数 hHMac [IN]HMAC 对象句柄。

pbHMac [OUT] 缓冲区指针，指向杂凑计算结果。

pulHMacLength [OUT] 杂凑计算结果的长度。

返回值 SAR\_OK: 成功。

其他： 错误码。

**GM/T 0016—2023**

**7.6.40** **关闭密码对象句柄**

原型 ULONG DEVAPI SKF\_CloseHandle(HANDLE hHandle)

功能描述 关闭会话密钥、密码杂凑对象、消息鉴别码对象、ECC 密钥协商句柄。

参数 hHandle [IN] 要关闭的对象句柄。

返回值 SAR\_OK: 成功。

其他： 错误码。

**7.7** **验证调试**

**7.7.1** **概述**

验证调试类接口函数仅用于在调试、测试和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。

**7.7.2 生成RSA 密钥对并导出**

原型 ULONG DEVAPISKF\_GenExtRSAKey(DEVHANDLEhDev,ULONG ulBitsLen,

RSAPRIVATEKEYBLOB\*pBlob)

功能描述 由设备生成RSA 密钥对并明文输出。

参数 hDev [IN] 设备句柄。

ulBitsLen [IN] 密钥模长。

pBlob [OUT] 返回的私钥数据结构。

返回值 SAR\_OK: 成功。

其他： 错误码。

注：生成的私钥只用于输出，接口内不做保留和计算。

**7.7.3 生成ECC 密钥对并导出**

原型

功能描述 参数

ULONG DEVAPI SKF\_GenExtECCKey(DEVHANDLE hDev,ULONG ulBitsLen, ECCPRIVATEKEYBLOB\*pPrivate\_key,ECCPUBLICKEYBLOB\*pPublic\_key,)

由设备生成ECC 密钥对并明文输出。

hDev [IN] 设备句柄。

ulBitsLen [IN] 密钥模长。

pPrivate\_key pPublick\_key

[OUT] 返回的私钥数据结构。

[OUT] 返回的公钥数据结构。

返回值 SAR\_OK: 成功。

其他： 错误码。

注：生成的私钥只用于输出，接口内不做保留和计算。

**GM/T 0016—2023**

**7.7.4** **RSA外来私钥运算**

函数原型 ULONG DEVAPI SKF\_ExtRSAPriKeyOperation(DEVHANDLE hDev,

RSAPRIVATEKEYBLOB\*pRSAPriKeyBlob,BYTE\*pblnput,ULONG ulInputLen,

BYTE\*pbOutput,ULONG\*pulOutputLen)

功能描述 直接使用外部传入的RSA 私钥对输入数据做私钥运算并输出结果。

参数 hDev [IN] 设备句柄。

pRSAPriKeyBlob [IN]RSA 私钥数据结构。

pbInput [IN] 指向待运算数据缓冲区。

ulInputLen [IN] 待运算数据的长度，应为公钥模长。

pbOutput [OUT]RSA 私钥运算结果，如果该参数为NULL, 则由pulOut-

putLen返回运算结果的实际长度。

pulOutputLen [IN,OUT] 输入时表示pbOutput 缓冲区的长度，输出时表示

RSA 私钥运算结果的实际长度。

返回值 SAR\_OK: 成功。

其他： 错误码。

**7.7.5** **ECC外来私钥解密**

函数原型 ULONG DEVAPI SKF\_ExtECCDecrypt(DEVHANDLE hDev,

ECCPRIVATEKEYBLOB\*pECCPriKeyBlob,PECCCIPHERBLOB pCipherText,

BYTE\*pbPlainText,ULONG\*pulPlainTextLen)

功能描述 使用外部传入的ECC 私钥对输入数据做解密运算并输出结果。

|  |  |  |
| --- | --- | --- |
| 参数 | hDev  pECCPriKeyBlob pCipherText  pbPlain Text | [IN] 设备句柄。  [IN]ECC 私钥数据结构。 [IN] 待解密的密文数据。  [OUT] 函数执行返回的明文数据，如果该参数为NULL, 则由 pulPlainTextLen 返回明文数据的实际长度。 |
|  | pulPlainTextLen | [IN,OUT] 输入时表示pbPlainText 缓冲区的长度，输出时表示 明文数据的实际长度。 |

返回值 SAR\_OK: 成功。

其他： 错误码。

**7.7.6** **ECC外来私钥签名**

函数原型 ULONG DEVAPI SKF\_ExtECCSign(DEVHANDLE hDev,

ECCPRIVATEKEYBLOB\*pECCPriKeyBlob,BYTE\*pbData,ULONG ulDataLen,

PECCSIGNATUREBLOB pSignaturc)

功能描述 使用外部传入的ECC 私钥对输入数据做签名运算并输出结果。

参数 hDev [IN] 设备句柄。

pECCPriKeyBlob [IN]ECC 私钥数据结构。

pbData [IN] 待签名数据。

ulDataLen [IN] 待签名数据的长度。

**GM/T 0016—2023**

pSignature [OUT] 签名值。

返回值 SAR\_OK: 成功。

其他： 错误码。

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， 预处理过程符合GM/T 0009—2023中第8章的规定。

**7.7.7** **外来对称密钥加密**

函数原型 ULONG DEVAPI SKF\_ExtKeyEncrypt(HANDLE hDev,BYTE\*pbKey,ULONG

ulKeyLen,BYTE\*pbData,ULONG ulDataLen,BYTE\*pbEncryptedData,ULONG

\*pulEncryptedLen)

功能描述 使用外来加密密钥对指定数据进行加密，被加密的数据只包含单一分组，加密后的密文

保存到指定的缓冲区中。

参数 hDev [IN] 设备句柄。

pbKey [IN] 外来密钥。

ulKeyLen [IN] 外来密钥长度。

pbData [IN] 待加密数据。

ulDataLen [IN] 待加密数据长度。

pbEncryptedData [OUT] 加密后的数据缓冲区指针，可为NULL, 用于获得加密后数

据长度。

pulEncryptedLen [IN,OUT] 输入时表示结果数据缓冲区长度，输出时表示结果数

据实际长度。

返回值 SAR\_OK: 成功。

其他： 错误码。

**7.7.8** **外来对称密钥解密**

函数原型 ULONG DEVAPI SKF\_ExtKeyDecrypt(HANDLE hDev,BYTE\*pbKey,ULONG

ulKeyLen,BYTE\*pbData,ULONG ulDataLen,BYTE\*pbEncryptedData,ULONG

\*pulEncryptedLen)

功能描述 使用外来解密密钥对指定数据进行解密，被解密的数据只包含单一分组，解密后的密文

保存到指定的缓冲区中。

参数 hDev [IN] 设备句柄。

pbKey [IN] 外来密钥。

ulKeyLen [IN] 外来密钥长度。

pbData [IN] 待解密数据。

ulDataLen [IN] 待解密数据长度。

pbEncryptedData [OUT] 解密后的数据缓冲区指针，可为NULL, 用于获得解密后数

据长度。

pulEncryptedLen [IN,OUT] 输入时表示结果数据缓冲区长度，输出时表示结果数

据实际长度。

返回值 SAR\_OK: 成功。

其他： 错误码。

**GM/T 0016—2023**

**7.7.9** **带外部密钥的杂凑运算**

原型 ULONG DEVAPI SKF\_ExtHmac(HANDLE hDev,BYTE\*pbKey,ULONG ulKey-

Length,BYTE\*pbData,ULONG ulDataLength,ULONG uAlgID,BYTE \*pbHash-

Data,ULONG \*pulHashLen)

功能描述 单包数据的带密钥的密码杂凑运算。

参数 hDev [IN] 设备句柄。

pbKEY [IN] 外部密钥。

ulKeyLength [IN] 外部密钥的长度。

pbData [IN] 缓冲区指针，指向明文数据。

ulDataLength [IN] 明文数据的长度。

uAlgID [IN] 密码杂凑算法标识。

pbHashData [OUT] 密码杂凑数据缓冲区指针，当此参数为NULL 时，由pulHash-

Len 返回密码杂凑结果的长度。

pulHashLen [IN,OUT] 输入时表示结果数据缓冲区长度，输出时表示结果数据实

际长度。

返回值 SAR\_OK: 成功。

其他： 错误码。

**7.7.10** **设备命令传输**

原型 ULONG DEVAPI SKF\_Transmit(DEVHANDLE hDev,BYTE\*pbCommand,

ULONG ulCommandLen,BYTE\*pbData,ULONG\*pulDataLen)

功能描述 将命令直接发送给设备，并返回结果。

参数 hDev [IN] 设备句柄。

pbCommand [IN] 设备命令。 ulCommandLen [IN] 命令长度。

pbData [OUT] 返回结果数据。

pulDataLen [IN,OUT] 输入时表示结果数据缓冲区长度，输出时表示结果数据实

际长度。

返回值 SAR\_OK: 成功。

其他： 错误码。

**8** **接口使用要求**

**8.1** **设备使用阶段**

设备的使用分成两个阶段。

a) 出厂阶段：设备出厂时，预置设备认证密钥，在此阶段除修改设备认证密钥及创建应用操作 外，禁止其他操作。

b) 应用阶段：已创建了应用的设备进入应用阶段。在此阶段，可进行所有操作。

**GM/T 0016—2023**

**8.2** **权限管理**

**8.2.1** **权限分类**

权限分为设备权限、用户权限和管理员权限。

a) 设备权限：通过设备认证后获得设备权限。

b) 用户权限：用户PIN 码验证通过后，获得用户权限，用户权限只作用于其所在的应用。

c) 管理员权限：管理员PIN 码验证通过后，获得管理员权限，管理员权限只作用于其所在的 应用。

**8.2.2** **权限使用**

权限的使用遵循以下要求。

a) 设备权限仅用于创建应用、删除应用和修改设备认证密钥。

b) 创建和删除容器应已取得用户权限。

c) 创建文件的权限在创建应用时指定。

d) 文件的读写权限在创建文件时指定。

e) 容器内私钥的使用应已取得用户权限。

1) 用户PIN 码和管理员PIN 码均具有最大重试次数，在创建应用时设定。当验证PIN 码错误 次数达到最大重试次数后，PIN 码即锁死。

g) 用 户PIN 码的解锁应已取得管理员权限。

h) 用户PIN 码的修改应已取得用户权限，管理员PIN 码的修改应已取得管理员权限。

**8.2.3** **设备认证**

应通过设备认证后才能在设备内创建和删除应用。

设备认证使用分组密码算法和设备认证密钥进行。认证的流程如下：

a) 被认证方调用SKF\_GenRandom 函数从设备获取8字节随机数RND, 并用0x00 将其填充至 密码算法的分块长度，组成数据块D0;

b) 被认证方对 D0 加密，得到加密结果D1, 并调用SKF\_DevAuth(), 将 D1 发送至设备；

c) 设备收到D1 后，验证D1 是否正确，正确则通过设备认证，否则设备认证失败。

**8.2.4 PIN码安全要求**

PIN 码的使用符合以下要求：

a)PIN 码长度应不少于6个字符；

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

c)PIN 码在设备中应安全存储，不可从设备中导出。

**8.3** **其他安全要求**

智能密码钥匙是一类密码模块，在物理安全、非入侵安全、敏感安全参数管理、自测试、生命周期保 障方面应符合GM/T 0028—2014中第8章的规定。

**GM/T 0016—2023**

**附** **录** **A**  **(规范性)** **错误代码定义**

**A.1** **错误代码定义**

本文件定义的错误代码如表A.1 所示。

**表A.1 错误代码定义和说明**

|  |  |  |
| --- | --- | --- |
| 宏描述 | 预定义值 | 说明 |
| SAR\_OK | 0x00000000 | 成功 |
| SAR\_FAIL | 0x0A000001 | 失败 |
| SAR\_UNKNOWNERR | 0x0A000002 | 异常锆误 |
| SAR\_NOTSUPPORTYF TERR | 0x0A000003 | 不支持的服务 |
| SAR\_FILEERR | 0x0A000004 | 文件操作错误 |
| SAR\_INVALIDHANDLEERR | D⁰000 | 无效的句柄 |
| SAR\_INVALIDPARAMERR | 0x0A00BBBB | 无效的委数 |
| SAR\_READFILEERR | 0x0 007 | 读理谓误 |
| SAR\_WRITEFIIEERR | 0x0AQ00008 | 文件错误 |
| SAR\_NAMELENERR | 0x0A000009 | 名称长度错误 |
| SAR\_KEYUSAGEERR | 0x0A00000A | 密钮用途误 |
| SAR\_MODULUSLENERR | 0x0A00000B | 模的报要错联 |
| SAR\_NOTINKTIALIZEERR | 0k0A00000C | 未初始化 |
| SAR\_OBJERR | 0x0A00000D | 对象错误 |
| SAR\_MEMORYERR | 0x0A00000E | 内存错误 |
| SAR\_TIMEOUTERR | 0x0A00000F | 超时 |
| SAR\_INDATALENERR | 0x0A000010 | 输入数据长度错误 |
| SAR\_INDATAERR | 0x0A000011 | 输人数据错误 |
| SAR\_GENRANDERR | 0x0A000012 | 生成随机数错误 |
| SAR\_HASHOBJERR | 0x0A000013 | HASH对象错 |
| SAR\_HASHERR | 0x0A000014 | HASH运算错误 |
| SAR\_GENRSAKEYERR | 0x0A000015 | 产生RSA密钥错 |
| SAR\_RSAMODULUSLENERR | 0x0A000016 | RSA密钥模长错误 |
| SAR\_CSPIMPRTPUBKEYERR | 0x0A000017 | CSP服务导入公钥错误 |
| SAR\_RSAENCERR | 0x0A000018 | RSA加密错误 |
| SAR\_RSADECERR | 0x0A000019 | RSA解密错误 |
| SAR\_HASHNOTEQUALERR | 0x0A00001A | HASH值不相等 |

**GM/T 0016—2023**

**表A.1 错误代码定义和说明(** **续**)

|  |  |  |
| --- | --- | --- |
| 宏描述 | 预定义值 | 说明 |
| SAR\_KEYNOTFOUNTERR | 0x0A00001B | 密钥未发现 |
| SAR\_CERTNOTFOUNTERR | 0x0A00001C | 证书未发现 |
| SAR\_NOTEXPORTERR | 0x0A00001D | 对象未导出 |
| SAR\_DECRYPTPADERR | 0x0A00001E | 解密时做补丁错误 |
| SAR\_MACLENERR | 0x0A00001F | MAC长度错误 |
| SAR\_BUFFER\_TO0\_SMALL | 0x0A000020 | 缓冲区不足 |
| SAR\_KEYINFOTYPEERR | 0x0A000021 | 密钥类型错误 |
| SAR\_NOT\_EVENTERR | 0x0A000022 | 无事件错误 |
| SAR\_DEVICE\_REMOVED | 0x0A000023 | 设备已移除 |
| SAR PIN\_INCORRECT | 0x0A000024 | PIN不正确 |
| SAR\_PIN\_LOCKED | 0x0A000025 | PIN被锁死 |
| SAR\_PIN\_INVALID | 0x0A000026 | PIN无效 |
| SAR\_PIN\_LEN\_RANGE | 0x0A000027 | PIN长度错误 |
| SAR\_USER\_ALREADY\_LOGGED\_IN | 0x0A000028 | 用户已经登录 |
| SAR\_USER\_PIN\_NOT\_INITIALIZED | 0x0A000029 | 没有初始化用户口令 |
| SAR USER TYPE\_INVALID | 0x0A00002A | PIN类型错误 |
| SAR APPLICATION\_NAME\_INVALID | 0x0A00002B | 应用名称无效 |
| SAR\_APPLICATION\_EXISTS | 0x0A00002C | 应用已经存在 |
| SAR\_USER\_NOT\_LOGGED\_IN | 0x0A00002D | 用户没有登录 |
| SAR\_APPLICATION\_NOT\_EXISTS | 0x0A00002E | 应用不存在 |
| SAR FILE ALREADY\_EXIST | 0x0A00002F | 文件已经存在 |
| SAR\_NO\_ROOM | 0x0A000030 | 空间不足 |
| SAR\_FILE\_NOT\_EXIST | 0x0A000031 | 文件不存在 |
| SAR\_REACH\_MAX\_CONTAINER\_COUNT | 0x0A000032 | 已达到最大可管理容器数 |
| 扩展错误码 | | |
| SAR\_AUTH\_BLOCKED | 0x0A000033 | 密钥已被锁住 |
| SAR INVALIDCONTAINERERR | 0x0A000035 | 无效容器 |
| SAR CONTAINER\_NOT\_EXISTS | 0x0A000036 | 容器不存在 |
| SAR CONTAINER EXISTS | 0x0A000037 | 容器已存在 |
| SAR\_KEYNOUSAGEERR | 0x0A000039 | 密钥未被使用 |
| SAR\_FILEATTRIBUTEERR | 0x0A00003A | 文件操作权限错误 |
| SAR\_DEVNOAUTH | 0x0A00003B | 设备未认证 |

**GM/T 0016—2023**

**附** **录** **B**  **(规范性)** **SM9** **应用接口**

**B.1 数据结构**

**B.1.1** SM9签名主私钥数据结构：

a) 类型定义

#define SM9\_MODULUS\_BITS\_LEN 256

#define SM9\_MODULUS\_BYTESLEN((SM9\_MODULUS BITS\_LEN+7)/8)

typedef struct Struct \_SM9MASTPRIVATBKEYBLOB{ ULONG ulBitL,ep

BYTE cbPfivateKey [SM9\_MODULUS\_BYTES\_LEN]; }SM9MASTPRIVATEKEYBLOB;

typedef SM9MAsTPRIVATEKEYBLOB SM9SIGNMASTPRIVATEKEYBLOB

\*PSM9 SIGNMASTPRIVATEKEYBLOB

typedef SM9MASTPRIVATEREYBEOBSMOENCMASTPRIVATEKEYBLOB,

\*PSM9ENCMASTPRIV 座 K OB ,

其中，shgyoDULus rTeS EEN为 算数的学K 化 32。

b) 数据项晶述见表B 压

**表B.1** **SNi9** **签名主私钥数据结构**

|  |  |  |  |
| --- | --- | --- | --- |
| 数据项 | 类理 | 意义 | 备注 |
| ulBitLen | ONS5 | 模数的位长度 | 值为256 |
| cbPrivateKey | BYFF数组 | 签备毛密铒 | 有限域的整数 |

**B.1.2** SM9签名重公钥数据结构：

a) 类型定义

#define SM9\_XCOQRDINATE\_BITS\_LEN SM9\_MODULUS BITS LEN

#define SM9\_XCOORDINATE\_BYTES LEN((SM9 xcOORDINATE\_BITS\_LEN+7)/8)

=define SM9\_YCOORDINATE BITS\_LEN SM9\_MODULUS\_BITS\_LEN

=define SM9\_YCOORDINATE\_BYTES\_LEN((SM9\_YCOORDINATE\_BITS\_LEN+7)/8)

typedef struct Struct\_SM9KEYBLOB2{ ULONG ulCompressTypc; ULONG ulBitLen;

BYTE cbXACoordinate[SM9\_XCOORDINATE\_BYTES\_LEN];

BYTE cbXBCoordinate[SM9\_XCOORDINATE\_BYTES\_LEN];

BYTE cbYACoordinate[SM9\_YCOORDINATE\_BYTES\_LEN];

BYTE cbYBCoordinate[SM9\_YCOORDINATE\_BYTES\_LEN]; }SM9KEYBLOB2,\*PSM9KEYBLOB2;

typedef SM9KEYBLOB2 SM9SIGNMASTPUBLICKEYBLOB,

**GM/T 0016—2023**

\*PSM9SIGNMASTPUBLICKEYBLOB;

其中，SM9\_XCOORDINATE\_BYTES\_LEN 为 SM9 签名主公钥 X 坐标的字节长度，值为32; SM9\_YCOORDINATE\_BYTES\_LEN 为 SM9 签名主公钥Y 坐标的字节长度，值为32。

b) 数据项描述见表B.2。

**表B.2 SM9 签名主公钥数据结构**

|  |  |  |  |
| --- | --- | --- | --- |
| 数据项 | 类型 | 意义 | 备注 |
| ulCompressType | ULONG | 压缩类型 | 值为2、3或4。当为4时cbYACoordinate、 cbYBCoordinate有效，当为其他值时  cbYACoordinate、cbYBCoordinate无效 |
| ulBitLen | ULONG | 模数的实际位长度 | 值为256 |
| cbXACoordinate | BYTE数组 | 曲线上点的X坐标高维 | 有限域上的整数 |
| cbXBCoordinate | BYTE数组 | 曲线上点的X坐标低维 | 有限域上的整数 |
| cbYACoordinate | BYTE数组 | 曲线上点的Y坐标高维 | 有限域上的整数 |
| cbYBCoordinate | BYTE数组 | 曲线上点的Y坐标低维 | 有限域上的整数 |

**B.1.3** SM9加密主公钥数据结构：

a) 类型定义

typedef struct Struct\_SM9KEYBLOB1{ ULONG ulCompressType;

ULONG ulBitLen;

BYTE cbXCoordinate[SM9\_XCOORDINATE\_BYTES\_LEN];

BYTE cbYCoordinate[SM9\_YCOORDINATE\_BYTES\_LEN]; }SM9KEYBLOB1,\*PSM9KEYBLOB1;

typedef SM9KEYBLOB1 SM9ENCMASTPUBLICKEYBLOB, \*PSM9ENCMASTPUBLICKEYBLOB;

b) 数据项描述见表 B.3。

**表B.3 SM9加密主公钥数据结构**

|  |  |  |  |
| --- | --- | --- | --- |
| 数据项 | 类型 | 意义 | 备注 |
| ulCompressType | ULONG | 压缩类型 | 值为2、3或4。当为4时cbYCoordinate有效，当为 其他值时cbYCoordinate无效 |
| ulBitLen | ULONG | 模数的位长度 | 值为256 |
| cbXCoordinate | BYTE数组 | 曲线上点的X坐标 | 有限域上的整数 |
| cbYCoordinate | BYTE数组 | 曲线上点的Y坐标 | 有限域上的整数 |

**B.1.4** SM9 用户签名密钥数据结构：

a) 类型定义

typedef SM9KEYBLOB1 SM9USERSIGNPRIVATEKEYBLOB, \*PSM9USERSIGNPRIVATEKEYBLOB;

b) 数据项描述见表B.4。

**GM/T 0016—2023**

**表B.4 SM9用户签名密钥数据结构**

|  |  |  |  |
| --- | --- | --- | --- |
| 数据项 | 类型 | 意义 | 备注 |
| ulCompressType | ULONG | 压缩类型 | 值为2、3或4。当为4时cbYCoordinatc有效，当 为其他值时cbYCoordinate无效 |
| ulBitLcn | ULONG | 模数的位长度 | 值为256 |
| cbXCoordinate | BYTE数组 | 曲线上点的X坐标 | 有限域上的整数 |
| cbYCoordinate | BYTE数组 | 曲线上点的Y坐标 | 有限域上的整数 |

**B.1.5** SM9用户加密密钥数据结构：

a) 类型定义

typedef SM9KEYBLOB2 SM9USERENCPRIVATEKEYBLOB, \*PSM9USERENCPRIVATEKEYBLOB;

b) 数据项描述见表B.5。

**表B.5** **SM9用户加密密钥数据结构**

|  |  |  |  |
| --- | --- | --- | --- |
| 数据项 | 类型 | 意义 | 备注 |
| ulCompress Type | ULONG | 压缩类型 | 值为2、3或4。当为4时cbYACoordinate、 cbYBCoordinate有效，当为其他值时  cbYACoordinate、cbYBCoordinate无效 |
| ulBitLen | ULONG | 模数的实际位长度 | 值为256 |
| cbXACoordinate | BYTE数组 | 曲线上点的X坐标高维 | 有限域上的整数 |
| cbXBCoordinate | BYTE数组 | 曲线上点的X坐标低维 | 有限域上的整数 |
| cbYACoordinate | BYTE数组 | 曲线上点的Y坐标高维 | 有限域上的整数 |
| cbYBCoordinate | BYTE数组 | 曲线上点的Y坐标低维 | 有限域上的整数 |

**B.1.6** SM9签名数据结构：

a) 类型定义

typedef struct BYTE BYTE BYTE

Struct\_SM9SIGNATUREBLOB{

h[SM9\_MODULUS\_BYTES\_LEN];

cbSXCoordinate[SM9\_XCOORDINATE\_BYTES\_LEN]; cbSYCoordinate[SM9\_XCOORDINATE\_BYTES\_LEN];

}SM9SIGNATUREBLOB,\*PSM9SIGNATUREBLOB;

b) 数据项描述见表B.6。

**表B.6 SM9 签名数据结构**

|  |  |  |  |
| --- | --- | --- | --- |
| 数据项 | 类型 | 意义 | 备注 |
| h | BYTE数组 | 签名结果的h部分 |  |
| cbSXCoordinate | BYTE数组 | 签名结果的S部分的X坐标 | 有限域上的整数 |
| cbSYCoordinato | BYTE数组 | 签名结果的S部分的Y坐标 | 有限域上的整数 |

**GM/T 0016—2023**

**B.1.7** SM9密文数据结构：

a) 类型定义

typedef struct Struct\_SM9CIPHERBLOB{

ULONG ulAsymAlgId;

BYTE cbXCoordinate[SM9\_XCOORDINATE\_BYTES\_LEN]; BYTE cbYCoordinate[SM9\_YCOORDINATE\_BYTES\_LEN];

BYTE cbC3[32]; ULONG ulC2Len;

BYTE cbC2[1];

}SM9CIPHERBLOB,\*PSM9CIPHERBLOB:

b) 数据项描述见表B.7。

**表** **B.7** **SM9** **密文数据结构**

|  |  |  |  |
| --- | --- | --- | --- |
| 数据项 | 类型 | 意义 | 备注 |
| ulAsymAlgId | ULONG | SM9加密算法标识 |  |
| cbXCoordinate | BYTE数组 | 加密结果的C1部分的X坐标 | 有限域上的整数 |
| cbYCoordinate | BYTE数组 | 加密结果的C1部分的Y坐标 | 有限域上的整数 |
| cbC3 | BYTE数组 | 加密结果的C3部分 | — |
| ulC2Len | ULONG | 加密结果的C2部分的字节长度 |  |
| cbC2 | BYTE数组 | 加密结果的C2部分 |  |

**B.1.8** SM9密钥封装数据结构：

a) 类型定义

typedef struct Struct\_SM9KEYPACKAGEBLOB{

BYTE cbXCoordinate[SM9\_XCOORDINATE\_BYTES\_LEN];

BYTE cbYCoordinate[SM9\_YCOORDINATE\_BYTES\_LEN]; }SM9KEYPACKAGEBLOB,\*PSM9KEYPACKAGEBLOB;

b) 数据项描述见表B.8。

**表** **B.8 SM9 密钥封装数据结构**

|  |  |  |  |
| --- | --- | --- | --- |
| 数据项 | 类型 | 意义 | 备注 |
| cbXCoordinate | BYTE数组 | 封装密文的X坐标 | 有限域上的整数 |
| cbYCoordinate | BYTE数组 | 封装密文的Y坐标 | 有限域上的整数 |

**B.1.9** SM9 用户签名密钥密钥封装保护结构：

a) 类型定义

typedef struct ULONG

ULONG ULONG BYTE

Struct\_SM9SIGNENCAPSULATEDKEYBLOB{

Version;

ulSymmAlgID;

ulBits;

cbEncryptedPriKey[64];

**GM/T 0016—2023**

|  |  |
| --- | --- |
| SM9SIGNMASTPUBLICKEYBLOB | signMastPubKey; |
| SM9ENCMASTPUBLICKEYBLOB | tmpEncMastPubKey; |
| BYTE | ucUserID[256]; |
| ULONG | ulUserlDLen; |

SM9KEYPACKAGEBLOB keyPackageBlob;

}SM9SIGNENCAPSULATEDKEYBLOB,\*PSM9SIGNENCAPSULATEDKEYBLOB;

b) 数据项描述见表B.9。

**表B.9 SM9用户签名密钥保护结构**

|  |  |  |  |
| --- | --- | --- | --- |
| 数据项 | 类型 | 意义 | 备注 |
| Version | ULONG | 版本号，本版本为1 | 一 |
| ulSymmAlgID | ULONG | 对称算法标识 | 应为ECB模式(不填充 补满) |
| ulBits | ULONG | 模数的位长度 | 值为256 |
| cbEncryptedPrivKey | BYTE数组 | 对称算法加密的用户签名 私钥 | 加密原文为SM9  USERSIGNPRIVAT  EKEYBLOB结构中的 cbXCoordinate|  cbYCoordinate |
| signMasterPubKey | SM9SIGNMASTPUBLICKEYBLOB | KGC用于生成用户签名 私钥的签名主公钥 |  |
| tmpEncMastPubKey | SM9ENCMASTPUBLICKEYBLOB | 用户生成的临时加密主 公钥 |  |
| ucUserID | BYTE数组 | 用户标识 |  |
| ulUserIDLen | ULONG | 用户标识长度 |  |
| keyPackageBlob | SM9KEYPACKAGEBLOB | 用临时加密主公钥和用户 标识使用SM9密钥封装  的对称密钥的封装密文 | 用户根据ulSymm  AlgID解封装出特定长 度的对称密钥 |

**B.1.10** SM9用户签名密钥数字信封保护结构：

a) 类型定义

typedef struct Struct\_SM9SIGNENVELOPEDKEYBLOB{

|  |  |
| --- | --- |
| ULONG | Version |
| ULONG | ulSymmAlgID; |
| ULONG | ulBits; |
| BYTE | cbEncryptedPriKey[64]; |
| SM9SIGNMASTPUBLICKEYBLOB | signMastPubKey; |
| SM9ENCMASTPUBLICKEYBLOB | tmpEncMastPubKey; |
| BYTE | ucUserID[256]; |
| ULONG | ulUserIDLen; |

**GM/T 0016—2023**

SM9CIPHERBLOB cipherBlob;

}SM9SIGNENVELOPEDKEYBLOB,\*PSM9SIGNENVELOPEDKEYBLOB;

b) 数据项描述见表B.10。

**表** **B.10** **SM9用户签名密钥数字信封保护结构**

|  |  |  |  |
| --- | --- | --- | --- |
| 数据项 | 类型 | 意义 | 备注 |
| Version | ULONG | 版本号，本版本为1 |  |
| ulSymmAlgID | ULONG | 对称算法标识 | 应为ECB模式(不填充补满) |
| ulBits | ULONG | 模数的位长度 | 值为256 |
| cbEncryptedPrivKey | BYTE数组 | 对称算法加密的用 户签名密钥密文 | 加密原文为  SM9USERSIGNPRIVATEKEY  BLOB结构中的cbXCoordinatell cbYCoordinate |
| signMasterPubKey | SM9SIGNMASTPUBLICKEYBLOB | KGC用于生成用  户签名密钥的签名 主公钥 |  |
| tmpEncMastPubKey | SM9ENCMASTPUBLICKEYBLOB | 用户生成的临时加 密主公钥 |  |
| ucUserID | BYTE数组 | 用户标识 |  |
| ulUserIDLen | ULONG | 用户标识长度 |  |
| cipherBlob | SM9KEYPACKAGEBLOB | 用临时加密主公钥 和用户标识加密的 对称密钥密文 | 加密模式应为SGD\_SM9 3 KDF 或者SGD\_SM9 3 ECB模式 |

**B.1.11** SM9用户加密密钥封装保护结构：

a) 类型定义

typedef struct SKF\_SM9ENCENCAPSULATEDKEYBLOB{

|  |  |
| --- | --- |
| ULONG | Version; |
| ULONG | ulSymmAlgID; |
| ULONG | ulBits; |
| BYTE | cbEncryptedPriKey[SM9\_MODULUS\_BITS\_LEN\*4]; |
| SM9ENCMASTPUBLICKEYBLOB | encMastPubKey; |
| SM9ENCMASTPUBLICKEYBLOB | tmpEncMastPubKey; |
| BYTE | ucUserID[256]; |
| ULONG | ulUserIDLen; |
| SM9KEYPACKAGEBLOB | keyPackageBlob; |

}SM9ENCENCAPSULATEDKEYBLOB,\*PSM9ENCENCAPSULATEDKEYBLOB;

b) 数据项描述见表B.11。

**GM/T 0016—2023**

**表B.11 用户加密密钥对保护结构**

|  |  |  |  |
| --- | --- | --- | --- |
| 数据项 | 类型 | 意义 | 备注 |
| Version | ULONG | 版本号，本版本为1 |  |
| ulSymmAlgID | ULONG | 对称算法标识 | 应为ECB模式(不填充补满) |
| ulBits | ULONG | 模数的位长度 | 值为256 |
| cbEncryptedPrivKey | BYTE数组 | 对称算法加密的用 户加密密钥密文 | 加密原文为  SM9USERENCPRIVATEKEYBL  OB结构中的cbXACoordinatell  obXBCoordinatell cbYACoordinatel cbYBCoordinatc |
| cncMastPubKcy | SM9ENCMASTPUBLICKEYBLOB | KGC用于生成用 户加密密钥的的 加密主公钥 |  |
| tmpEncMastPuBKey | SM9ENCMASTPCECREYBEOB | 户生成的临时 加上公健 |  |
| ucUserID | BYTE数组 | 用 标 识 |  |
| ulUserIDLer | ULONG | 用识长度 |  |
| keyPackage | SM9KEYPARACEEBLOB | 用 密主公 钥和用ID生成  的脚钥封装密交 | 用萨根据ulSymmAlgID解封装出 特定长度的对称密钥 |

**B.1.12** SM9 用户加密密钥数字信封保护结构

a) 类型定义

lypedef struc SKF SM9ENCENVELOPEDKEYBLOB{

|  |  |
| --- | --- |
| ULONG ULONG ULONG BYTE | Version;  ulSymmAlgID; ulBits;  cbEneryptedPriKey[1 28]; |
| SM9ENCMASTPUBLICKEYBLOB SM9ENCMASTPUBLICKEYBLOB BYTE  ULONG  SM9CIPHERBLOB | encMastPubKey,  tmpEncMastPubKey; ucUserID[256];  ulUserIDLen cipherBlob; |

}SM9ENCENVELOPEDKEYBLOB,\*PSM9ENCENVELOPEDKEYBLOB;

b) 数据项描述见表B.12。

**GM/T 0016—2023**

**表** **B.12** **SM9用户加密密钥数字信封保护结构**

|  |  |  |  |
| --- | --- | --- | --- |
| 数据项 | 类型 | 意义 | 备注 |
| Version | ULONG | 版本号，本版本为1 |  |
| ulSymmAlgID | ULONG | 对称算法标识 | 应为ECB模式(不填充补满) |
| ulBits | ULONG | 模数的实际位长度 | 值为256 |
| cbEncryptedPrivKey | BYTE数组 | 对称算法加密的用 户加密密钥密文 | 加密原文为  SM9USERENCPRIVATEKEYB  LOB结构中的cbXACoordinatell  cbXBCoordinatel|cbYACoordinatell cbYBCoordinate |
| encMasterPubKey | SM9ENCMASTPUBLICKEYBLOB | KGC用于生成用户 加密密钥的的签名 主公钥 |  |
| tmpMasterPubKey | SM9ENCMASTPUBLICKEYBLOB | 用户生成的临时加 密主公钥 |  |
| ucUscrID | BYTE数组 | 用户标识 |  |
| ulUserIDLcn | ULONG | 用户标识长度 |  |
| cipherBlob | SM9CIPHERBLOB | 用临时加密主公钥 和用户ID加密的对 称密钥密文 | 加密模式应为SGD\_SM9 3 KDF 或者SGD\_SM9 3 ECB |

**B.2** **SM9** **接口**

**B.2.1** **导入SM9** **签名主公钥**

函数原型 ULONG DEVAPI SKF\_ImportSM9SignMastPubKey(HAPPLICATION hApplica-

tion,PSM9SIGNMASTPUBLICKEYBLOB pSignMastPubKeyBlob)

功能描述 导入SM9 签名主公钥。

参数 phApplication [IN] 应用句柄。

pSignMastPubKeyBlob [IN]SM9签名主公钥数据结构。

返回值 SAR\_OK: 成功。

其他： 错误码。 注：权限要求：已取得管理员权限。

**B.2.2** **导入SM9** **加密主公钥**

函数原型 ULONG DEVAPI SKF\_ImportSM9EncMastPubKey(HAPPLICATION hApplica-

tion,PSM9ENCMASTPUBLICKEYBLOB pEncMastPubKeyBlob)

**GM/T 0016—2023**

功能描述 导入SM9 加密主公钥。

参 数 phApplication [IN] 应用句柄。

pEncMastPubKeyBlob [IN]SM9加密主公钥数据结构。

返回值 SAR\_OK: 成功。

其他： 错误码。 注：权限要求：已取得管理员权限。

**B.2.3** **导** **出SM9** **签名主公钥**

函数原型 ULONG DEVAPI SKF ExporISM9SignMastPubKey(HAPPLICATION hApplica-

tion,PSM9SIGXMASTPUBLICKEYBLOB pSignMasbPubKeyBlob)

功能描述 导出SM9 签名主公钥

参 数 phApplieation [IN] 应用句柄。

pSignMastIubKeyBlob [OUT]SM9 签名主公钥数据结构。

返回值 SAR\_OK: 成功。

其 他 错误码。

**B.2.4** **导出SM9** **加密主公钥**

函数原型 uloNG DEVNPTSRJ Expo EncN ubKey CUIAPertaeATION hAplica-

tion,SM9 ENCMASTPUBLI YBLO ncMastPubKeyBlob )

|  |  |  |
| --- | --- | --- |
| 功能描述 | 导 入SM9 加密主公确 |  |
| 参数 | phApplication  EncMastPibKeyBlob | 要应排句栖  0U1 SMo加密主公钥数据结构。 |
| 返回值 | SAR OK: 成功。  其 他 错误码。 |  |
|  |

**B.2.5** **生** **成SM9** **临时加密主密钥对**

函数原型 ULONG DEVAPISKF \_GenSM9 EncMastKcyPair (HAPPLICATION hApplication,

PSM9ENCMASTPUBLICKEYBLOB pEncMastPubKeyBlob)

功能描述 生 成SM9 临时加密主密钥对并输出加密主公钥。

参 数 phApplication [IN] 应用句柄。

pEncMastPubKeyBlob [OUT] 返 回SM9 临时加密主公钥数据结构。

返回值 SAR\_OK: 成功。

其他： 错误码。 注：权限要求：已取得用户权限。

**GM/T 0016—2023**

**B.2.6** **导入SM9** **用户签名密钥对密钥封装保护结构**

函数原型 ULONG DEVAPI SKF\_ImportSM9SignEncapsulatedKey(HCONTAINER hContainer,

PSM9SIGNENCAPSULATEDKEYBLOB pEncapsulatedKeyBlob)

功能描述 导入使用用户签名密钥对保护结构1保护的SM9 用户签名私钥和签名主公钥。

参数 hContainer [IN] 密钥容器句柄。

pEncapsulatedKeyBlob [IN] 用户签名密钥密钥封装保护结构。

返回值 SAR\_OK: 成功。

其他： 错误码。 注：权限要求：已取得用户权限。

**B.2.7** **导入SM9** **用户签名密钥对数字信封保护结构**

函数原型

功能描述 参数

ULONG DEVAPI SKF\_ImportSM9SignEnvelopedKey(HCONTAINER hContainer,

PSM9SIGNENVELOPEDKEYBLOB pEnvelopedKeyBlob)

导入使用用户签名密钥对保护结构2保护的SM9 用户签名私钥和签名主公钥。

hContainer [IN] 密钥容器句柄。

pEnvelopedKeyBlob [IN] 用户签名密钥数字信封保护结构。

返回值 SAR\_OK: 成功。

其他： 错误码。 注：权限要求：已取得用户权限。

**B.2.8** **导入SM9** **用户加密密钥对密钥封装保护结构**

函数原型 ULONG DEVAPI SKF\_ImportSM9EncEncapsulatedKey(HCONTAINER hContainer, PSM9ENCENCAPSULATEDKEYBLOB pEncapsulatedKeyBlob)

功能描述 导入使用用户加密密钥对保护结构1保护的SM9 用户加密私钥和加密主公钥。

参数 hContainer [IN] 密钥容器句柄。

pEncapsulatedKeyBlob [IN] 用户加密密钥密钥封装保护结构。

返回值 SAR\_OK: 成功。

其他： 错误码。 注：权限要求：已取得用户权限。

**B.2.9** **导入SM9** **用户加密密钥对数字信封保护结构**

函数原型 ULONG DEVAPI SKF\_ImportSM9EncEnvelopedKey(HCONTAINER hContainer, PSM9ENCENVELOPEDKEYBLOB pEnvelopedKeyBlob)

功能描述 导入使用用户加密密钥对保护结构2保护的SM9 用户加密私钥和加密主公钥。

参数 hContainer

pEnvelopedKeyBlob

[IN] 密钥容器句柄。

[IN] 用户加密密钥数字信封保护结构。

返回值 SAR\_OK: 成功。

其他： 错误码。 注：权限要求：已取得用户权限。

**GM/T 0016—2023**

**B.2.10 SM9生成密钥协商参数并输出**

函数原型 ULONG DEVAPI SKF\_SM9GenerateAgreementData(HCONTAINER hContainer,

ULONG ulAlgID,BYTE \*pucResponsorID,ULONG ulResponsorIDLen,

PSM9KEYBLOB1 pSponsorTempSM9PubKeyBlob,HANDLE \*phAgreementHandlc)

功能描述 使用SM9 密钥协商算法，为计算会话密钥而产生协商参数，返回临时SM9 密钥对的公钥

及协商句柄。

参数 hContainer [IN] 容器句柄。

ulAlgID [IN] 会话密钥算法标识。

pucResponsorID [IN] 响应方的ID。

ulResponsorIDLen [IN] 响应方ID的长度。

pSponsorTempSM [OUT] 发起方临时公钥。

9PubKeyBlob

phAgreemen- [OUT] 返回的密钥协商句柄。

tHandle

返回值 SAR\_OK: 成功。

其他： 错误码。

注：权限要求：已取得用户权限。为协商会话密钥，协商的发起方首先调用本函数。

**B.2.11 SM9产生协商数据并计算会话密钥**

函数原型： ULONG DEVAPI SKF\_SM9GenerateAgreementDataAndKey(

HANDLE hContainer,ULONG ulAlgID,PSM9KEYBLOB1 pSponsorTempSM9Pub- KeyBlob,BYTE \*pucSponsorID,ULONG ulSponsorIDLen,PSM9KEYBLOB1 pRe- sponsorTempSM9PubKeyBlob,HANDLE\*phKeyHandle)

功能描述： 使 用SM9 密钥协商算法，产生协商参数并计算会话密钥，输出临时公钥，并返回产生的 密钥句柄。

参数：

hContainer ulAlgID

pSponsorTempSM9PubKeyBlob pucSponsorID

ulSponsorIDLen

[IN] 容器句柄。

[IN] 会话密钥算法标识。 [IN] 发起方的临时公钥。 [IN] 发起方的ID。

[IN] 发起方ID的长度。

pResponsorTempSM9PubKeyBlob phKeyHandle

[OUT] 响应方的临时公钥。

[OUT] 返回的会话密钥句柄。

返回值 SAR\_OK: 成功。

其他： 错误码。

注：权限要求：已取得用户权限。本函数由响应方调用。

**GM/T 0016—2023**

**B.2.12** **SM9计算会话密钥并确认**

函数原型： ULONG DEVAPI SKF\_SM9GenerateKey(HANDLE hAgreementHandle,

PSM9KEYBLOB1 pResponsorTempSM9PubKeyBlob,HANDLE\*phKeyHandle)

功能描述： 使用SM9 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： hAgreementHandle [IN] 密钥协商句柄。

pResponsorTempSM9PubKeyBlob [IN] 响应方临时公钥。

phKeyHandle [OUT] 返回的密钥句柄。

返回值 SAR\_OK: 成功。

其他： 错误码。

注：权限要求：已取得用户权限。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。

**B.2.13 SM9签名**

函数原型 ULONG DEVAPI SKF\_SM9SignData(HCONTAINER hContainer,BYTE \*puc-

Data,ULONG ulDataLen,PSM9SIGNATUREBLOB pSignature)

功能描述 SM9 数字签名。采用SM9 算法和用户私钥，对指定数据pbData进行数字签名。签名

后的结果存放到pSignature中 。

参数 hContainer [IN] 密钥容器句柄。

pucData [IN] 待签名的数据。

ulDataLen [IN] 待签名的数据长度。

pSignature [OUT] 签名值。

返回值 SAR\_OK: 成功。

其他： 错误码。

注：权限要求：已取得用户权限。待签名数据pucData为待签名消息原文的SM3杂凑值，ulDataLen为SM3杂凑函 数的输出杂凑长度，值为32。

**B.2.14 SM9验签**

函数原型 ULONG DEVAPI SKF\_SM9Verify(HAPPLICATION hApplication,

PSM9SIGNMASTPUBLICKEYBLOB pSignMastPubKeyBlob,BYTE \*pucUserID, ULONG ulUserIDLen,BYTE \*pucData,ULONG ulDataLen,PSM9

SIGNATUREBLOB pSignature)

功能描述 用SM9 签名者ID 对数据进行验签。

参数 hApplication [IN] 应用句柄。

pSignMastPubKeyBlob [IN] 签名主公钥，若为NULL, 则使用内部主公钥。

pUserID [IN] 签名者ID。

ulUserIDLen [IN] 签名者ID的长度。

pbData [IN] 待验证签名的数据。

ulDataLen [IN] 待验证签名的数据长度。

**GM/T 0016—2023**

pSignature [IN] 待验证签名值。

返回值 SAR\_OK: 成功。

其他： 错误码。

注：待验证签名的数据pucData为待验证签名的消息原文的SM3杂凑值，ulDataLen为SM3杂凑函数的输出杂 凑长度，值为32。

**B.2.15 SM9密钥封装**

函数原型 ULONG DEVAPI SKF\_SM9KeyEncapsulate(HAPPLICATION hApplication,

ULONG ulAlgID,PSM9ENCMASTPUBLICKEYBLOB pEncPubKeyBlob,

BYTE \*pucUserID,ULONG ulUserIDLen,

PSM9KEYPACKAGEBLOB pKeyPackageBlob,HANDLE \*phSessionKey)

功能描述 生成会话密钥并使用接收者标识进行密钥封装。

参数 phApplication [IN] 应用句柄。

ulAlgID [IN] 封装密钥算法标识。

pEncPubKeyBlob [IN] 加密主公钥，若输入NULL, 则使用内部主公钥。

pUserID [IN] 接收者用户标识。

ulUserIDLen [IN] 接收者用户标识的长度。

pKeyPackageBlob [OUT] 封装后的密文。

phSessionKey [OUT] 会话密钥句柄。

返回值 SAR\_OK: 成功。

其他： 错误码。

**B.2.16 SM9密钥解封装**

函数原型 ULONG DEVAPI SKF\_SM9KeyDecapsulate(HCONTAINER hContainer,

ULONG ulAlgID,PSM9KEYPACKAGEBLOB pKeyPackageBlob,

HANDLE \*phSessionKey)

功能描述 接收者对封装密文解封装并返回会话密钥句柄。

参数 hContaincr [IN] 密钥容器句柄。

ulAlgID [IN] 会话密钥算法标识。

pKeyPackageBlob [IN] 封装密文。

phSessionKey [OUT] 会话密钥句柄。

返回值 SAR\_OK: 成功。

其他： 错误码。 注：权限要求：已取得用户权限。

**GM/T 0016—2023**

**B.2.17** **SM9加密**

函数原型 ULONG DEVAPI SKF\_SM9Encrypt(HAPPLICATION hApplication,

PSM9ENCMASTPUBLICKEYBLOB pEncPubKeyBlob,BYTE \*pucUserID,

ULONG ulUserIDLen,ULONG ulAsymAlgId, BYTE \*pucIV,ULONG ulIVLen,

BYTE \*pucPlainText,ULONG ulPlainTextLen,

PSM9CIPHERBLOB pCipherText,ULONG \*pulCipherTextLen)

功能描述 使用外部传入的用户标识对输入数据做加密运算并输出结果。

参数 hApplication [IN] 应用句柄。

pEncPubKeyBlob [IN]SM9 加密主公钥，若为NULL, 则使用内部加密主公

钥。

pucUscrID [IN] 用户ID。

ulUserIDLen [IN] 用户ID的长度。

ulAsymAlgId [IN]SM9 加密算法标识。

pucIV [IN]SM9 加密算法内部IV 向量，当为NULL 时，由设备内

部生成。当不为NULL 并且ulAsymAlgId为SGD\_SM9\_ 3\_CBC 、SGD\_SM9 3 CFB 、SGD\_SM9 3 OFB时有效。

ulIVLen [IN]SM9 加密算法内部IV向量长度。当pucIV不为

NULL 并且ulAsymAlgId为SGD\_SM9 3 CBC、 SGD\_SM9 3 CFB、SGD\_SM9 3 OFB 时有效。

pbPlainText [IN] 待加密的明文数据。

ulPlainTextLen [IN] 待加密明文数据的长度。

pCipherText [OUT]SM9 密文数据结构，当为NULL 时 ，pulCipherTex-

tLen输出pCipherText的字节长度。

pulCipherTextLen [IN,OUT] 输入为pCipherText 缓冲区长度，输出为SM9

密文数据结构实际长度。

返回值 SAR\_OK: 成功。

其他： 错误码。

**B.2.18 SM9解密**

函数原型 int SKF\_SM9Decrypt(

HANDLE hContainer,

PSM9CIPHERBLOB pCipherText BYTE \*pucPlainText,

ULONG \*pulPlainTextLen)

描述 使用用户私钥对输入密文进行解密运算并输出结果。

参数 hContainer [IN] 容器句柄

pCipherText [IN] 密文

pucPlainText [OUT] 指向解密结果的缓冲区，如果此参数为NULL 时，由

pulPlainTextLen返回解密结果的长度。

**GM/T 0016—2023**

返回值

pulPlainTextLen

SAR\_OK: 其他：

[IN,OUT] 输入时表示pucPlainText 缓冲区的长度，输出时 表示解密结果的长度。

成功。

失败，返回错误代码。



**GM/T 0016—2023**

**附** **录** **C** **(规范性)**

**VPN相关接口**

**C.1 计算IKE工作密钥**

int SKF\_GenerateKeywithIKE( HANDLE hContainer,

BYTE \*pucSponsorNonce,

|  |  |  |
| --- | --- | --- |
|  | UINT BYTE UINT BYTE  UINT | uiSponsorNonceLength, \*pucResponseNonce,  uiResponseNonceLength, \*pucSponsorCookie,  uiSponsorCookieLength, |
| 原型： | BYTE | \*pucResponseCookie, |

UINT uiResponseCookieLength,

UINT uiPrfAlgID

UINT uiKeyBitsD,

HANDLE \*phKeyHandleD, UINT uiKeyBitsA,

HANDLE \*phKeyHandleA, UINT uiKeyBitsE,

HANDLE \*phKeyHandleE);

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

密钥句柄。

参数： hContainer[in]

pucSponsorNonce[in]

uiSponsorNonceLength[in]

pucResponseNonce[in

uiResponseNonceLength[in] pucSponsorCookie[in]

uiSponsorCookieLength[in] pucResponseCookie[in]

uiResponseCookieLength[in] uiPrfAlgID[in]

uiKeyBitsD[in]

phKeyHandleD[out] uiKeyBitsA[in]

phKeyHandleA[out] uiKeyBitsE[in]

phKeyHandleE[out]

容器句柄

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

SKEYID\_d 密钥长度

返回的SKEYID\_d 密钥句柄 SKEYID\_a 密钥长度

返回的SKEYID\_a 密钥句柄 SKEYID\_e 密钥长度

返回的SKEYID\_e 密钥句柄

**GM/T 0016—2023**

返回值： 0

非 0

成功

失败，返回错误代码

注：IKE 一阶段(主模式)消息3和消息4交互完成后，参与通信的双方各自调用本函数，计算后续工作密钥 SKEYID\_d、SKEYID\_a、SKEYID\_e。IKE一阶段(主模式)计算IKE工作密钥的过程符合GM/T 0022—2023 中<5.1.2.1>的规定，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 e。

**C.2 计** **算IKE 工作密钥并用外部ECC 公钥加密输出**

原型：

描述：

int SKF \_GeneratcKeywithEPK \_ IKE(

HANDLE hContainer,

BYTE \*pucSponsosNonce,

UINT uiSponsorNonceLength BYTE \*pucResponseNonce,

UInT uikesponscNoncol ength

BYTE \*pucSponsoCookic

uikT uispo foCaolbecLeng

BYTE \*pucRespQnseCookic

UmNT uiResponeCookicLeng

UINT uiPrfAIpD

UINT uiFccAlgID

ECCPUBLICKEYBLOB\*pucPublicKey, UINT uiKeyBitsD

ECCCipher tpucKeylD

HANDLE \*phKeyHandleD , UINT uiKeyBitsA

ECCCipher \*pucKeyA,

HANDLE \*phKeyHandleA, UINT uiKeyBitsE,

ECCCipher \*pucKeyE,

HANDLE phKeyHandleE);

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

**GM/T 0016—2023**

|  |  |  |
| --- | --- | --- |
| 参数： | hContainer [in]  pucSponsorNonce [in]  uiSponsorNonceLength [in] pucResponseNonce [in  uiResponseNonceLength [in] pucSponsorCookic [in  uiSponsorCookieLength [in] pucResponseCookie [in]  uiResponseCookieLength [in] uiPrfAlgID [in]  uiEccAlgID [in]  pucPublicKey [in] uiKeyBitsD [in] | 容器句柄  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  外部ECC 公钥的算法标识 输入的外部ECC 公钥结构 返回的SKEYID\_d 密钥长度  缓冲区指针，用于存放返回的SKEYID\_d 密 |
|  | pucKeyD [out]  phKeyHandleD [out] uiKeyBitsA [in] | 钥密文  返回的SKEYID\_d 密钥句柄 SKEYID\_a 密钥长度  缓冲区指针，用于存放返回的 SKEYID\_A 密 |
|  | pucKeyA [out]  phKeyHandleA [out] uiKeyBitsE [in] | 钥密文  返回的SKEYID\_a 密钥句柄 SKEYID\_e 密钥长度  缓冲区指针，用于存放返回的SKEYID\_E 密 |
|  | pucKeyE [out]  phKeyHandleE [out] | 钥密文  返回的SKEYID\_e 密钥句柄 |
| 返回值： | 0  非0 | 成功  失败，返回错误代码 |

**注：**IKE 一阶段(主模式)消息3和消息4交互完成后，参与通信的双方各自调用本函数，计算后续工作密钥 SKEYID\_d、SKEYID\_a、SKEYID\_c。IKE一阶段(主模式)计算IKE 工作密钥的过程符合GM/T 0022— 2023中<5.1.2.1>的规定，输入的密钥参数按顺序为Ni\_b、Nr\_b、CKY-I、CKY-R,返回的密钥密文及密钥句柄按 顺序为SKEYID\_d、SKEYID\_a、SKEYID\_e。

**GM/T 0016—2023**

**C.3 计算IPSEC会话密钥**

int SKF\_GenerateKeywithIPSEC(

HANDLE hContainer,

BYTE \*pucProtocolID,

UINT uiProtocolIDLength, BYTE \*pucSpi

UINT uiSpiLength

BYTE \*pucSponsorNonce,

UINT uiSponsorNonceLength,

原型：

BYTE \*pucResponseNonce,

UINT uiResponseNonceLength, HANDLE phKeyHandle,

UINT uiPrfAlgID

UINT uiKeyBitsEnc,

HANDLE \*phKeyHandleEnc, UINT uiKeyBitsMac,

HANDLE phKeyHandleMac);

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

会话密钥句柄。

参数： hContainer [in] 容器句柄

pucProtocolID [in] 协议ID

uiProtocolIDLength [in] 协议ID 长度

pucSpi [in] 安全参数索引SPI

uiSpiLength [in] 安全参数索引SPI 长度

pucSponsorNonce [in] 发起方nonce 载荷主体

uiSponsorNonceLength [in] 发起方nonce 载荷主体长度

pucResponseNonce [in] 响应方nonce 载荷主体

uiResponseNonceLength [in] 响应方nonce 载荷主体长度

hKeyHandle[in] 输入的SKEYID\_d 密钥句柄

uiPrfAlgID [in] PRF 算法标识

uiKeyBitsEnc [in] 返回的加密密钥长度

phKeyHandleEnc[out] 返回的加密密钥句柄

uiKeyBitsMac [in] 返回的杂凑密钥长度

phKeyHandleMac[out] 返回的杂凑密钥句柄

返回值： 0 成功

非0 失败，返回错误代码

**GM/T 0016—2023**

注：IKE二阶段(快速模式)消息交互完成后，参与通信的双方各白调用本函数，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。IKE二阶段(快速模式)计算IPSEC会话密钥的过程符合GM/ T 0022—2023中<5.1.2.2>的规定，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。本函数在C.1计算IKE工作密钥函数调用之后调用，并将该函数返回的密钥句柄之 一(SKEYID\_d)作为输入。

**C.4 计算IPSEC会话密钥并用外部ECC公钥加密输出**

int SKF\_GenerateKeywithEPK\_IPSEC(

HANDLE hContainer,

BYTE \*pucProtocolID,

UINT uiProtocolIDLength,

BYTE \*pucSpi,

UINT uiSpiLength,

BYTE \*pucSponsorNonce,

UINT uiSponsorNonceLength,

BYTE \*pucResponseNonce,

原型： UINT uiResponseNonceLength,

HANDLE hKeyHandle, UINT uiPrfAlgID,

UINT uiEccAlgID,

ECCPUBLICKEYBLOB\*pucPublicKey,

UINT uiKeyBitsEnc,

ECCCipher \*pucKeyEnc

HANDLE\*phKeyHandleEnc, UINT uiKeyBitsMac,

ECCCipher \*pucKeyMac

HANDLE \*phKeyHandleMac);

描述： 使用IKE 二阶段(快速模式)交换得到的密钥计算参数计算IPSEC 会话密钥，并用外部

ECC 公钥加密输出，同时返回会话密钥句柄。

参数： hContainer[in]

pucProtocolID[in]

uiProtocolIDLength[in] pucSpi[in]

uiSpiLength [in]

pucSponsorNonce[in]

uiSponsorNonceLength[in] pucResponseNonce[in]

uiResponseNonceLength[in] hKeyHandle[in]

uiPrfAlgID [in] uiEccAlgID[in]

容器句柄 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

**GM/T 0016—2023**

pucPublicKey[in]

uiKeyBitsEnc [in]

pucKeyEnc [out]

phKeyHandlcEnc[out] uiKeyBitsMac [in]

pucKeyMac [out]

phKeyHandleMac[out]

返回值： 0

非0

输入的外部ECC 公钥结构 返回的加密密钥长度

缓冲区指针，用于存放返回的加密密钥密文 返回的加密密钥句柄

返回的杂凑密钥长度

缓冲区指针，用于存放返回的杂凑密钥密文 返回的杂凑密钥句柄

成功

失败，返回错误代码

注：IKE 二阶段(快速模式)消息交互完成后，参与通信的双方各白调用本函数，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。IKE二阶段(快速模式)计算IPSEC会话密钥的过程符合 GM/T 0022—2023中<5.1.2.2>的规定，输入的密钥参数按顺序为protocol、SPI、Ni\_b、Nr\_b,返回的密钥密文和 密钥句柄按顺序为加密密钥和杂凑密钥(用于完整性校验)。本函数在C.1计算IKE工作密钥函数调用之后 调用，并将该函数返回的密钥句柄之一(SKEYID\_d)作为输入。

**C.5 计算SSL 工作密钥**

int SKF\_GeneratcKeywithSSL( HANDLE hContainer,

HANDLE hKeyPreMaster,

BYTE \*pucClientRandom,

UINT uiClientRandomLength, BYTE \*pucServerRandom,

UINT uiServerRandomLength, UINT uiPrIAlgID,

UINT uiKeyBitsClientMac,

HANDLE\*phKeyHandleClientMac,

原型： UINT uiKeyBitsServerMac,

HANDLE\*phKeyHandleServerMac, UINT uiKeyBitsClientEnc,

HANDLE \*phKeyHandleClientEnc, UINT uiKeyBitsServerEnc

HANDLE \*phKeyHandleServerEnc

BYTE \*pucClientIV,

ULONG ulClientIVLength, BYTE \*pucServerIV,

ULONG ulServerIVLength );

**GM/T 0016—2023**

描述： 使 用SSL 握手协议得到的预主密钥计算SSL 工作密钥，同时返回工作密钥句柄。

参数： hContainer[in] 容器句柄

hKeyPreMaster[in] 预主密钥pre\_master\_secret 密钥句柄

pucClientRandom[in] 客户端随机数

uiClientRandomLength[in] 客户端随机数长度

pucServerRandom[in] 服务端随机数

uiServerRandomLength[in] 服务端随机数长度

uiPrfAlgID [in] PRF 算法标识

uiKeyBitsClientMac[in] 客户端杂凑密钥长度

phKeyHandleClientMac[out] 客户端杂凑密钥句柄

uiKeyBitsServerMac[in] 服务端杂凑密钥长度

phKeyHandleServerMac[out] 服务端杂凑密钥句柄

uiKeyBitsClientEnc[in] 客户端加密密钥长度

phKeyHandleClientEnc[out] 客户端加密密钥句柄

uiKeyBitsServerEnc[in] 服务端加密密钥长度

phKeyHandleServerEnc[out] 服务端加密密钥句柄

pucClientIV[out] 缓冲区指针，用于存放返回的客户端IV

uiClientIV Length[in] 客户端初始向量长度

pucServeIV[out] 缓冲区指针，用于存放返回的服务端IV

uiServerIV Length[in] 服务端初始向量长度

返回值： 0 成功

非0 失败，返回错误代码

注：SSL握手协议消息交互完成后，参与通信的双方各自调用本函数，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)、cli- ent\_write\_key(客户端加密密钥)、server\_write\_key(服务端加密密钥)。SSL 计算工作密钥的过程符合GM/ T 0024—2023中6.5.2的规定。

预主密钥句柄是调用RSA生成并导出会话密钥接口、ECC生成并导出会话密钥接口或者密钥协商接口得到 的句柄。在调用接口时传入的算法标识为0x00000000。

**GM/T 0016—2023**

**C.6** **计** **算SSL** **工作密钥并用外部ECC** **公钥加密输出**

int SKF\_GenerateKeywithEPK\_SSL( HANDLE hContainer,

HANDLE hKeyPreMaster,

BYTE \*pucClientRandom,

UINT uiClientRandomLength, BYTE \*pucServerRandom,

UINT uiServerRandomLength, UINT uiPrfAlgID,

UINT uiEccAlgID ,

ECCPUBLICKEYBLOB \*pucPublicKey,

UINT uiKeyBisClientMac, ECCCipher pucKeyClientMac, HANDLEphKeyHandleClientMac, UINT uiKcyBitsScrvcrMac,

原型：

ECCcipher “pucKeyServerMae, HAXDLE\* phKeyHadeSeryerMac UINT uiKeylisChentne,

ecCCipher puckeyCrentEn HANDLE\* phkeyHlandleClierttnc UINT uiKeyBitsServerEnc,

ECCCipher \*pucKeySceverEnc ,

HANDLE\*phl eyHandeSovorEnc, BYTE\* pucClientN

ULONG uiClientVLength,

BYTE pucServerlV,

ULONG uiServerIV Length);

使 用SSL 握于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

描述： 参数：

输出。

hContainer[in ]

hKcyPreMaster[in]

pucClientRandom[in]

uiClientRandomLength[in] pucServerRandom[in]

uiServcrRandomLength[in] uiPrfAlgID [in]

uiEccAlgID [in] pucPublicKey[in]

uiKcyBitsClientMac[in]

pucKeyClientMac[out]

phKeyHandleClientMac[out]

容器句柄

预主密钥句柄 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

外部ECC 公钥的算法标识 输入的外部ECC 公钥结构 客户端杂凑密钥长度

客户端杂凑密钥

客户端杂凑密钥句柄

**GM/T 0016—2023**

uiKeyBitsServerMac[in

pucKeyServerMac[out]

phKeyHandleServerMac[out]

uiKeyBitsClientEnc[in]

pucKeyClientEnc[out]

phKeyHandleClientEnc[out] uiKeyBitsServerEnc[in]

pucKeyServerEnc[out]

phKeyHandleServerEnc[out] pucClientIV[out]

uiClientIV Length[in] pucServerlV [out]

uiServerlVLength[in]

返回值： 0

非0

服务端杂凑密钥长度 服务端杂凑密钥

服务端杂凑密钥句柄 客户端加密密钥长度 客户端加密密钥

客户端加密密钥句柄 服务端加密密钥长度 服务端加密密钥

服务端加密密钥句柄

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

注：SSL握手协议消息交互完成后，参与通信的双方各自调用本函数，计算SSL记录层协议的工作密钥并返回密 钥密文和密钥句柄：client\_write\_MAC\_secret(客户端杂凑密钥)、server\_writc\_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)。SSL计算工作密钥的过程符合 GM/T 0024—2023中6.5.2的规定。

预主密钥句柄是调用RSA生成并导出会话密钥接口、ECC生成并导出会话密钥接口或者密钥协商接口得到 的句柄。在调用接口时传入的算法标识为0x00000000。

**GM/T 0016—2023**

**附** **录** **D**  **(资料性)** **SM9 编程范例**

编程范例供应用开发商参考，应用开发商可根据实际情况进行相应的修改后使用。

DEVHANDLE

HAPPLICATION HCONTAINER

SM9SIGNMASTPUBLICKEYBLOB

SM9ENCMASTPUBLICKEYBLOB

SM9ENCMASTPUBIICKEYBLOB

SM9SIGNENCAPSULATEDKEYBLOB SM9ENCENCAPSULATEDKEYBLOB SM9SIGNENVELOPEDKEYBLOB

SM9ENCENVELOPEDKEYBLOB

SM9KEYBLOB1 SM9KEYBLOB1

SM9KEYPACKAGEEEOB

HANDLE HANDLE BYTE

BYTE

BYTE

BYTE

BYTE

ULONG BYTE

SM9SIGNATUREBTQB

BYTE\* ULONG BYTE

PSM9CIPHERBLOB ULONG

ULONG

//打开设备，创建应用和容器

hDev

hApplication; hContainer;

signMastPubKeyBlob;

encMastPubKeyBlob;

tempEneMastPubKeyBlob ; signEncapsulatedlKeyBlob encEncapsulatedKeyBlob; signEnvelopedKeyBlob;

cncEnvelopedKeyBlob;

SponSorTempSM 9PabKeyBlob

Tesponsor FempSM9PubKeyBlob;

koyPackagcBlob hSangementhal hKs andle;

re orID [MAX USER\_ID \_LN];

rdsponsorIDLe

sponsorIDIMX CSERDLEN];

sponsorHDLon

uscrHDIMAX USER IDLEM;

userIDLen

digest[32];

signature; pPlainText;

plainTexiLen;

IV[16];

pCipher;

cipherLen; ret;

/\*1.导入 KGC 的主公钥\*/

//从KGC 获取到签名主公钥signMastPubKeyBlob 和加密主公钥 encMastPubKeyBlob ret=SKF\_ImportSM9SignMastPubKey(hApplication,&signMastPubKeyBlob);

if(ret !=SAR\_OK){ //错误处理

}

**GM/T 0016—2023**

ret =SKF \_ImportSM9EncMastPubKey(hApplication,&encMastPubKeyBlob); if(ret !=SAR\_OK){

//错误处理

}

/\*2.申请用户密钥\*/

//生成SM9 临时加密主密钥对

ret=SKF\_GenSM9EncMastKeyPair(hAplication,&tempEncMastPubKeyBlob);

if(ret !=SAR\_OK){ //错误处理

}

//将用户标识和tempEncMastPubKeyBlob发送给KGC申请用户密钥 //KGC 生成密钥封装保护结构(或数字信封保护结构)并下发给用户 //导入用户签名密钥和加密密钥密钥封装保护结构

ret=SKF\_ImporlSM9SignEncapsulatedKey(hContainer,&signEncapsulatedKeyBlob); if(ret !=SAR\_OK){

//错误处理 }

ret=SKF\_ImporlSM9EncEncapsulatedKey(hContainer,&encEncapsulatedKeyBlob); if(ret !=SAR\_OK){

//错误处理 }

//若KGC 下发的是数字信封保护结构，则导入用户签名密钥和加密密钥密钥封装保护结构

ret =SKF\_ImportSM9SignEnvelopedKey(hContainer,&-signEnvelopedKeyBlob); if(ret!=SAR\_OK){

//错误处理 }

ret =SKF\_ImportSM9EncEnvelopedKey(hContainer,&encEnvelopedKeyBlob); if(ret!=SAR\_OK){

//错误处理

}

/\*3.密钥协商\*/

//生成密钥协商参数并输出，本过程由发起方调用

ret=SKF\_SM9GenerateAgreementData(hContainer,SGD\_SM4\_CBC,responsorID,respon- sorIDLen,&sponsorTempSM9PubKeyBlob,hAgreementHandle);

if(ret !=SAR\_OK){

//错误处理 }

//产生协商数据并计算会话密钥，本过程由响应方调用 ret=SKF\_SM9GenerateAgreementDataAndKey(

hContainer,SGD\_SM4\_CBC,&sponsorTempSM9PubKeyBlob,sponsorID,sponsorIDLen, &responsorTempSM9PubKeyBlob,

&hKeyHandle);

**GM/T 0016—2023**

if(ret !=SAR\_OK){ //错误处理

}

//计算会话密钥并确认，本过程由发起方调用

ret =SKF\_SM9GenerateKey(hAgreementHandle,&.responsorTempSM9PubKeyBlob,&hKey- Handle);

if(ret!=SAR\_OK){

//错误处理 }

/\*4.签名\*/

//计算待签数据的SM3 杂凑值并存入digest 中

ret =SKF\_SM9SignData(hContainer,digest,32,&signature);

if(ret !=SAR\_OK){ //错误处理

}

//若hApplication中已导入KGC 签名主公钥，则签名主公钥可传入NULL

ret =SKF\_SM9Verify(hApplication,&signMastPubKeyBlob,userID,userIDLen,digest,32, &signaturc);

if(ret!=SAR\_OK){

//签名验证失败或错误 }else{

//签名验证成功

}

/\*5.SM9 密钥封装\*/

//设置userID为密钥解封装方的标识

//若hApplication中已导入KGC 加密主公钥，则加密主公钥可传入NULL

ret =SKF\_SM9KeyEncapsulate(hApplication,SGD\_SM4\_CBC,&encMastPubKeyBlob,use- rID,userIDLen,&keyPackageBlob,&hKeyHandle);

if(ret !=SAR\_OK){ //错误处理

}

//将keyPackageBlob 发送给密钥解封装方 //密钥解封装，本过程由密钥解封装方调用

ret =SKF\_KeyDecapsulate(hContainer,SGD\_SM4\_CBC,&keyPackageBlob,&hKeyHandle); if(ret !=SAR\_OK){

//错误处理 }

/\*6.加密\*/

//设置pPlainText 指向待加密数据，plainTextLen 为待加密数据长度

//若hApplication中已导入KGC 加密主公钥，则加密主公钥可传入NULL //IV 可传入NULL, 则设备内部产生IV

ret =SKF\_SM9Encrypt (hApplication,&encMastPubKeyBlob,userID,userIDLen,

**GM/T 0016—2023**

SGD\_SM9 3 CBC,IV,16,pPlainText,plainTextLen,NULL,&cipherLen); if(ret !=SAR\_OK){

//错误处理 }

pCipher =(PSM9CIPHERBLOB)malloc(cipherLen); if(pCipher ==NULL){

//错误处理 }

ret =SKF\_SM9Encrypt (hApplication,&encMastPubKeyBlob,userID,userIDLen, SGD\_SM9 3 CBC,IV,16,pPlainText,plainTextLen,pCipher,&cipherLen);

if(ret !=SAR\_OK){ //错误处理

}

//解密

ret =SKF\_SM9Decrypt(hContainer,pCipher,NULL,&plainTextLen);

if(ret !=SAR\_OK){ //错误处理

}

pPlainText=(BYTE\*)malloc(plainTextLen); if(pPlainText ==NULL){

//错误处理 }

ret=SKF\_SM9Decrypt(hContainer,pCipher,pPlainText,&plainTextLen); if(ret!=SAROK){

//错误处理 }

free(pCipher);

free(pPlainText);

//关闭容器、应用和设备

**GM/T 0016—2023**

**参** **考** **文** **献**

[1]GM/T 0009—2023 SM2 密码算法使用规范

[2]GM/T 0022—2023 IPSec VPN技术规范

[3]GM/T 0024—2023 SSL VPN技术规范

[4]GM/T 0080—2020 SM9密码算法使用规范