**多卡融合项目国际国密算法密钥接口说明**

文档修订记录

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 版本编号或者更改记录编号 | \*变化  状态 | 简要说明（变更内容和变更范围） | 日期 | 变更人 | 批准日期 | 批准人 |
| V1.0 | A | 新建 | 2017-03-20 | 牛林东 |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

目录

[1．简介 4](#_Toc479190912)

[1.1. 密级 4](#_Toc479190913)

[1.2. 编写目的 4](#_Toc479190914)

[1.3. 读者对象 4](#_Toc479190915)

[1.4. 说明 4](#_Toc479190916)

[1.5. 加密流程图 5](#_Toc479190917)

[2. 接口说明 6](#_Toc479190918)

[1.1．申请密钥(300001) 6](#_Toc479190919)

[1.2 . 对称算法加解密数据(300002) 6](#_Toc479190920)

[1.3．将ZEK/ZAK从ZMK转为LMK加密 7](#_Toc479190921)

[1.4．将ZEK/ZAK从LMK转为ZMK加密 7](#_Toc479190922)

[1.5．将ZPK由ZMK转为LMK加密(300005) 8](#_Toc479190923)

[1.6．将ZPK由LMK转为ZMK加密(300006) 8](#_Toc479190924)

[1.5 . 用EDK密钥加解密数据 8](#_Toc479190925)

[1.6 . 产生一对RSA密钥(300011) 9](#_Toc479190926)

[1.7 . 公钥加密(300012) 9](#_Toc479190927)

[1.8 . 私钥解密(300013) 10](#_Toc479190928)

[1.9 . 导入私钥(300014)(备份用) 11](#_Toc479190929)

[2.0 . 将由公钥加密的PIN转换成ZPK加密(300015) 11](#_Toc479190930)

[3. 密钥类型表 12](#_Toc479190931)

[4. 密钥方案表 13](#_Toc479190932)

[5. 错误码定义 13](#_Toc479190933)

1．简介

* 1. 密级

项目内部公开。

* 1. 编写目的

《虚拟账务系统国际算法密钥接口》是各渠道交换通信报文时的参考，供项目各子系统加解密组报文使用，可根据实际使用情况修改或增加。

* 1. 读者对象

项目经理、业务分析师、系统设计师、系统开发工程师、系统测试工程师

* 1. 说明

本接口基于国际通用的对称或非对称加密算法，采用银联三级密钥体系构成一机一密，符合目前主流POS厂商或其他加密渠道的协议，接口格式为若干个键/值(key, value)对的集合，可根据实际情况采用JSON，XML等其他报文格式作为传输方式。

* 1. 加密流程图



2. 接口说明

2.1．申请密钥(300001)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入参数 | TAG名 | 长度 | 类型 | 备注 |
| 模式 | mode | 2 | char | 0－产生密钥。  1－产生密钥并在ZMK下加密。 |
| 密钥类型 | keyType | 3 | char | 密钥类型 |
| 密钥方案 | keyProg | 1 | char | 密钥长度/LMK下加密密钥的方案。  增加：  N=AES-192  R=AES-256 |
| ZMK | key | 16H或32H或1A+32H或1A+48H或1A+64H | char | 仅当模式为1时才显示该ZMK域。 |
| 密钥方案（ZMK） | keyProj | 1 | char | 加密输出密钥的密钥方案。仅当模式为1时才显示该域。 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输出参数 | TAG名 | 长度 | 类型 | 备注 |
| 应答码 | retCode | 2 | char | 参见错误码 |
| 密钥（LMK） | keyLMK | 16H或 1A+32H或1A+48H或1A+64H | char | LMK加密下的密钥。 |
| 密钥（ZMK） | keyZMK | 16H或 1A+32H或1A+48H或1A+64H | char | ZMK加密下的密钥。 |

2.2 . 对称算法加解密数据(300002)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入参数 | TAG名 | 长度 | 类型 | 备注 |
| EncryptFlag | encryptFlag | 1 | char | 1－解密  2－加密 |
| Mode | mode | 2 | char | 01－DES\_ECB  02－DES\_CBC |
| 密钥类型 | keyType | 3 | char | “008”---ZAK  “00A”—ZEK |
| Key length | keyLength | 1 | char | 0 = 单长度密钥  1 = 双长度密钥 |
| 密钥 | key | 16/32 | char | 该密钥由LMK加密的密文送入 |
| Iv | iv | 16 | char | 初始向量。 Mode=02时，该域存在。 |
| 数据长度 | dataLength | 4 | char | 加解密数据长度 定长4位 |
| 数据 | data | 2\*n | char |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输出参数 | TAG名 | 长度 | 类型 | 备注 |
| 应答码 | retCode | 2 | char | 参见错误码 |
| 数据长度 | dataLength | 4 | char | 定长4位 |
| 数据 | data | N | char |  |

2.3．将ZEK/ZAK从ZMK转为LMK加密(300003)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入参数 | TAG名 | 长度 | 类型 | 备注 |
| 标志 | flag | 1 | char | 0为ZEK，1为ZAK。 |
| ZMK | zmkKey | 16H或32H或1A+32H或1A+48H | char | LMK对（04-05）下加密的ZMK。 |
| ZEK或ZAK | zkKey | 16H或1A+32H或1A+48H | char | ZMK下加密的ZEK或ZAK。  （标志位为0时为ZEK，1为ZAK） |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输出参数 | TAG名 | 长度 | 类型 | 备注 |
| 应答码 | retCode | 2 | char | 参见错误码 |
| ZEK | zekLmkKey | 16H或1A+32H或1A+48H | char | 转换后的ZEK |
| ZAK | zakLmkKey | 16H或1A+32H或1A+48H | char | 转换后的ZAK。 |

2.4．将ZEK/ZAK从LMK转为ZMK加密(300004)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入参数 | TAG名 | 长度 | 类型 | 备注 |
| 标志 | flag | 1 | char | 0为ZEK，1为ZAK。 |
| ZMK | zmkKey | 16H或32H或1A+32H或1A+48H | char | LMK对（04-05）下加密的ZMK。 |
| ZEK或ZAK | zkKey | 16H或1A+32H或1A+48H | char | LMK下加密的ZEK或ZAK。  （标志位为0时为ZEK，1为ZAK） |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输出参数 | TAG名 | 长度 | 类型 | 备注 |
| 应答码 | retCode | 2 | char | 参见错误码 |
| ZEK | zekZmkKey | 16H或1A+32H或1A+48H | char | 转换后的ZEK |
| ZAK | zakZmkKey | 16H或1A+32H或1A+48H | char | 转换后的ZAK。 |

2.5．将ZPK由ZMK转为LMK加密(300005)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入参数 | TAG名 | 长度 | 类型 | 备注 |
| ZMK | zmkKey | 16H或32H或1A+32H或1A+48H | char | LMK对（04-05）下加密的ZMK。 |
| ZPK | zpkZmkKey | 16H或1A+32H或1A+48H | char | ZMK下加密的ZPK。 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输出参数 | TAG名 | 长度 | 类型 | 备注 |
| 应答码 | retCode | 2 | char | 参见错误码 |
| ZPK | zpkLmkKey | 16H或1A+32H或1A+48H | char | 转换后的ZPK；由LMK06-07对加密。 |

2.6．将ZPK由LMK转为ZMK加密(300006)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入参数 | TAG名 | 长度 | 类型 | 备注 |
| ZMK | zmkKey | 16H或32H或1A+32H或1A+48H | char | LMK对（04-05）下加密的ZMK。 |
| ZPK | zpkLmkKey | 16H或1A+32H或1A+48H | char | LMK下加密的ZPK。 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输出参数 | TAG名 | 长度 | 类型 | 备注 |
| 应答码 | retCode | 2 | char | 参见错误码 |
| ZPK | zpkZmkKey | 16H或1A+32H或1A+48H |  | 转换后的ZPK；由ZMK加密。 |

2.7 . 用EDK密钥加解密数据(300007)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入参数 | TAG名 | 长度 | 类型 | 备注 |
| EncryptFlag | encryptFlag | 1 | char | 0－加密  1－解密 |
| ~~Mode~~ | ~~mode~~ | ~~1~~ | ~~char~~ | **~~G:~~**~~ECB模式~~  ~~H:CBC模式~~  ~~此处的标志位改为字母表示~~  ~~为了兼容原有指令，当此项为空时，默认为ECB模式~~ |
| EDK | key | 16 or 32  1A+32 or 1A+48 | char | LMK24~25加密  007密钥策略 |
| ~~IV-CBC~~ | ~~ivCbc~~ | ~~16~~ | ~~char~~ | ~~仅当模式为1时，有此项~~ |
| 数据长度 | dataLength | 4 | char | 加解密数据长度 定长4位 |
| 数据 | data | 2\*n | char |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输出参数 | TAG名 | 长度 | 类型 | 备注 |
| 应答码 | retCode | 2 | char | 参见错误码 |
| 数据 | data | N | char |  |

2.8 . 使用带入的密钥进行数据加解密计算 (300008)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入参数 | TAG名 | 长度 | 类型 | 备注 |
| 消息块号 | msgNo | 1 | char | 0=仅一块  1=第一块。  2=中间块。  3=最后块。 |
| 密钥模式 | KeyMode | 1 | char | 密钥模式：  0=DES加密  1=DES解密  2=AES加密  3=AES解密 |
| 密钥加密模式 | EncryptMode | 1 | char | 密钥加密模式：   1=ECB   2=CBC |
| 密钥类型 | keyType | 1 | char | 密钥类型：   0=ZEK   1=TEK  2 = ZMK |
| 密钥 | key | 16H/1A+32H/1A+3H/1A+48H/1A+64H | char | LMK30-31对下保护的ZEK  LMK32-33对下保护的TEK |
| 输入消息类型 | inMsgType | 1 | char | 消息类型：   0=消息数据为二进制   1=消息数据为扩展十六进制 |
| 输出消息类型 | outMsgType | 1 | char | 消息类型：   0=消息数据为二进制   1=消息数据为扩展十六进制 |
| IV | iv | 16H/32H | char | 初始值，仅当密钥加密模式为2,3,4时有此域。  当密钥模式为0,1时，该域为16H  当密钥模式为2,3,4,5,6,7时，该域为32H |
| 数据长度 | dataLength | 3H | char | 将要作MAC的消息长度（若消息类型为二进制，则为随后域长度；若消息类型为扩展十六进制，则为随后域长度的一半）。 |
| 数据 | data | 2\*n | char | 二进制或扩展十六进制形式的消息块。 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输出参数 | TAG名 | 长度 | 类型 | 备注 |
| 应答码 | retCode | 2 | char | 参见错误码 |
| 数据长度 | dataLength | 4 | char | 定长4位 |
| 数据 | data | N | char |  |
| IV数据域 | iv | 32 | char |  |

2.9．随机生成密钥(300009)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入参数 | TAG名 | 长度 | 类型 | 备注 |
| 密钥方案 | keyProj | 1 | char | 0 – DES密钥  1 – SM1密钥 |
| 模式 | mode | 1 | char | 0－产生密钥，返回密钥密文和校验值  1－产生密钥，并保存在指定的索引  2 – 产生密钥，返回密钥密文和校验值，并保存在指定的索引 |
| 密钥类型 | keyType | 3 | char | 密钥类型 |
| 密钥方案（LMK） | keyProg | 1 | char |  |
| 密钥索引 | keyIndex | 1A+3H | char | 仅当模式为1或2时，显示该域 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输出参数 | TAG名 | 长度 | 类型 | 备注 |
| 应答码 | retCode | 2 | char | 参见错误码 |
| 密钥（LMK） | keyLMK | 16H或 1A+32H或1A+48H | char | 仅当模式为0或2时，显示该域，LMK加密下的密钥。 |
| 密钥校验值 | checkKey | 16H | char | 仅当模式为0或2时，密钥校验值。 |

2.10．数据加解密计算(300010)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入参数 | TAG名 | 长度 | 类型 | 备注 |
| 加密模式标识 | enMode | 1 | char | 0=离散密钥3DES加密   1=离散密钥3DES解密   2=做1DES\_MAC（密钥的左8字节）(ANSI X9.19 MAC)   3=做3DES\_MAC(ANSI X9.19 MAC)   4=离散密钥DES加密   5=离散密钥DES解密   6=过程密钥DES加密   7=过程密钥DES解密   8=过程密钥3DES加密   9=过程密钥3DES解密 |
| 方案ID | enProj | 2 | char | 加解密算法模式：   01=ECB   02=CBC   03=CFB   04=OFB |
| 根密钥类型 | rootKeyType | 3 | char | 产生卡片密钥的发卡行密钥类型：   109= MK-AC   209= MK-SMI   309= MK-SMC   509= MK-DN |
| 根密钥 | rootKey | 32H/1A+32H/1A+3H | char | 暂时先实现1A+3H  用来产生卡片密钥的发卡行MK-SMI主密钥。1A+3H表示使用K+3位索引方式读取加密机内保存密钥。 |
| 离散次数 | disperseNum | 1 | char | 指定对根密钥离散的次数（1-3次） |
| 离散数据 | disperseData | 16\*n | char | 离散卡片密钥的数据，其中n代表离散次数 |
| 过程数据 | processData | 16 | char | 用于产生过程密钥的数据，仅当加密模式标识为6，7，8，9时有此域.当加密模式为8，9时，此域为（00||00||00||00||00||00|| ATC）的形式，ATC为4H |
| 数据填充标识 | fillFlag | 2 | char | 01=不添补数据   02=添补数据（仅当加密模式标识为2或3时,此域为02，在数据后填充0x80,0x00 …） |
| 数据长度 | dataLength | 3 | char | 数据长度/2 （实际长度应为8的倍数） |
| 数据 | data | n\*16 | char | 数据明文/密文（CBC加密时IV为前8字节,若仅为8字节数据，则加解密的数据为0000…） |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输出参数 | TAG名 | 长度 | 类型 | 备注 |
| 应答码 | retCode | 2 | char | 参见错误码 |
| 数据长度 | dataLength | 3 | char | 数据长度/2 |
| 数据 | data | n\*16 | char | 数据明文/密文 |

2.11 . 产生一对RSA密钥(300011)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入参数 | TAG名 | 长度 | 类型 | 备注 |
| 密钥长度 | keyLength | 4 | char | 比特长度：“0320”－“4096”，应为8的整倍数。 |
| 私钥索引 | keyIndex | 2 | char | “01”－“20”：密码机内保存新生成的密钥。  “99”：不保存新生成的密钥 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输出参数 | TAG名 | 长度 | 类型 | 备注 |
| 应答码 | retCode | 2 | char | 参见错误码 |
| 私钥长度 | priLength | 4 | char |  |
| 私钥 | priKey | N | char |  |
| 公钥 | pubKey | N | char |  |

2.12 . 公钥加密(300012)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入参数 | TAG名 | 长度 | 类型 | 备注 |
| 填充方式 | fillMode | 1 | char | “0”：如果数据长度小于密钥长度，加密时先在数据前面补0x00，以使数据长度等于密钥的长度，然后再进行加密；  “1”：PKCS填充方式（一般情况下使用此方式） |
| 公钥索引 | keyIndex | 2 | char | “00”－“20”  “99”为外带公钥 |
| 明文数据长度 | dataLength | 4 | char | 明文数据的字节数4位定长 |
| 明文 | data | n |  | 用于加密的明文数据 |
| 公钥 | pubKey | n |  | NS.1 DER编码方式，当公钥索引为“99”时存在 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输出参数 | TAG名 | 长度 | 类型 | 备注 |
| 应答码 | retCode | 2 | char | 参见错误码 |
| 数据长度 | dataLength | 4 | char | 定长4位 |
| 加密数据 | secretData | N | char |  |

2.13 . 私钥解密(300013)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入参数 | TAG名 | 长度 | 类型 | 备注 |
| 填充方式 | fillMode | 1 | char | “0”：如果数据长度小于密钥长度，加密时先在数据前面补0x00，以使数据长度等于密钥的长度，然后再进行加密；  “1”：PKCS填充方式（一般情况下使用此方式） |
| 私钥索引 | keyIndex | 2 | char | “00”－“20”  “99”为外带私钥 |
| 私钥长度 | keyLength | 4 | char | 私钥密文字节数，当私钥索引为“99”时存在 |
| 私钥 | priKey | 2\*n | char | 加密机保护的私钥密文，当私钥索引为“99”时存在 |
| 密文数据长度 | dataLength | 4 | char | 密文数据的字节数4位定长 |
| 明文 | secretData | 2\*n |  | 用于解密的密文数据 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输出参数 | TAG名 | 长度 | 类型 | 备注 |
| 应答码 | retCode | 2 | char | 参见错误码 |
| 数据长度 | dataLength | 4 | char | 定长4位 |
| 解密数据 | data | N | char |  |

2.14 . 导入私钥(300014)(备份用)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入参数 | TAG名 | 长度 | 类型 | 备注 |
| 私钥索引 | index | 2 | char | “00”－“20” |
| 私钥长度 | keyLength | 4 | char |  |
| 私钥 | priKey | n | char |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输出参数 | TAG名 | 长度 | 类型 | 备注 |
| 应答码 | retCode | 2 | char | 参见错误码 |

2.15 . 将由公钥加密的PIN转换成ZPK加密(300015)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入参数 | TAG名 | 长度 | 类型 | 备注 |
| 私钥索引 | keyIndex | 2 | char | 指定的私钥，用于解密PIN数据密文 99 私钥外带 |
| 填充方式 | fillMode | 1 | char | “0”：如果数据长度小于密钥长度，加密时先在数据前面补0x00，以使数据长度等于密钥的长度，然后再进行加密；  “1”：PKCS填充方式（一般情况下使用此方式） |
| 密文 | sercetData | n |  | 经公钥加密的数据密文 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输出参数 | TAG名 | 长度 | 类型 | 备注 |
| 应答码 | retCode | 2 | char | 参见错误码 |
| 数据长度 | dataLength | 4 | char | 定长4位 |
| 数据 | data | N | char |  |

2.16 . 生成摘要(300016)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入参数 | TAG名 | 长度 | 类型 | 备注 |
| Mech | mech | 1 | char | 0－SHA-1  1－MD5 |
| 摘要长度 | abstractLength | 4 | char |  |
| 摘要 | abstract | n | char |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输出参数 | TAG名 | 长度 | 类型 | 备注 |
| 应答码 | retCode | 2 | char | 参见错误码 |
| 摘要 | digest | N | char |  |

2.17 . 银联POS 算MAC (300017)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入参数 | TAG名 | 长度 | 类型 | 备注 |
| 密钥类型 | keyType | 1 | char | 0－TAK（终端认证密钥）  1－ZAK（区域认证密钥） |
| Key length | keyLength | 1 | char | 0 = 单长度密钥  1 = 双长度密钥 |
| 密钥 | key | 16H或32H或1A+32H | char | 对应LMK对下加密的密钥。  TAK－LMK对（16-17）下加密  ZAK－LMK对（26-27）下加密 |
| 消息长度 | dataLength | 4 |  | 报文的字符数/2 |
| 消息 | data | 2n | char | 须做MAC的数据。 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输出参数 | TAG名 | 长度 | 类型 | 备注 |
| 应答码 | retCode | 2 | char | 参见错误码 |
| MAC | mac | 16 | char |  |

2.18 . 产生SM2密钥对(300018)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入参数 | TAG名 | 长度 | 类型 | 备注 |
| 密钥长度 | keyLength | 4 | char | 比特长度：应为256 |
| 密钥类型 | keyType | 1 | char | 0：协商1：签名；2：加密；3：签名、加密和协商 |
| 密钥索引 | keyIndex | 2 | char | ““01”－“20”：密码机内保存新生成的密钥。  “99”：不保存新生成的密钥。 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输出参数 | TAG名 | 长度 | 类型 | 备注 |
| 应答码 | retCode | 2 | char | 参见错误码 |
| 密钥长度 | secretLength | 4 | char |  |
| 密钥密文 | secretKey | n | char |  |
| 公钥明文X | pubKeyX | 64 | char |  |
| 公钥明文Y | pubKeyY | 64 | char |  |

2.19 . 用SM2私钥做签名(300019)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入参数 | TAG名 | 长度 | 类型 | 备注 |
| 密钥索引 | keyIndex | 2 | char | ““01”－“20”：密码机内保存新生成的密钥。  “99”：不保存新生成的密钥。 |
| 外带密钥长度 | secretLength | 4 | char | 仅当密钥索引为99时有此域，下一个域长度 |
| 外带密钥 | secretKey | n | char | 仅当密钥索引为99时有此域，SM2密钥密文 |
| HASH算法 | hashAlgorithm | 2 | char | 01：不做HASH，此时数据长度必须是32字节  02：用SM3在内部做HASH |
| 用户标识长度 | usrFlagLength | 4 | char | 仅当HASH算法为02时有此域 |
| 用户标识 | usrFlag | n | char | 仅当HASH算法为02时有此域 |
| 数据长度 | dataLength | 4 | char |  |
| 数据 | data | n | char | 用于运算的数据。  如果[HASH算法]的值是01,该数据应该是HASH过后的数据，且长度必须为32字节。 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输出参数 | TAG名 | 长度 | 类型 | 备注 |
| 应答码 | retCode | 2 | char | 参见错误码 |
| 签名结果的R部分 | signatureR | 64 | char | 不足256位的在高位补0 |
| 签名结果的S部分 | signatureS | 64 | char | 不足256位的在高位补0 |

2.20 . 用SM2公钥做验签(300020)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入参数 | TAG名 | 长度 | 类型 | 备注 |
| 密钥索引 | keyIndex | 2 | char | ““01”－“20”：密码机内保存新生成的密钥。  “99”：不保存新生成的密钥。 |
| 公钥明文X | pubKeyX | 64 | char | 仅当公钥索引为99时有该域  不足256位的在高位补0 |
| 公钥明文Y | pubKeyY | 64 | char | 仅当公钥索引为99时有该域  不足256位的在高位补0 |
| 签名结果R | signatureR | 64 | char | 不足256位的在高位补0 |
| 签名结果S | signatureS | 64 | char | 不足256位的在高位补0 |
| HASH算法 | hashAlgorithm | 2 | char | 01：不做HASH，此时数据长度必须是32字节  02：用SM3在内部做HASH |
| 用户标识长度 | usrFlagLength | 4 | char | 仅当HASH算法为02时有此域 |
| 用户标识 | usrFlag | n | char | 仅当HASH算法为02时有此域 |
| 数据长度 | dataLength | 4 | char |  |
| 数据 | data | n | char | 用于运算的数据。  如果[HASH算法]的值是01,该数据应该是HASH过后的数据，且长度必须为32字节。 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输出参数 | TAG名 | 长度 | 类型 | 备注 |
| 应答码 | retCode | 2 | char | 参见错误码 |

2.21 . 用SM2公钥做SCE加密(C1C2C3) (300021)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入参数 | TAG名 | 长度 | 类型 | 备注 |
| 密钥索引 | keyIndex | 2 | char | ““01”－“20”：密码机内保存新生成的密钥。  “99”：不保存新生成的密钥。 |
| 公钥明文X | pubKeyX | 64 | char | 仅当公钥索引为99时有该域  不足256位的在高位补0 |
| 公钥明文Y | pubKeyY | 64 | char | 仅当公钥索引为99时有该域  不足256位的在高位补0 |
| 数据长度 | dataLength | 4 | char |  |
| 数据 | data | n | char |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输出参数 | TAG名 | 长度 | 类型 | 备注 |
| 应答码 | retCode | 2 | char | 参见错误码 |
| 密文长度 | secretLength | 4 | char | 等于32+32+明文数据长度+32 |
| 密文 | secretData | n | char | C1X+C1Y+C2+C3 |

2.22 . 用SM2私钥做SCE解密(C1C2C3) (300022)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入参数 | TAG名 | 长度 | 类型 | 备注 |
| 密钥索引 | keyIndex | 2 | char | ““01”－“20”：密码机内保存新生成的密钥。  “99”：不保存新生成的密钥。 |
| 外带密钥长度 | secretLength | 4 | char | 仅当密钥索引为99时有此域，下一个域长度 |
| 外带密钥 | secretKey | n | char | 仅当密钥索引为99时有此域，SM2密钥密文 |
| 密文长度 | secretDataLength | 4 | char | 等于32+32+明文数据长度+32 |
| 密文 | secretData | n | char | C1X+C1Y+C2+C3 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输出参数 | TAG名 | 长度 | 类型 | 备注 |
| 应答码 | retCode | 2 | char | 参见错误码 |
| 数据长度 | dataLength | 4 | char |  |
| 数据 | data | n | char |  |

2.23．生成密钥SM4密钥(300023)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入参数 | TAG名 | 长度 | 类型 | 备注 |
| 模式 | mode | 2 | char | 0－产生密钥。  1－产生密钥并在ZMK下加密。  2-产生密钥并保存在指定的索引下  3-产生密钥并保存在指定的索引下以及在ZMK下加密 |
| 密钥类型 | keyType | 3 | char | 密钥类型 |
| 密钥方案 | keyProg | 1 | char | 密钥长度/LMK下加密密钥的方案。  增加：  N=AES-192  R=AES-256 |
| ZMK | key | 1A+32H或1A+3H | char | 仅当模式为1时才显示该ZMK域。 |
| 密钥索引 | keyIndex | 1A+3H | char | 产生密钥存放位置。仅当模式为2，3时才显示该域。 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输出参数 | TAG名 | 长度 | 类型 | 备注 |
| 应答码 | retCode | 2 | char | 参见错误码 |
| 密钥（LMK） | keyLMK | 1A+32H | char | LMK加密下的密钥，’S’+32H的密文。 |
| 密钥（ZMK） | keyZMK | 1A+32H | char | ZMK加密下的密钥，’S’+32H的密文。模式为1/3由此域 |
| 校验值 | checkKey | 16 | char |  |

2.24 . SM4算法加解密数据(300024)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入参数 | TAG名 | 长度 | 类型 | 备注 |
| EncryptFlag | encryptFlag | 1 | char | 1－解密  2－加密 |
| Mode | mode | 2 | char | 01 –SM4-ECB  02 --SM4-CBC |
| 密钥类型 | keyType | 3 | char |  |
| Key length | keyLength | 1 | char | 0 = 单长度密钥  1 = 双长度密钥 |
| 密钥 | key | 1A+32 | char | ‘S’表示SM4密钥  该密钥由LMK加密的密文送入 |
| Key校验值 | checkKey | 16 | char |  |
| Iv | iv | 32 | char | 初始向量。 Mode=02时，该域存在。 |
| 数据长度 | dataLength | 4 | char | 加解密数据长度，长度必须是16的倍数 |
| 数据 | data | 2\*n | char |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输出参数 | TAG名 | 长度 | 类型 | 备注 |
| 应答码 | retCode | 2 | char | 参见错误码 |
| 数据长度 | dataLength | 4 | char | 定长4位 |
| 数据 | data | N | char |  |

2.25．随机产生SM1密钥(300025)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入参数 | TAG名 | 长度 | 类型 | 备注 |
| 密钥类型 | keyType | 3 | char | 001 – ZPK （LMK0607加密）  008 – ZAK (LMK2627加密)  00A – ZEK(LMK3031加密) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输出参数 | TAG名 | 长度 | 类型 | 备注 |
| 应答码 | retCode | 2 | char | 参见错误码 |
| 密钥（LMK） | keyLMK | 32H | char | LMK下加密的密钥 |
| 校验值 | checkKey | 16 | char |  |

2.26 . SM1数据加解密(300026)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入参数 | TAG名 | 长度 | 类型 | 备注 |
| Mode | mode | 2 | char | 运算方式  0：SM1 ECB加密  1：SM1 ECB解密  2：SM1 CBC加密  3：SM1 CBC解密  4：SM4 ECB加密  5：SM4 ECB解密  6：SM4 CBC加密  7：SM4 CBC解密解密指令涉及安全问题，这里不提供。 |
| 密钥类型 | keyType | 3 | char |  |
| Key length | keyLength | 1 | char | 0 = 单长度密钥  1 = 双长度密钥 |
| 密钥 | key | 1A+32 | char | ‘S’表示SM4密钥  该密钥由LMK加密的密文送入 |
| Key校验值 | checkKey | 16 | char |  |
| Iv | iv | 16 | char | 仅当mode等于2、3 、6、7 时有此域 |
| 数据长度 | dataLength | 4 | char | 加解密数据长度，长度必须是16的倍数 |
| 数据 | data | 2\*n | char |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输出参数 | TAG名 | 长度 | 类型 | 备注 |
| 应答码 | retCode | 2 | char | 参见错误码 |
| 数据长度 | dataLength | 4 | char | 定长4位 |
| 数据 | data | N | char |  |

2.27 . SM3计算 (300027)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入参数 | TAG名 | 长度 | 类型 | 备注 |
| 模式 | mode | 1 | char | 1-只对数据进行SM3摘要  2-对用户ID、公钥、及数据按照SM2的规范进行摘要 |
| 算法标识 | algorithmFlag | 1 | char | 3-SM3 |
| 公钥明文X | pubKeyX | 64 | char | 仅当模式为2有该域  不足256位的在高位补0 |
| 公钥明文Y | pubKeyY | 64 | char | 仅当模式为2有该域  不足256位的在高位补0 |
| 用户标识长度 | userFlagLength | 4 | char | 仅当模式为2有该域 |
| 用户标识 | userFlag | n |  | 仅当模式为2有该域 |
| 数据长度 | dataLength | 4 | char |  |
| 数据 | data | n | char |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输出参数 | TAG名 | 长度 | 类型 | 备注 |
| 应答码 | retCode | 2 | char | 参见错误码 |
| HSAH值 | hashData | 64 | char | Hash结果：见HASH结果长度标志 |

2.28 . 用私钥签名(300028)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入参数 | TAG名 | 长度 | 类型 | 备注 |
| 填充方式 | fillMode | 1 | char | “0”：如果数据不是密钥长度的整倍数，后面补0x00  “1”：PKCS填充方式（一般情况下使用此方式） |
| 密钥索引 | keyIndex | 2 | char | ““01”－“20”：密码机内保存新生成的密钥。  “99”：不保存新生成的密钥。 |
| 私钥长度 | priKeyLength | 4 | char | 私钥密文字节数，当私钥索引为“99”时存在 |
| 私钥密文 | priKey | n | char | 加密机保护的私钥，当私钥索引为“99”时存在 |
| 数据长度 | dataLength | 4 | char |  |
| 数据 | data | n | char | 用于签名的数据 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输出参数 | TAG名 | 长度 | 类型 | 备注 |
| 应答码 | retCode | 2 | char | 参见错误码 |
| 签名长度 | signatureLength | 64 | char |  |
| 签　名 | signature | 64 | char |  |

2.29 . 用公钥验证(300029)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入参数 | TAG名 | 长度 | 类型 | 备注 |
| 填充方式 | fillMode | 1 | char | “0”：如果数据不是密钥长度的整倍数，后面补0x00  “1”：PKCS填充方式（一般情况下使用此方式） |
| 签名长度 | signatureLength | 4 | char |  |
| 签　名 | signature | n | char |  |
| 数据长度 | dataLength | 4 | char |  |
| 数据 | data | n | char | 用于签名的数据 |
| 公　钥 | pubKey | n | char |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输出参数 | TAG名 | 长度 | 类型 | 备注 |
| 应答码 | retCode | 2 | char | 参见错误码 |

2.30 . 私钥加密(300030)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入参数 | TAG名 | 长度 | 类型 | 备注 |
| 填充方式 | fillMode | 1 | char | “0”：如果数据长度小于密钥长度，加密时先在数据前面补0x00，以使数据长度等于密钥的长度，然后再进行加密；  “1”：PKCS填充方式（一般情况下使用此方式） |
| 私钥索引 | keyIndex | 2 | char | “00”－“20”  “99”为外带私钥 |
| 私钥长度 | keyLength | 4 | char | 私钥密文字节数，当私钥索引为“99”时存在 |
| 私钥 | priKey | 2\*n | char | 加密机保护的私钥密文，当私钥索引为“99”时存在 |
| 明文数据长度 | dataLength | 4 | char | 明文数据的字节数4位定长 |
| 明文 | data | n |  | 用于加密的明文数据 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输出参数 | TAG名 | 长度 | 类型 | 备注 |
| 应答码 | retCode | 2 | char | 参见错误码 |
| 数据长度 | dataLength | 4 | char | 定长4位 |
| 加密数据 | secretData | N | char |  |

2.31 . 公钥解密(300031)（只有软加密，加密机无此指令）

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输入参数 | TAG名 | 长度 | 类型 | 备注 |
| 填充方式 | fillMode | 1 | char | “0”：如果数据长度小于密钥长度，加密时先在数据前面补0x00，以使数据长度等于密钥的长度，然后再进行加密；  “1”：PKCS填充方式（一般情况下使用此方式） |
| 公钥索引 | keyIndex | 2 | char | “00”－“20”  “99”为外带公钥 |
| 公钥 | pubKey | n | char | 加密机保护的公钥密文，当公钥索引为“99”时存在 |
| 密文数据长度 | dataLength | 4 | char | 密文数据的字节数4位定长 |
| 明文 | secretData | 2\*n |  | 用于解密的密文数据 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 输出参数 | TAG名 | 长度 | 类型 | 备注 |
| 应答码 | retCode | 2 | char | 参见错误码 |
| 数据长度 | dataLength | 4 | char | 定长4位 |
| 解密数据 | data | N | char |  |

3. 密钥类型表

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| LMK对/变种 | LMK对代码 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| 04-05 | 00 | ZMK | ZMK  (Comp) | KML |  |  |  |  |  |  |  |
| 06-07 | 01 | ZPK |  |  |  |  |  |  |  |  |  |
| 14-15 | 02 | PVK  TPK  TMK |  |  |  | CVK  CSCK |  |  |  |  |  |
| 16-17 | 03 | TAK |  |  |  |  |  |  |  |  |  |
| 18-19 | 04 |  |  |  |  |  |  |  |  |  |  |
| 20-21 | 05 |  |  |  |  |  |  |  |  |  |  |
| 22-23 | 06 | WWK |  |  |  |  |  |  |  |  |  |
| 24-25 | 07 | EDK |  |  |  |  |  |  |  |  |  |
| 26-27 | 08 | ZAK |  |  |  |  |  |  |  |  |  |
| 28-29 | 09 | BDK | MK-AC | MK-SMI | MK-SMC | MK-DAK | MK-DN |  |  |  |  |
| 30-31 | 0A | ZEK |  |  |  |  |  |  |  |  |  |
| 32-33 | 0B |  |  |  |  |  |  |  |  |  |  |
| 34-35 | 0C | RSA-  SK |  |  |  |  |  |  |  |  |  |
| 36-37 | 0D |  |  |  |  |  |  |  |  |  |  |
| 38-39 | 0E |  |  |  |  |  |  |  |  |  |  |

4. 密钥方案表

|  |  |
| --- | --- |
| **密钥方案标志** | **注释** |
| Z | 使用ANSI X9.17方式加密的单倍长度DES密钥 |
| U | 使用变量方式对双倍长度密钥进行的加密。用于在LMK下对密钥的加密；  可用于密钥的输入和输出。 |
| T | 使用变量方式对三倍长度密钥进行的加密。用于在LMK下对密钥的加密；  可用于密钥的输入和输出。 |
| X | 使用ANSI X9.17方式对双倍长度密钥进行的加密，只能用于输入和输出  密钥。这种方式可以通过“CS”命令使之有效。 |
| Y | 使用ANSI X9.17方式对三倍长度密钥进行的加密，只能用于输入和输出  密钥。这种方式可以通过“CS”命令使之有效。 |
| S | 使用LMK 对SM4密钥进行加密。能用于输入和输出  密钥。这种方式可以通过“CS”命令使之有效。 |

5. 错误码定义

|  |  |
| --- | --- |
| **代码** | **具体描述** |
| 00 | 没有错误。 |
| 01 | 确认错误。/输入密钥奇偶校验错误警告。 |
| 02 | 密钥长度错误（不符合算法要求）。 |
| 04 | 无效密钥类型错误。 |
| 05 | 无效密钥长度标识。 |
| 10 | 源密钥奇偶校验错。 |
| 11 | 目的密钥奇偶校验错。/密钥为全0。 |
| 12 | 用户存储区域的内容无效。重启、掉电或重写。 |
| 13 | 主密钥奇偶校验错。 |
| 14 | LMK对02-03加密下的PIN失效。 |
| 15 | 无效的输入数据（无效的格式，无效的字符，或者没有提供足够的数据）。 |
| 16 | 控制台或打印机没有准备好或者没有连接好。 |
| 17 | HSM不在授权状态，或者不能清除PIN输出，或者两种情况都有。 |
| 18 | 没有装载文档格式定义。 |
| 19 | 指定的Diebold表无效。 |
| 20 | PIN块没有包含有效值。 |
| 21 | 无效的索引值，或者索引/块数导致了溢出。 |
| 22 | 无效的账号。 |
| 23 | 无效的PIN块格式代码。 |
| 24 | PIN长度小于4或大于12。 |
| 25 | 十进制表错误。 |
| 26 | 无效的密钥方案。 |
| 27 | 不匹配的密钥长度。 |
| 28 | 无效的密钥类型。 |
| 29 | 密钥函数被禁止。 |
| 30 | 参考数无效。 |
| 31 | 没有足够的请求入口以提供批量处理。 |
| 33 | LMK密钥转换存储区被破坏。 |
| 40 | 无效的固件校验和。 |
| 41 | 内部的硬件/软件错：RAM已坏，无效的错误代码，等等。 |
| 42 | DES错误。 |
| 47 | DSP错误；报告给管理员。 |
| 49 | 私钥错误；报告给管理员。 |
| 60 | 无此命令。 |
| 74 | 无效摘要信息语法（仅仅无哈希模式）。 |
| 75 | 无效公钥/私钥对。 |
| 76 | 公钥长度错误。 |
| 77 | 明文数据块错误。 |
| 78 | 私钥长度错误。 |
| 79 | 哈希算法对象标识错误。 |
| 80 | 数据长度错误。MAC数据（或其它数据）的长度大于或小于指定的长度。 |
| 81 | 证书偏移值与长度错。 |
| 90 | HSM接收的请求信息中的数据奇偶校验错。 |
| 91 | 纵向冗余校验（LRC）字符不匹配对输入数据所计算出的值  （当HSM接收到一个透明的异步包时）。 |
| 92 | 计数值（命令/数据域）不在有效范围内，或者不正确  （当HSM接收到一个透明的异步包时）。 |

错误码说明

|  |  |
| --- | --- |
| 错误代码 | 说明 |
| 00 | 正确。 |
| 01 | 校验值错/无密钥/验签失败。 |
| 03 | 密钥类型错。 |
| 04 | 无效的密钥类型/模式标志错。 |
| 05 | ID标识错。 |
| 06 | 非法的偏移量。 |
| 07 | MAC数据填充标志错。 |
| 09 | MAC长度标志错。 |
| 10 | 密钥奇偶校验错。 |
| 12 | 用户索引下无密钥/密钥索引错。 |
| 13 | LMK错，报告给管理员。 |
| 14 | 解密后数据填充格式错。 |
| 15 | 输入数据错误。 |
| 16 | 打印机没准备好。 |
| 17 | 密码机没有授权。 |
| 18 | 打印格式没定义。 |
| 28 | 密钥类型错。 |
| 33 | 密钥索引无效。 |
| 34 | 离散次数错。 |
| 36 | 算法标志错/算法转换选择错。 |
| 42 | 源/目的加密算法选择错。 |
| 46 | 数据长度错。 |
| 80 | 数据长度错。 |
| 81 | 密钥长度错/密文长度错/缺少分隔符。 |
| 83 | 写密钥失败。 |
| 84 | 读密钥失败。 |
| 85 | SM2密钥索引下无密钥。 |
| 89 | SM2签名失败。 |
| 97 | SM2加解密失败。 |