阿里云
云盾加密服务

API参考

金融加密机

20180831

法律声明

阿里云提醒您在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。
2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。
3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。
4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的”现状“、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。
5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含”阿里云”、Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。
6. 阿里云文档中所有内容，包括但不限于图片、架构设计、页面布局、文字描述，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表本文档中的内容。此外，未经阿里云事先书面同意，任何人不得为了任何

营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含”阿里云”、Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。

7. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

格式	说明	样例
	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	 **禁止**： 重置操作将丢失用户配置数据。
	该类警示信息可能导致系统重大变更甚至故障，或者导致人身伤害等结果。	 **警告**： 重启操作将导致业务中断，恢复业务所需时间约10分钟。
	用于警示信息、补充说明等，是用户必须了解的内容。	 **说明**： 导出的数据中包含敏感信息，请妥善保存。
	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	 **说明**： 您也可以通过按**Ctrl** + **A**选中全部文件。
>	多级菜单递进。	**设置** > **网络** > **设置网络类型**
粗体	表示按键、菜单、页面名称等UI元素。	单击 **确定**。
courier字体	命令。	执行 cd /d C:/windows 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	bae log list --instanceid *Instance_ID*
[]或者[a|b]	表示可选项，至多选择一个。	ipconfig *[-all|-t]*
{}或者{a|b}	表示必选项，至多选择一个。	swich *{stand | slave}*

目录

1 API概览

该章节介绍加密实例的业务方面的API接口，包括对称加密、非对称加密等应用业务的API接口。加密服务API适用于金融应用服务， 此API为常用金融类通用接口。

API	描述
getInstance（通过环境变量指定配置信息）	此方法在实现实例化之外，同时完成与密码机连接的相关参数设置。此方法不需传入参数，但需通过系统环境变量**TACFG_INTERFACE_J**来指定配置信息。
getInstance（通过参数指定配置信息）	此方法在实现实例化之外，同时完成与密码机连接的相关参数设置。此方法需传入参数来指定配置文件路径或配置信息。
updateConfigure	重置此对象中的配置信息。
getDeviceSerial	获取设备序列号。
genWorkKey	生成工作密钥。
generateZmkLetter	生成ZMK密钥信封。
genWorkKeyEnc	生成工作密钥，并输出在ZMK下加密的密文。
genRandom	随机数据生成。
generalDataEnc	通用数据加密。
dataEnc	数据加密。
dataDec	数据解密。
generalDataDec	通用数据解密。
calMAC	生成数据MAC。
verifyMAC	验证交易数据MAC/TAC。
generalCalMAC（String）	通用生成数据MAC。
generalCalMAC（byte[]）	通用生成数据MAC。
generalVerifyMAC	校验数据MAC。
verifyCipherAndGenARPC	PBOC验证ARQC并产生ARPC。
encryptScript	PBOC脚本加密。
calScriptMAC	计算脚本MAC。

API	描述
verifyARQC	验证ARQC。
genARPC	产生ARPC。
exportWorkKey	LMK加密的密钥转换成ZMK下加密导出。
importWorkKey	导入工作密钥，导入ZMK加密的密钥，即ZMK加密的密钥转换为LMK下加密。
loadPrintFormat	将自定义的打印格式数据装在到HSM中。
printKeyElement	随机产生一个密钥成份，通过连接于密码机的打印机打印出明文，并返回成份的密文。需满足主机服务的信函打印权限。
diverAndGenNewKey	分散产生新密钥，可选的存储到加密机内。
genCVV	卡校验值的计算，VISA CVV的产生和验证。
verifyCVV	校验VISA CVV。
genRandPIN	产生一个随机PIN码。
encPINByLmk	LMK加密一个明文PIN。
encPinByZpk	使用ZPK加密明文PIN数据。
decPinByZpk	解密ZPK加密的PIN密文数据。
weakPinSet	设置密码机设备中弱口令集合属性（同时设置应用下配置的所有密码机的属性）。
weakPinCheck	弱口令检查。
transferPinLMK2ZPK	LMK加密的PIN密文转为ZPK加密。
transferPinZPK12ZPK2	将PIN由ZPK1加密转换为ZPK2加密。
transferPinZPK2LMK	将PIN由ZPK加密转换为LMK加密。
transferPin	将PIN由TPK1/ZPK1加密转换为TPK2/ZPK2加密。
transferPinSm2ToZpk（私钥使用索引号）	SM2公钥加密的数字PIN密文转为ZPK加密。
transferPinSm2ToZpk（私钥使用密文）	SM2公钥加密的数字PIN密文转为ZPK加密。
transferPinRsaToZpk（私钥使用索引号）	RSA公钥加密的数字PIN密文转为ZPK加密。
transferPinRsaToZpk（私钥使用密文）	RSA公钥加密的数字PIN密文转为ZPK加密。
RsaPublicKeyEnc	RSA公钥加密。

API	描述
RsaPublicKeyEnc（杂凑算法）	RSA数据加密。
RsaPrivateKeyDec	RSA数据解密。
RsaPrivateKeyDec（杂凑算法）	RSA数据解密。
genRsaSignature	RSA计算签名（EW）。
genRsaSignature（OAEP计算模式）	RSA计算签名（EW）OAEP计算模式。
verifyRsaSignature	RSA验证签名（EY）。
verifyRsaSignature（OAEP计算模式）	RSA验证签名（EY）OAEP计算模式。
SM2PublicKeyEnc	SM2公钥加密。
EccPublicKeyEnc	ECC算法公钥数据加密运算。
SM2PrivateKeyDec	SM2私钥解密。
EccPrivateKeyDec	ECC算法私钥数据解密运算。
genSM2Signature（密钥使用索引号）	SM2计算签名（E5）。
genSM2Signature（密钥使用密文）	SM2计算签名（E5）。
genEccSignature（密钥使用索引号）	ECC算法签名运算。
genEccSignature（密钥使用密文）	ECC算法签名运算。
verifySM2Signature（密钥使用索引号）	SM2验证签名（E6）。
verifyEccSignature（密钥使用索引号）	ECC算法验签名运算。
verifySM2Signature（密钥使用密文）	SM2验证签名（E6）。
verifyEccSignature（密钥使用密文）	ECC算法验签名运算。
generateHASH	计算数据得到数据摘要，默认采用国密SM3算法进行摘要运算。
generateHASH	计算数据得到数据摘要，采用国密SM3算法进行摘要运算并且传公钥形式。
generateHASH	计算数据得到数据摘要，支持多种hash算法模式。
generateHASH	小报文数据摘要值计算。
genRSAKeyPair	随机生成RSA密钥对。
getSM2PublicKey	获取SM2公钥明文。

API	描述
generateSm2KeyPair	产生国密SM2-256 曲线的SM2秘钥对，可选的储存在加密机中。
generateEccKeyPair	产生ECC密钥对。
getKeyInfo	获取密钥信息。
bufferEncrypt	通用大数据加密，可用于加密文件数据等通用格式的数据。
bufferEncrypt（初始化向量固定为全0）	通用大数据加密，可用于加密文件数据等通用格式的数据（初始化向量固定为全0）。
bufferDecrypt	通用大数据加密，可用于解密文件数据等通用格式的数据。
bufferDecrypt（初始化向量固定为全0）	通用大数据加密，可用于解密文件数据等通用格式的数据（初始化向量固定为全0）。
bigDatageneralEnc	使用DEK进行大包数据加密，接口中完成数据的拆包与重组。
bigDatageneralDnc	使用DEK进行大包数据解密，接口中完成数据的拆包与重组。
sm2ExportSymmkey	SM2公钥保护导出一条对称秘钥。
sm2ImportSymmkey	SM2公钥保护导入对称秘钥。
generateSignature	SM2私钥签名运算（对数据摘要运算）。
verifySignature	SM2公钥验签运算（针对数据摘要）。
rsaImportSymmkey	RSA公钥保护导入一条对称密钥。
rsaExportSymmkey	RSA公钥加密导出一条对称密钥。
sm2ExportSymmkey	SM2公钥保护导出一条对称秘钥。
getRSAPublicKey	根据密钥索引获取RSA公钥。
generalCalMAC	使用ZAK计算银联POS MAC。
generalCalMAC	使用ZAK计算银联POS MAC。
generalRSApublicKeyMAC	计算RSA公钥MAC。
generalSM2publicKeyMAC	计算RSA公钥MAC。
generalEccPublicKeyMAC	计算外部ECC公钥的MAC。
transferPintpk2PublicKey	TPK加密PIN转为公钥加密。

API	描述
encExportKeyByProKey	保护密钥（可分散 ）加密导出一条密钥（可分散）。
symmEncExportRSAkey	保护密钥加密导出RSA密钥对。
symmEncImportRSAkey	保护密钥加密保护导入RSA密钥对。
transferAscPinSm2ToZpk	非对称公钥加密的字符PIN密文转为ZPK加密。
transferAscPinSm2ToZpk（PIN组成格式为PIN明文块）	公钥加密的字符PIN密文转为ZPK加密 （公钥加密的PIN组成格式为PIN明文块）。
transferAscPinSm2ToZpk	非对称公钥加密的字符PIN密文转为ZPK加密。
genWorkKeyEnc	重载更新密钥函数，LMK密钥标识与ZMK密钥标识分开。
generateZPKCharPIN	ZPK加密字符PIN密文。
transferZpkCharPin	字符PINBLOCK转为其他密钥加密，不支持分散。
blocksEncrypt	多段数据加密，此方法以固定的ECB模式加密输入数据列表中的各段数据。
blocksDecrypt	多段数据解密，此方法以固定的ECB模式解密输入数据列表中的各段数据。
genRandomKey	产生一条随机密钥，可选的存储到密码机内（KR指令）。
deriveSymmKey	KD指令密钥分散。
disGenNewKey	分散产生新密钥，可选的存储到密码机内。
exportKeyByTranserKey	传输密钥保护导出一条密钥（KH）。
exportKeyByTranserKey	传输密钥保护导出一条密钥（KH）。
importKeyByTranserKey	传输密钥保护导出一条密钥（KH）。
encExportKeyByProKey	保护密钥加密导出一条密钥（通用（ SH））。
encImportKeyByProKey	保护密钥加密导入一条密钥（通用）。
deleteKey	删除内部指定索引的密钥（ KF）。
importSymKey	导入存储一条对称密钥。
disperKeyOutputComponentCipher	分散密钥输出子密钥的多个成分密文。
encExportKey	KMC（Kdek）加密导出多条应用密钥（G2）。

API	描述
encConfidentialData	KMC（Sdek）加密敏感数据（G3）。
encData	KMC（Senc）加密数据（G4）。
encDataMAC	KMC（Scmac）计算数据C-MAC（G5）。
testDataMAC	KMC（Srmac）验证数据R-MAC（G6）。
externalAuthentication	外部认证（G7）。
proEncKMCSessionKey	保护密钥加密出导出KMC三条会话密钥（G8）。
proRSAKey	KMC（Sdek）保护导出一对RSA密钥（GF）。
proSM2Key	KMC（Sdek）保护导出一对SM2密钥（G0）。
tKEncPINChangeKMC	TK加密的PIN密文转为KMC（Sdek）下加密（GD）。
eMVchackingARQC	EMV4.X验证ARQC/TC/AAC，可选的产生ARPC（KW）。
eMVchackingPIN	EMV4.X脚本安全报文/PIN修改（KY）。
pbocchackingPIN	PBOC脱机PIN修改/加密（KX）。
numberPINBLOCKEnc	数字PINBLOCK转加密（D7）。
s7_numberPINBLOCKEnc	数字PINBLOCK转加密（通用（S7））。
transferPinCipherTpk2ublicKey	将字符PIN由由TPK加密转为公钥加密（N5）。
numberPINPriEnc	将字数字PIN从从ZPK下加密转换到私有算法加密（CB）。
cipherCompositionSynKey	由密文成份合成一个密钥（A4）。
makeOneZPK	产生一个ZPK（IA）。
zpkSynZMKToLMK	ZPK从ZMK加密转换为LMK加密（FA）。
zpkSynLMKToZMK	ZPK从从LMK加密转换为ZMK加密（GC）。
makeOneZEK	产生一个ZEK/ZAK（FI）。
zekSynZMKToLMK	ZPK从ZMK加密转换为LMK加密（FA）。
zekSynLMKToZMK	ZEK/ZAK从LMK加密转换为ZMK加密（FM）。
makeOneTMK	产生一个TMK，TPK，PVK（HC）。
makeOneTAK	产生一个TAK（HA）。

API	描述
takSynZMKToLMK	将TAK从ZMK下加密转为LMK下加密（MI）。
takCountDataMAC	TAK计算数据MAC（MA）。
checkDataMAC	TAK验证数据MAC（MC）。
zpkCalculateMAC	银联应用系统，在线分发ZPK验证密钥，密钥类型只支持ZPK。
calculateMAC	ZAK/TAK产生X9.9和和X9.19的文报文MAC（MS）。
transferPinTpkToLmk	将PIN由TPK加密转换为LMK加密（JC）。
transferPinTpkToZpk	将PIN由TPK加密转换为ZPK加密（CA）。
produceIbmPinOffset	产生IBM PIN Offset（DE）。
ibmGetPin	使用IBM方式得到一个PIN（EE）。
checkIbmTerminalPin	校验一个用IBM方式的终端PIN（DA）。
checkIbmExchangePin	校验一个用IBM方式的交换PIN（EA）。
produceVisaPvv	产生VISA PVV（DG）。
pvvCheckZpkPinBlock	PVV校验ZPK加密的PINBLOCK（EC）。
produceOrCheckCsc	生成或者校验美国运通的CSC（RY）。
inDataHash	对一个数据块进行哈希运算（GM）。
lodingRsaKeyOld	装载RSA密钥对，兼容旧版本保留（EK）。
lodingRsaKey	装载RSA密钥对（EJ）。
rsaExportSymKey	RSA公钥保护导出一条对称密钥，RACAL兼容（GK）。
rsaImportSymKey	RSA公钥保护导入一条对称密钥，RACAL兼容（GI）。
encAESGCM256	AES-GCM-256加密。
decAESGCM256	AES-GCM-256解密。
calcHMAC	计算数据HMAC-明文密钥。
symEncExportSM2Key	保护密钥（对称）加密导出一对SM2密钥（TT）。
symImportSm2KeyPair	保护密钥（对称）加密对导入一对SM2密钥。

API	描述
getHsmFunctionState	获取密码机运行状态。
generateZmkLetter	生成ZMK密钥信封。
generateZmkLetter	生成ZMK密钥信封（由外部指定编码格式）。
eccSignature	ECC私钥签名运算。
eccVerify	Ecc公钥验签运算。
symmTransformCipher	数据转加密（通用）。
EccEciesEncrypt	Ecc NISTP256曲线加密运算。
EccEciesDecrypt	Ecc NISTP256曲线解密运算。
printPINDate	打印PIN/PIN请求数据。
generateLengthPIN	产生指定长度的随机字符PIN。
transferLmkToZpk	将字符PIN由LMK加密转换为ZPK加密。
transferZpkToLmk	将字符PIN由ZPK加密转换为LMK加密。
printPINBolckDate	打印LMK加密字符PIN。
getAsymmKeyPublicKeyAndPrivateKey	返回RSA或ECC算法指定索引下的私钥和DER编码公钥。

2 调用方式

该章节介绍加密服务API调用方式。

加密服务API接口调用方式

加密服务API支持通过以下两种方式调用：

- 配置文件形式传递属性

```
hsmGeneralFinance instance = hsmGeneralFinance.getInstance("/etc/cfg.ini");
```

- 配置信息字符串传递属性

```
hsmGeneralFinance instance = hsmGeneralFinance.getInstance(
    "{" +
    "[LOGGER];" +
    "logsw = error, debug, info, warn;" +
    "logPath = /log/;" +
    "[HOST 1];" +
    "hsmModel = SJJ1310;" +
    "host = 192.168.9.124;" +
    "port = 8018;" +
    "}"
    );
```

表 2-1: 配置说明

配置项		说明
[LOGGER]	logsw	输出日志的类型。
	logPath	输出日志的路径。
[HOST 1]	hsmModel	密码机型号：*SJJ1310*。
	linkNum	连接数，取值范围：1~64。
	host	密码机的IP。
	port	端口，固定为8018端口。
	timeout	设置为默认值。
	encodetype	设置为默认值。
	msgheadlength	设置为默认值。

加密服务API接口实例化及接口函数调用说明

说明：

配置文件以文件形式传入，文件所在地址为："./cacipher.ini"。

```
public static void main(String[] args)
  throws ConfigurationException, TAException {
 // 第一步实例化接口
 // 标识本地所在配置文件地址
 hsmGeneralFinance hsm = hsmGeneralFinance.getInstance("./cacipher.ini");
 // 声明参数(取值范围请参照帮助文档)
 int algType = 1 ;    // 算法模式
 int Key = 1;         // 密钥索引为1的密钥
 int PadFlag  = 2;    // 填充模式
 String IV = "31323334353637383930313233343536";    // 初始向量IV
 byte[] data = "测试大包数据加解密的测试数据".getBytes();
 // 第二步调用接口内部封装方法
 // 以大包数据做加密运算为例
 try{
  byte[] result = hsm.bigDatageneralEnc(algType, Key, PadFlag, IV, data);
  //result 返回加密结果
  } catch (TAException e){
   //TAException 为接口内部封装的异常信息，
   //用来接收并返回接口运行异常时抛出的异常信息。
   e.printStackTrace();
   }
 }
```

3 API接口

3.1 getInstance（通过环境变量指定配置信息）

此方法在实现实例化之外，同时完成与密码机连接的相关参数设置。此方法不需传入参数，但需通过系统环境变量**TACFG_INTERFACE_J**来指定配置信息。

```
public static hsmGeneralFinance getInstance()
                    throws cn.tass.exceptions.TAException
```

请求参数

无

返回参数

返回本类的实例。

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.2 getInstance（通过参数指定配置信息）

此方法在实现实例化之外，同时完成与密码机连接的相关参数设置。此方法需传入参数来指定配置文件路径或配置信息。

```
public static hsmGeneralFinance getInstance(java.lang.String conf)
                    throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
conf	String	是	配置信息： • 文件路径 • 配置信息内容

返回参数

返回本类的实例。

异常处理

程序运行中出错则抛出异常。

cn.tass.exceptions.TAException：接口自定义的异常。

3.3 updateConfigure

重置此对象中的配置信息。

```
public boolean updateConfigure(java.lang.String config)
                        throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
conf	String	是	配置信息： • 文件路径 • 配置信息内容

返回参数

无

异常处理

程序运行中出错则抛出异常。

cn.tass.exceptions.TAException

3.4 getDeviceSerial

获取设备序列号。

```
public java.lang.String[] getDeviceSerial()
```

请求参数

无

返回参数

设备序列号。

异常处理

无

3.5 genWorkKey

生成工作密钥。

```
public java.lang.String[] genWorkKey(java.lang.String keyType,
                                     char keyFlag,
                                     int storeKeyIndex,
                                     java.lang.String storeKeyLabel)
                              throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
keyType	String	是	密钥类型，相应密钥类型的密钥类型号，支持密钥类型代码和密钥类型名称两种格式。以下左侧为密钥类型代码，右侧为密钥类型名称： • 000：ZMK/KEK • 001：ZPK • 002：PVK/TPK/TMK • 003：TAK • 008：ZAK • 009：BDK • 00A：ZEK/DEK • 00B：TEK • 011：KMC • 109：MK-AC/MDK • 10C：HMAC • 209：MK-SMI • 309：MK-SMC • 402：CVK • 409：MK-DAK • 509：MK-DN
keyFlag	char	是	在LMK下加密的密钥密文标识： • *Z*：单倍长DES密钥 • *X*：双倍长DES密钥 • *Y*：三倍长DES密钥 • *R*：16字节SM4密钥 • *P*：16字节SM1密钥 • *L*：16字节AES密钥

名称	类型	是否必须	描述
			• *M*：24字节AES密钥 • *N*：32字节AES密钥
storeKeyIndex	int	是	密钥存储索引，范围为：1~2048 **说明**： 如果设置的取值不在1~2048，则表示不存储。
storeKeyLabel	String	是	密钥存储标签，用于在密钥内部存储时标记密钥的标签说明，0-16个ASII字符。 **说明**： 只有**storeKeyIndex**取值为1~2048时生效。

返回参数

- 0号索引下：密钥在LMK下加密的密文
- 1号索引下：密钥校验值

异常处理

程序运行中出错则抛出异常。

`cn.tass.exceptions.TAException`：接口自定义的异常。

3.6 generateZmkLetter

生成ZMK密钥信封。

```
public java.lang.String[] generateZmkLetter(java.lang.String SerilNo,
                                            char keyAlg,
                                            int hashAlgFlag)
                                     throws cn.tass.exceptions.TAException
```

此方法为定制功能，生成一个ZMK并按内置的格式打印两个密钥信封，信封上分别包括ZMK的左右两部分值，一级根据指定**SerilNo**生成的ZMK目标装入设备的检验值。功能如下：

- 第一成份为ZMK左半部分, 第二成份为ZMK右半部分。
- 设备序列号由用户自定义，长度限制为128位ASCII字符。
- 密钥检验值为ZMK密钥加密一组值为全0数据生成，取左4字节。

- 设备校验值生成过程：

  1. SerilNo通过Hash运算，得到SerilNo的Hash值。
  2. 将Hash值左右两部分进行异或。
  3. 使用ZMK加密异或结果，取左4字节作为设备检验值，ZMK加密异或结果时采用PBOC 2.0 数据加解密的填充方式。

请求参数

名称	类型	是否必须	描述
SerilNo	String	是	设备序列号，暂限制长度为1~128。
keyAlg	char	是	密钥算法标识： • *Z*：单倍长DES密钥 • *X*：双倍长DES密钥 • *Y*：三倍长DES密钥 • *R*：16字节SM4密钥 • *P*：16字节SM1密钥 • *L*：16字节AES密钥 • *M*：24字节AES密钥 • *N*：32字节AES密钥
hashAlgFlag	int	是	计算设备检验值时使用的HASH算法标识。 • *1*：SHA_1 • *2*：MD5 • *3*：ISO 10118_2 • *5*：SHA_224 • *6*：SHA_256 • *7*：SHA_384 • *8*：SHA_512 • *20*：SM3_256

返回参数

- 0号索引下：密钥在LMK下加密的密文
- 1号索引下：密钥校验值
- 2号索引下：设备检验值

异常处理

程序运行中出错则抛出异常。

cn.tass.exceptions.TAException

3.7 genWorkKeyEnc

生成工作密钥，并输出在ZMK下加密的密文。

```
public java.lang.String[] genWorkKeyEnc(java.lang.String keyType,
                                        char keyFlag,
                                        java.lang.Object ZMKKey,
                                        int storeKeyIndex,
                                        java.lang.String storeKeyLabel)
                                 throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
keyType	String	是	密钥类型，相应密钥类型的密钥类型号，支持密钥类型代码和密钥类型名称两种格式。以下左侧为密钥类型代码，右侧为密钥类型名称： • 000：ZMK/KEK • 001：ZPK • 002：PVK/TPK/TMK • 003：TAK • 008：ZAK • 009：BDK • 00A：ZEK/DEK • 00B：TEK • 011：KMC • 109：MK-AC/MDK • 10C：HMAC • 209：MK-SMI • 309：MK-SMC • 402：CVK • 409：MK-DAK • 509：MK-DN
keyFlag	char	是	密钥算法标识： • Z：单倍长DES密钥

名称	类型	是否必须	描述
			• *X*：双倍长DES密钥 • *Y*：三倍长DES密钥 • *R*：16字节SM4密钥 • *P*：16字节SM1密钥 • *L*：16字节AES密钥
ZMKKey	-	-	ZMK密钥索引或密文
storeKeyIndex	int	是	密钥存储索引，范围为：1~2048 **说明**： 如果设置的取值不在1~2048，则表示不存储。
storeKeyLabel	String	是	密钥存储标签，用于在密钥内部存储时标记密钥的标签说明，0-16个ASII字符。 **说明**： 只有**storeKeyIndex**取值为1~2048时生效。

返回参数

- 0：LMK加密的密钥密文
- 1：ZMK加密的密钥密文
- 2：新产生密钥的校验值

异常处理

程序运行中出错则抛出异常。

`cn.tass.exceptions.TAException`

3.8 genRandom

随机数据生成。

```
public java.lang.String genRandom(int length)
```

```
                    throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
length	int	是	待生成随机数的长度，范围为：1~2048

返回参数

生成的随机数。

异常处理

程序运行中出错则抛出异常。

cn.tass.exceptions.TAException：接口自定义的异常。

3.9 generalDataEnc

通用数据加密。

```
public byte[] generalDataEnc(int algType,
                             java.lang.String keyType,
                             java.lang.Object key,
                             java.lang.String disperFactor,
                             int sessionType,
                             java.lang.String sessionFactor,
                             int padFlag,
                             byte[] inData,
                             java.lang.String IV)
                      throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
algType	int	是	加密算法模式： • *0*：ECB模式加密 • *1*：CBC模式加密 • *2*：CFB模式加密 • *3*：OFB模式加密 • *4*：CRT模式加密 (16字节分组长度处理)
keyType	String	是	用于加密数据的源密钥类型，支持密钥类型代码和密钥类型名称两种格式。 以下左侧为密钥类型代码，右侧为密钥类型名称：

名称	类型	是否必须	描述
			• 000：KEK • 109：MDK • 309：MK-SMC • 00A：ZEK/DEK • 00B：TEK • 011：KMC
key	-	是	用户加密数据的密钥的索引或密文： • 传入参数数据类型为int型时，通过密钥索引调用密钥（取值范围：1~2048）。 • 传入参数类型为String时，按LMK加密的密钥密文处理。
disperFactor	int	是	密钥分散因子。n级分散因子进行串联，且每级分散因子必须为16个字节。
sessionType	int	是	会话密钥产生模式： • *0*：不产生会话密钥。 • *1*：ECB模式加密8字节会话密钥因子，得8字节会话密钥。 • *2*：ECB模式加密16字节会话密钥因子，得16字节会话密钥。 • *3*：密钥的左右8字节异或，得8字节会话密钥。 • *4*：取密钥的左8字节作为会话密钥。 • *5*：CBC模式加密16字节会话密钥因子，得16字节会话密钥。
sessionFactor	String	是	会话密钥因子： • **sessionType**为*1*时，该域为8字节（16H） • **sessionType**为*2*时，该域为16字节（32H） • **sessionType**为*5*时，该域为16字节（32H）
padFlag	int	是	PAD填充标识： • *0*：PBOC 2.0填充模式 • *1*：ISO/IEC 9797-1的 PADDING模式2

名称	类型	是否必须	描述
			• *2*：ISO/IEC 9797-1的 PADDING模式1 • *3*：ANSI X9.23 • *4*：PKCS#5 • *5*：NoPadding模式 • *10*：PBOC3.0 • *11*：左填充+ISO/IEC 9797-1
inData	byte[]	是	输入的明文数据
IV	String	否	初始向量，仅当**algType**取值为*0*、*1*、*2*、*3*、*4*时需存在该域： • 若密钥算法为128分组，该域为16 字节（32H） • 若密钥算法为64分组，该域为8字节（16H）

返回参数

加密之后的密文数据。

异常处理

程序运行中出错则抛出异常。

`cn.tass.exceptions.TAException`：接口自定义的异常。

3.10 dataEnc

数据加密。

```
public java.lang.String dataEnc(int algType,
                                 java.lang.String srcKeyType,
                                 java.lang.Object srcKey,
                                 java.lang.String disperFactor,
                                 int sessionType,
                                 java.lang.String sessionFactor,
                                 int padFlag,
                                 java.lang.String inData,
                                 java.lang.String IV)
                          throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
algType	int	是	加密算法模式：

名称	类型	是否必须	描述
			• *0*：ECB模式加密 • *1*：CBC模式加密 • *2*：CFB模式加密 • *3*：OFB模式加密
srcKeyType	String	是	源密钥类型。 • KEK • MDK • MK-SMC • ZEK/DEK • TEK • EDK
srcKey	-	是	用户加密数据的密钥的索引或密文。
disperFactor	int	是	密钥分散因子。 长度为n*16的HEX字符串（n*16H，n取值为0~8，每级分散因子长度为16H即8字节）。
sessionType	int	是	会话密钥产生模式： • *0*：不产生会话密钥 • *1*：ECB模式加密8字节会话密钥因子，得8字节会话密钥 • *2*：ECB模式加密16字节会话密钥因子，得16字节会话密钥 • *3*：密钥的左右8字节异或，得8字节会话密钥 • *4*：取密钥的左8字节作为会话密钥
sessionFactor	String	否	会话密钥因子：仅当**sessionType**为*1*或2时有效，该域通常为2字节ATC。
padFlag	int	是	PAD填充标识： • *0*：PBOC 2.0填充模式 • *1*：ISO/IEC 9797-1的 PADDING模式2 • *2*：ISO/IEC 9797-1的 PADDING模式1 • *3*：ANSI X9.23 • *4*：PKCS#5 • *5*：NoPadding模式

名称	类型	是否必须	描述
			• *10*：PBOC3.0 • *11*：左填充+ISO/IEC 9797-1
inData	byte[]	是	待加密的数据。
IV	String	否	初始向量，仅当**algType**取值为*0*、*1*、*2*、*3*时需存在该域： • 若密钥算法为128分组，该域为16 字节（32H） • 若密钥算法为64分组，该域为8字节（16H）

返回参数

加密之后的密文数据。

异常处理

程序运行中出错则抛出异常。

`cn.tass.exceptions.TAException`：接口自定义的异常。

3.11 dataDec

数据解密。

```
public java.lang.String dataDec(int algType,
                                 java.lang.String srcKeyType,
                                 java.lang.Object srcKey,
                                 java.lang.String disperFactor,
                                 int sessionType,
                                 java.lang.String sessionFactor,
                                 int padFlag,
                                 java.lang.String cipher,
                                 java.lang.String IV)
                          throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
algType	int	是	加密算法模式： • *0*：ECB模式加密 • *1*：CBC模式加密 • *2*：CFB模式加密

名称	类型	是否必须	描述
			• *3*：OFB模式加密
srcKeyType	String	是	用于解密数据的源密钥类型，支持类型包括： • MK-SMC • ZEK/DEK • TEK • EDK
srcKey	-	是	密钥的索引或密文。
disperFactor	int	是	密钥分散因子。n级分散因子进行串联，且每级分散因子必须为16个字节。
sessionType	int	是	会话密钥产生模式： • *0*：不产生会话密钥 • *1*：ECB模式加密8字节会话密钥因子，得8字节会话密钥 • *2*：ECB模式加密16字节会话密钥因子，得16字节会话密钥 • *3*：密钥的左右8字节异或，得8字节会话密钥 • *4*：取密钥的左8字节作为会话密钥
sessionFactor	String	否	会话密钥因子：仅当**sessionType**为*1*或*2*时有效，该域通常为2字节ATC。
padFlag	int	是	PAD填充标识： • *0*：PBOC 2.0填充模式 • *1*：ISO/IEC 9797-1的 PADDING模式2 • *2*：ISO/IEC 9797-1的 PADDING模式1 • *3*：ANSI X9.23 • *4*：PKCS#5 • *5*：NoPadding模式 • *10*：PBOC3.0 • *11*：左填充+ISO/IEC 9797-1
cipher	String	是	待解密的数据密文。
IV	String	否	初始向量，仅当**algType**取值为*0*、*1*、*2*、*3*时需存在该域：

名称	类型	是否必须	描述
			• 若密钥算法为128分组，该域为16 字节（32H） • 若密钥算法为64分组，该域为8字节（16H）

返回参数

解密后的明文。

异常处理

程序运行中出错则抛出异常。

cn.tass.exceptions.TAException：接口自定义的异常。

3.12 generalDataDec

通用数据解密。

```
public byte[] generalDataDec(int algType,
                             java.lang.String keyType,
                             java.lang.Object key,
                             java.lang.String disperFactor,
                             int sessionType,
                             java.lang.String sessionFactor,
                             int padFlag,
                             byte[] inData,
                             java.lang.String IV)
                      throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
algType	int	是	加密算法模式： • *0*：ECB模式加密 • *1*：CBC模式加密 • *2*：CFB模式加密 • *3*：OFB模式加密 • *4*：CRT模式加密 (16字节分组长度处理)
keyType	String	是	密钥类型：*MK-SMC*、*ZEK*、*DEK*、*TEK*。
key	-	是	用户加密数据的密钥的索引或密文。
disperFactor	int	是	密钥分散因子。n*32H（n取值为0-8，每级分散因子长度为32H即是16字节）。

名称	类型	是否必须	描述
sessionType	int	是	会话密钥产生模式： • *0*：不产生会话密钥 • *1*：ECB模式加密8字节会话密钥因子，得8字节会话密钥 • *2*：ECB模式加密16字节会话密钥因子，得16字节会话密钥 • *3*：密钥的左右8字节异或，得8字节会话密钥 • *4*：取密钥的左8字节作为会话密钥 • *5*：CBC模式加密16字节会话密钥因子，得16字节会话密钥
sessionFactor	String	是	会话密钥因子： • **sessionType**为*1*时，该域为8字节（16H） • **sessionType**为*2*时，该域为16字节（32H） • **sessionType**为*5*时，该域为16字节（32H）
padFlag	int	是	PAD填充标识，取值如下： • *0*：PBOC 2.0填充模式 • *1*：ISO/IEC 9797-1的PADDING模式2 • *2*：ISO/IEC 9797-1的PADDING模式1 • *3*：ANSI X9.23 • *4*：PKCS#5 • *5*：NoPadding模式 • *10*：PBOC3.0 • *11*：左填充+ISO/IEC 9797-1
inData	byte[]	是	待解密的数据。
IV	String	否	初始向量，仅当**algType**取值为*0*、*1*、*2*、*3*时需存在该域： • 若密钥算法为128分组，该域为16 字节（32H） • 若密钥算法为64分组，该域为8字节（16H）

返回参数

解密之后的数据。

异常处理

程序运行中出错则抛出异常。

cn.tass.exceptions.TAException：程序运行异常。

3.13 calMAC

生成数据MAC。

```
public java.lang.String calMAC(int algType,
                               int valueType,
                               java.lang.String srcKeyType,
                               java.lang.Object srcKey,
                               java.lang.String disperFactor,
                               int sessionType,
                               java.lang.String sessionFactor,
                               int padFlag,
                               java.lang.String inData,
                               java.lang.String IV)
                        throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
algType	int	是	MAC算法模式： • *1*：ISO9797-1 MAC 算法模式1 • *3*：ISO9797-1 MAC 算法模式3，限密钥标识为X/U算法
valueType	int	是	MAC取值方式，按前个域模式产生的密文值输出下述结果作为MAC： • 1~8：输出密文值的左n字节 • 10：输出16字节MAC，限密钥标识为P/L/R • 11~18：输出密文值的右n字节 • 21~28：左右异或后取左n字节输出 • 31~38：左右异或后取右n字节输出 • 44：四字节异或，最后输出4字节
srcKeyType	String	是	源密钥类型，支持密钥类型代码和密钥类型名称两种格式。支持类型如下：

名称	类型	是否必须	描述
			• *MDK* • *MK-SMI* • *KEK* • *KMC* • *ZAK* • *TAK*
srcKey	-	是	密钥的索引或密文。
disperFactor	int	是	密钥分散因子。n级分散因子进行串联，且每级分散因子必须为16个字节。
sessionType	int	是	会话密钥产生模式： • *0*：不产生会话密钥。 • *1*：ECB模式加密8字节会话密钥因子，得8字节会话密钥。 • *2*：ECB模式加密16字节会话密钥因子，得16字节会话密钥。 • *3*：密钥的左右8字节异或，得8字节会话密钥。 • *4*：取密钥的左8字节作为会话密钥。
sessionFactor	String	否	会话密钥因子，仅当**sessionType**为*1*或2时有效，该域通常为2字节ATC。
padFlag	int	是	PAD填充标识，取值如下： • *0*：PBOC 2.0填充模式 • *1*：ISO/IEC 9797-1的PADDING模式2 • *2*：ISO/IEC 9797-1的PADDING模式1 • *3*：ANSI X9.23 • *4*：PKCS#5 • *5*：NoPadding模式 • *10*：PBOC3.0 • *11*：左填充+ISO/IEC 9797-1
inData	String	是	需要计算MAC的数据。
IV	String	否	初始向量。 • 若密钥算法为128分组（密钥标识 P/L/R），该域为16 字节（32H）。

名称	类型	是否必须	描述
			• 若密钥算法为64分组，该域为8字节（16H）。

返回参数

数据MAC值。

异常处理

程序运行中出错则抛出异常。

cn.tass.exceptions.TAException：接口自定义异常。

3.14 verifyMAC

验证交易数据MAC/TAC。

```
public boolean verifyMAC(int algType,
                         int valueType,
                         java.lang.String srcKeyType,
                         java.lang.Object srcKey,
                         java.lang.String disperFactor,
                         int sessionType,
                         java.lang.String sessionFactor,
                         int padFlag,
                         java.lang.String inData,
                         java.lang.String IV,
                         java.lang.String MAC)
                  throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
algType	int	是	MAC算法模式： • *1*：ISO9797-1 MAC 算法模式1 • *3*：ISO9797-1 MAC 算法模式3，限密钥标识为X/U算法
valueType	int	是	MAC取值方式： • 1~8：输出密文值的左n字节 • 10：输出16字节MAC，限密钥标识为P/L/R • 11~18：输出密文值的右n字节 • 21~28：左右异或后取左n字节输出 • 31~38：左右异或后取右n字节输出

名称	类型	是否必须	描述
			• 44：四字节异或，最后输出4字节
srcKeyType	String	是	源密钥类型，支持密钥类型代码和密钥类型名称两种格式。支持类型如下： • *MDK* • *MK-SMI* • *KEK* • *KMC* • *ZAK* • *TAK*
srcKey	-	是	密钥的索引或密文。
disperFactor	int	是	密钥分散因子。n级分散因子进行串联，且每级分散因子必须为16个字节。
sessionType	int	是	会话密钥产生模式： • *0*：不产生会话密钥 • *1*：ECB模式加密8字节会话密钥因子，得8字节会话密钥 • *2*：ECB模式加密16字节会话密钥因子，得16字节会话密钥 • *3*：密钥的左右8字节异或，得8字节会话密钥 • *4*：取密钥的左8字节作为会话密钥
sessionFactor	String	否	会话密钥因子，仅当**sessionType**为*1*或*2*时有效，该域通常为2字节ATC。
padFlag	int	是	PAD填充标识，取值如下： • *0*：PBOC 2.0填充模式 • *1*：ISO/IEC 9797-1的PADDING模式2 • *2*：ISO/IEC 9797-1的PADDING模式1 • *3*：ANSI X9.23 • *4*：PKCS#5 • *5*：NoPadding模式 • *10*：PBOC3.0 • *11*：左填充+ISO/IEC 9797-1
inData	String	是	需要计算MAC的数据。

名称	类型	是否必须	描述
IV	String	否	初始向量。 128 位分组（密钥标识 P/L/R/M/N）时，该域 16 字节（32H）；否则该域为 8 字节（16H）

返回值

- 验证成功返回true。
- 验证失败返回false。

异常处理

程序运行中出错则抛出异常。

cn.tass.exceptions.TAException：接口自定义的异常。

3.15 generalCalMAC（String）

通用生成数据MAC。

```
public java.lang.String generalCalMAC(int algType,
                                       int valueType,
                                       java.lang.String srcKeyType,
                                       java.lang.Object srcKey,
                                       java.lang.String disperFactor,
                                       int sessionType,
                                       java.lang.String sessionFactor,
                                       int padFlag,
                                       java.lang.String inData,
                                       java.lang.String IV)
                                 throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
algType	int	是	MAC算法模式： • *1*：ISO9797-1 MAC 算法模式1 • *3*：ISO9797-1 MAC 算法模式3，限密钥标识为X/U算法
valueType	int	是	MAC取值方式： • *1~8*：输出密文值的左n字节 • *10*：输出16字节MAC，限密钥标识为P/L/R

名称	类型	是否必须	描述
			• *11~18*：输出密文值的右n字节 • *21~28*：左右异或后取左n字节输出 • *31~38*：左右异或后取右n字节输出 • *44*：四字节异或，最后输出4字节
srcKeyType	String	是	源密钥类型，支持密钥类型代码和密钥类型名称两种格式。支持类型如下： • *MDK* • *MK-SMI* • *KEK* • *KMC* • *ZAK* • *TAK*
srcKey	-	是	密钥的索引或密文。 索引取值范围：1~2048。
disperFactor	int	是	密钥分散因子。n级分散因子进行串联，且每级分散因子必须为16个字节。
sessionType	int	是	会话密钥产生模式： • *0*：不产生会话密钥。 • *1*：ECB模式加密8字节会话密钥因子，得8字节会话密钥。 • *2*：ECB模式加密16字节会话密钥因子，得16字节会话密钥。 • *3*：密钥的左右8字节异或，得8字节会话密钥。 • *4*：取密钥的左8字节作为会话密钥。 • *5*：CBC模式加密16字节会话密钥因子，得16字节会话密钥。
sessionFactor	String	否	会话密钥因子，仅当**sessionType**为以下取值时有效： • sessionType为*1*时，该域为8字节（16H）。 • sessionType为2时，该域为16字节（32H）。

名称	类型	是否必须	描述
			• sessionType为5时，该域为16字节（32H）。
padFlag	int	是	PAD填充标识，取值如下： • *0*：PBOC 2.0填充模式 • *1*：ISO/IEC 9797-1的PADDING模式2 • *2*：ISO/IEC 9797-1的PADDING模式1 • *3*：ANSI X9.23 • *4*：PKCS#5 • *5*：NoPadding模式 • *10*：PBOC3.0 • *11*：左填充+ISO/IEC 9797-1
inData	String	是	需要计算MAC的数据长度（0~1968字节数）。
IV	String	否	初始向量。 128 位分组（密钥标识 P/L/R/M/N）时，该域 16 字节（32H）；否则该域为 8 字节（16H）

返回值

计算出的MAC值。

异常处理

程序运行中出错则抛出异常。

`cn.tass.exceptions.TAException`：接口自定义的异常。

3.16 generalCalMAC（byte[]）

通用生成数据MAC。

```
public byte[] generalCalMAC(int algType,
                            int valueType,
                            java.lang.String srcKeyType,
                            java.lang.Object srcKey,
                            java.lang.String disperFactor,
                            int sessionType,
                            java.lang.String sessionFactor,
                            int padFlag,
                            byte[] inData,
                            java.lang.String IV)
```

```
                throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
algType	int	是	MAC算法模式： • *1*：ISO9797-1 MAC 算法模式1 • *3*：ISO9797-1 MAC 算法模式3，限密钥标识为X/U算法
valueType	int	是	MAC取值方式： • *1~8*：输出密文值的左n字节。 • *10*：输出16字节MAC，限密钥标识为P/L/R。 • *11~18*：输出密文值的右n字节。 • *21~28*：左右异或后取左n字节输出。 • *31~38*：左右异或后取右n字节输出。 • *44*：四字节异或，最后输出4字节。
srcKeyType	String	是	源密钥类型，支持密钥类型代码和密钥类型名称两种格式。支持类型如下： • *MDK* • *MK-SMI* • *KEK* • *KMC* • *ZAK* • *TAK*
srcKey	-	是	密钥的索引或密文。
disperFactor	int	是	密钥分散因子。n级分散因子进行串联，且每级分散因子必须为16个字节。
sessionType	int	是	会话密钥产生模式： • *0*：不产生会话密钥。 • *1*：ECB模式加密8字节会话密钥因子，得8字节会话密钥。 • *2*：ECB模式加密16字节会话密钥因子，得16字节会话密钥。 • *3*：密钥的左右8字节异或，得8字节会话密钥。

名称	类型	是否必须	描述
			• *4*：取密钥的左8字节作为会话密钥。 • *5*：CBC模式加密16字节会话密钥因子，得16字节会话密钥。
sessionFactor	String	否	会话密钥因子，仅当**sessionType**为以下取值时有效： • **sessionType**为*1*时，该域为8字节（16H）。 • **sessionType**为*2*时，该域为16字节（32H）。 • **sessionType**为*5*时，该域为16字节（32H）。
padFlag	int	是	PAD填充标识，取值如下： • *0*：PBOC 2.0填充模式 • *1*：ISO/IEC 9797-1的PADDING模式2 • *2*：ISO/IEC 9797-1的PADDING模式1 • *3*：ANSI X9.23 • *4*：PKCS#5 • *5*：NoPadding模式 • *10*：PBOC3.0 • *11*：左填充+ISO/IEC 9797-1
inData	String	是	需要计算MAC的数据。
IV	String	否	初始向量。 128 位分组（密钥标识 P/L/R/M/N）时，该域 16 字节（32H）；否则该域为 8 字节（16H）

返回值

计算出的MAC值。

异常处理

程序运行中出错则抛出异常。

cn.tass.exceptions.TAException：接口自定义的异常。

3.17 generalVerifyMAC

校验数据MAC。

```
public boolean generalVerifyMAC(int algType,
                                int valueType,
                                java.lang.String srcKeyType,
                                java.lang.Object srcKey,
                                java.lang.String disperFactor,
                                int sessionType,
                                java.lang.String sessionFactor,
                                int padFlag,
                                java.lang.String inData,
                                java.lang.String IV,
                                java.lang.String desMac)
                         throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
algType	int	是	MAC算法模式： • *1*：ISO9797-1 MAC 算法模式1 • *3*：ISO9797-1 MAC 算法模式3，限密钥标识为X/U算法
valueType	int	是	MAC取值方式： • *1~8*：输出密文值的左n字节。 • *10*：输出16字节MAC，限密钥标识为P/L/R。 • *11~18*：输出密文值的右n字节。 • *21~28*：左右异或后取左n字节输出。 • *31~38*：左右异或后取右n字节输出。 • *44*：四字节异或，最后输出4字节。
srcKeyType	String	是	源密钥类型，支持密钥类型代码和密钥类型名称两种格式。支持类型如下： • *MDK* • *MK-SMI* • *KEK* • *KMC* • *ZAK* • *TAK*
srcKey	-	是	密钥的索引或密文。

名称	类型	是否必须	描述
disperFactor	int	是	密钥分散因子。n级分散因子进行串联，且每级分散因子必须为16个字节。
sessionType	int	是	会话密钥产生模式： • *0*：不产生会话密钥。 • *1*：ECB模式加密8字节会话密钥因子，得8字节会话密钥。 • *2*：ECB模式加密16字节会话密钥因子，得16字节会话密钥。 • *3*：密钥的左右8字节异或，得8字节会话密钥。 • *4*：取密钥的左8字节作为会话密钥。 • *5*：CBC模式加密16字节会话密钥因子，得16字节会话密钥。
sessionFactor	String	否	会话密钥因子，仅当**sessionType**为以下取值时有效： • **sessionType**为*1*时，该域为8字节（16H）。 • **sessionType**为*2*时，该域为16字节（32H）。 • **sessionType**为*5*时，该域为16字节（32H）。
padFlag	int	是	PAD填充标识，取值如下： • *0*：PBOC 2.0填充模式 • *1*：ISO/IEC 9797-1的PADDING模式2 • *2*：ISO/IEC 9797-1的PADDING模式1 • *3*：ANSI X9.23 • *4*：PKCS#5 • *5*：NoPadding模式 • *10*：PBOC3.0 • *11*：左填充+ISO/IEC 9797-1
inData	String	是	输入数据，直接对0~1968字节输入数据进行加密运算。
IV	String	否	初始向量。

名称	类型	是否必须	描述
			128 位分组（密钥标识 P/L/R/M/N）时，该域 16 字节（32H）；否则该域为 8 字节（16H）

返回值

- 验证成功返回true。
- 验证失败返回false。

异常处理

程序运行中出错则抛出异常。

cn.tass.exceptions.TAException：接口自定义的异常。

3.18 verifyCipherAndGenARPC

PBOC验证ARQC并产生ARPC。

```
public java.lang.String[] verifyCipherAndGenARPC(
                                java.lang.Object MDKSrcKey,
                                java.lang.String serialNum,
                                java.lang.String ATC,
                                java.lang.String transactionData,
                                java.lang.String ARQC,
                                java.lang.String ARC)
                          throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
MDKSrcKey	-	是	MDK源密钥索引或密文，用于运算的MK-AC/MDK源密钥索引或LMK下加密的密文。
serialNum	String	是	PAN或PAN序列号，用于分散MDK产生卡片UDK的分散因子。账号+账号应用序列号取最右16个数字，若小于16个则后对齐左补0。
ATC	String	是	应用交易计数器。2字节。
transactionData	String	是	明文交易数据。
ARQC	String	是	待验证的ARQC/TC/AAC，或用于产生ARPC。
ARC	String	是	用于产生ARPC。

返回值

- String[0]：ARPC。
- String[1]：ARQC验证失败时存在，输出密码机运算的ARQC。

异常处理

程序运行中出错则抛出异常。

cn.tass.exceptions.TAException：接口自定义的异常。

3.19 encryptScript

PBOC脚本加密。

```
public java.lang.String encryptScript(java.lang.Object MDKSrcKey,
                                      java.lang.String serialNum,
                                      java.lang.String ATC,
                                      java.lang.String data)
                               throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
MDKSrcKey	-	是	MDK源密钥索引或密文，用于运算的MK-AC/MDK源密钥索引或LMK下加密的密文。
serialNum	String	是	PAN或PAN序列号，用于分散MDK产生卡片UDK的分散因子。账号+账号应用序列号取最右16个数字，若小于16个则后对齐左补0。
ATC	String	是	应用交易计数器。 长度：2字节
data	String	是	输入数据。 数据长度：0~984 字节

返回值

加密后的数据密文。

异常处理

程序运行中出错则抛出异常。

cn.tass.exceptions.TAException：接口自定义的异常。

3.20 calScriptMAC

计算脚本MAC。

```
public java.lang.String calScriptMAC(java.lang.Object MDKSrcKey,
                                      java.lang.String serialNum,
                                      java.lang.String ATC,
                                      java.lang.String data)
                               throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
MDKSrcKey	-	是	MDK源密钥索引或密文，用于运算的MK-AC/MDK源密钥索引或LMK下加密的密文。
serialNum	String	是	PAN或PAN序列号，用于分散MDK产生卡片UDK的分散因子。账号+账号应用序列号取最右16个数字，若小于16个则后对齐左补0。
ATC	String	是	应用交易计数器。 长度：2字节
data	String	是	输入数据。 数据长度：0~984 字节

返回值

脚本MAC数据。

异常处理

程序运行中出错则抛出异常。

cn.tass.exceptions.TAException：接口自定义的异常。

3.21 verifyARQC

验证ARQC。

```
public boolean verifyARQC(java.lang.Object MDKSrcKey,
                          java.lang.String serialNum,
                          java.lang.String ATC,
                          java.lang.String transactionData,
                          java.lang.String ARQC)
```

```
                throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
MDKSrcKey	-	是	MDK源密钥索引或密文，用于运算的MK-AC/MDK源密钥索引或LMK下加密的密文。
serialNum	String	是	PAN或PAN序列号，用于分散MDK产生卡片UDK的分散因子。账号+账号应用序列号取最右16个数字，若小于16个则后对齐左补0。
ATC	String	是	应用交易计数器，用于产生交易会话密钥。长度：2字节
transactionData	String	是	交易数据。
ARQC	String	是	待验证的ARQC。

返回值

- 验证成功返回true。
- 验证失败返回false。

异常处理

程序运行中出错则抛出异常。

cn.tass.exceptions.TAException：接口自定义的异常。

3.22 genARPC

产生ARPC。

```
public java.lang.String genARPC(java.lang.Object MDKSrcKey,
                                 java.lang.String serialNum,
                                 java.lang.String ATC,
                                 java.lang.String ARQC,
                                 java.lang.String ARC)
                          throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
MDKSrcKey	-	是	MDK源密钥索引或密文，用于运算的MK-AC/MDK源密钥索引或LMK下加密的密文。

名称	类型	是否必须	描述
serialNum	String	是	PAN或PAN序列号，用于分散MDK产生卡片UDK的分散因子。账号+账号应用序列号取最右16个数字，若小于16个则后对齐左补0。
ATC	String	是	应用交易计数器。 长度：2字节
ARQC	String	是	交易密文数据：ARQC、TC、AAC等。
ARC	String	是	用于产生ARPC。

返回值

产生的ARPC。

异常处理

程序运行中出错则抛出异常。

`cn.tass.exceptions.TAException`：接口自定义的异常。

3.23 exportWorkKey

LMK加密的密钥转换成ZMK下加密导出。

```
public java.lang.String[] exportWorkKey(
                                java.lang.String targetKeyType,
                                java.lang.Object zmkKey,
                                java.lang.Object targetKey,
                                char targetKeyFlag)
                         throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
targetKeyType	String	是	待导出密钥的密钥类型，支持密钥类型代码和密钥类型名称两种格式。 以下左侧为密钥类型代码，右侧为密钥类型名称： • 000：ZMK/KEK • 001：ZPK • 002：PVK/TPK/TMK • 003：TAK

名称	类型	是否必须	描述
			• 008：ZAK • 009：BDK • 109：MK-AC/MDK • 209：MK-SMI • 309：MK-SMC • 402：CVK • 409：MK-DAK • 509：MK-DN • 00A：ZEK/DEK • 00B：TEK • 10C：HMAC • 011：KMC
zmkKey	-	是	ZMK密钥索引或密文。
targetKey	-	是	待导出密钥的索引或LMK加密的密钥密文。
targetKeyFlag	char	是	在ZMK下加密的密钥标识，取值如下： • *Z*：单倍长DES密钥 • *X*：双倍长DES密钥 • *Y*：三倍长DES密钥 • *P*：16字节SM1密钥 • *L*：16字节AES密钥 • *R*：16字节SM4密钥

返回值

- [0]：ZMK下加密的密钥密文。
- [1]：密钥校验值。

异常处理

程序运行中出错则抛出异常。

cn.tass.exceptions.TAException

3.24 importWorkKey

导入工作密钥，导入ZMK加密的密钥，即ZMK加密的密钥转换为LMK下加密。

```
public java.lang.String[] importWorkKey(
                          java.lang.String importedKeyType,
```

```
                    java.lang.Object zmkKey,
                    java.lang.String importKeyCipherByZmk,
                    char importedKeyAlgFlag,
                    int storeKeyIndex,
                    java.lang.String storeKeyLabel)
                throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
targetKeyType	String	是	待导出密钥的密钥类型，支持密钥类型代码和密钥类型名称两种格式。 以下左侧为密钥类型代码，右侧为密钥类型名称： • 000：ZMK/KEK • 001：ZPK • 002：PVK/TPK/TMK • 003：TAK • 008：ZAK • 009：BDK • 109：MK-AC/MDK • 209：MK-SMI • 309：MK-SMC • 402：CVK • 409：MK-DAK • 509：MK-DN • 00A：ZEK/DEK • 00B：TEK • 10C：HMAC • 011：KMC
zmkKey	-	是	ZMK密钥索引或密文。
importKeyCipherByZmk	String	是	ZMK下加密的密钥密文。
importedKeyAlgFlag	char	是	在LMK下加密的密钥密文标识，取值如下： • *Z* • *X* • *Y* • *U* • *T*

名称	类型	是否必须	描述
			• *P* • *L* • *R*
storeKeyIndex	int	是	密钥存储索引，取值范围：1~2048。 **说明**： 其他值表示不存储
storeKeyLabel	String	否	密钥存储标签，用于在密钥内部存储时标记密钥的标签说明。 长度：0~16个ASCII字符 **说明**： 仅当**storeKeyIndex**取值为*1~2048*时生效。

返回值

- [0]：LMK下加密的密钥密文。
- [1]：密钥校验值。

异常处理

程序运行中出错则抛出异常。

cn.tass.exceptions.TAException

3.25 loadPrintFormat

将自定义的打印格式数据装在到HSM中。

```
public boolean loadPrintFormat(java.lang.String data)
                        throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
data	String	是	打印格式符号表中定义的符号和常量。

返回值

- 成功返回true。
- 失败返回false。

异常处理

程序运行中出错则抛出异常。

`cn.tass.exceptions.TAException`：接口自定义的异常。

3.26 printKeyElement

随机产生一个密钥成份，通过连接于密码机的打印机打印出明文，并返回成份的密文。需满足主机服务的信函打印权限。

```
public java.lang.String[] printKeyElement(
                          java.lang.String keyType,
                          char keyAlgFlag,
                          java.lang.String... printFields)
                   throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
keyType	String	是	密钥类型，支持密钥类型代码和密钥类型名称两种格式。 以下左侧为密钥类型代码，右侧为密钥类型名称： • 000：ZMK • 001：ZPK • 002：PVK/TPK/TMK • 003：TAK • 008：ZAK • 109：MDK • 402：CVK • 00A：ZEK
keyAlgFlag	char	是	在LMK下加密的密钥密文标识，取值如下： • *Z* • *X* • *Y* • *U*

名称	类型	是否必须	描述
			• *T*
printFields	String	是	打印域0,1,2,3... **说明**： 不能包含分号（；）字符。

返回值

- [0]：密钥密文。
- [1]：密钥校验值。

异常处理

程序运行中出错则抛出异常。

cn.tass.exceptions.TAException：接口自定义的异常。

3.27 diverAndGenNewKey

分散产生新密钥，可选的存储到加密机内。

```
public java.lang.String[] diverAndGenNewKey(
                                java.lang.String srcKeyType,
                                java.lang.Object srcKey,
                                java.lang.String subKeyType,
                                char subKeyAlgFlag,
                                int disperAlgType,
                                java.lang.String disperFactor,
                                java.lang.String IV,
                                int storeKeyIndex,
                                java.lang.String storeKeyLabel)
                            throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
srcKeyType	String	是	源密钥类型，支持密钥类型代码和密钥类型名称两种格式。 以下左侧为密钥类型代码，右侧为密钥类型名称： • 000：ZMK/KEK • 002：PVK/TPK/TMK • 007：EDK

名称	类型	是否必须	描述
			• 008：ZAK • 109：MK-AC/MDK • 209：MK-SMI • 309：MK-SMC • 409：MK-DAK • 509：MK-DN • 00A：ZEK/DEK • 011：KMC
srcKey	-	是	源密钥索引或密文。
subKeyType	String	是	子密钥类型，支持密钥类型代码和密钥类型名称两种格式。 以下左侧为密钥类型代码，右侧为密钥类型名称： • 000：ZMK/KEK • 008：ZAK • 109：MK-AC/MDK • 209：MK-SMI • 309：MK-SMC • 409：MK-DAK • 509：MK-DN • 00A：ZEK/DEK • 011：KMC
subKeyAlgFlag	char	是	子密钥标识，取值如下： • *X* • *U* • *P* • *R* • *L* • *N*
disperAlgType	String	是	分散算法模式，取值如下： • *0*：PBOC子密钥分散算法，8字节分散因子D，使用源密钥对16字节[D||D的非]采用源密钥的算法标识进行ECB模式加密。

名称	类型	是否必须	描述
			• *1*：ECB模式加密16字节分散因子。 • *2*：ECB模式加密16字节分散因子，并复制扩展为32字节长度密钥。 **说明**： 仅限**subKeyFlag**为*N*。 • *3*：CBC模式加密16字节分散因子。 • *4*：ECB模式加密分散因子，分散因子必须为8字节的倍数，且至少16字节。截取加密结果的前后各8字节作为子密钥。 **说明**： 仅限**subKeyFlag**为*X*或者*U*。 • *5*：CBC模式加密分散因子，分散因子必须为16字节的倍数。截取加密结果的最后16字节作为子密钥。 **说明**： 仅限**subKeyFlag**为*L*。
disperFactor	String	是	分散因子，长度为n*m*2的HEX字符串。 n为分散级数，取值为1~8。 m*2H为每级分散因子长度，m取值如下： • 当**disperAlgType**为*0*时，m为8H。 • 当**disperAlgType**为*1*或*2*或*3*时，m为16H。 • 当**disperAlgType**为*4*时，分散因子为n*16H。
IV	String	是	初始向量，当**disperAlgType**为*5*时有效。
storeKeyIndex	int	是	密钥存储索引，取值范围：1~2048。 **说明**： 其他值表示不存储。
storeKeyLabel	String	否	密钥存储标签，长度为0~16。 当**storeKeyIndex**取值在1~2048范围内时生效。

返回值

- [0]：密钥密文。
- [1]：密钥校验值。

异常处理

程序运行中出错则抛出异常。

`cn.tass.exceptions.TAException`：接口自定义的异常。

3.28 genCVV

卡校验值的计算，VISA CVV的产生和验证。

```
public java.lang.String genCVV(java.lang.Object CVK,
                               java.lang.String mainAccount,
                               int deadTime,
                               int serviceCode)
                        throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
CVK	String	是	CVK密钥索引或LMK下加密的VISA CVK A/B。
mainAccount	int	是	卡的主账号。
deadTime	int	是	卡的过期时间。例如： 9301表示93年一月
serviceCode	int	是	卡的服务码。

返回值

产生的CVV。

异常处理

程序运行中出错则抛出异常。

`cn.tass.exceptions.TAException`：接口自定义的异常。

3.29 verifyCVV

校验VISA CVV。

```
public boolean verifyCVV(java.lang.Object CVK,
```

```
                    java.lang.String CVV,
                    java.lang.String mainAccount,
                    int validity,
                    int serviceCode)
                throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
CVK	-	是	CVK密钥索引或LMK下加密的VISA CVK A/B。
CVV	String	是	待校验的CVV。
mainAccount	String	是	卡的主账号。
validity	int	是	卡的过期时间。例如： 9301表示93年一月
serviceCode	int	是	卡的服务码。

返回值

- 验证成功返回true。
- 验证失败返回false。

异常处理

程序运行中出错则抛出异常。

cn.tass.exceptions.TAException：接口自定义的异常。

3.30 genRandPIN

产生一个随机PIN码。

```
public java.lang.String genRandPIN(java.lang.String account,
                                   int pinLen)
                            throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
account	String	是	用户主账号有效位的最右12个数字。
pinLen	int	是	PIN长度，取值范围为4~12。如果取值小于4或大于12，则默认长度为4。

返回值

LMK加密的PIN密文。

异常处理

程序运行中出错则抛出异常。

cn.tass.exceptions.TAException：接口自定义的异常。

3.31 encPINByLmk

LMK加密一个明文PIN。

```
public java.lang.String encPINByLmk(java.lang.String pinPlainText,
                                    java.lang.String account)
                             throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
pinPlainText	String	是	PIN明文，PIN明文左对齐，右边填充一个字符（F）。
account	String	是	用户主账号有效位的最右12个数字。

返回值

LMK加密的PIN密文。

异常处理

程序运行中出错则抛出异常。

cn.tass.exceptions.TAException：接口自定义的异常。

3.32 encPinByZpk

使用ZPK加密明文PIN数据。

```
public java.lang.String encPinByZpk(java.lang.Object ZPKKey,
                                    int PINBLOCKForm,
                                    java.lang.String account,
                                    java.lang.String pin)
```

```
                    throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
ZPKKey	-	是	ZPK密钥索引或密文。
PINBLOCKFo rm	int	是	PINBLOCK格式取值。
account	String	是	用户主账号，传空时内部自动设为全0。
pin	String	是	明文PIN字符串。

返回值

ZPK加密的PIN密文。

异常处理

程序运行中出错则抛出异常。

cn.tass.exceptions.TAException。

3.33 decPinByZpk

解密ZPK加密的PIN密文数据。

```
public java.lang.String decPinByZpk(
                        java.lang.Object srcZPKKey,
                        java.lang.String srcPINBLOCKCipher,
                        int PINBLOCKForm,
                        java.lang.String account)
                    throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
srcZPKKey	-	是	用于解密PIN的密钥索引或密文。
srcPINBLOC KCipher	String	是	ZPK加密的PIN密文数据。
PINBLOCKFo rm	int	是	数字PINBLOCK格式取值。
account	String	是	用户主账号。

返回值

解密后的明文PIN数据。

异常处理

程序运行中出错则抛出异常。

cn.tass.exceptions.TAException。

3.34 weakPinSet

设置密码机设备中弱口令集合属性（同时设置应用下配置的所有密码机的属性）。

```
public boolean weakPinSet(
                    java.util.ArrayList<java.lang.String> weakPinSet)
                  throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
weakPinSet	ArrayList	是	弱口令集合数据：N个等长的弱口令串连。

返回值

- 成功时返回true。
- 失败（或部分设备设置失败）时返回false。

异常处理

程序运行中出错则抛出异常。

cn.tass.exceptions.TAException。

3.35 weakPinCheck

弱口令检查。

```
public boolean weakPinCheck(int keyType,
                            java.lang.Object key,
                            int pinblockType,
                            java.lang.String pinCipher,
                            java.lang.String pan,
                            java.util.ArrayList<java.lang.String> addWeakPinSet)
```

```
                    throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
keyType	int	是	密钥类型。
key	-	是	密钥索引或密文。
pinblockType	int	是	PINBLOCK格式。
pinCipher	String	是	PIN密文。
pan	String	是	用户主账号。
addWeakPinSet	ArrayList	是	弱口令集合数据：N个等长的弱口令串连。

返回值

- 检查成功时返回true。
- 检查失败时返回false。

异常处理

程序运行中出错则抛出异常。

cn.tass.exceptions.TAException。

3.36 transferPinLMK2ZPK

LMK加密的PIN密文转为ZPK加密。

```
public java.lang.String transferPinLMK2ZPK(java.lang.Object ZPKKey,
                                           int PINBLOCKForm,
                                           java.lang.String account,
                                           java.lang.String PINCipher)
                    throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
ZPKKey	-	是	目标ZPK密钥索引或密文。
PINBLOCKForm	int	是	PINBLOCK格式。
account	String	是	用户主账号。

名称	类型	是否必须	描述
			• 当**PINBLOCKForm**为*04*时，该域为18N，去除校验位的18位主账号，不足18位则右对齐左填字符（F）。 • 当**PINBLOCKForm**为其他值时，该域为12N，去除校验位的最右12位主账号。
PINCipher	String	是	LMK下加密的PIN密文。

返回值

目标ZPK下加密的PINBLOCK密文。

异常处理

程序运行中出错则抛出异常。

cn.tass.exceptions.TAException：接口自定义的异常。

3.37 transferPinZPK12ZPK2

将PIN由ZPK1加密转换为ZPK2加密。

```
public java.lang.String transferPinZPK12ZPK2(
                        java.lang.Object ZPK1,
                        java.lang.Object ZPK2,
                        java.lang.String srcPINBLOCKCipher,
                        int srcPINBLOCKForm,
                        int dstPINBLOCKForm,
                        java.lang.String account)
                    throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
ZPK1	-	是	源ZPK密钥索引或密文。
ZPK2	-	是	目标ZPK密钥索引或密文。
srcPINBLOCKCipher	String	是	源PINBLOCK密文，源ZPK密钥方案为： • R/P/L：32H • 其它：16H
srcPINBLOCKForm	int	是	源PINBLOCK格式。

名称	类型	是否必须	描述
dstPINBLOCKForm	int	是	目标PINBLOCK格式
account	String	是	用户主账号。 • 当**dstPINBLOCKForm**为*04*时，该域为18N，去除校验位的18位主账号，不足18位则右对齐左填字符（F）。 • 当**dstPINBLOCKForm**为其他值时，该域为12N，去除校验位的最右12位主账号。

返回值

ZPK2加密的PINBLOCK密文。

异常处理

程序运行中出错则抛出异常。

`cn.tass.exceptions.TAException`：接口自定义的异常。

3.38 transferPinZPK2LMK

将PIN由ZPK加密转换为LMK加密。

```
public java.lang.String transferPinZPK2LMK(
                        java.lang.Object srcZPKKey,
                        java.lang.String srcPINBLOCKCipher,
                        int PINBLOCKForm,
                        java.lang.String account)
                    throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
srcZPKKey	-	是	用于加密PIN的ZPK密钥索引或密文。
srcPINBLOCKCipher	String	是	在源ZPK下加密的PINBLOCK密文，源ZPK密钥方案为： • R/P/L：32H • 其它：16H
PINBLOCKForm	int	是	PINBLOCK格式。

名称	类型	是否必须	描述
account	String	是	用户主账号。 • 当**PINBLOCKForm**为*04*时，该域为18N，去除校验位的18位主账号，不足18位则右对齐左填字符（F）。 • 当**PINBLOCKForm**为其他值时，该域为12N，去除校验位的最右12位主账号。

返回值

LMK下加密的PIN密文。

异常处理

程序运行中出错则抛出异常。

`cn.tass.exceptions.TAException`：接口自定义的异常。

3.39 transferPin

将PIN由TPK1/ZPK1加密转换为TPK2/ZPK2加密。

```
public java.lang.String transferPin(java.lang.String srcKeyType,
                                    java.lang.Object srcKey,
                                    java.lang.String dstKeyType,
                                    java.lang.Object dstKey,
                                    java.lang.String srcPINBLOCKCipher,
                                    int srcPINBLOCKForm,
                                    java.lang.String srcAccount,
                                    int dstPINBLOCKForm,
                                    java.lang.String dstAccount)
                             throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
srcKeyType	String	是	源密钥类型，取值如下： • *1*：TPK • *2*：ZPK
srcKey	-	是	用于加密PIN的源TPK/ZPK密钥索引或密文。
dstKeyType	String	是	目标密钥类型，取值如下：

名称	类型	是否必须	描述
			• *1*：TPK/PVK • *2*：ZPK
dstKey	-	是	用于加密PIN的目标TPK/ZPK密钥索引或密文。
srcPINBLOCKCipher	String	是	在源TPK/ZPK下加密的PINBLOCK密文。
srcPINBLOCKForm	int	是	源PINBLOCK格式。
srcAccount	String	是	源账号，12N/18N。
dstPINBLOCKForm	int	是	目标PINBLOCK格式。
dstAccount	String	是	目标账号，12N/18N。

返回值

目标PINBLOCK密文。

异常处理

程序运行中出错则抛出异常。

cn.tass.exceptions.TAException

3.40 transferPinSm2ToZpk（私钥使用索引号）

SM2公钥加密的数字PIN密文转为ZPK加密。

```
public java.lang.String[] transferPinSm2ToZpk(int privateKey,
                                              int pinCipherType,
                                              int zpk,
                                              int pinblockType,
                                              java.lang.String pan,
                                              byte[] srcPinCipher)
                       throws cn.tass.exceptions.TAException
```

```
public java.lang.String[] transferPinSm2ToZpk(int privateKey,
                                              int pinCipherType,
                                              java.lang.String zpk,
                                              int pinblockType,
                                              java.lang.String pan,
                                              byte[] srcPinCipher)
```

```
                    throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
privateKey	int	是	使用密码机内部索引下的私钥。
pinCipherType	int	是	公钥加密的PIN组成格式。 • *0*：ID长度(2N)+ID码+PIN长度(2N)+PIN明文 • *1*：PIN明文块。
zpk	int	是	ZPK索引。
	String	是	ZPK在LMK下加密的密文值。
pinblockType	int	是	ZPK下加密的PINBLOCK格式代码。
pan	String	是	用户主账号。 • **pinblockType**取值为*4*时，该域为18N。 • **pinblockType**取值为其他值，该域为12N。
srcPinCipher	byte[]	是	输入公钥加密的PIN密文数据。

返回值

- 数组1号索引下为ZPK下加密的PINBLOCK密文。
- 数组2号索引下为ID码，仅当**pinCipherType**取值为*0*时存在。

异常处理

程序运行中出错则抛出异常。

cn.tass.exceptions.TAException

3.41 transferPinSm2ToZpk（私钥使用密文）

SM2公钥加密的数字PIN密文转为ZPK加密。

```
public java.lang.String[] transferPinSm2ToZpk(byte[] privateKey,
                                              int pinCipherType,
                                              int zpk,
                                              int pinblockType,
                                              java.lang.String pan,
                                              byte[] srcPinCipher)
```

```
                    throws cn.tass.exceptions.TAException
```

```
public java.lang.String[] transferPinSm2ToZpk(byte[] privateKey,
                                              int pinCipherType,
                                              java.lang.String zpk,
                                              int pinblockType,
                                              java.lang.String pan,
                                              byte[] srcPinCipher)
                    throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
privateKey	byte[]	是	由LMK加密的非对称私钥数据。
pinCipherType	int	是	公钥加密的PIN组成格式。
zpk	int	是	ZPK索引。
	String	是	ZPK在LMK下加密的密文值。
pinblockType	int	是	ZPK下加密的PINBLOCK格式。
pan	String	是	用户主账号。 • **pinblockType**取值为*4*时，该域为18N。 • **pinblockType**取值为其他值，该域为12N。
srcPinCipher	byte[]	是	输入公钥加密的PIN密文数据。

返回值

- 数组1号索引下为ZPK下加密的PINBLOCK密文。
- 数组2号索引下为ID码，仅当**pinCipherType**取值为*0*时存在。

异常处理

程序运行中出错则抛出异常。

cn.tass.exceptions.TAException

3.42 transferPinRsaToZpk（私钥使用索引号）

RSA公钥加密的数字PIN密文转为ZPK加密。

```
public java.lang.String[] transferPinRsaToZpk(int privateKey,
                                              int pinCipherType,
                                              int padFlagAsymm,
                                              int zpk,
                                              int pinblockType,
```

```
                                    java.lang.String pan,
                                    byte[] srcPinCipher)
                             throws cn.tass.exceptions.TAException
```

```
public java.lang.String[] transferPinRsaToZpk(int privateKey,
                                    int pinCipherType,
                                    int padFlagAsymm,
                                    java.lang.String zpk,
                                    int pinblockType,
                                    java.lang.String pan,
                                    byte[] srcPinCipher)
                             throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
privateKey	int	是	使用密码机内部索引下的私钥。
pinCipherType	int	是	公钥加密的PIN组成格式。
padFlagAsymm	int	是	RSA公钥加密PIN时的填充算法： • *1*：表示PKCS_1_1.5格式。 • *7*：表示使用在PIN数据块前补0x00的方式。
zpk	int	是	ZPK在密码机内的索引。
	String	是	ZPK在LMK下加密的密文值。
pinblockType	int	是	PINBLOCK格式。
pan	String	是	用户主账号。 • **pinblockType**取值为*4*时，该域为18N。 • **pinblockType**取值为其他值，该域为12N。
srcPinCipher	byte[]	是	输入公钥加密的PIN密文数据。

返回值

- 数组1号索引下为ZPK下加密的PINBLOCK密文。
- 数组2号索引下为ID码，仅当**pinCipherType**取值为*0*时存在。

异常处理

程序运行中出错则抛出异常。

`cn.tass.exceptions.TAException`

3.43 transferPinRsaToZpk（私钥使用密文）

RSA公钥加密的数字PIN密文转为ZPK加密。

```
public java.lang.String[] transferPinRsaToZpk(byte[] privateKey,
                                              int pinCipherType,
                                              int padFlagAsymm,
                                              int zpk,
                                              int pinblockType,
                                              java.lang.String pan,
                                              byte[] srcPinCipher)
                       throws cn.tass.exceptions.TAException
```

```
public java.lang.String[] transferPinRsaToZpk(byte[] privateKey,
                                              int pinCipherType,
                                              int padFlagAsymm,
                                              java.lang.String zpk,
                                              int pinblockType,
                                              java.lang.String pan,
                                              byte[] srcPinCipher)
                       throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
privateKey	byte[]	是	由LMK加密的非对称私钥数据。
pinCipherType	int	是	公钥加密的PIN组成格式。
padFlagAsymm	int	是	RSA公钥加密PIN时的填充算法： • *1*：表示PKCS_1_1.5格式。 • *7*：表示使用在PIN数据块前补0x00的方式。
zpk	int	是	ZPK在密码机内的索引。
	String	是	ZPK在LMK下加密的密文值。
pinblockType	int	是	PINBLOCK格式。
pan	String	是	用户主账号。 • **pinblockType**取值为*4*时，该域为18N。 • **pinblockType**取值为其他值，该域为12N。
srcPinCipher	byte[]	是	输入公钥加密的PIN密文数据。

返回值

- 数组1号索引下为ZPK下加密的PINBLOCK密文。

- 数组2号索引下为ID码，仅当**pinCipherType**取值为*0*时存在。

异常处理

程序运行中出错则抛出异常。

cn.tass.exceptions.TAException

3.44 RsaPublicKeyEnc

RSA公钥加密。

```
public byte[] RsaPublicKeyEnc(int padFlag,
                              byte[] indata,
                              int key)
                       throws cn.tass.exceptions.TAException
```

```
public byte[] RsaPublicKeyEnc(int padFlag,
                              byte[] indata,
                              byte[] key)
                       throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
padFlag	int	是	填充标识： • *0*：不填充。 • *1*：PKCS#1 V1.5 方法（EMSA-PKCS1-v1_5）。
indata	byte[]	是	待加密的数据。
key	int	是	RSA密钥索引。
	byte[]	是	RSA密钥密文。

返回值

加密后的数据。

异常处理

程序运行中出错则抛出异常。

cn.tass.exceptions.TAException：接口自定义的异常。

3.45 RsaPublicKeyEnc（杂凑算法）

RSA数据加密。

```
public byte[] RsaPublicKeyEnc(int mgfHashAlg,
                              byte[] OAEPdata,
                              byte[] indata,
                              int key)
                       throws cn.tass.exceptions.TAException
```

```
public byte[] RsaPublicKeyEnc(int mgfHashAlg,
                              byte[] OAEPdata,
                              byte[] indata,
                              byte[] key)
                       throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
mgfHashAlg	int	是	杂凑算法： • *1*：SHA1 • *2*：MD5 • *5*：SHA224 • *6*：SHA256 • *7*：SHA384 • *8*：SHA512
OAEPdata	byte[]	是	OAEP编码参数，取值范围：00~99。
indata	byte[]	是	待加密的数据。
key	int	是	RSA密钥索引。
	byte[]	是	RSA密钥密文。

返回值

加密后的数据。

异常处理

程序运行中出错则抛出异常。

cn.tass.exceptions.TAException：接口自定义的异常。

3.46 RsaPrivateKeyDec

RSA数据解密。

```
public byte[] RsaPrivateKeyDec(int padFlag,
                               int key,
                               byte[] inData)
                        throws cn.tass.exceptions.TAException
```

```
public byte[] RsaPrivateKeyDec(int padFlag,
                               byte[] key,
                               byte[] inData)
                        throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
padFlag	int	是	填充标识： • *0*：不填充。 • *1*：PKCS#1 V1.5 方法（EMSA-PKCS1-v1_5）。
key	int	是	用于解密数据的RSA密钥索引。
	byte[]	是	用于解密数据的RSA密钥密文。
indata	byte[]	是	待解密的数据。

返回值

解密后的数据。

异常处理

程序运行中出错则抛出异常。

`cn.tass.exceptions.TAException`：接口自定义的异常。

3.47 RsaPrivateKeyDec（杂凑算法）

RSA数据解密。

```
public byte[] RsaPrivateKeyDec(int MgfHashAlg,
                               byte[] OAEPdata,
                               int key,
                               byte[] inData)
                        throws cn.tass.exceptions.TAException
```

```
public byte[] RsaPrivateKeyDec(int MgfHashAlg,
                               byte[] OAEPdata,
```

```
                        byte[] key,
                        byte[] inData)
                 throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
mgfHashAlg	int	是	杂凑算法： • *1*：SHA1 • *2*：MD5 • *5*：SHA224 • *6*：SHA256 • *7*：SHA384 • *8*：SHA512
OAEPdata	byte[]	是	OAEP编码参数，取值范围：00~99。
key	int	是	用于解密数据的RSA密钥索引。
	byte[]	是	用于解密数据的RSA密钥密文。
indata	byte[]	是	待解密的数据。

返回值

解密后的数据。

异常处理

程序运行中出错则抛出异常。

`cn.tass.exceptions.TAException`：接口自定义的异常。

3.48 genRsaSignature

RSA计算签名（EW）。

```
public byte[] genRsaSignature(int hashFlag,
                        int padFlag,
                        byte[] indata,
                        int key)
                 throws cn.tass.exceptions.TAException
```

```
public byte[] genRsaSignature(int hashFlag,
                        int padFlag,
                        byte[] indata,
                        byte[] key)
```

```
                    throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
hashFlag	int	是	Hash算法标识： • *1*：SHA-1 • *2*：MD5 • *3*：ISO 10118-2(部分服务版本不支持) • *5*：SHA-224 • *6*：SHA-256 • *7*：SHA-384 • *8*：SHA-512
padFlag	int	是	填充标识： • *0*：不填充（外部自行填充，配合HASH算法-04）。 • *1*：PKCS#1 V1.5 方法（EMSA-PKCS1-v1_5）。
indata	byte[]	是	待计算摘要数据。 字节长度：0~1984
key	int	是	用于计算签名的RSA密钥索引。
	byte[]	是	用于计算签名的RSA密钥密文。

返回值

签名值数据。

异常处理

程序运行中出错则抛出异常。

`cn.tass.exceptions.TAException`

3.49 genRsaSignature（OAEP计算模式）

RSA计算签名（EW）OAEP计算模式。

```
public byte[] genRsaSignature(int hashFlag,
                              int mgfHashFlag,
                              byte[] OAEPdata,
                              byte[] indata,
                              int key)
```

```
                throws cn.tass.exceptions.TAException
```

```
public byte[] genRsaSignature(int hashFlag,
                              int mgfHashFlag,
                              byte[] OAEPdata,
                              byte[] indata,
                              byte[] key)
                       throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
hashFlag	int	是	Hash算法标识： • *1*：SHA-1 • *2*：MD5 • *3*：ISO 10118-2(部分服务版本不支持) • *5*：SHA-224 • *6*：SHA-256 • *7*：SHA-384 • *8*：SHA-512
mgfHashAlg	int	是	杂凑算法： • *1*：SHA1 • *2*：MD5 • *5*：SHA224 • *6*：SHA256 • *7*：SHA384 • *8*：SHA512
OAEPdata	byte[]	是	OAEP编码参数，取值范围：00~99。
indata	byte[]	是	待计算摘要数据。 字节长度：0~1984
key	int	是	用于计算签名的RSA密钥索引。
	byte[]	是	用于计算签名的RSA密钥密文。

返回值

计算的签名值数据。

异常处理

程序运行中出错则抛出异常。

cn.tass.exceptions.TAException

3.50 verifyRsaSignature

RSA验证签名（EY）。

```
public boolean verifyRsaSignature(int hashFlag,
                                  int padFlag,
                                  byte[] indata,
                                  int key,
                                  byte[] Signature)
                           throws cn.tass.exceptions.TAException
```

```
public boolean verifyRsaSignature(int hashFlag,
                                  int padFlag,
                                  byte[] indata,
                                  byte[] key,
                                  byte[] Signature)
                           throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
hashFlag	int	是	Hash算法标识： • *1*：SHA-1 • *2*：MD5 • *3*：ISO 10118-2 • *5*：SHA-224 • *6*：SHA-256 • *7*：SHA-384 • *8*：SHA-512
padFlag	int	是	填充标识： • *0*：不填充（外部自行填充，配合HASH算法-04）。 • *1*：PKCS#1 V1.5 方法（EMSA-PKCS1-v1_5）。
indata	byte[]	是	待计算摘要数据。 字节长度：0~1984
key	int	是	用于验证签名的RSA密钥索引。
	byte[]	是	用于验证签名的RSA密钥密文。
Signature	byte[]	是	待验证签名

返回值

- 验证成功返回true。
- 验证失败返回false。

异常处理

程序运行中出错则抛出异常。

cn.tass.exceptions.TAException

3.51 verifyRsaSignature（OAEP计算模式）

RSA验证签名（EY）OAEP计算模式。

```
public boolean verifyRsaSignature(int hashFlag,
                                  int MgfFlag,
                                  byte[] OAEPdata,
                                  byte[] indata,
                                  byte[] key,
                                  byte[] Signature)
                           throws cn.tass.exceptions.TAException
```

```
public boolean verifyRsaSignature(int hashFlag,
                                  int MgfFlag,
                                  byte[] OAEPdata,
                                  byte[] indata,
                                  int key,
                                  byte[] Signature)
                           throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
hashFlag	int	是	Hash算法标识： • *1*：SHA-1 • *2*：MD5 • *3*：ISO 10118-2 • *5*：SHA-224 • *6*：SHA-256 • *7*：SHA-384 • *8*：SHA-512
mgfHashAlg	int	是	杂凑算法： • *1*：SHA1 • *2*：MD5

名称	类型	是否必须	描述
			• *5*：SHA224 • *6*：SHA256 • *7*：SHA384 • *8*：SHA512
OAEPdata	byte[]	是	OAEP编码参数，取值范围：00~99。
indata	byte[]	是	待计算摘要数据。 字节长度：0~1984
key	int	是	RSA密钥索引。
	byte[]	是	RSA密钥密文。
Signature	byte[]	是	待验证签名

返回值

- 验证成功返回true。
- 验证失败返回false。

异常处理

程序运行中出错则抛出异常。

`cn.tass.exceptions.TAException`

3.52 SM2PublicKeyEnc

SM2公钥加密。

```
public byte[] SM2PublicKeyEnc(byte[] inData,
                              int key)
                       throws cn.tass.exceptions.TAException
```

```
public byte[] SM2PublicKeyEnc(byte[] inData,
                              byte[] key)
                       throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
indata	byte[]	是	待加密的数据。 字节长度：0~1900
key	int	是	用于加密数据的SM2密钥索引。

名称	类型	是否必须	描述
	byte[]		用于加密数据的SM2密钥密文。

返回值

加密后的数据密文。

异常处理

程序运行中出错则抛出异常。

cn.tass.exceptions.TAException

3.53 EccPublicKeyEnc

ECC算法公钥数据加密运算。

```
public byte[] EccPublicKeyEnc(byte[] inData,
                              int curveFlag,
                              int key)
                       throws cn.tass.exceptions.TAException
```

```
public byte[] EccPublicKeyEnc(byte[] inData,
                              int curveFlag,
                              byte[] key)
                       throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
indata	byte[]	是	待加密的数据。 字节长度：0~1900
curveFlag	int	是	椭圆曲线标识。
key	int	是	用于加密数据的SM2密钥索引。
	byte[]		用于加密数据的SM2密钥密文。

返回值

加密后的数据密文。

异常处理

程序运行中出错则抛出异常。

cn.tass.exceptions.TAException

3.54 SM2PrivateKeyDec

SM2私钥解密。

```
public byte[] SM2PrivateKeyDec(byte[] inData,
                               int key)
                        throws cn.tass.exceptions.TAException
```

```
public byte[] SM2PrivateKeyDec(byte[] inData,
                               byte[] key)
                        throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
indata	byte[]	是	待解密的数据。 字节长度：0097~1996
key	int	是	用于解密数据的SM2密钥索引。
	byte[]		用于解密数据的SM2密钥密文。

返回值

解密后的数据明文。

异常处理

程序运行中出错则抛出异常。

`cn.tass.exceptions.TAException`

3.55 EccPrivateKeyDec

ECC算法私钥数据解密运算。

```
public byte[] EccPrivateKeyDec(byte[] inData,
                               int curveFlag,
                               int key)
                        throws cn.tass.exceptions.TAException
```

```
public byte[] EccPrivateKeyDec(byte[] inData,
                               int curveFlag,
                               byte[] key)
```

```
                    throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
indata	byte[]	是	待解密的数据。 字节长度：0097~1996
curveFlag	int	是	椭圆曲线标识。
key	int	是	用于解密数据的ECC密钥索引。
	byte[]		用于解密数据的ECC密钥密文。

返回值

解密后的数据明文。

异常处理

程序运行中出错则抛出异常。

cn.tass.exceptions.TAException

3.56 genSM2Signature（密钥使用索引号）

SM2计算签名（E5）。

```
public byte[] genSM2Signature(byte[] userID,
                              int flag,
                              byte[] indata,
                              int key)
                       throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
userID	byte[]	是	用户标识。 字节长度：0~0032。
flag	int	是	签名编码格式： • *0*：签名值数据串（r、s序列） • *1*：DER编码格式（r、s序列编码），整数使用2的补码表示法
indata	byte[]	是	待计算摘要数据。 字节长度：0~1984

名称	类型	是否必须	描述
key	int	是	用于计算签名的密钥索引。 取值范围：1~64

返回值

计算的签名值。

异常处理

程序运行中出错则抛出异常。

cn.tass.exceptions.TAException

3.57 genSM2Signature（密钥使用密文）

SM2计算签名（E5）。

```
public byte[] genSM2Signature(byte[] userID,
                              int flag,
                              byte[] indata,
                              byte[] publicKey,
                              byte[] privateKey)
                       throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
userID	byte[]	是	用户标识。 字节长度：0~0032。
flag	int	是	签名编码格式： • *0*：签名值数据串（r、s序列） • *1*：DER编码格式（r、s序列编码），整数使用2的补码表示法
indata	byte[]	是	待计算摘要数据。 字节长度：0~1984
publicKey	byte[]	是	SM2公钥，外部传入SM2公钥密文。
privateKey	byte[]	是	LMK下加密的SM2私钥私钥密文。

返回值

计算的签名值。

异常处理

程序运行中出错则抛出异常。

cn.tass.exceptions.TAException

3.58 genEccSignature（密钥使用索引号）

ECC算法签名运算。

```
public byte[] genEccSignature(byte[] userID,
                              int encodeFlag,
                              int hashFlag,
                              byte[] indata,
                              int curveFlag,
                              int key)
                       throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
userID	byte[]	是	用户标识。 字节长度：0~0032。
encodeFlag	int	是	签名结果的编码格式： • *0*：签名值数据串（r、s序列）。 • *1*：DER编码格式（r、s序列编码），整数使用2的补码表示法。
hashFlag	int	是	摘要算法标识。
indata	byte[]	是	待计算摘要数据。 字节长度：0~1984
curveFlag	int	是	椭圆曲线标识
key	int	是	用于计算签名的密钥索引。 取值范围：1~64

返回值

计算的签名值。

异常处理

程序运行中出错则抛出异常。

cn.tass.exceptions.TAException

3.59 genEccSignature（密钥使用密文）

ECC算法签名运算。

```
public byte[] genEccSignature(byte[] userID,
                              int encodeFlag,
                              int hashFlag,
                              byte[] indata,
                              int curveFlag,
                              byte[] publicKey,
                              byte[] privateKey)
                       throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
userID	byte[]	是	用户标识。 字节长度：0~0032。
encodeFlag	int	是	签名结果的编码格式： • *0*：签名值数据串（r、s序列）。 • *1*：DER编码格式（r、s序列编码），整数使用2的补码表示法。
hashFlag	int	是	摘要算法标识。
indata	byte[]	是	待计算摘要数据。 字节长度：0~1984
curveFlag	int	是	椭圆曲线标识
publicKey	byte[]	是	用户公钥数据。
privateKey	byte[]	是	用于计算签名的私钥数据。

返回值

计算的签名值。

异常处理

程序运行中出错则抛出异常。

cn.tass.exceptions.TAException

3.60 verifySM2Signature（密钥使用索引号）

SM2验证签名（E6）。

```
public boolean verifySM2Signature(byte[] userID,
```

```
                    int flag,
                    byte[] indata,
                    int key,
                    byte[] Signature)
                throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
userID	byte[]	是	用户标识。 字节长度：0~0032。
flag	int	是	签名结果的编码格式： • *0*：签名值数据串（r、s序列）。 • *1*：DER编码格式（r、s序列编码），整数使用2的补码表示法。
indata	byte[]	是	待计算签名数据。 字节长度：0~1984
key	int	是	用于验证签名的密钥索引。
Signature	byte[]	是	待验证签名。 字节长度：64~80

返回值

- 验证成功返回true。
- 验证失败返回false。

异常处理

程序运行中出错则抛出异常。

`cn.tass.exceptions.TAException`

3.61 verifyEccSignature（密钥使用索引号）

ECC算法验签名运算。

```
public boolean verifyEccSignature(byte[] userID,
                    int encodeFlag,
                    int hashFlag,
                    byte[] indata,
                    int curveFlag,
                    int key,
                    byte[] Signature)
```

```
                    throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
userID	byte[]	是	用户标识。 字节长度：0~0032。
encodeFlag	int	是	签名结果的编码格式： • *0*：签名值数据串（r、s序列）。 • *1*：DER编码格式（r、s序列编码），整数使用2的补码表示法。
hashFlag	int	是	摘要算法标识。
indata	byte[]	是	待计算签名数据。 字节长度：0~1984
curveFlag	int	是	椭圆曲线标识
key	int	是	用于验证签名的密钥索引。 取值范围：1~64
Signature	byte[]	是	待验证签名。 字节长度：64~80

返回值

- 验证成功返回true。
- 验证失败返回false。

异常处理

程序运行中出错则抛出异常。

`cn.tass.exceptions.TAException`

3.62 verifySM2Signature（密钥使用密文）

SM2验证签名（E6）。

```
public boolean verifySM2Signature(byte[] userID,
                                  int flag,
                                  byte[] indata,
                                  byte[] Signature,
                                  byte[] publicKey)
```

```
                    throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
userID	byte[]	是	用户标识。 字节长度：0~0032。
flag	int	是	签名结果的编码格式： • *0*：签名值数据串（r、s序列）。 • *1*：DER编码格式（r、s序列编码），整数使用2的补码表示法。
indata	byte[]	是	待计算签名数据。 字节长度：0~1984
Signature	byte[]	是	待验证签名。 字节长度：64~80
publicKey	byte[]	是	SM2公钥密文。

返回值

- 验证成功返回true。
- 验证失败返回false。

异常处理

程序运行中出错则抛出异常。

cn.tass.exceptions.TAException

3.63 verifyEccSignature（密钥使用密文）

ECC算法验签名运算。

```
public boolean verifyEccSignature(byte[] userID,
                                  int encodeFlag,
                                  int hashFlag,
                                  byte[] indata,
                                  int curveFlag,
                                  byte[] publicKey,
                                  byte[] Signature)
```

```
                    throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
userID	byte[]	是	用户标识。 字节长度：0~0032。
encodeFlag	int	是	签名结果的编码格式： • *0*：签名值数据串（r、s序列）。 • *1*：DER编码格式（r、s序列编码），整数使用2的补码表示法。
hashFlag	int	是	摘要算法标识。
indata	byte[]	是	待计算签名数据。 字节长度：0~1984
curveFlag	int	是	椭圆曲线标识
Signature	byte[]	是	待验证签名。 字节长度：64~80
publicKey	byte[]	是	用于验证签名的公钥DER编码数据。

返回值

- 验证成功返回true。
- 验证失败返回false。

异常处理

程序运行中出错则抛出异常。

cn.tass.exceptions.TAException

3.64 generateHASH

计算数据得到数据摘要，默认采用国密SM3算法进行摘要运算。

```
public byte[] generateHASH(byte[] data)
```

```
                throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
data	byte[]	是	计算数据摘要的数据。

返回值

返回经过H123对大数据进行摘要运算的结果。

异常处理

程序运行中出错则抛出异常。

cn.tass.exceptions.TAException

3.65 generateHASH

计算数据得到数据摘要，采用国密SM3算法进行摘要运算并且传公钥形式。

```
public byte[] generateHASH(byte[] data,
                           byte[] sm2PublicKey)
                    throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
data	byte[]	是	计算数据摘要的数据。
sm2PublicKey	byte[]	是	SM2公钥明文。

返回值

返回经过H123对大数据进行摘要运算的结果。

异常处理

程序运行中出错则抛出异常。

cn.tass.exceptions.TAException

3.66 generateHASH

计算数据得到数据摘要，支持多种hash算法模式。

```
public byte[] generateHASH(byte[] data,
                           int hashAlgFlag,
```

```
                byte[] userId,
                byte[] publicKey)
        throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
data	byte[]	是	计算数据摘要的数据。
hashAlgFlag	int	是	Hash算法标识： • *1*：SHA-1 • *2*：MD5 • *3*：ISO 10118-2 • *5*：SHA-224 • *6*：SHA-256 • *7*：SHA-384 • *8*：SHA-512 • *20*：SM3-256
userID	byte[]	是	用户标识。 **说明**： 当且仅当**hashAlgFlag**为*20*时有效，并且字节长度不能大于32。
PublicKey	byte[]	是	SM2公钥明文。 **说明**： 仅当**hashAlgFlag**为*20*时有效

返回值

返回经过H123对大数据进行摘要运算的结果。

异常处理

程序运行中出错则抛出异常。

cn.tass.exceptions.TAException

3.67 generateHASH

小报文数据摘要值计算。

```
public byte[] generateHASH(int hashAlgFlag,
```

```
                    byte[] data,
                    byte[] ID,
                    byte[] pubKey)
            throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
hashAlgFlag	int	是	Hash算法标识： • *1*：SHA-1 • *2*：MD5 • *3*：ISO 10118-2 • *5*：SHA-224 • *6*：SHA-256 • *7*：SHA-384 • *8*：SHA-512 • *20*：SM3-256
data	byte[]	是	待计算摘要数据。 字节长度：0~4096
ID	byte[]	否	用户ID，若**ID**取值为*null*，则默认采用国密局发布的默认ID（1234567812345678），且**pubKey**取值为空。 **说明**： 仅当**hashAlgFlag**为*20*时有效。
publicKey	byte[]	否	SM2公钥明文。 **说明**： 仅当**hashAlgFlag**为*20*时有效

返回值

数据摘要值。

异常处理

程序运行中出错则抛出异常。

`cn.tass.exceptions.TAException`

3.68 genRSAKeyPair

随机生成RSA密钥对。

```
public java.util.ArrayList<byte[]> genRSAKeyPair(
                                   int keyUseWay,
                                   int keyLen,
                                   int exponent,
                                   int storeIndex,
                                   java.lang.String keyLabel)
                            throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
keyUseWay	int	是	密钥用途： • *0*：签名密钥 • *1*：密钥管理密钥 • *2*：不限，建议使用此项
keyLen	int	是	密钥模长。 取值范围1024~2048，且必须为8的倍数。
exponent	int	是	公钥指数 取值：3或65537
storeIndex	int	是	秘钥储存索引，取值如下： • 1~64：储存在加密机内。 • 0：表示不储存。
keyLabel	String	是	密钥存储标签，当**storeIndex**取值在*1~64*之间生效。

返回值

公钥数据、LMK加密的私钥密文数据。

异常处理

程序运行中出错则抛出异常。

cn.tass.exceptions.TAException

3.69 getSM2PublicKey

获取SM2公钥明文。

```
public byte[] getSM2PublicKey(int storeIndex)
                    throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
storeIndex	int	是	sm2公钥储存在加密机中的索引。 取值范围：1~64

返回值

SM2公钥明文。

异常处理

程序运行中出错则抛出异常。

`cn.tass.exceptions.TAException`

3.70 generateSm2KeyPair

产生国密SM2-256 曲线的SM2秘钥对，可选的储存在加密机中。

```
public java.util.ArrayList<byte[]> generateSm2KeyPair(
                                int storeIndex,
                                java.lang.String storeLabel)
                          throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
storeIndex	int	是	sm2公钥储存在加密机中的索引。 取值范围：1~64
storeLabel	String	是	密钥存储标签，最大长度为16个 ASCII 字符。 **说明**： 当**storeIndex**取值为*1~64*之间时生效。

返回值

- 0：产生的SM2公钥。
- 1：产生的SM2私钥。

异常处理

程序运行中出错则抛出异常。

cn.tass.exceptions.TAException

3.71 generateEccKeyPair

产生ECC密钥对。

```
public java.util.ArrayList<byte[]> generateEccKeyPair(
                                    int storeIndex,
                                    int curveFlag,
                                    java.lang.String storeLabel)
                             throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
storeIndex	int	是	密钥索引，取值： • 1~64：表示储存在加密机中。 • 其他值表示不储存
curveFlag	int	是	椭圆曲线参数标识。 • *07*：国密-256新曲线，SM2 • *20*：prime192v1 • *21*：prime192v2 • *22*：prime192v3 • *23*：Secp192K1 • *30*：NISTP256 • *31*：Brainpoolp256r1 • *32*：frp256v1 • *33*：Secp256K1 • *44*：Curve25519(X25519)
storeLabel	String	是	密钥存储标签，最大长度为16个 ASCII 字符。

名称	类型	是否必须	描述
			说明： 当**storeIndex**取值为*1~64*之间时生效。

返回值

公钥DER编码数据、LMK加密的私钥数据。

异常处理

程序运行中出错则抛出异常。

cn.tass.exceptions.TAException

3.72 getKeyInfo

获取密钥信息。

```
public hsmGeneralFinance.SymmKeyAttribute getKeyInfo(int keyIndex)
                        throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
keyIndex	int	是	需要获取的密钥。

返回值

SymmKeyAttribute定义的密钥属性对象。

异常处理

程序运行中出错则抛出异常。

cn.tass.exceptions.TAException

3.73 bufferEncrypt

通用大数据加密，可用于加密文件数据等通用格式的数据。

```
public byte[] bufferEncrypt(int keyIndex,
                            java.lang.String transformation,
                            byte[] initVector,
                            byte[] inData)
```

```
                throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
keyIndex	int	是	密钥索引。 取值范围：1~2048。
transformation	String	是	转换算法参数，格式为：算法/模式/填充方式。 例如：AES/CBC/PKCS5Padding
initVector	byte[]	是	初始化向量，CBC/CFB/OFB等运算模式所需的初始化向量。
inData	byte[]	是	输入待加密的明文数据。

返回值

输出加密后的密文数据。

异常处理

程序运行中出错则抛出异常。

`cn.tass.exceptions.TAException`

3.74 bufferEncrypt（初始化向量固定为全0）

通用大数据加密，可用于加密文件数据等通用格式的数据（初始化向量固定为全0）。

```
public byte[] bufferEncrypt(int keyIndex,
                            java.lang.String transformation,
                            byte[] inData)
                     throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
keyIndex	int	是	密钥索引。 取值范围：1~2048。
transformation	String	是	转换算法参数，格式为：算法/模式/填充方式。 例如：AES/CBC/PKCS5Padding
inData	byte[]	是	输入待加密的明文数据。

返回值

输出加密后的密文数据。

异常处理

程序运行中出错则抛出异常。

cn.tass.exceptions.TAException

3.75 bufferDecrypt

通用大数据加密，可用于解密文件数据等通用格式的数据。

```
public byte[] bufferDecrypt(int keyIndex,
                            java.lang.String transformation,
                            byte[] initVector,
                            byte[] inData)
                     throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
keyIndex	int	是	密钥索引。 取值范围：1~2048。
transformation	String	是	转换算法参数，格式为：算法/模式/填充方式。 例如：AES/CBC/PKCS5Padding
initVector	byte[]	是	初始化向量，CBC/CFB/OFB等运算模式所需的初始化向量。
inData	byte[]	是	输入待解密的密文数据。

返回值

输出解密后的明文数据。

异常处理

程序运行中出错则抛出异常。

cn.tass.exceptions.TAException

3.76 bufferDecrypt（初始化向量固定为全0）

通用大数据加密，可用于解密文件数据等通用格式的数据（初始化向量固定为全0）。

```
public byte[] bufferDecrypt(int keyIndex,
                            java.lang.String transformation,
                            byte[] inData)
                     throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
keyIndex	int	是	密钥索引。 取值范围：1~2048。
transformation	String	是	转换算法参数，格式为：算法/模式/填充方式。 例如：AES/CBC/PKCS5Padding
inData	byte[]	是	输入待解密的密文数据。

返回值

输出解密后的明文数据。

异常处理

程序运行中出错则抛出异常。

`cn.tass.exceptions.TAException`

3.77 bigDatageneralEnc

使用DEK进行大包数据加密，接口中完成数据的拆包与重组。

```
public byte[] bigDatageneralEnc(int algType,
                                java.lang.Object Key,
                                int PadFlag,
                                java.lang.String IV,
                                byte[] data)
                         throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
algType	int	是	算法模式： • *0*：ECB • *1*：CBC

名称	类型	是否必须	描述
			• *2*：CFB • *3*：OFB
Key	int	是	密钥索引。 取值范围：0~2048
	String		LMK 下加密的密钥密文
PadFlag	int	是	填充模式： • *0*：PBOC 2.0填充模式 • *1*：ISO/IEC 9797-1的 PADDING模式2 • *2*：ISO/IEC 9797-1的 PADDING模式1 • *3*：ANSI X9.23 • *4*：PKCS#5 • *5*：NoPadding模式 • *10*：PBOC3.0 • *11*：左填充+ISO/IEC 9797-1
IV	String	是	初始向量，仅当**algType**取值为*1*、*2*、*3*时生效。 • 当ZEK 密钥标识为*Z*/*X*/*U*/*Y*/*T* 时，该域为16H。 • 当ZEK 密钥标识为*R*/*P*/*L*时，该域为32H。
data	byte[]	是	加密数据。

返回值

加密结果。

异常处理

程序运行中出错则抛出异常。

cn.tass.exceptions.TAException

3.78 bigDatageneralDnc

使用DEK进行大包数据解密，接口中完成数据的拆包与重组。

```
public byte[] bigDatageneralDnc(int algType,
                                java.lang.Object Key,
```

```
                    int PadFlag,
                    java.lang.String IV,
                    byte[] data)
                throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
algType	int	是	算法模式： • *0*：ECB • *1*：CBC • *2*：CFB • *3*：OFB
Key	int	是	密钥索引。 取值范围：0~2048
	byte[]		LMK 下加密的密钥密文
PadFlag	int	是	填充模式： • *0*：PBOC 2.0填充模式 • *1*：ISO/IEC 9797-1的 PADDING模式2 • *2*：ISO/IEC 9797-1的 PADDING模式1 • *3*：ANSI X9.23 • *4*：PKCS#5 • *5*：NoPadding模式 • *10*：PBOC3.0 • *11*：左填充+ISO/IEC 9797-1
IV	String	是	初始向量，仅当**algType**取值为*1*、*2*、*3*时生效。 • 当ZEK 密钥标识为*Z*/*X*/*U*/*Y*/*T* 时，该域为16H。 • 当ZEK 密钥标识为*R*/*P*/*L*时，该域为32H。
data	byte[]	是	需要解密的数据。

返回值

解密结果。

异常处理

程序运行中出错则抛出异常。

`cn.tass.exceptions.TAException`

3.79 sm2ExportSymmkey

SM2公钥保护导出一条对称秘钥。

```
public java.lang.String[] sm2ExportSymmkey(
                                byte[] sm2PublicKey,
                                java.lang.String keyType,
                                char keyFlag,
                                java.lang.Object symmKey,
                                java.lang.String deriveFactor)
                            throws java.lang.NumberFormatException,
                            cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
sm2PublicKey	byte[]	是	外部传入SM2公钥
keyType	String	是	密钥类型，相应密钥类型的密钥类型号，支持密钥类型代码和密钥类型名称两种格式。以下左侧为密钥类型代码，右侧为密钥类型名称： • 000：KEK • 00A：DEK • 109：MDK
keyFlag	char	是	密钥LMK下加密的密钥标识： • *Z*：单倍长DES密钥 • *X*：双倍长DES密钥 • *Y*：三倍长DES密钥 • *R*：16字节SM4密钥 • *P*：16字节SM1密钥 • *L*：16字节AES密钥 • *M*：24字节AES密钥 • *N*：32字节AES密钥
symmKey	int	是	加密机内部对称秘钥索引。 取值范围：1~2048

名称	类型	是否必须	描述
	String		外部传入秘钥密文。
deriveFactor	String	是	导出秘钥的分散因子。 取值范围：长度为32的倍数（16个字节）

返回值

- 0：被导出的对称秘钥数据块密文。
- 1：被导出对称秘钥的校验值。

异常处理

程序运行中出错则抛出异常。

```
java.lang.NumberFormatException
```

```
cn.tass.exceptions.TAException
```

3.80 sm2ImportSymmkey

SM2公钥保护导入对称秘钥。

```
public java.lang.String[] sm2ImportSymmkey(
                                java.lang.String keyType,
                                char keyFlag,
                                int storeKeyIndex,
                                java.lang.String storeKeyLabel,
                                java.lang.String check,
                                byte[] data,
                                java.lang.Object PrivateKey)
                          throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
keyType	String	是	密钥类型，相应密钥类型的密钥类型号，支持密钥类型代码和密钥类型名称两种格式。以下左侧为密钥类型代码，右侧为密钥类型名称： • 000：KEK • 00A：DEK • 109：MDK
keyFlag	char	是	密钥LMK下加密的密钥标识：

名称	类型	是否必须	描述
			• *Z*：单倍长DES密钥 • *X*：双倍长DES密钥 • *Y*：三倍长DES密钥 • *R*：16字节SM4密钥 • *P*：16字节SM1密钥 • *L*：16字节AES密钥 • *M*：24字节AES密钥 • *N*：32字节AES密钥
storeKeyIndex	int	是	密钥存储索引。 取值范围：1~2048，其他值时表示不存储。
storeKeyLabel	String	是	密钥存储标签，用于密钥存储在加密机时的说明标签，0~16个ASII字符。。 **说明**： 仅在**storeKeyIndex**取值为1~2048时生效。
check	String	是	导入秘钥的校验值，SM2保护导出对称秘钥时的校验值。
data	byte[]	是	导入秘钥的密文，SM2保护导出对称秘钥的数据密文。
PrivateKey	int	是	标识存储到密码机内的目标索引号。 取值范围：1~64
	byte[]		表示外部传入SM2私钥。

返回值

- 0：导入秘钥的密文（LMK加密下输出）。
- 1：导入秘钥的校验值。

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.81 generateSignature

SM2私钥签名运算（对数据摘要运算）。

```
public byte[] generateSignature(byte[] data,
                                java.lang.Object eccPrivateKey,
                                int codeFormat)
                         throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
data	byte[]	是	待签名的数据，字节长度为32字节。
eccPrivateKey	int	是	SM2公钥在密码机内存储的索引号。 取值范围：0001~0064
	byte[]		外部密钥传入。
codeFormat	int	是	签名编码格式： • 0：签名值数据串（r、s序列） • 1：DER编码格式（r、s序列编码）

返回值

返回数字签名结果。

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.82 verifySignature

SM2公钥验签运算（针对数据摘要）。

```
public boolean verifySignature(int signCodeFormat,
                               byte[] signature,
                               byte[] srcData,
                               java.lang.Object eccPublicKey)
```

```
                    throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
signCodeFormat	int	是	签名编码格式： • 0：签名值数据串（r、s序列） • 1：DER编码格式（r、s序列编码）
signature	byte[]	是	待验证的签名。 字节长度：0064~0080
srcData	byte[]	是	待签名的数据，字节长度为32字节。
eccPublicKey	int	是	SM2公钥在密码机内存储的索引号。 取值范围：0001~0064
	byte[]		外部传入SM2公钥。

返回值

- 成功返回true。
- 失败返回false。

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.83 rsaImportSymmkey

RSA公钥保护导入一条对称密钥。

```
public java.lang.String[] rsaImportSymmkey(
                          java.lang.String keyType,
                          char keyFlag,
                          int storeKeyIndex,
                          java.lang.String storeKeyLabel,
                          java.lang.String check,
                          byte[] data,
                          java.lang.Object PrivateKey)
```

```
                    throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
keyType	String	是	RSA公钥加密导入的对称密钥类型。 • 000：KEK • 00A：DEK • 109：MDK
keyFlag	char	是	RSA公钥加密导入的对称密钥算法类型标识： • *Z*：8字节DES密钥 • *X*：16字节3DES密钥 • *U*：16字节3DES密钥 • *Y*：24字节3DES密钥 • *T*：24字节3DES密钥 • *P*：16字节SM1密钥 • *R*：16字节SM4密钥 • *L*：16字节AES密钥
storeKeyIndex	int	是	导入对称密钥的存储索引。
storeKeyLabel	String	是	导入密钥标签。
check	String	是	导入秘钥的校验值。
data	byte[]	是	密钥密文，在RSA公钥下加密的密钥密文。
PrivateKey	-	是	RSA公钥密文或索引。

返回值

返回类型为String数组，数组下标的含义：

- 0：导入的对称密钥在LMK下加密的密文数据。
- 1：被导入密钥的校验值。

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.84 rsaExportSymmkey

RSA公钥加密导出一条对称密钥。

```
public java.lang.Object[] rsaExportSymmkey(
                                java.lang.Object RsaPublicKey,
                                java.lang.String keyType,
                                java.lang.Object symmKey,
                                java.lang.String deriveFactor,
                                byte[] publickeyAuthData)
                            throws java.lang.NumberFormatException,
                            cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
RsaPublicKey	-	是	RSA公钥。
keyType	String	是	被保护导出的密钥的类型： • 000：KEK • 00A：DEK • 109：MDK
symmKey	-	是	被导出密钥的密钥索引或密文。
deriveFactor	String	是	导出秘钥的分散因子。
publickeyAuthData	byte[]	是	认证数据，用于计算公钥 MAC 的额外的数据，不能包含分号（；）。 字节长度：0~128

返回值

返回类型为Object数组，数组下标的含义：

- 0：被导出的对称秘钥数据块密文。
- 1：被导出对称秘钥的校验值。

异常处理

程序运行中出错则抛出异常。

```
java.lang.NumberFormatException
```

```
cn.tass.exceptions.TAException
```

3.85 sm2ExportSymmkey

SM2公钥保护导出一条对称秘钥。

```
public java.lang.String[] sm2ExportSymmkey(
                                java.lang.Object sm2PublicKey,
                                java.lang.String keyType,
                                char keyFlag,
                                java.lang.Object symmKey,
                                java.lang.String deriveFactor)
                            throws java.lang.NumberFormatException,
                            cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
sm2PublicKey	-	是	传入的SM2公钥或者索引
keyType	String	是	密钥类型，相应密钥类型的密钥类型号，支持密钥类型代码和密钥类型名称两种格式。以下左侧为密钥类型代码，右侧为密钥类型名称： • 000：KEK • 00A：DEK • 109：MDK
keyFlag	char	是	密钥LMK下加密的密钥标识： • *Z*：单倍长DES密钥 • *X*：双倍长DES密钥 • *Y*：三倍长DES密钥 • *R*：16字节SM4密钥 • *P*：16字节SM1密钥 • *L*：16字节AES密钥 • *M*：24字节AES密钥 • *N*：32字节AES密钥
symmKey	int	是	加密机内部对称秘钥索引。

名称	类型	是否必须	描述
			取值范围：1~2048
	String		外部传入秘钥密文。
deriveFactor	String	是	导出秘钥的分散因子。 取值范围：长度为32的倍数（16个字节）

返回值

- 0：被导出的对称秘钥数据块密文。
- 1：被导出对称秘钥的校验值。

异常处理

程序运行中出错则抛出异常。

```
java.lang.NumberFormatException
```

```
cn.tass.exceptions.TAException
```

3.86 getRSAPublicKey

根据密钥索引获取RSA公钥。

```
public byte[] getRSAPublicKey(int keyIndex)
                    throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
keyIndex	int	是	密钥索引。

返回值

返回RSA公钥。

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.87 generalCalMAC

使用ZAK计算银联POS MAC。

```
public java.lang.String generalCalMAC(java.lang.Object key,
                                       java.lang.String inData)
                                throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
key	int	是	ZAK类型密钥索引。 取值范围：1~2048
	String		LMK下加密的ZAK密钥密文。
inData	String	是	计算MAC的数据。 取值16进制字符串：0~4096字节数

返回值

返回生成的MAC密文。

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.88 generalCalMAC

使用ZAK计算银联POS MAC。

```
public java.lang.String generalCalMAC(java.lang.Object key,
                                       byte[] inData)
                                throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
key	int	是	ZAK类型密钥索引。

名称	类型	是否必须	描述
			取值范围：1~2048
	String		LMK下加密的ZAK密钥密文。
inData	byte[]	是	计算MAC的数据。 长度：0~4096字节数

返回值

返回生成的MAC密文。

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.89 generalRSApublicKeyMAC

计算RSA公钥MAC。

```
public byte[] generalRSApublicKeyMAC(byte[] rsaPublicKey,
                                     byte[] AuthenticationData)
                              throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
rsaPublicKey	byte[]	是	公钥，ASN.1格式的DER编码，包含模、指数e序列。
AuthenticationData	byte[]	是	认证数据。

返回值

使用LMK分组对公钥和认证数据计算的MAC。

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.90 generalSM2publicKeyMAC

计算RSA公钥MAC。

```
public byte[] generalSM2publicKeyMAC(
                              byte[] sm2PublicKey,
                              java.lang.String AuthenticationData)
                            throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
sm2PublicKey	byte[]	是	SM2公钥。
AuthenticationData	String	是	认证数据，用于计算公钥MAC的额外数据，不能包含分号（；）。

返回值

返回MAC值。

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.91 generalEccPublicKeyMAC

计算外部ECC公钥的MAC。

```
public byte[] generalEccPublicKeyMAC(
                              int curveFlag,
                              byte[] publicKey,
                              java.lang.String AuthenticationData)
                            throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
curveFlag	int	是	ECC曲线标识。

名称	类型	是否必须	描述
publicKey	byte[]	是	ECC公钥数据。
AuthenticationData	String	是	认证数据，用于计算公钥MAC的额外数据，不能包含分号（；）。

返回值

返回公钥MAC值。

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.92 transferPintpk2PublicKey

TPK加密PIN转为公钥加密。

```
public byte[] transferPintpk2PublicKey(java.lang.Object tpkKey,
                                       java.lang.String tpkPIN,
                                       int PINBLOCK,
                                       java.lang.String pan,
                                       int publicKeyFlag,
                                       byte[] publicKey,
                                       byte[] keyCheck,
                                       byte[] publicKeyMAC,
                                       int publicKeyPINBLOCK,
                                       int publicKeyPadding)
                                throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
tpkKey	int	是	密钥索引。 取值范围：1~2048
	String		外部传入密钥密文。
tpkPIN	String	是	TPK加密的PIN密文。 • TPK密钥为8字节分组长度时，长度为16H。 • TPK密钥为16字节分组长度时，长度为32H
PINBLOCK	int	是	PINBLOCK格式。

名称	类型	是否必须	描述
pan	String	是	账号PAN。 • 当**PINBLOCK**取值为*4*时，若长度不满足18N，右侧填充F。 • 当**PINBLOCK**取值为其他值，PAN取值12N。
publicKeyFlag	int	是	公钥算法标识。 • *1*：RSA 公钥 • *7*：SM2 公钥
publicKey	byte[]	是	公钥，ASN.1格式DER编码。
keyCheck	byte[]	是	计算公钥MAC认证数据。
publicKeyMAC	byte[]	是	公钥MAC值，长度4字节。
publicKeyPINBLOCK	int	是	PINBLOCK格式，当**PINflag**取值为*10*时存在。
publicKeyPadding	int	是	RSA公钥加密的填充模式，当且仅当**publicKeyFlag**取值为*1*时存在该域。 • *1*：PKCS#1 v1.5 填充方式。 • *7*：在 PIN 数据块前面补 0x00，以使数据长度等于RSA 密钥模长。

返回值

公钥加密的 PIN 数据块密文。

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.93 encExportKeyByProKey

保护密钥（可分散 ）加密导出一条密钥（可分散）。

```
public java.lang.String[] encExportKeyByProKey(
                          int encAlgModel,
                          java.lang.String proKeyType,
                          java.lang.Object proKey,
                          java.lang.String proKeyDisperFactor,
                          int sessionKeyType,
```

```
                    java.lang.String sessionKeyFactor,
                    java.lang.String exportKeyType,
                    java.lang.Object exportKey,
                    java.lang.String exporKeyDisperFactor,
                    java.lang.String keyHead,
                    char extendFlag,
                    int PADFlag,
                    java.lang.String IV)
                  throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
encAlgModel	int	是	加密算法模式，标识保护密钥加密被导出密钥时的算法模式。 • *0*：ECB • *1*：CBC
proKeyType	String	是	保护密钥类型，支持密钥类型代码和密钥类型名称两种格式。 以下左侧为密钥类型代码，右侧为密钥类型名称： • 000：ZMK/KEK • 001：KMC • 109：MDK
proKey	int	是	加密机内部对称密钥索引。 取值范围：1~2048
	String		外部传入密钥密文。
proKeyDisperFactor	String	是	保护密钥分散因子，n个分散因子的串联，每个分散因子必须为16个字节。
sessionKeyType	int	是	会话密钥产生模式： • *00*：不产生会话密钥。 • *01*：ECB 模式加密8字节会话密钥因子，得8字节会话密钥。 • *02*：ECB 模式加密16字节会话密钥因子，得16字节会话密钥。 • *03*：密钥的左右8字节异或，得8字节会话密钥。 • *04*：取密钥的左8字节做为会话密钥。

名称	类型	是否必须	描述
			• *05*：CBC模式加密16字节会话密钥因子，得16字节会话密钥。
sessionKeyFactor	String	是	会话密钥因子，仅当**sessionKeyType**取值为*01*/*02*/*05*时生效。 • **sessionKeyType**为*01*时，该域为8字节（16H），适用于产生PBOC规范的单长度会话密钥。 取值：6字节0x00||2字节ATC。 • **sessionKeyType**为*02*时，该域为16字节（32H），适用于产生PBOC规范的双长度会话密钥。 取值：6字节0x00||2字节ATC||6字节0x00||2字节ATC的非 • **sessionKeyType**为*05*时，该域为16字节（32H），适用于产生GP规范SCP02的卡片会话密钥。 取值：2字节密钥类型||2字节卡计数器||12字节0x00
exportKeyType	String	是	被导出密钥的类型: • 000：KEK • 109：MK-AC/MDK • 209：MK-SMI • 309：MK-SMC • 409：MK-DAK • 509：MK-DN • 011：KMC • 00A：DEK
exportKey	int	是	加密机内部对称密钥索。 取值范围：1~2048
	String		外部传入密钥密文。
exporKeyDisperFactor	String	是	导出密钥分散因子。 取值：16字节的整数倍长度

名称	类型	是否必须	描述
keyHead	String	是	密钥头，IC卡内存储此密钥的密钥头，通常为密钥属性，用于计算密钥密文。 取值：0~32字节
extendFlag	char	是	• **extendFlag**取值为*P*时，采用*PADFlag*填充模式。 • **extendFlag**取值不为*P*时，则默认采用填充模式00和一个分组全00的IV（CBC加密模式）。
PADFlag	int	否	**extendFlag**取值为*P*时存在。 • *0*：PBOC2.0填充模式 • *1*：ISO/IEC9797-1的PADDING模式2 • *2*：ISO/IEC9797-1的PADDING模式1 • *3*：ANSIX9.23 • *4*：PKCS#5 • *5*：NoPadding模式 • *10*：PBOC3.0 • *11*：左填充+ISO/IEC9797-1
IV	String	否	当**extendFlag**取值*P*，且**encAlgModel**为*1*时存在。 • 若密钥算法为128分组，该域为16字节（32H）。 • 若密钥算法为64分组，该域为8字节（16H）。

返回值

- 0：密钥数据块密文。
- 1：被导出密钥或子密钥的校验值。

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.94 symmEncExportRSAkey

保护密钥加密导出RSA密钥对。

```
public java.util.ArrayList<byte[]> symmEncExportRSAkey(
                                    int encAlgModel,
                                    java.lang.String proKeyType,
                                    java.lang.Object proKey,
                                    java.lang.String proKeyDisperFactor,
                                    int keyIndex,
                                    byte[] priKeyData,
                                    char extendFlag,
                                    int padFlag,
                                    int outputFormat,
                                    java.lang.String IV)
                                throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
encAlgModel	int	是	加密算法模式。 • *0*：ECB • *1*：CBC
proKeyType	String	是	保护密钥类型，支持密钥类型代码和密钥类型名称两种格式。 以下左侧为密钥类型代码，右侧为密钥类型名称： • 000：KEK • 109：MDK
proKey	int	是	用于加密保护 RSA 的保护密钥索引。
	String		外部传入密钥密文。
proKeyDisperFactor	String	是	分散因子，n个分散因子的串联，每个分散因子必须为16个字节。
keyIndex	int	是	导出RSA密钥索引。 • 当取值为1~64之间时，表示导出加密机内部索引位置的RSA密钥对。

名称	类型	是否必须	描述
			• 当取值为99时，**priKeyData**存在表示外部传入的RSA私钥密文（Lmk下加密的私钥密文）。
priKeyData	byte[]	是	LMK加密的私钥密文。 **说明**： 当且仅当**keyIndex**取值为*99*时有效。
extendFlag	char	是	扩展标识： • **extendFlag**取值为*P*时，标识下面三个域取值有效。 • **extendFlag**取值不为*P*时，标识下面三个域取值无效，HSM默认采用填充模式01、明文DER编码格式输出公钥和一个分组全00 的IV（CBC加密模式）。
PADFlag	int	否	填充模式，当且仅当**extendFlag**取值为*P*时有效。
outputFormat	int	否	输出格式，当且仅当**extendFlag**取值为*P*时有效。 • *0*：公钥明文DER格式输出，ASN.1格式DER编码（模，指数序列）。 • *1*：m及e采用分量密文形式输出。 • *2*：公钥明文DER编码格式输出，私钥密文输出。
IV	String	否	当**extendFlag**取值*P*，且**encAlgModel**为*1*时存在。 • 若密钥算法为128分组，该域为16字节（32H）。 • 若密钥算法为64分组，该域为8字节（16H）。

返回值

私钥密文各分量（结果参照**extendFlag**和*outputFormat*取值）。

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.95 symmEncImportRSAkey

保护密钥加密保护导入RSA密钥对。

```
public byte[] symmEncImportRSAkey(int encAlgModel,
                                  java.lang.String proKeyType,
                                  java.lang.Object proKey,
                                  java.lang.String proKeyDisperFactor,
                                  int keyIndex,
                                  java.lang.String RSAkeyLable,
                                  byte[] pubKeyData,
                                  byte[] d,
                                  byte[] P,
                                  byte[] Q,
                                  byte[] dP,
                                  byte[] dQ,
                                  byte[] qInv)
                           throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
encAlgModel	int	是	加密算法模式。 • *0*：ECB • *1*：CBC
proKeyType	String	是	保护密钥类型，支持密钥类型代码和密钥类型名称两种格式。 以下左侧为密钥类型代码，右侧为密钥类型名称： • 000：KEK • 109：MDK
proKey	int	是	用于加密保护 RSA 的保护密钥索引。
	String		外部传入密钥密文。
proKeyDisperFactor	String	是	分散因子，n个分散因子的串联，每个分散因子必须为16个字节。
keyIndex	int	是	导出RSA密钥索引。

名称	类型	是否必须	描述
			• 当取值为1~64之间时，表示导出加密机内部索引位置的RSA密钥对。 • 当取值为99时，**priKeyData**存在表示外部传入的RSA私钥密文（Lmk下加密的私钥密文）。
RSAkeyLable	String	是	RSA密钥标签，当且仅当**keyIndex**取值为1~64之间时有效。 取值范围：0~16个ASCII字符。
pubKeyData	byte[]	是	要导入的RSA密钥的公钥明文，ASN.1格式DER编码（模，指数序列）格式。
d	byte[]	是	私钥指数d
P	byte[]	是	私钥分量P
Q	byte[]	是	私钥分量Q
dP	byte[]	是	私钥分量dP
dQ	byte[]	是	私钥分量dQ
qInv	byte[]	是	私钥分量qInv

返回值

LMK加密的私钥，包括m，e，d和5个CRT成份。

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.96 transferAscPinSm2ToZpk

非对称公钥加密的字符PIN密文转为ZPK加密。

```
public java.lang.String[] transferAscPinSm2ToZpk(
                                    int asymmAlg,
                                    java.lang.Object privateKey,
                                    int pinCipherType,
                                    int padFlagAsymm,
                                    java.lang.Object zpk,
                                    int pinblockType,
                                    java.lang.String pan,
                                    byte[] srcPinCipher,
                                    char T,
                                    int macType,
```

```
                                int macPad)
                        throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
asymmAlg	int	是	公钥算法标识。 • *1*：RSA • *7*：SM2
privateKey	int	是	用于加密保护 RSA 的保护密钥索引。
	String		外部传入密钥密文。
pinCipherType	int	是	公钥加密PIN组成格式。 • *00*：ID长度（2N）+ID码+PIN长度（2N）+PIN明文 • *01*：PIN明文块 • *02*：PIN与账号分别左对齐，以填充0x00方式扩展为32H，再异或后得到PIN数据块
padFlagAsymm	int	是	公钥加密的填充模式，标识公钥加密PIN数据块时采用的填充模式，仅当**asymmAlg**为*01*时存在。 • *01*：PKCS#1v1.5填充方式 • *07*：在PIN数据块前面补0x00，以使数据长度等于RSA密钥模长
zpk	-	是	ZPK密钥，用于加密PIN的ZPK密钥索引或密文。
pinblockType	int	是	PINBLOCK格式。
pan	String	是	账号PAN。 • 数据块格式为04时，该域为18N，去除校验位的18位主账号，不足18位则右对齐左填F。 • 当PIN数据块格式为其他值时，该域为12N。
srcPinCipher	byte[]	是	公钥加密的PIN密文。

名称	类型	是否必须	描述
T	char	是	MAC计算扩展标识，当且仅当取值为*T*时，**macType**和**macPad**两个域的取值有效。
macType	int	是	MAC算法模式： • *1*：ISO9797-1MAC算法模式1 • *3*：ISO9797-1MAC算法模式3（限ZPK标识为X/U）
macPad	int	是	计算MAC数据块的PAD标识。

返回值

返回值为公钥加密的字符PIN密文转为ZPK加密[0]。

- 当返回长度为2时，且**pinCipherType**取值为*0*或*2*时，返回数组下标[1]为ID码。
- 当返回长度为2时，且MAC计算扩展域标识为*T*时，返回数组下标[1]为数据块MAC值。
- 当**pinCipherType**取值为*0*或*2*，MAC计算扩展域标识为*T*时，则返回数组下标[1]为ID码、返回数组下标[2]为数据块MAC值。

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.97 transferAscPinSm2ToZpk（PIN组成格式为PIN明文块）

公钥加密的字符PIN密文转为ZPK加密 （公钥加密的PIN组成格式为PIN明文块）。

```
public java.lang.String transferAscPinSm2ToZpk(
                                    int asymmAlg,
                                    java.lang.Object privateKey,
                                    int padFlagAsymm,
                                    java.lang.Object zpk,
                                    int pinblockType,
                                    java.lang.String pan,
                                    byte[] srcPinCipher)
                             throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
asymmAlg	int	是	公钥算法标识。

名称	类型	是否必须	描述
			• *1*：RSA • *7*：SM2
privateKey	int	是	私钥索引，用于解密（公钥加密的）PIN密文的私钥索引号。 取值范围：0001~0064
	String		私钥密文。
padFlagAsymm	int	是	公钥加密的填充模式，标识公钥加密PIN数据块时采用的填充模式，仅当**asymmAlg**为*01*时存在。 • *01*：PKCS#1v1.5填充方式 • *07*：在PIN数据块前面补0x00，以使数据长度等于RSA密钥模长
zpk	-	是	ZPK密钥，用于加密PIN的ZPK密钥索引或密文。
pinblockType	int	是	PINBLOCK格式。
pan	String	是	账号PAN。 • 数据块格式为04时，该域为18N，去除校验位的18位主账号，不足18位则右对齐左填F。 • 当PIN数据块格式为其他值时，该域为12N。
srcPinCipher	byte[]	是	公钥加密的PIN密文。

返回值

返回值为ZPK加密的PIN数据块密文。

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.98 transferAscPinSm2ToZpk

非对称公钥加密的字符PIN密文转为ZPK加密。

```
public java.lang.String[] transferAscPinSm2ToZpk(
                                    int asymmAlg,
                                    java.lang.Object privateKey,
                                    int pinCipherType,
                                    int padFlagAsymm,
                                    java.lang.Object zpk,
                                    int pinblockType,
                                    java.lang.String pan,
                                    byte[] srcPinCipher)
                            throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
asymmAlg	int	是	公钥算法标识。 • *1*：RSA • *7*：SM2
privateKey	int	是	私钥索引，用于解密（公钥加密的）PIN密文的私钥索引号。 取值范围：0001~0064
	String		私钥密文。
pinCipherType	int	是	公钥加密PIN组成格式。 • *00*：ID长度（2N）+ID码+PIN长度（2N）+PIN明文 • *02*：PIN与账号分别左对齐，以填充0x00方式扩展为32H，再异或后得到PIN数据块
padFlagAsymm	int	是	公钥加密的填充模式，标识公钥加密PIN数据块时采用的填充模式，仅当**asymmAlg**为*01*时存在。 • *01*：PKCS#1v1.5填充方式

名称	类型	是否必须	描述
			• *07*：在PIN数据块前面补0x00，以使数据长度等于RSA密钥模长
zpk	-	是	ZPK密钥，用于加密PIN的ZPK密钥索引或密文。
pinblockType	int	是	PINBLOCK格式。
pan	String	是	账号PAN。 • 数据块格式为04时，该域为18N，去除校验位的18位主账号，不足18位则右对齐左填F。 • 当PIN数据块格式为其他值时，该域为12N。
srcPinCipher	byte[]	是	公钥加密的PIN密文。

返回值

返回值为数组。

- [0]：ZPK加密的PIN密文
- [1]：ID码

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.99 genWorkKeyEnc

重载更新密钥函数，LMK密钥标识与ZMK密钥标识分开。

```
public java.lang.String[] genWorkKeyEnc(java.lang.String keyType,
                                        char LmkKeyFlag,
                                        java.lang.Object ZMKKey,
                                        char ZMKKeyFlag,
                                        int storeKeyIndex,
                                        java.lang.String storeKeyLabel)
```

```
                    throws java.lang.Exception
```

请求参数

名称	类型	是否必须	描述
keyType	String	是	密钥类型，相应密钥类型的密钥类型号，支持密钥类型代码和密钥类型名称两种格式。以下左侧为密钥类型代码，右侧为密钥类型名称： • 000：ZMK/KEK • 001：ZPK • 002：PVK/TPK/TMK • 003：TAK • 008：ZAK • 009：BDK • 00A：ZEK/DEK • 00B：TEK • 011：KMC • 109：MK-AC/MDK • 10C：HMAC • 209：MK-SMI • 309：MK-SMC • 402：CVK • 409：MK-DAK • 509：MK-DN
LmkKeyFlag	char	是	LMK密钥标识。
ZMKKey	-	是	ZMK密钥索引或密文。
ZMKKeyFlag	char	是	ZMK下加密的密钥密文标识。
storeKeyIndex	int	是	密钥存储索引。存储范围为1~2048，其他值表示不存储。
storeKeyLabel	String	是	密钥存储标签，用于在密钥存储内部存储时标记密钥的说明标签，0~16个ASII字符。

返回值

返回值为数组。

- [0]：LMK加密的密钥密文

- [1]：ZMK加密的密钥密文
- [2]：新产生密钥的校验值

异常处理

程序运行中出错则抛出异常。

```
java.lang.Exception
```

3.100 generateZPKCharPIN

ZPK加密字符PIN密文。

```
public java.lang.String generateZPKCharPIN(java.lang.Object key,
                                           java.lang.String PIN,
                                           int PINBLOCK,
                                           java.lang.String PAN)
                                    throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
Key	-	是	用于加密PIN的ZPK的密钥索引或密文。
PIN	String	是	字符PIN明文。
PINBLOCK	int	是	字符PINBLOCK格式。 • *0*：PIN与账号分别左对齐，以填充0x00方式扩展为48H（采用64位分组算法）或64H（采用128位分组算法），再异或后得到PIN数据块。 • *2*：PIN与账号分别左对齐，以填充0x00方式扩展为32H，再异或后得到PIN数据块。
PAN	String	是	账号PAN。 当**PINBLOCK**取值为*2*时，**PAN**取值不能超过16个数字。

返回值

PIN由ZPK加密密文。

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.101 transferZpkCharPin

字符PINBLOCK转为其他密钥加密，不支持分散。

```
public java.lang.String[] transferZpkCharPin(java.lang.Object zpkKey,
                                             java.lang.String keyType,
                                             java.lang.Object key,
                                             java.lang.String srcPIN,
                                             int srcPINBLOCK,
                                             java.lang.String srcPAN,
                                             int PINBLOCK,
                                             java.lang.String PAN)
                      throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
zpkKey	-	是	用于加密PIN的源ZPK的密钥索引或密文。
keyType	String	是	目标PINBLOCK加密密钥类型，支持密钥类型代码和密钥类型名称两种格式。 以下左侧为密钥类型代码，右侧为密钥类型名称： • 001：ZPK • 002：TPK/PVK
key	-	是	目的PINBLOCK加密的密钥索引或密文。
srcPIN	String	是	源PINBLOCK密文。
srcPINBLOCK	int	是	源PINBLOCK数据组成格式。 • *0*：PIN与账号分别左对齐，以填充0x00方式扩展为48H（采用64位分组算法）或64H（采用128位分组算法），再异或后得到PIN数据块。 • *2*：PIN与账号分别左对齐，以填充0x00方式扩展为32H，再异或后得到PIN数据块。
srcPAN	String	是	源账号PAN。

名称	类型	是否必须	描述
			当**srcPINBLOCK**取值为*2*时，该域最大长度不能超过16个数字。
PINBLOCK	int	是	目的加密PINBLOCK数据组成格式。 • *0*：PIN与账号分别左对齐，以填充0x00方式扩展为48H（采用64位分组算法）或64H（采用128位分组算法），再异或后得到PIN数据块。 • *2*：PIN与账号分别左对齐，以填充0x00方式扩展为32H，再异或后得到PIN数据块。
PAN	String	是	目的账号PAN。 当**srcPINBLOCK**取值为*2*时，该域最大长度不能超过16个数字。

返回值

返回值为数组。

- [0]：PIN明文长度（字节数）。
- [1]：目的密钥下加密的随机字符PIN密文。

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.102 blocksEncrypt

多段数据加密，此方法以固定的ECB模式加密输入数据列表中的各段数据。

```
public java.util.ArrayList<byte[]> blocksEncrypt(
                                java.lang.String keyType,
                                java.lang.Object key,
                                java.lang.String disperFactor,
                                int sessionType,
                                java.lang.String sessionFactor,
                                java.util.ArrayList<byte[]> inData,
                                int paddingMode)
```

```
                    throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
keyType	String	是	工作密钥的密钥类型。 • 000：KEK • 109：MDK • 309：MK-SMC • 00A：ZEK/DEK • 00B：TEK • 011：KMC
key	int	是	工作密钥索引。 取值范围：1~2048
	String		LMK加密的会话密钥密文
disperFactor	String	是	密钥分散因子，n个分散因子的串联，每个分散因子必须为16个字节。
sessionType	int	是	会话密钥生成模式： • *00*：不产生会话密钥。 • *01*：ECB模式加密8字节会话密钥因子，得8字节会话密钥。 • *02*：ECB模式加密16字节会话密钥因子，得16字节会话密钥。 • *03*：密钥的左右8字节异或，得8字节会话密钥。 • *04*：取密钥的左8字节做为会话密钥。 • *05*：CBC模式加密16字节会话密钥因子，得16字节会话密钥。
sessionFactor	String	是	会话密钥因子，仅当**sessionType**取值为01/02/05时存在。 • **sessionType**为*01*时，该域为8字节（16H），适用于产生PBOC规范的单长度会话密钥，仅限3DES算法的密钥。 取值：6字节0x00||2字节ATC

名称	类型	是否必须	描述
			• **sessionType**为*02*时，该域为16字节（32H），适用于产生PBOC规范的双长度会话密钥。 取值：6字节0x00||2字节ATC||6字节0x00||2字节ATC的非 • **sessionType**为*05*时，该域为16字节（32H），适用于产生GP规范SCP02的卡片会话密钥， 取值：2字节密钥类型||2字节卡计数器||12字节0x00
inData	byte[]	是	输入明文数据列表，可以使用以下方式： • 多余多段数据，需要分别放入集合当中，并需要满足集合的泛型模式，数据已byte[]方式增加进集合。 • 支持一整段数据直接以byte格式填充进集合进行加解密。
paddingMode	int	是	数据填充模式，本方法均按16字节分组的方式进行填充。 • *0*：PBOC的数据加解密填充规范，非强制的填充'80'00... • *1*：ISO/IEC9797-1的PADDING模式2，强制性填充'80'00... • *2*：ISO/IEC9797-1的PADDING模式1，非强制性填充'00'00... • *3*：ANSIX9.23定义的填充算法，强制性填充'00...'n

返回值

密文数据列表。

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.103 blocksDecrypt

多段数据解密，此方法以固定的ECB模式解密输入数据列表中的各段数据。

```
public java.util.ArrayList<byte[]> blocksDecrypt(
                                java.lang.String keyType,
                                java.lang.Object key,
                                java.lang.String disperFactor,
                                int sessionType,
                                java.lang.String sessionFactor,
                                java.util.ArrayList<byte[]> inData,
                                int paddingMode)
                            throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
keyType	String	是	工作密钥的密钥类型。 • 309：MK-SMC • 00A：ZEK/DEK • 00B：TEK
key	int	是	工作密钥索引。 取值范围：1~2048
	String		LMK加密的会话密钥密文
disperFactor	String	是	密钥分散因子，n个分散因子的串联，每个分散因子必须为16个字节。
sessionType	int	是	会话密钥生成模式： • *00*：不产生会话密钥。 • *01*：ECB模式加密8字节会话密钥因子，得8字节会话密钥。 • *02*：ECB模式加密16字节会话密钥因子，得16字节会话密钥。 • *03*：密钥的左右8字节异或，得8字节会话密钥。 • *04*：取密钥的左8字节做为会话密钥。

名称	类型	是否必须	描述
			• *05*：CBC模式加密16字节会话密钥因子，得16字节会话密钥。
sessionFactor	String	是	会话密钥因子，仅当**sessionType**取值为01/02/05时存在。 • **sessionType**为*01*时，该域为8字节（16H），适用于产生PBOC规范的单长度会话密钥，仅限3DES算法的密钥。 取值：6字节0x00||2字节ATC • **sessionType**为*02*时，该域为16字节（32H），适用于产生PBOC规范的双长度会话密钥。 取值：6字节0x00||2字节ATC||6字节0x00||2字节ATC的非 • **sessionType**为*05*时，该域为16字节（32H），适用于产生GP规范SCP02的卡片会话密钥， 取值：2字节密钥类型||2字节卡计数器||12字节0x00
inData	byte[]	是	输入密文数据列表。
paddingMode	int	是	数据填充模式，本方法均按16字节分组的方式进行填充。 • *0*：PBOC的数据加解密填充规范，非强制的填充'80'00... • *1*：ISO/IEC9797-1的PADDING模式2，强制性填充'80'00... • *2*：ISO/IEC9797-1的PADDING模式1，非强制性填充'00'00... • *3*：ANSIX9.23定义的填充算法，强制性填充'00...'n

返回值

解密数据列表。

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.104 genRandomKey

产生一条随机密钥，可选的存储到密码机内（KR指令）。

```
public java.lang.String[] genRandomKey(java.lang.String keyType,
                                       char keyFlag,
                                       char keyStoreFlag,
                                       int keyIndex,
                                       java.lang.String keyTag)
                                throws java.lang.Exception
```

请求参数

名称	类型	是否必须	描述
keyType	String	是	密钥类型，相应密钥类型的密钥类型号，支持密钥类型代码和密钥类型名称两种格式。以下左侧为密钥类型代码，右侧为密钥类型名称： • 000：ZMK/KEK • 001：ZPK • 002：PVK/TPK/TMK • 003：TAK • 008：ZAK • 009：BDK • 00A：ZEK/DEK • 00B：TEK • 011：KMC • 109：MK-AC/MDK • 10C：HMAC • 209：MK-SMI • 309：MK-SMC • 402：CVK • 409：MK-DAK • 509：MK-DN
KeyFlag	char	是	LMK密钥标识。

名称	类型	是否必须	描述
			• *Z*：8字节DES密钥 • *X*/*U*：16字节3DES密钥 • *Y*/*T*：24字节3DES密钥 • *P*：16字节SM1密钥 • *R*：16字节SM4密钥 • *L*：16字节AES密钥 • *M*：24字节AES密钥 • *N*：32字节AES密钥
keyStoreFlag	char	否	密钥存储标识。 • 取值*K*，表明密钥产生后存储在加密机中，当选择此值，后续参数**密钥索引**、**密钥标签长度**、**密钥标签**必须存在。 • 此项如果为空（没有任何数据），表明密钥不保存加密机中，而是由LMK加密后输出密文。
keyIndex	int	否	密钥索引，存储到密码机内的密钥索引号，仅当**keyStoreFlag**存在时需要。 取值范围：0001~2048
keyTag	String	否	密钥标签，用于在密钥内部存储时标记密钥的标签说明，仅当**keyStoreFlag**存在时需要。 0~16个ASCII字符

返回值

密钥密文（LMK加密的密钥密文）、校验值（密钥校验值）

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

```
java.lang.Exception
```

3.105 deriveSymmKey

KD指令密钥分散。

```
public java.lang.String[] deriveSymmKey(java.lang.String srcKeyType,
                                        java.lang.Object srcKey,
                                        java.lang.String tarKeyType,
                                        char tarKeyAlg,
                                        int deriveFlag,
                                        java.lang.String iv,
                                        java.lang.String deriveFactor,
                                        char keyStoreFlag,
                                        int tarIndex,
                                        java.lang.String tarKeyTag)
                                 throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
srcKeyType	String	是	源密钥类型，支持密钥类型代码和密钥类型名称两种格式。 以下左侧为密钥类型代码，右侧为密钥类型名称： • 000：ZMK/KEK • 002：PVK/TPK/TMK • 007：EDK • 008：ZAK • 00A：ZEK/DEK • 011：KMC • 109：MK-AC/MDK • 209：MK-SMI • 309：MK-SMC • 409：MK-DAK • 509：MK-DN
srcKey	int	是	用于分散产生新密钥的源密钥索引。
	String		用于分散产生新密钥的源密钥密文。

名称	类型	是否必须	描述
tarKeyType	String	是	子密钥类型标识： • 000：ZMK/KEK • 008：ZAK • 00A：ZEK/DEK • 011：KMC • 109：MK-AC/MDK • 209：MK-SMI • 309：MK-SMC • 409：MK-DAK • 509：MK-DN
tarKeyAlg	char	是	子密钥算法标识： • *X*/*U*：16字节3DES密钥 • *P*：16字节SM1密钥 • *R*：16字节SM4密钥 • *L*：16字节AES密钥 • *N*：32字节AES密钥
deriveFlag	int	是	分散算法标识： • *0*：PBOC子密钥分散算法用于分散产生应用子密钥。8字节分散因子D，使用源密钥对16字节[D|D的非]采用源密钥的算法标识进行ECB模式加密。 • *1*：ECB模式加密16字节分散因子。 • *2*：ECB模式加密16字节分散因子，并复制扩展为32字节长度密钥。 **说明**： 只有在**tarKeyAlg**为*N*时支持。 • *3*：CBC模式加密16字节分散因子。 • *4*：ECB模式加密分散因子，分散因子必须为8字节的倍数，且至少16字节。截取加密结果的前后各8字节作为子密钥。 **说明**： 只有在**tarKeyAlg**为*X*/*U*时支持。

名称	类型	是否必须	描述
			• *5*：CBC模式加密分散因子，分散因子必须为16字节的倍数。截取加密结果的最后16字节作为子密钥。 **说明**： 只有在**srcKeyType**和**tarKeyAlg**都为*L*时支持。
iv	String	是	初始向量。 **说明**： 当**disperAlgType**为*5*时有效。
deriveFactor	String	是	分散因子数据，n级分散因子串联。 • 当**deriveFlag**为*0*时，每级分散因子为8字节（16H）。 • 当**deriveFlag**为*1*/*2*/*3*时，每级分散因子为16字节（32H）。 • 当**deriveFlag**为*4*时，分散因子为n*16H，并且只当做一级分散。
keyStoreFlag	char	否	密钥存储标识。 • 取值*K*，表明密钥产生后存储在加密机中，当选择此值，后续域**密钥索引**、**密钥标签长度**、**密钥标签**必须存在。 • 此项为空（没有任何数据），表明密钥不保存加密机中，而是由LMK加密后输出密文。
tarIndex	int	否	密钥存储索引，取值范围1~2048，其他值表示不存储。
tarKeyTag	String	否	子密钥存储标签，该域用于在密钥内部存储时标记密钥的标签说明，0~16个ASCII字符。 **说明**： 仅当**keyStoreFlag**存在时存在。

返回值

返回String[]：

- [0]：密钥密文
- [1]：密钥校验值

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.106 disGenNewKey

分散产生新密钥，可选的存储到密码机内。

```
public java.lang.String[] disGenNewKey(java.lang.String srcKeyType,
                                       java.lang.Object srcKey,
                                       java.lang.String subKeyType,
                                       char subKeyFlag,
                                       int disAlgModel,
                                       java.lang.String IV,
                                       int disperSeries,
                                       java.lang.String disperFactor,
                                       char keyStoreFlag,
                                       int keyIndex,
                                       java.lang.String keyLabel)
                                throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
srcKeyType	String	是	源密钥类型，用于分散产生子密钥的源密钥类型。 • 000：ZMK/KEK • 008：ZAK • 00A：ZEK/DEK • 011：KMC • 109：MK-AC/MDK • 209：MK-SMI • 309：MK-SMC • 409：MK-DAK • 509：MK-DN
srcKey	int	是	用于分散产生新密钥的源密钥索引。

名称	类型	是否必须	描述
	String		用于分散产生新密钥的源密钥密文。
subKeyType	String	是	子密钥类型标识： • 000：ZMK/KEK • 008：ZAK • 00A：ZEK/DEK • 011：KMC • 109：MK-AC/MDK • 209：MK-SMI • 309：MK-SMC • 409：MK-DAK • 509：MK-DN
subKeyFlag	char	是	子密钥算法标识： • *X*/*U*：16字节3DES密钥 • *P*：16字节SM1密钥 • *R*：16字节SM4密钥 • *L*：16字节AES密钥 • *N*：32字节AES密钥
disAlgModel	int	是	分散算法标识： • *0*：PBOC子密钥分散算法用于分散产生应用子密钥。8字节分散因子D，使用源密钥对16字节[D|D的非]采用源密钥的算法标识进行ECB模式加密。 • *1*：ECB模式加密16字节分散因子。 • *2*：ECB模式加密16字节分散因子，并复制扩展为32字节长度密钥。 **说明**： 只有在**tarKeyAlg**为*N*时支持。 • *3*：CBC模式加密16字节分散因子。 • *4*：ECB模式加密分散因子，分散因子必须为8字节的倍数，且至少16字节。截取加密结果的前后各8字节作为子密钥。 **说明**： 只有在**tarKeyAlg**为*X*/*U*时支持。

名称	类型	是否必须	描述
			• *5*：CBC模式加密分散因子，分散因子必须为16字节的倍数。截取加密结果的最后16字节作为子密钥。 **说明**： 只有在**srcKeyType**和**tarKeyAlg**都为*L*时支持。
iv	String	是	初始向量。 **说明**： 当**disperAlgType**为*5*时有效。
disperSeries	int	是	分散级数。
deriveFactor	String	是	分散因子数据，n级分散因子串联。 • 当**deriveFlag**为*0*时，每级分散因子为8字节（16H）。 • 当**deriveFlag**为*1*/*2*/*3*时，每级分散因子为16字节（32H）。 • 当**deriveFlag**为*4*时，分散因子为n*16H，并且只当做一级分散。
keyStoreFlag	char	否	密钥存储标识。 • 取值*K*，表明密钥产生后存储在加密机中，当选择此值，后续域**密钥索引**、**密钥标签长度**、**密钥标签**必须存在。 • 此项为空（没有任何数据），表明密钥不保存加密机中，而是由LMK加密后输出密文。
tarIndex	int	否	密钥存储索引，取值范围1~2048，其他值表示不存储。
tarKeyTag	String	否	子密钥存储标签，该域用于在密钥内部存储时标记密钥的标签说明，0~16个ASCII字符。 **说明**： 仅当**keyStoreFlag**存在时存在。

返回值

返回String[]：

- [0]：密钥密文
- [1]：密钥校验值

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.107 exportKeyByTranserKey

传输密钥保护导出一条密钥（KH）。

```
public java.lang.String[] exportKeyByTranserKey(
                          int encAlgModel,
                          int macAlgModel,
                          int macValueType,
                          int proKeyType,
                          java.lang.Object proKey,
                          java.lang.String proDisperFactor,
                          java.lang.String exportKeyType,
                          java.lang.Object exportKey,
                          java.lang.String exporKeyDisperFactor,
                          int macKeyType,
                          java.lang.Object macKey,
                          java.lang.String macKeyDisperFactor,
                          java.lang.String keyHead,
                          java.lang.String commandHead,
                          java.lang.String random,
                          java.lang.String IV)
                        throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
encAlgModel	int	是	加密算法模式： • *00*：ECB • *01*：CBC
macAlgModel	int	是	MAC算法模式： • *01*：ISO9797-1MAC算法模式1，使用MAC密钥CBC模式加密数据，取最后一段密文。

名称	类型	是否必须	描述
			• *03*：ISO9797-1MAC算法模式3，限密钥标识为X/U等同于ANSIX9.19，MAC密钥16字节，KL对数据DESCBC加密运算，最后一段结果KRDES解密，再KLDES加密，得8字节MAC结果。
macValueType	int	是	MAC取值方式:按前个域模式产生的密文值输出下述结果作为MAC。 • 01~08：输出MAC值的左n字节（n取值为第2个数字） • 11~18：输出MAC值的右n字节 • 21~28：左右异或后取左n字节输出 • 31~38：左右异或后取右n字节输出 • 44：四字节异或，最后输出4字节 • 10：密钥标识为P/L/R时输出完整的16字节MAC值
proKeyType	int	是	保护密钥类型： • 000：KEK • 109：MDK
proKey	int	是	加密保护被导出密钥的密钥索引。
	String		加密保护被导出密钥的密钥密文。
proDisperFactor	String	是	保护密钥分散因子n个分散因子的串联，每个分散因子必须为8个字节。用于产生卡片传输密钥或卡片的应用主控密钥。
exportKeyType	String	是	导出密钥类型： • 000：KEK • 00A：DEK • 011：KMC • 109：MK-AC/MDK • 209：MK-SMI • 309：MK-SMC • 409：MK-DAK • 509：MK-DN

名称	类型	是否必须	描述
exportKey	int	是	导出密钥的密钥索引。
	String		导出密钥的密钥密文。
exporKeyDisperFactor	String	是	导出密钥分散因子，长度为n*16H。
macKeyType	int	是	MAC密钥类型： • 999：与保护密钥同 • 000：KEK • 109：MDK
macKey	int	否	用于计算密文MAC的密钥索引。 **说明**： **macKeyType**取值为*999*时，此参数不设置。
	String		用于计算密文MAC的密钥密文。 **说明**： **macKeyType**取值为*999*时，此参数不设置。
macKeyDisperFactor	String	否	MAC密钥分散因子，长度为n*16H。 **说明**： **macKeyType**取值为*999*时，此参数不设置。
keyHead	String	否	密钥头，长度为n*16H。
commandHead	String	是	命令头。
random	String	是	随机数，用于计算密文MAC通常为4字节，最大不超过32字节。
IV	String	是	• 128位分组（密钥标识P/L/R）时，该域16字节（32H）。 • 其他情况下，该域为8字节（16H）。

返回值

密钥数据块密文、密文MAC、校验值

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.108 exportKeyByTranserKey

传输密钥保护导出一条密钥（KH）。

```
public java.lang.String[] exportKeyByTranserKey(
                          int encAlgModel,
                          int macAlgModel,
                          int macValueType,
                          java.lang.String proKeyType,
                          java.lang.Object proKey,
                          java.lang.String proDisperFactor,
                          java.lang.String exportKeyType,
                          java.lang.Object exportKey,
                          java.lang.String exporKeyDisperFactor,
                          java.lang.String macKeyType,
                          java.lang.Object macKey,
                          java.lang.String macKeyDisperFactor,
                          java.lang.String keyHead,
                          java.lang.String commandHead,
                          java.lang.String random,
                          java.lang.String IV)
                        throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
encAlgModel	int	是	加密算法模式： • *00*：ECB • *01*：CBC
macAlgModel	int	是	MAC算法模式： • *01*：ISO9797-1MAC算法模式1，使用MAC密钥CBC模式加密数据，取最后一段密文。 • *03*：ISO9797-1MAC算法模式3，限密钥标识为X/U等同于ANSIX9.19，MAC密钥16字节，KL对数据DESCBC加密运算，最后一段结果KRDES解

名称	类型	是否必须	描述
			密，再KLDES加密，得8字节MAC结果。
macValueType	int	是	MAC取值方式:按前个域模式产生的密文值输出下述结果作为MAC。 • 01~08：输出MAC值的左n字节（n取值为第2个数字） • 11~18：输出MAC值的右n字节 • 21~28：左右异或后取左n字节输出 • 31~38：左右异或后取右n字节输出 • 44：四字节异或，最后输出4字节 • 10：密钥标识为P/L/R时输出完整的16字节MAC值
proKeyType	String	是	保护密钥类型： • 000：KEK • 109：MDK
proKey	int	是	加密保护被导出密钥的密钥索引。
	String		加密保护被导出密钥的密钥密文。
proDisperFactor	String	是	保护密钥分散因子n个分散因子的串联，每个分散因子必须为8个字节。用于产生卡片传输密钥或卡片的应用主控密钥。
exportKeyType	String	是	导出密钥类型： • 000：KEK • 00A：DEK • 011：KMC • 109：MK-AC/MDK • 209：MK-SMI • 309：MK-SMC • 409：MK-DAK • 509：MK-DN
exportKey	int	是	导出密钥的密钥索引。
	String		导出密钥的密钥密文。

名称	类型	是否必须	描述
exporKeyDisperFactor	String	是	导出密钥分散因子，长度为n*16H。
macKeyType	String	是	MAC密钥类型： • 999：与保护密钥同 • 000：KEK • 109：MDK
macKey	int	否	用于计算密文MAC的密钥索引。 **说明**： **macKeyType**取值为*999*时，此参数不设置。
	String		用于计算密文MAC的密钥密文。 **说明**： **macKeyType**取值为*999*时，此参数不设置。
macKeyDisperFactor	String	否	MAC密钥分散因子，长度为n*16H。 **说明**： **macKeyType**取值为*999*时，此参数不设置。
keyHead	String	否	密钥头，长度为n*16H。
commandHead	String	是	命令头。
random	String	是	随机数，用于计算密文MAC通常为4字节，最大不超过32字节。
IV	String	是	• 128位分组（密钥标识P/L/R）时，该域16字节（32H）。 • 其他情况下，该域为8字节（16H）。

返回值

密钥数据块密文、密文MAC、校验值

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.109 importKeyByTranserKey

传输密钥保护导出一条密钥（KH）。

```
public java.lang.String[] importKeyByTranserKey(
                                int inputType,
                                int encAlgModel,
                                int macAlgModel,
                                int macValueType,
                                java.lang.String proKeyType,
                                java.lang.Object proKey,
                                java.lang.String proKeyDisperFactor,
                                java.lang.String imporKeyType,
                                char imporKeyFlag,
                                char imporKeyStoreFlag,
                                int imporKeyIndex,
                                java.lang.String imporKeyLabel,
                                java.lang.String macKeyType,
                                java.lang.Object macKey,
                                java.lang.String macKeyDisperFactor,
                                java.lang.String keyHead,
                                java.lang.String commandHead,
                                java.lang.String random,
                                java.lang.String IV,
                                java.lang.String cipher,
                                java.lang.String cipherMAC)
                            throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
inputType	int		输入类型： • *0*：仅密文 • *1*：密文+MAC
encAlgModel	int	是	加密算法模式： • *00*：ECB • *01*：CBC
macAlgModel	int	是	MAC算法模式： • *01*：ISO9797-1MAC算法模式1，使用MAC密钥CBC模式加密数据，取最后一段密文。

名称	类型	是否必须	描述
			• *03*：ISO9797-1MAC算法模式3，限密钥标识为X/U等同于ANSIX9.19，MAC密钥16字节，KL对数据DESCBC加密运算，最后一段结果KRDES解密，再KLDES加密，得8字节MAC结果。
macValueType	int	是	MAC取值方式：按**macAlgModel**产生的密文值输出下述结果作为MAC。 • 01~08：输出MAC值的左n字节（n取值为第2个数字） • 11~18：输出MAC值的右n字节 • 21~28：左右异或后取左n字节输出 • 31~38：左右异或后取右n字节输出 • 44：四字节异或，最后输出4字节 • 10：密钥标识为P/L/R时输出完整的16字节MAC值
proKeyType	String	是	保护密钥类型： • 000：KEK • 109：MDK
proKey	int	是	用于加密保护被导入密钥的密钥索引。
	String		用于加密保护被导入密钥的密钥密文。
proDisperFactor	String	是	保护密钥分散因子n个分散因子的串联，每个分散因子必须为8个字节。用于产生卡片传输密钥或卡片的应用主控密钥。
imporKeyType	String	是	被导入密钥类型： • 00A：DEK • 011：KMC • 109：MK-AC/MDK
imporKeyFlag	char		导入密钥标识 • *Z*：8字节DES密钥 • *X*/*U*：16字节3DES密钥 • *Y*/*T*：24字节3DES密钥 • *P*：16字节SM1密钥

名称	类型	是否必须	描述
			• *R*：16字节SM4密钥 • *L*：16字节AES密钥 • *M*：24字节AES密钥 • *N*：32字节AES密钥
imporKeyStoreFlag	char	否	导入密钥存储标识。 • 取值为*K*，表明存储到HSM中某索引。 • 不设置，输出LMK下加密的密钥。
imporKeyIndex	int	否	导入密钥索引，存储到密码机内的密钥索引号，取值范围：0001~2048。 **说明**： 仅当**imporKeyStoreFlag**存在时设置。
imporKeyLabel	String	否	导入密钥标签，用于标记被导入密钥的标签说明，0~16个ASCII字符。 **说明**： 仅当**imporKeyStoreFlag**存在时设置。
macKeyType	String	否	MAC密钥类型： • 999：与保护密钥同 • 000：KEK • 109：MDK **说明**： 仅当**inputType**为*1*时存在。
macKey	int	否	用于计算密文MAC的密钥索引。 **说明**： 仅当**inputType**为*1*并且**macKeyType**取值不为*999*时，此参数存在。
	String		用于计算密文MAC的密钥密文。 **说明**：

名称	类型	是否必须	描述
			仅当**inputType**为*1*并且**macKeyType**取值不为*999*时，此参数存在。
macKeyDisperFactor	String	否	n个分散因子的串联，每个分散因子必须为8个字节。  **说明**： 仅当**inputType**为*1*并且**macKeyType**取值不为*999*时，此参数存在。
keyHead	String	否	密钥头，长度为n*2H。
commandHead	String	否	命令头，取值00~20（即0-32字节）。  **说明**： 仅当**inputType**为*1*时存在。
random	String	否	随机数，IC卡生成的随机数，用于验证密文MAC通常为4字节，最大不超过32字节。  **说明**： 仅当**inputType**为*1*时存在。
IV	String	否	用于计算密文MAC的初始向量。 • 128位分组（密钥标识P/L/R）时，该域16字节（32H）。 • 其他情况下，该域为8字节（16H）。  **说明**： 仅当**inputType**为*1*时存在。
cipher	String	是	密钥数据块密文。
cipherMAC	String	否	密文MAC，仅当安全报文类型为1时存在。如果MAC不满16H，则左对齐后右补字符0；MAC校验时忽略右边为0的位。

返回值

新密钥密文(LMK)、密钥校验值

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.110 encExportKeyByProKey

保护密钥加密导出一条密钥（通用（ SH ））。

```
public java.lang.String[] encExportKeyByProKey(
                          int encAlgModel,
                          int proKeyType,
                          java.lang.Object proKey,
                          java.lang.String proKeyDisperFactor,
                          int sessionKeyType,
                          java.lang.String sessionKeyFactor,
                          java.lang.String exportKeyType,
                          java.lang.Object exportKey,
                          java.lang.String exporKeyDisperFactor,
                          java.lang.String keyHead,
                          char extendFlag,
                          int PADFlag,
                          java.lang.String IV)
                        throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
encAlgModel	int	是	加密算法模式： • *00*：ECB • *01*：CBC
proKeyType	int	是	保护密钥类型： • 000：ZMK/KEK • 109：MDK • 011：KMC
proKey	int	是	保护密钥，使用密钥索引。
	String		保护密钥，使用密钥密文。
proKeyDisperFactor	String	是	保护密钥分散因子。
sessionKeyType	int	是	会话密钥模式： • *00*：不产生会话密钥。

名称	类型	是否必须	描述
			• *01*：ECB模式加密8字节会话密钥因子，得8字节会话密钥。 • *02*：ECB模式加密16字节会话密钥因子，得16字节会话密钥。 • *03*：密钥的左右8字节异或，得8字节会话密钥。 • *04*：取密钥的左8字节做为会话密钥。 • *05*：CBC模式加密16字节会话密钥因子，得16字节会话密钥。
sessionKeyFactor	String	否	会话密钥因子。 **说明**： 仅当**sessionKeyType**为*01*/*02*/*05*时存在。 • **sessionKeyType**为*01*时，该值为8字节（16H），适用于产生PBOC规范的单长度会话密钥。 取值：6字节0x00||2字节ATC • **sessionKeyType**为*02*时，该域为16字节（32H），适用于产生PBOC规范的双长度会话密钥。 取值：6字节0x00||2字节ATC||6字节0x00||2字节ATC的非 • **sessionKeyType**为*05*时，该域为16字节（32H），适用于产生GP规范SCP02的卡片会话密钥。 取值：2字节密钥类型||2字节卡计数器||12字节0x00
exportKeyType	String	是	导出密钥类型： • 000：KEK • 00A：DEK • 011：KMC • 109：MK-AC/MDK • 209：MK-SMI

名称	类型	是否必须	描述
			• 309：MK-SMC • 409：MK-DAK • 509：MK-DN
exportKey	int	是	被导出密钥的密钥索引。
	String		被导出密钥的密钥密文。
exporKeyDisperFactor	String	是	导出密钥分散因子。
keyHead	String	是	密钥头，IC卡内存储此密钥的密钥头，通常为密钥属性。
extendFlag	char	否	扩展标识。
PADFlag	int	否	PAD标识。
IV	String	否	仅当**extendFlag**为*P*且**encAlgModel**为*01*时存在。 • 若密钥算法为128分组，该域为16字节（32H）。 • 若密钥算法为64分组，该域为8字节（16H）。

返回值

密钥数据块密文、校验值

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.111 encImportKeyByProKey

保护密钥加密导入一条密钥（通用）。

```
public java.lang.String[] encImportKeyByProKey(
                          int encAlgModel,
                          int proKeyType,
                          java.lang.Object proKey,
                          java.lang.String proKeyDisperFactor,
                          int sessionKeyType,
                          java.lang.String sessionKeyFactor,
```

```
                    java.lang.String imporKeyType,
                    char imporKeyFlag,
                    java.lang.String imporKeyCipher,
                    char imporKeyStoreFlag,
                    int imporKeyIndex,
                    java.lang.String imporKeyLabel,
                    java.lang.String keyHead,
                    char extendFlag,
                    int PADFlag,
                    java.lang.String IV,
                    java.lang.String KEYCV)
                throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
encAlgModel	int	是	加密算法模式： • *00*：ECB • *01*：CBC
proKeyType	int	是	保护密钥类型： • 000：KEK • 109：MDK • 011：KMC
proKey	int	是	保护密钥，使用密钥索引。
	String		保护密钥，使用外部存储密文。
proKeyDisperFactor	String	是	保护密钥分散因子。
sessionKeyType	int	是	会话密钥模式： • *00*：不产生会话密钥。 • *01*：ECB模式加密8字节会话密钥因子，得8字节会话密钥。 • *02*：ECB模式加密16字节会话密钥因子，得16字节会话密钥。 • *03*：密钥的左右8字节异或，得8字节会话密钥。 • *04*：取密钥的左8字节做为会话密钥。 • *05*：CBC模式加密16字节会话密钥因子，得16字节会话密钥。
sessionKeyFactor	String	否	会话密钥因子。

名称	类型	是否必须	描述
			说明： 仅当**sessionKeyType**为*01*/*02*/*05*时存在。 • **sessionKeyType**为*01*时，该值为8字节（16H），适用于产生PBOC规范的单长度会话密钥。 取值：6字节0x00||2字节ATC • **sessionKeyType**为*02*时，该域为16字节（32H），适用于产生PBOC规范的双长度会话密钥。 取值：6字节0x00||2字节ATC||6字节0x00||2字节ATC的非 • **sessionKeyType**为*05*时，该域为16字节（32H），适用于产生GP规范SCP02的卡片会话密钥。 取值：2字节密钥类型||2字节卡计数器||12字节0x00
imporKeyType	String	是	导入密钥类型： • 000：ZMK/KEK • 001：ZPK • 002：PVK/TPK/TMK • 003：TAK • 008：ZAK • 009：BDK • 00A：DEK • 00B：TEK • 011：KMC • 109：MK-AC/MDK • 209：MK-SMI • 309：MK-SMC • 402：CVK • 409：MK-DAK • 509：MK-DN

名称	类型	是否必须	描述
imporKeyFlag	char	是	导入密钥标识： • *Z*：8字节DES密钥 • *X*/*U*：16字节3DES密钥 • *Y*/*T*：24字节3DES密钥 • *P*：16字节SM1密钥 • *R*：16字节SM4密钥 • *L*：16字节AES密钥
imporKeyCipher	String	是	导入密钥密文。
imporKeyStoreFlag	char	否	导入密钥存储标识。如果取值为*K*，则表明存储到HSM中某索引，必须存在后续域。
imporKeyIndex	int	否	导入密钥索引，存储到密码机内的密钥索引号。 取值范围：0001~2048 **说明**： 仅当**imporKeyStoreFlag**为*K*时存在。
imporKeyLabel	int	是	导入密钥标签，用于标记被导入密钥的标签说明，0~16个ASCII字符。
keyHead	String	是	密钥头，IC卡内存储此密钥的密钥头，通常为密钥属性。若取值为全00，则密码机不验证密钥头的有效性。
extendFlag	char	否	扩展标识，取值为*P*。HSM默认采用填充模式00去除填充，一个分组全00的IV（CBC加密模式），不校验KEYCV。） • 若该域存在，则下面的扩展域存在。 • 若该域不存在，则后面的扩展域均不存在。
PADFlag	int	否	PAD标识，标识加密被导出密钥块前的填充规则。 取值范围：00~05或10~11 **说明**： 当且仅当**extendFlag**为*P*时存在。

名称	类型	是否必须	描述
IV	String	否	仅 当**extendFlag**为*P*且**encAlgModel**为*01*时存在。 • 若密钥算法为128分组，该域为16字节（32H）。 • 若密钥算法为64分组，该域为8字节（16H）。
KEYCV	String	否	被导入密钥的校验值KEYCV，用于校验被导入密钥的合法性。如果校验值不满16H，则左对齐后右补字符0，校验时忽略右边为0的位；若输入全0则不进行验证。 **说明**： 当且仅当**extendFlag**为*P*时存在。

返回值

新密钥密文(LMK)、校验值

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.112 deleteKey

删除内部指定索引的密钥（ KF ）。

```
public java.lang.String[] deleteKey(int keyType,
                                    int keyIndex)
                             throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
keyType	int	是	密钥类型，指定待删除密钥的类型。 • *00*：对称密钥 • *01*：RSA密钥对 • *02*：SM2密钥对

名称	类型	是否必须	描述
keyIndex	int	是	密钥索引号 ，待删除的密钥的索引号。 取值范围：1~2048

返回值

返回错误码。

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.113 importSymKey

导入存储一条对称密钥。

```
public java.lang.String[] importSymKey(java.lang.String keyType,
                                       java.lang.String keyCipher,
                                       java.lang.String keyCV,
                                       int keyIndex,
                                       java.lang.String keyLabel)
                                throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
keyType	String	是	密钥类型： • 000：ZMK/KEK • 001：ZPK • 002：PVK/TPK/TMK • 003：TAK • 008：ZAK • 009：BDK • 00A：ZEK/DEK • 00B：TEK • 011：KMC • 109：MK-AC/MDK • 10C：HMAC • 209：MK-SMI • 309：MK-SMC • 402：CVK

名称	类型	是否必须	描述
			• 409：MK-DAK • 509：MK-DN
keyCipher	String	是	密钥密文，:LMK加密的密钥密文。
keyCV	String	是	密钥校验值，校验通过后内部存储。如果输入全0则不验证，直接存储覆盖。
keyIndex	int	是	密钥索引号 ，存储到密码机内的密钥索引号。 取值范围：1~2048
keyLabel	String	是	密钥标签，用于在密钥内部存储时标记密钥的标签说明，0~16 个ASCII 字符。

返回值

校验值。

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.114 disperKeyOutputComponentCipher

分散密钥输出子密钥的多个成分密文。

```
public java.lang.String[][] disperKeyOutputComponentCipher(
                                java.lang.String srcKeyType,
                                java.lang.Object srcKey,
                                java.lang.String subKeyType,
                                char subKeyFlag,
                                int disperAlgModel,
                                int disperSeries,
                                java.lang.String disperFactor,
                                int keyComponentNum)
                          throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
srcKeyType	String	是	源密钥类型： • 000：ZMK/KEK • 008：ZAK

名称	类型	是否必须	描述
			• 00A：ZEK/DEK • 011：KMC • 109：MK-AC/MDK • 10C：HMAC • 209：MK-SMI • 309：MK-SMC • 409：MK-DAK • 509：MK-DN
srcKey	int	是	源密钥，使用密钥索引。
	String		源密钥，使用外部存储密文。
subKeyType	String	是	子密钥类型： • 000：ZMK/KEK • 008：ZAK • 00A：ZEK/DEK • 011：KMC • 109：MK-AC/MDK • 10C：HMAC • 209：MK-SMI • 309：MK-SMC • 409：MK-DAK • 509：MK-DN
subKeyFlag	char	是	子密钥标识： • *X*/*U*：16字节3DES密钥 • *P*：16字节SM1密钥 • *R*：16字节SM4密钥 • *L*：16字节AES密钥 • *N*：32字节AES密钥
disperAlgModel	int	是	分散算法模式： • *0*：PBOC子密钥分散算法用于分散产生应用子密钥。8字节分散因子D，使用源密钥对16字节[D|D的非]采用源密钥的算法标识进行ECB模式加密。 • *1*：ECB模式加密16字节分散因子

名称	类型	是否必须	描述
			• *2*：ECB模式加密16字节分散因子，并复制扩展为32字节长度密钥。 **说明**： 仅限**subKeyFlag**为*N* • *3*：CBC模式加密16字节分散因子。
disperSeries	int	是	分散级数。 取值范围：01~08
disperFactor	String	是	分散因子，n级分散因子串联。 • 当**disperAlgModel**为*0*时，每级分散因子为8字节（16H）。 • 当**disperAlgModel**为*1*/*2*/*3*时，每级分散因子为16字节（32H）。
keyComponentNum	int	是	密钥成份个数。 取值范围：2~8

返回值

分散产生新密钥，并输出子密钥的密文和多个成分密文，成分密文采用子密钥的类型所对应的LMK加密。

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.115 encExportKey

KMC（Kdek）加密导出多条应用密钥（G2）。

```
public java.lang.String[] encExportKey(int encAlgModel,
                                       java.lang.Object proKey,
                                       java.lang.String keyData,
                                       java.lang.String cardCounter,
                                       int keyHeadLength,
                                       int disperSeries,
                                       java.lang.String disperFactor,
                                       java.util.List<java.util.List<java.lang.Object>> UDK)
```

```
                    throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
encAlgModel	int	是	加密算法模式： • *00*：ECB • *01*：CBC
proKey	-	是	保护密钥，用于加密保护被导出密钥的源密钥（厂商KMC）索引或密文。
keyData	String	是	密钥派生数据。
cardCounter	String	是	卡序列计数器（SCP02）。
keyHeadLength	int	是	每个密钥头长度，被导出密钥的单个密钥头的长度。 取值范围：00~20（即0-32字节）
disperSeries	int	是	分散级数。 取值范围：01~08
disperFactor	String	是	分散因子，n级分散因子串联。
keyderivationnum	-	否	要导出密钥个数NUM
keyderivationType1	-	否	导出密钥1类型
keyderivationKey1	-	否	导出密钥1
keyHead1	-	否	密钥头1
keyderivationTypen	-	否	导出密钥n类型
keyderivationKeyn	-	否	导出密钥n
keyHeadn	-	否	密钥头n
UDK	-	是	包含导出密钥类型，导出密钥，密钥头

返回值

- String[0]：密钥块密文

- String[1]：子密钥1校验值
- String[n]：子密钥n校验值

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.116 encConfidentialData

KMC（Sdek）加密敏感数据（G3）。

```
public java.lang.String encConfidentialData(
                                int encAlgModel,
                                java.lang.Object srcKey,
                                java.lang.String keyData,
                                java.lang.String cardCounter,
                                java.lang.String confData,
                                int paddingFlag)
                         throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
encAlgModel	int	是	加密算法模式： • *00*：ECB • *01*：CBC
srcKey	-	是	KMC源密钥，用于加密数据的源密钥（发行商KMC）索引或密文。
keyData	String	是	密钥派生数据。
cardCounter	String	是	卡序列计数器（SCP02）。
confData	String	是	机密数据。 长度取值：0000~03D8（即0~984字节）
paddingFlag	int	是	PAD标识标识加密前数据的填充规则。 取值范围：00~05

返回值

数据密文。

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.117 encData

KMC（Senc）加密数据（G4）。

```
public java.lang.String encData(int encAlgModel,
                                java.lang.Object srcKey,
                                java.lang.String keyData,
                                java.lang.String cardCounter,
                                java.lang.String confData,
                                int paddingFlag)
                         throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
encAlgModel	int	是	加密算法模式： • *00*：ECB • *01*：CBC
srcKey	-	是	KMC源密钥，用于加密数据的源密钥（发行商KMC）索引或密文。
keyData	String	是	密钥派生数据。
cardCounter	String	是	卡序列计数器（SCP02）。
confData	String	是	机密数据。 长度取值：0000~03D8（即0~984字节）
paddingFlag	int	是	PAD标识标识加密前数据的填充规则。 取值范围：00~05

返回值

数据密文。

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.118 encDataMAC

KMC（Scmac）计算数据C-MAC（G5）。

```
public java.lang.String encDataMAC(int macAlgModel,
                                   java.lang.Object srcKey,
                                   java.lang.String keyData,
                                   java.lang.String cardCounter,
                                   java.lang.String confData,
                                   int paddingFlag,
                                   int ICVFlag,
                                   java.lang.String ICV)
                            throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
macAlgModel	int	是	MAC算法模式： • *01*：ISO9797-1模式1，即FullTripleDESMAC • *03*：ISO9797-1模式3，即SingleDesFinalTripleDesMAC
srcKey	-	是	KMC源密钥，用于加密数据的源密钥（发行商KMC）索引或密文。
keyData	String	是	密钥派生数据。
cardCounter	String	是	卡序列计数器（SCP02）。
confData	String	是	机密数据。 长度取值：0000~03D8（即0~984字节）
paddingFlag	int	是	PAD标识标识加密前数据的填充规则。 取值范围：00~05
ICVFlag	int	是	ICV使用模式： • *0*：直接使用ICV域参与MAC运算。 • *1*：使用Sc-mac左8字节加密ICV域后参与MAC运算。
ICV	String	是	初始化向量，8字节。

返回值

输出的MAC值。

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.119 testDataMAC

KMC（Srmac）验证数据R-MAC（G6）。

```
public java.lang.String[] testDataMAC(int macAlgModel,
                                      java.lang.Object srcKey,
                                      java.lang.String keyData,
                                      java.lang.String cardCounter,
                                      java.lang.String confData,
                                      int paddingFlag,
                                      int ICVFlag,
                                      java.lang.String ICV,
                                      java.lang.String testMAC)
                               throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
macAlgModel	int	是	MAC算法模式： • *01*：ISO9797-1模式1，即FullTripleDESMAC • *03*：ISO9797-1模式3，即SingleDesFinalTripleDesMAC
srcKey	-	是	KMC源密钥，用于加密数据的源密钥（发行商KMC）索引或密文。
keyData	String	是	密钥派生数据。
cardCounter	String	是	卡序列计数器（SCP02）。
confData	String	是	机密数据。 长度取值：0000~03D8（即0~984字节）
paddingFlag	int	是	PAD标识标识加密前数据的填充规则。 取值范围：00~05
ICVFlag	int	是	ICV使用模式： • *0*：直接使用ICV域参与MAC运算。

名称	类型	是否必须	描述
			• *1*：使用Sc-mac左8字节加密ICV域后参与MAC运算。
ICV	String	是	初始化向量，8字节。
testMAC	String	是	待验证MAC，待验证的应答报文R-MAC，验证时忽略大小写。

返回值

输出的MAC值。

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.120 externalAuthentication

外部认证（G7）。

```
public java.lang.String externalAuthentication(
                                java.lang.Object srcKey,
                                java.lang.String keyData,
                                java.lang.String hostChallenge,
                                java.lang.String cardChallenge,
                                java.lang.String cardCipher)
                            throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
srcKey	-	是	KMC源密钥，用于加密数据的源密钥（发行商KMC）索引或密文。
keyData	String	是	密钥派生数据。
hostChallenge	String	是	8字节终端随机数。
cardChallenge	String	是	对SCP02，2字节计数器+6字节卡片随机数
cardCipher	String	是	卡片认证密文，8字节，用于主机对卡片的认证。

返回值

输出的MAC值。

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.121 proEncKMCSessionKey

保护密钥加密出导出KMC三条会话密钥（G8）。

```
public java.lang.String[] proEncKMCSessionKey(
                            int encAlgModel,
                            java.lang.String proKeyType,
                            java.lang.Object proKey,
                            java.lang.String prodisperFactor,
                            int proKeysessionType,
                            java.lang.String proKeysessionFactor,
                            java.lang.Object keyderivationKey,
                            java.lang.String keydata,
                            java.lang.String cardCounter)
                          throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
encAlgModel	int	是	加密算法模式： • *00*：ECB • *01*：CBC
proKeyType	String	是	保护密钥类型： • 000：KEK • 109：MDK
proKey	-	是	保护密钥，用于加密保护三条会话密钥的密钥索引或密文。
prodisperFactor	String	是	分散因子，n个分散因子的串联，每个分散因子必须为16个字节。
proKeysessionType	int	是	保护密钥的会话密钥产生模式： • *00*：不产生会话密钥。 • *01*：ECB模式加密8字节会话密钥因子，得8字节会话密钥。

名称	类型	是否必须	描述
			• *02*：ECB模式加密16字节会话密钥因子，得16字节会话密钥。 • *03*：密钥的左右8字节异或，得8字节会话密钥。 • *04*：取密钥的左8字节做为会话密钥。 • *05*：CBC模式加密16字节会话密钥因子，得16字节会话密钥。
proKeysessionFactor	String	是	保护密钥的会话密钥因子。 **说明**：仅当**proKeysessionType**为*01*/*02*/*05*时存在。 • **proKeysessionType**为*01*时，该域为8字节（16H），适用于产生PBOC规范的单长度会话密钥。 取值：6字节0x00||2字节ATC • **proKeysessionType**为*02*时，该域为16字节（32H），适用于产生PBOC规范的双长度会话密钥。 取值：6字节0x00||2字节ATC||6字节0x00||2字节ATC的非 • **proKeysessionType**为*05*时，该域为16字节（32H），适用于产生GP规范SCP02的卡片会话密钥。 取值：2字节密钥类型||2字节卡计数器||12字节0x00
keyderivationKey	-	是	导出密钥，被导出KMC的密钥索引或密文。
keydata	String	是	卡片个人化密钥数据keydata，由6字节KMCID和4字节芯片序号（CSN）组成，取6个最低有效字节用于分散产生Kenc/Kmac/Kdek。
cardCounter	String	是	卡片计数器。

返回值

- String[0]：Senc密文
- String[1]：Senc校验值
- String[2]：Scmac密文
- String[3]：Scmac校验值
- String[4]：Sdek密文
- String[5]：Sdek校验值

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.122 proRSAKey

KMC（Sdek）保护导出一对RSA密钥（GF）。

```
public java.util.List<byte[]> proRSAKey(int encAlgModel,
                                        java.lang.Object proKey,
                                        java.lang.String keyData,
                                        java.lang.String cardCounter,
                                        char keyStoreFlag,
                                        int tarIndex,
                                        java.lang.String privateKey)
                                 throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
encAlgModel	int	是	加密算法模式： • *00*：ECB • *01*：CBC
proKey	-	是	保护密钥，用于加密保护RSA密钥的KMC密钥索引或密文。
keyData	String	是	密钥派生数据，由6字节KMCID和4字节芯片序号（CSN）组成，取6个最低有效字节用于分散产生卡片Kdek。
cardCounter	String	是	卡序列计数器（SCP02），用于由Kdek产生Sdek会话密钥：Kdek3DES-CBC加密[‘0181’+2字节Counter+12个’00’]。

名称	类型	是否必须	描述
keyStoreFlag	char	否	RSA密钥索引标识。 • 取值为*K*，则**tarIndex**为4N模式。 • 取值为空，则**tarIndex**为2N模式。
tarIndex	int	是	RSA密钥索引号，RSA公钥在密码机内存储的索引号。取值： • 1~64 • *99*：2N模式，标识私钥使用**privateKey**。 • *9999*：4N模式，标识私钥使用**privateKey**。
privateKey	String	否	私钥数据，仅当**tarIndex**为*99*或*9999*时，存在LMK加密的私钥（包括m，e，d和5个CRT成份）。

返回值

- String[0]：公钥
- String[1]：私钥指数d
- String[2]：私钥分量P
- String[3]：私钥分量Q
- String[4]：私钥分量dP
- String[5]：私钥分量dQ
- String[6]：私钥分量qInv

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.123 proSM2Key

KMC（Sdek）保护导出一对SM2密钥（G0）。

```
public java.util.List<byte[]> proSM2Key(int encAlgModel,
                                        java.lang.Object proKey,
                                        java.lang.String keyData,
                                        java.lang.String cardCounter,
                                        int tarKeyAlg,
```

```
                                int tarIndex,
                                java.lang.String SM2PublicKey,
                                java.lang.String SM2privateKey)
                        throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
encAlgModel	int	是	加密算法模式： • *00*：ECB • *01*：CBC
proKey	-	是	保护密钥，用于加密保护SM2密钥对的KMC密钥索引或密文。
keyData	String	是	密钥派生数据，由6字节KMCID和4字节芯片序号（CSN）组成，取6个最低有效字节用于分散产生卡片Kdek。
cardCounter	String	是	卡序列计数器（SCP02），用于由Kdek产生Sdek会话密钥：Kdek3DES-CBC加密[‘0181’ +2字节Counter+12个’00’]。
tarKeyAlg	int	是	曲线标识07-国密-256新曲线
tarIndex	int	是	SM2密钥索引号，标识密码机内的密钥索引位置。取值： • 0001~0064 • *9999*：标识密钥使用**SM2PublicKey**和**SM2privateKey**。
SM2PublicKey	String	否	SM2公钥，仅当**tarIndex**为*9999*时存在。
SM2privateKey	String	否	私钥数据，仅当**tarIndex**为*9999*时，存在LMK加密的私钥密文。

返回值

- String[0]：公钥
- String[1]：私钥d密文

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.124 tKEncPINChangeKMC

TK加密的PIN密文转为KMC（Sdek）下加密（GD）。

```
public java.lang.String tKEncPINChangeKMC(
                                java.lang.String srcKeyType,
                                java.lang.Object srcKey,
                                java.lang.Object srcKMCKey,
                                java.lang.String keyData,
                                java.lang.String cardCounter,
                                java.lang.String srcPINBLOCKCipher,
                                int srcPINBLOCKFormat,
                                java.lang.String srcPINBLOCKId,
                                int tarPINBLOCKFormat,
                                java.lang.String tarPINBLOCKId)
                            throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
srcKeyType	String	是	源密钥类型： • 000：ZMK/KEK • 00A：ZEK/DEK • 00B：TEK
srcKey	int	是	源密钥，用于加密PINBLOCK的源密钥索引。
	String		源密钥，用于加密PINBLOCK的源密钥密文。
srcKMCKey	int	是	目的KMC密钥，用于加密PINBLOCK的目的主密钥（IMK）索引。
	String		目的KMC密钥，用于加密PINBLOCK的目的主密钥（IMK）密文。
keyData	String	是	密钥派生数据，由6字节KMCID和4字节芯片序号（CSN）组成，取6个最低有效字节用于分散产生卡片Kdek。

名称	类型	是否必须	描述
cardCounter	String	是	卡序列计数器（SCP02），用于由Kdek产生Sdek会话密钥：Kdek3DES-CBC加密[‘0181’+2字节Counter+12个’00’]。
srcPINBLOCKCipher	String	是	源PINBLOCK密文。 • 在源密钥下加密的PINBLOCK密文源密钥方案为R/P/L/M/N时为32H。 • 其他情况下为16H。
srcPINBLOCKFormat	int	是	源PINBLOCK格式，源密钥下加密PIN数据块的格式代码。
srcPINBLOCKId	String	是	源账号。
tarPINBLOCKFormat	int	是	目标PINBLOCK格式，目标密钥下加密PIN数据块的格式代码。
tarPINBLOCKId	String	是	目标账号，即用户主账号。 • 当目标PIN数据块格式为04时，该域为18N，去除校验位的18位主账号，不足18位则右对齐左填F。 • 当PIN数据块格式为其他值时，该域为12N，去除校验位的最右12位主账号。

返回值

- String[0]：公钥
- String[1]：私钥d密文

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.125 eMVchackingARQC

EMV4.X验证ARQC/TC/AAC，可选的产生ARPC（KW）。

```
public java.lang.String[] eMVchackingARQC(
                                int modelAlg,
                                int planId,
                                java.lang.Object MKAC,
                                java.lang.String IVAC,
                                java.lang.String PANOrder,
```

```
                    java.lang.String parameteBH,
                    java.lang.String ATC,
                    java.lang.String transctionsData,
                    java.lang.String applicationCipher,
                    java.lang.String ARC,
                    java.lang.String CSU,
                    java.lang.String data)
                throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
modelAlg	int	是	模式标志： • *0*：仅执行ARQC验证。 • *1*：执行ARQC验证，和EMV4.1方法一ARPC产生。 • *2*：仅执行EMV4.1方法一ARPC产生。 • *3*：执行ARQC验证，和EMV4.1方法二ARPC产生。 • *4*：仅执行EMV4.1方法二ARPC产生。
planId	int	是	方案ID，用于加密PINBLOCK的源密钥索引或密文。 • *0*：VIS1.4.0或M/Chip4使用EMV4.1卡密钥离散方法A及EMV2000会话密钥离散方式。 • *1*：VIS1.4.0或M/Chip4使用EMV4.1卡密钥离散方法B及EMV2000会话密钥离散方式。 • *2*：VIS1.4.0或M/Chip4使用EMV4.1卡密钥离散方法A及EMV通用会话密钥离散方式。 • *3*：VIS1.4.0或M/Chip4使用EMV4.1卡密钥离散方法B及EMV通用会话密钥离散方式。 • *9*：PBOC2.0
MKAC	-	是	MK-AC，发行商的应用主密钥索引或密文。
IV-AC	String	否	IV-AC，用于离散产生卡片会话密钥的初始向量（EMV2000过程密钥离散方式使用）。

名称	类型	是否必须	描述
			说明： 仅当**planId**为*0*或*1*时存在。
PANOrder	String	是	PAN/PAN序列号，离散卡片密钥使用的帐号或者帐号序列号，该数据的填充由应用完成。
parameteBH	String	否	B/H参数，用于产生会话密钥（EMV2000过程密钥离散方式使用）0=B为2，H为161=B为4，H为8。 **说明**： 仅当**planId**为*0*或*1*时存在分支因子和数高参数。
ATC	String	是	ATC，用于产生会话密钥。
transctionsData	String	否	交易数据，用于计算ARQC的明文数据。 **说明**： 仅当**modelAlg**为*0*/*1*/*3*时存在。
applicationCipher	String	是	ARQC/TC/AAC，待验证的应用密文ARQC/TC/AAC，以及用于ARPC的计算。
ARC	String	否	ARC，用于计算ARQC。 **说明**： 仅当**modelAlg**为*1*/*2*时存在认证应答码。
CSU	String	否	CSU，用于计算ARPC。 **说明**： 仅当**modelAlg**为*3*/*4*时存在卡状态更新值。
data	String	否	认证数据，用于计算ARPC。 数据长度：0~8 **说明**：

名称	类型	是否必须	描述
			仅当**modelAlg**为*3*/*4*时存在专有的认证数据。

返回值

- String[0]：公钥
- String[1]：私钥d密文

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.126 eMVchackingPIN

EMV4.X脚本安全报文/PIN修改（KY）。

```
public java.lang.String[] eMVchackingPIN(
                                int modelAlg,
                                int planId,
                                java.lang.Object mksmi,
                                java.lang.String ivsmi,
                                java.lang.String panOrder,
                                java.lang.String parameteBH,
                                java.lang.String atc,
                                java.lang.String ac,
                                java.lang.String proData,
                                java.lang.Object mksmc,
                                java.lang.String ivsmc,
                                java.lang.Object tk,
                                int offset,
                                java.lang.String cipherData,
                                int srcPINType,
                                java.lang.Object srcPINEncKey,
                                int srcPinblockAlg,
                                int tarPinblockAlg,
                                java.lang.String id,
                                java.lang.Object mkac)
                            throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
modelAlg	int	是	模式标志： • *0*：仅计算MAC，使用EMV4.1卡密钥离散方法A及EMV2000会话密钥离散方式。

名称	类型	是否必须	描述
			• *2*：MAC计算和数据加密。 • *4*：MAC计算和PINCHANGE。 • *5*：仅计算MAC，使用方案ID指定算法。
planId	int	是	方案ID，用于加密PINBLOCK的源密钥索引或密文。 • *0*：VIS1.4.0或M/Chip4使用EMV4.1卡密钥离散方法A及EMV2000会话密钥离散方式。（与**modelAlg**为*0*相同） • *1*：M/Chip4使用EMV4.1卡密钥离散方法A及EMV2000会话密钥离散方式。 • *4*：CCD使用EMV4.1卡密钥离散方法B及EMV2000会话密钥离散方式。 • *5*：VIS1.4.0使用EMV4.1卡密钥离散方法A及EMV通用会话密钥离散方式。 • *6*：M/Chip4使用EMV4.1卡密钥离散方法A及EMV通用会话密钥离散方式。 • *7*：CCD使用EMV4.1卡密钥离散方法B及EMV通用会话密钥离散方式。 • *9*：PBOC2.0
mksmi	-	是	MK-SMI，用于计算MAC的主密钥索引或密文。
ivsmi	String	否	IV-SMI **说明**： 仅当**modelAlg**为*0*或**planId**为*0*/*1*/*4*时存在。
panOrder	String	是	PAN/PAN序列号，**planId**为*4*/*7*时，为**PAN/PAN序列号长度**标识的长度。
parameteBH	String	否	B/H参数，用于产生会话密钥0=B为2，H为161=B为4，H为8。 **说明**： 仅当**modelAlg**为*0*或**planId**为*0*/*1*/*4*时存在分支因子和数高参数。

名称	类型	是否必须	描述
atc	String	否	ATC，用于按EMV2000和PBOC会话密钥离散方式产生会话密钥。  **说明**： 仅当**modelAlg**为*0*或**planId**为*0*/*1*/*4*/*9*时存在应用交易序号。
ac	String	否	AC，用于按EMV通用会话密钥离散方式产生会话密钥。  **说明**： 仅当**planId**为*5*/*6*/*7*时存在应用密文。
proData	String	是	明文数据，用于参与MAC计算的明文数据。
mksmc	-	否	MK-SMC，用于加密计算的主密钥索引或密文。  **说明**： 仅当**modelAlg**为*2*/*4*时存在。
ivsmc	String	否	IV-SMC  **说明**： 仅当**modelAlg**为*2*/*4*且**planId**为*0*/*1*/*4*时存在。
tk	-	否	TK（ZEK/DEK），输入的密文数据的源加密密钥的索引或密文。  **说明**： 仅当**modelAlg**为2时存在。
offset	int	否	偏移量，在计算MAC时，将MK-SMC卡片会话密钥加密的密文数据插入到参与MAC计算的明文数据的位置。  **说明**： 仅当**modelAlg**为*2*/*4*时存在。

名称	类型	是否必须	描述
cipherData	String	是	密文数据，用于参与MAC计算的明文数据。
srcPINType	int	否	源PIN加密密钥类型。 • *0*：ZPK • *1*：TPK **说明**： 仅当**modelAlg**为*4*时存在。
srcPINEncKey	-	否	源PIN加密密钥，用于计算ARQC。 **说明**： 仅当**modelAlg**为*1*/*4*时存在认证应答码。
srcPinblockAlg	int	否	源PINBLOCK格式。 **说明**： 仅当**modelAlg**为*4*时存在。
tarPinblockAlg	int	否	目的PINBLOCK格式，指定MK-SMC会话密钥加密PIN块时的格式。 • *34*：标准EMV格式 • *35*：Mastercard格式 • *41*：VISA/PBOC不使用当前PIN • *42*：VISA/PBOC使用当前PIN **说明**： 仅当**modelAlg**为*4*时存在。
id	String	否	账号，PIN转换中使用的帐号（不包含校验位）。当源PINBLOCK=4时为18H，不足18位左补字符F。 **说明**： 仅当**modelAlg**为*4*时存在。
mkac	-	否	MK-AC，用来产生PINBLOCK的填充（UDK-A）发卡行应用主密钥的索引或密文。

名称	类型	是否必须	描述
			说明： 仅 当**modelAlg**为*4*且**tarPinblockAlg**为*41*/*42*时存在。

返回值

- String[0]：MAC
- String[1]：重新加密的密文数据长度
- String[2]：重新加密的密文数据

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.127 pbocchackingPIN

PBOC脱机PIN修改/加密（KX）。

```
public java.lang.String pbocchackingPIN(
                                int planId,
                                java.lang.String appKeyType,
                                java.lang.Object applicationMainKey,
                                java.lang.String panOrder,
                                java.lang.String atc,
                                int pinblockFormat1,
                                int pinInputAlg,
                                java.lang.String proPINNew,
                                java.lang.String proPINOld,
                                java.lang.Object srcPINEncKey,
                                int pinblockFormat2,
                                java.lang.String newPinCipher,
                                java.lang.String oldPinCipher,
                                java.lang.String id)
                            throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
planId	int	是	方案ID，指定密钥分散算法和会话密钥离散算法及填充模式的规范标识。 *9*：PBOC3.0
appKeyType	String	是	应用主密钥类型：

名称	类型	是否必须	描述
			109：MDK/MK-AC
applicationMainKey	-	是	应用主密钥，用于加密PIN块的卡应用主密钥索引或密文。该密钥按指定规范定义的算法进行分散和产生会话密钥后再加密PIN块。
panOrder	String	是	PAN/PAN序列号，用于分散MDK产生卡片UDK的分散因子账号+账号应用序列号取最右16个数字，若小于16个则后对齐左补0。
atc	String	否	ATC，2字节，应用交易计数器用于计算交易会话密钥。
pinblockFormat1	int	是	PINBLOCK格式1： • *34*：标准EMV格式 • *35*：Mastercard格式 • *41*：VISA/PBOC不使用当前PIN • *42*：VISA/PBOC使用当前PIN
pinInputAlg	int	是	PIN输入模式： • *1*：明文PIN • *2*：ZPK加密的密文PIN • *3*：TPK加密的密文PIN
proPINNew	String	否	明文PIN（新） **说明**： 仅当**pinInputAlg**为*1*时存在。
proPINOld	String	否	明文PIN（旧） **说明**： 仅当**pinInputAlg**为*1*且**pinblockFormat1**为*42*时存在。
srcPINEncKey	-	否	源PIN加密密钥，用于加密PIN的源ZPK/TPK密钥索引或密文。 **说明**：

名称	类型	是否必须	描述
			仅当**pinInputAlg**不为*1*时存在。
pinblockFormat2	int	否	PINBLOCK格式2，ZPK/TPK下加密PIN数据块的格式代码。 **说明**： 仅当**pinInputAlg**不为*1*时存在。
newPinCipher	String	否	PIN密文（新），在TPK/ZPK下加密的新PINBLOCK密文，密钥方案为R/P/L/M/N时该域为32H，否则16H。 **说明**： 仅当**pinInputAlg**不为*1*时存在。
oldPinCipher	String	否	PIN密文（旧） **说明**： 仅当**pinInputAlg**不为*1*且**pinblockFormat1**为*42*时存在。
id	String	否	账号，PIN转换中使用的帐号（不包含校验位）。当源PINBLOCK=4时为18H，不足18位左补字符F。

返回值

PIN密文数据

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.128 numberPINBLOCKEnc

数字PINBLOCK转加密（D7）。

```
public java.lang.String numberPINBLOCKEnc(
                        java.lang.Object srcKey,
                        int srcKeyDisperSeries,
                        java.lang.String srcKeyDisperFactor,
                        int srcKeySessionAlg,
                        java.lang.String srcKeySessionFactor,
```

```
                    java.lang.Object tarKey,
                    int tarKeyDisperSeries,
                    java.lang.String tarKeyDisperFactor,
                    int tarKeySessionAlg,
                    java.lang.String tarKeySessionFactor,
                    int maxPINLength,
                    java.lang.String srcPINBLOCKCipher,
                    int srcPINBLOCKFormat,
                    java.lang.String srcId,
                    int tarPINBLOCKFormat,
                    java.lang.String tarId)
                throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
srcKey	-	是	源ZPK密钥，用于加密PIN的源ZPK密钥索引或密文。
srcKeyDisperSeries	int	是	源密钥分散级数。 取值范围：00~08
srcKeyDisperFactor	String	是	源密钥分散因子，n级分散因子串联，每级分散因子为8字节（16H）。
srcKeySessionAlg	int	是	源密钥会话密钥模式： • *00*：不产生会话密钥。 • *01*：ECB模式加密8字节会话密钥因子[6字节‘00’||2字节ATC]，得8字节会话密钥。 • *02*：ECB模式加密16字节会话密钥因子[6字节‘00’||2字节ATC||6字节‘00’||2字节ATC的非]，得16字节会话密钥。 • *03*：密钥的左右8字节异或，得8字节会话密钥。 • *04*：取密钥的左8字节做为会话密钥。
srcKeySessionFactor	String	否	源密钥会话密钥因子，通常为2字节ATC。 **说明**： 仅当**srcKeySessionAlg**为*01*或*02*时存在。
tarKey	-	是	目标ZPK密钥

名称	类型	是否必须	描述
tarKeyDisperSeries	int	是	目标密钥分散级数
tarKeyDisperFactor	Stromg	是	目标密钥分散因子
tarKeySessionAlg	int	是	目标密钥会话密钥模式： • *00*：不产生会话密钥。 • *01*：ECB模式加密8字节会话密钥因子[6字节‘00’||2字节ATC]，得8字节会话密钥。 • *02*：ECB模式加密16字节会话密钥因子[6字节‘00’||2字节ATC||6字节‘00’||2字节ATC的非]，得16字节会话密钥。 • *03*：密钥的左右8字节异或，得8字节会话密钥。 • *04*：取密钥的左8字节做为会话密钥。
tarKeySessionFactor	String	是	目标密钥会话密钥因子
maxPINLength	int	是	最大PIN长度
srcPINBLOCKCipher	String	是	源PINBLOCK密文
srcPINBLOCKFormat	int	是	源PINBLOCK格式
srcId	String	是	源账号
tarPINBLOCKFormat	int	是	目标PINBLOCK格式
tarId	String	是	目标账号

返回值

PIN密文数据

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.129 s7_numberPINBLOCKEnc

数字PINBLOCK转加密（通用（S7））。

```
public java.lang.String s7_numberPINBLOCKEnc(
                        java.lang.String srcKeyType,
                        java.lang.Object srcKey,
                        int srcKeyDisperSeries,
                        java.lang.String srcKeyDisperFactor,
                        int srcKeySessionAlg,
                        java.lang.String srcKeySessionFactor,
                        java.lang.String tarKeyType,
                        java.lang.Object tarKey,
                        int tarKeyDisperSeries,
                        java.lang.String tarKeyDisperFactor,
                        int tarKeySessionAlg,
                        java.lang.String tarKeySessionFactor,
                        int maxPINLength,
                        java.lang.String srcPINBLOCKCipher,
                        int srcPINBLOCKFormat,
                        java.lang.String srcId,
                        int tarPINBLOCKFormat,
                        java.lang.String tarId)
                    throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
srcKey	-	是	源ZPK密钥，用于加密PIN的源ZPK密钥索引或密文。
srcKeyDisperSeries	int	是	源密钥分散级数。 取值范围：00~08
srcKeyDisperFactor	String	是	源密钥分散因子，n级分散因子串联，每级分散因子为8字节（16H）。
srcKeySessionAlg	int	是	源密钥会话密钥模式： • *00*：不产生会话密钥。 • *01*：ECB模式加密8字节会话密钥因子[6字节‘00’||2字节ATC]，得8字节会话密钥。 • *02*：ECB模式加密16字节会话密钥因子[6字节‘00’||2字节ATC||6字

名称	类型	是否必须	描述
			节‘00’ ||2字节ATC的非]，得16字节会话密钥。 • *03*：密钥的左右8字节异或，得8字节会话密钥。 • *04*：取密钥的左8字节做为会话密钥。
srcKeySessionFactor	String	否	源密钥会话密钥因子，通常为2字节ATC。 **说明**： 仅当**srcKeySessionAlg**为*01*或*02*时存在。
tarKey	-	是	目标ZPK密钥
tarKeyDisperSeries	int	是	目标密钥分散级数
tarKeyDisperFactor	Stromg	是	目标密钥分散因子
tarKeySessionAlg	int	是	目标密钥会话密钥模式： • *00*：不产生会话密钥。 • *01*：ECB模式加密8字节会话密钥因子[6字节‘00’ ||2字节ATC]，得8字节会话密钥。 • *02*：ECB模式加密16字节会话密钥因子[6字节‘00’ ||2字节ATC||6字节‘00’ ||2字节ATC的非]，得16字节会话密钥。 • *03*：密钥的左右8字节异或，得8字节会话密钥。 • *04*：取密钥的左8字节做为会话密钥。
tarKeySessionFactor	String	是	目标密钥会话密钥因子
maxPINLength	int	是	最大PIN长度
srcPINBLOCKCipher	String	是	源PINBLOCK密文
srcPINBLOCKFormat	int	是	源PINBLOCK格式

名称	类型	是否必须	描述
srcId	String	是	源账号
tarPINBLOCKFormat	int	是	目标PINBLOCK格式
tarId	String	是	目标账号

返回值

PIN密文数据

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.130 transferPinCipherTpk2ublicKey

将字符PIN由由TPK加密转为公钥加密（N5）。

```
public byte[] transferPinCipherTpk2ublicKey(java.lang.Object tpkKey,
                                            java.lang.String tpkPIN,
                                            int PINBLOCK,
                                            java.lang.String pan,
                                            int publicKeyFlag,
                                            int publicindex,
                                            byte[] publicKey,
                                            byte[] keyCheck,
                                            byte[] publicKeyMAC,
                                            int PINflag,
                                            java.lang.String ID,
                                            int publicKeyPINBLOCK,
                                            int publicKeyPadding)
                                     throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
tpkKey	-	是	TPK密钥，用于加密PIN的TPK密钥索引或密文。
tpkPIN	String	是	在TPK密钥下加密的PINBLOCK密文，ExpandedHex格式。
PINBLOCK	int	是	字符PINBLOCK格式。
pan	String	是	账号PAN。

名称	类型	是否必须	描述
publicKeyFlag	int	是	公钥算法标识。
publicindex	int	是	公钥索引号。
publicKey	byte[]	是	公钥，ASN.1格式DER编码。
keyCheck	byte[]	是	认证数据，用于计算公钥MAC的额外的数据，不能包含分号（；）字符。
publicKeyMAC	byte[]	是	公钥MAC值，用于验证公钥的合法可信。
PINflag	int	是	公钥加密PIN组成格式。 • *00*：ID长度(2N)+ID码+PIN长度(2N)+PIN明文 • *01*：PIN明文块 • *02*：PIN长度(2N)+PIN明文+ID长度(2N)+ID码
ID	String	否	ID码，ExpandedHex格式。 **说明**： 仅当**PINflag**为*00*/*02*时存在。
publicKeyPINBLOCK	int	是	
publicKeyPadding	int	是	公钥加密的填充模式。

返回值

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.131 numberPINPriEnc

将字数字PIN从从ZPK下加密转换到私有算法加密（CB）。

```
public byte[] numberPINPriEnc(java.lang.Object srcKey,
                              java.lang.String tarKeyType,
                              java.lang.Object tarKey,
                              int maxPINLength,
                              java.lang.String srcPINBLOCKCipher,
                              int srcPINBLOCKFormat,
```

```
                    java.lang.String srcId,
                    int priPINEncAlg,
                    java.lang.String date)
            throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
srcKey	-	是	源ZPK密钥，用于加密PIN的源ZPK密钥索引或密文。
tarKeyType	String	是	目标PIN加密密钥类型： • 001：ZPK • 002：PVK
tarKey	-	是	目标PIN加密密钥
maxPINLength	int	是	最大PIN长度
srcPINBLOCKCipher	String	是	源PINBLOCK密文
srcPINBLOCKFormat	int	是	源PINBLOCK格式
srcId	String	是	源账号
priPINEncAlg	int	是	私有PIN加密算法标识： • *01*：乌海银行特殊DES算法 • *02*：浙商银行电话项目私有算法 • *03*：山西农信PINBLOCK加密算法 • *04~99*：预留，当前版本不支持
date	String	否	日期，格式为yyyymmdd。 **说明**： 仅当**priPINEncAlg**为*01*时存在。

返回值

目标PINBLOCK密文。

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.132 cipherCompositionSynKey

由密文成份合成一个密钥（A4）。

```
public java.lang.String[] cipherCompositionSynKey(
                                int keyCompositionNumbenr,
                                java.lang.String keyType,
                                char keyAlg,
                                java.lang.String... keyComposition)
                          throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
keyCompositionNumbenr	int	是	密钥成分个数，用于加密PIN的源ZPK密钥索引或密文。
keyType	String	是	密钥类型： • 000：ZMK • 001：ZPK • 002：PVK/TPK/TMK • 003：TAK • 008：ZAK • 00A：ZEK • 109：MDK • 402：CVK
keyAlg	char	是	密钥标识，在LMK下加密的密钥密文标识：*Z*/*X*/*Y*/*U*/*T*。
keyComposition	String	是	密钥成份集合，长度等于**keyCompositionNumbenr**。

返回值

- String[0]：密钥密文
- String[1]：密钥校验值

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.133 makeOneZPK

产生一个ZPK（IA）。

```
public java.lang.String[] makeOneZPK(java.lang.Object zmkCipher,
                                     char KeyAlgZMK,
                                     char keyAlgLMK,
                                     int keySynTept)
                              throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
zmkCipher	-	是	ZMK密文，ZMK密钥索引或LMK006~008下加密的ZMK密文。
KeyAlgZMK	char	是	密钥标识（ZMK）
keyAlgLMK	char	是	密钥标识（LMK）
keySynTept	int	是	密钥校验值类型（KCV） • *0*：KCV 16H • *1*：KCV 6H

返回值

- String[0]：ZPK密文（ZMK）
- String[1]：ZPK密文（ZMK）
- String[2]：密钥校验值

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.134 zpkSynZMKToLMK

ZPK从ZMK加密转换为LMK加密（FA）。

```
public java.lang.String[] zpkSynZMKToLMK(
```

```
                        java.lang.Object zmkCipher,
                        java.lang.Object zpkCipherZMK,
                        char KeyAlgZMK,
                        char keyAlgLMK,
                        int keySynTept)
                throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
zmkCipher	-	是	ZMK密文，ZMK密钥索引或LMK006~008下加密的ZMK密文。
zpkCipherZMK	-	是	ZPK密文（ZMK），ZMK密钥索引。
KeyAlgZMK	char	是	密钥标识（ZMK）
keyAlgLMK	char	是	密钥标识（LMK）
keySynTept	int	是	密钥校验值类型（KCV） • *0*：KCV 16H • *1*：KCV 6H

返回值

- String[0]：ZPK密文（LMK）
- String[1]：密钥校验值

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.135 zpkSynLMKToZMK

ZPK从从LMK加密转换为ZMK加密（GC）。

```
public java.lang.String[] zpkSynLMKToZMK(
                        java.lang.Object zmkCipher,
                        java.lang.Object zpkcipherLMK,
                        char KeyAlgZMK,
                        char keyAlgLMK,
                        int keySynTept)
```

```
                    throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
zmkCipher	-	是	ZMK密文（LMK），ZMK密钥索引或LMK006~008下加密的ZMK密文。
zpkcipherLMK	-	是	ZPK密文（LMK），ZPK密钥索引或LMK009~011加密的ZPK密钥密文。
KeyAlgZMK	char	是	密钥标识（ZMK）
keyAlgLMK	char	是	密钥标识（LMK）
keySynTept	int	是	密钥校验值类型（KCV） • *0*：KCV 16H • *1*：KCV 6H

返回值

- String[0]：ZPK密文（ZMK）
- String[1]：密钥校验值

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.136 makeOneZEK

产生一个ZEK/ZAK（FI）。

```
public java.lang.String[] makeOneZEK(int keyTypeAlg,
                                     java.lang.Object zmkCipher,
                                     char KeyAlgZMK,
                                     char keyAlgLMK,
                                     int keySynTept)
                              throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
keyTypeAlg	int	是	密钥类型标志 • *0*：ZEK

名称	类型	是否必须	描述
			• *1*：ZAK
zmkCipher	-	是	ZMK密文，ZMK密钥索引或LMK006~008下加密的ZMK密文。
KeyAlgZMK	char	是	密钥标识（ZMK）
keyAlgLMK	char	是	密钥标识（LMK）
keySynTept	int	是	密钥校验值类型（KCV） • *0*：KCV 16H • *1*：KCV 6H

返回值

- String[0]：密钥密文（ZMK）
- String[1]：密钥密文（LMK）
- String[2]：密钥校验值

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.137 zekSynZMKToLMK

ZPK从ZMK加密转换为LMK加密（FA）。

```
public java.lang.String[] zekSynZMKToLMK(
                                int keyTypeAlg,
                                java.lang.Object zmkCipher,
                                java.lang.Object zekCipherZMK,
                                char KeyAlgZMK,
                                char keyAlgLMK,
                                int keySynTept)
                         throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
keyTypeAlg	int	是	密钥类型标志： • *0*：ZEK • *1*：ZAK

名称	类型	是否必须	描述
zmkCipher	-	是	ZMK密文（LMK），ZMK密钥索引或LMK006~008下加密的ZMK密文。
zekCipherZMK	-	是	ZEK/ZAK密文（ZMK），ZMK密钥索引。
KeyAlgZMK	char	是	密钥标识（ZMK）
keyAlgLMK	char	是	密钥标识（LMK）
keySynTept	int	是	密钥校验值类型（KCV） • *0*：KCV 16H • *1*：KCV 6H

返回值

- String[0]：ZPK密文（LMK）
- String[1]：密钥校验值

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.138 zekSynLMKToZMK

ZEK/ZAK从LMK加密转换为ZMK加密（FM）。

```
public java.lang.String[] zekSynLMKToZMK(
                                int keyTypeAlg,
                                java.lang.Object zmkCipher,
                                java.lang.Object zekcipherLMK,
                                char KeyAlgZMK,
                                char keyAlgLMK,
                                int keySynTept)
                          throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
keyTypeAlg	int	是	密钥类型标志： • *0*：ZEK • *1*：ZAK

名称	类型	是否必须	描述
zmkCipher	-	是	ZMK密文（LMK），ZMK密钥索引或LMK006~008下加密的ZMK密文。
zekcipherLMK	-	是	ZEK/ZAK密文（LMK），ZEK密钥索引或LMK009~011加密的ZEK密钥密文。
KeyAlgZMK	char	是	密钥标识（ZMK）
keyAlgLMK	char	是	密钥标识（LMK）
keySynTept	int	是	密钥校验值类型（KCV） • *0*：KCV 16H • *1*：KCV 6H

返回值

- String[0]：ZPK密文（ZMK）
- String[1]：密钥校验值

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.139 makeOneTMK

产生一个TMK，TPK，PVK（HC）。

```
public java.lang.String[] makeOneTMK(java.lang.Object tmkCipher,
                                     char KeyAlgZMK,
                                     char keyAlgLMK,
                                     int keySynTept)
                              throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
tmkCipher	-	是	TMK密文，TMK密钥索引或LMK021~023下加密的TMK密文。
KeyAlgZMK	char	是	密钥标识（ZMK）
keyAlgLMK	char	是	密钥标识（LMK）
keySynTept	int	是	密钥校验值类型（KCV）

名称	类型	是否必须	描述
			• *0*：KCV 16H • *1*：KCV 6H

返回值

- String[0]：密钥密文（TMK）
- String[1]：密钥密文（LMK）
- String[2]：密钥校验值

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.140 makeOneTAK

产生一个TAK（HA）。

```
public java.lang.String[] makeOneTAK(java.lang.Object tmkCipher,
                                     char KeyAlgZMK,
                                     char keyAlgLMK,
                                     int keySynTept)
                              throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
tmkCipher	-	是	TMK密文，TMK密钥索引或LMK021~023下加密的TMK密文。
KeyAlgZMK	char	是	密钥标识（ZMK）
keyAlgLMK	char	是	密钥标识（LMK）
keySynTept	int	是	密钥校验值类型（KCV） • *0*：KCV 16H • *1*：KCV 6H

返回值

- String[0]：密钥密文（TMK）
- String[1]：密钥密文（LMK）

- String[2]：密钥校验值

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.141 takSynZMKToLMK

将TAK从ZMK下加密转为LMK下加密（MI）。

```
public java.lang.String[] takSynZMKToLMK(
                                java.lang.Object zmkCipher,
                                java.lang.Object takCipherZMK,
                                char KeyAlgZMK,
                                char keyAlgLMK,
                                int keySynTept)
                         throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
zmkCipher	-	是	ZMK密文（LMK），ZMK密钥索引或LMK006~008下加密的ZMK密文。
takCipherZMK	-	是	TAK密文（ZMK），ZMK下加密的TAK密文。
KeyAlgZMK	char	是	密钥标识（ZMK）
keyAlgLMK	char	是	密钥标识（LMK）
keySynTept	int	是	密钥校验值类型（KCV） • *0*：KCV 16H • *1*：KCV 6H

返回值

- String[0]：TAK密文（LMK）
- String[1]：密钥校验值

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.142 takCountDataMAC

TAK计算数据MAC（MA）。

```
public java.lang.String takCountDataMAC(java.lang.Object takKey,
                                        java.lang.String data)
                                 throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
takKey-TAK	-	是	TAK密钥索引或LMK024~026下加密的TAK密文。
data	-	是	数据，要产生MAC所用的数据，最大4096字节。

返回值

MAC

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.143 checkDataMAC

TAK验证数据MAC（MC）。

```
public java.lang.String checkDataMAC(java.lang.Object takKey,
                                     java.lang.String AMC,
                                     java.lang.String data)
                              throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
takKey-TAK	-	是	TAK密钥索引或LMK024~026下加密的TAK密文。

名称	类型	是否必须	描述
data	-	是	数据，要产生MAC所用的数据，最大4096字节。

返回值

MAC

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.144 zpkCalculateMAC

银联应用系统，在线分发ZPK验证密钥，密钥类型只支持ZPK。

```
public java.lang.String zpkCalculateMAC(int Flag,
                                        java.lang.String IV,
                                        java.lang.Object Key,
                                        byte[] data,
                                        int inputMacLength)
                                 throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
Flag	int	是	报文标识块。
IV	String	否	用于计算MAC/TAC的初始向量。 • 当ZAK密钥标识为*Z*/*X*/*U*时，该域为16H。 • 当ZAK密钥标识为*R*/*P*/*L*时，该域为32H。 **说明**： 仅当**Flag**为*2*/*3*时有此域。
Key	int	是	ZPK密钥索引。 取值范围：0~2048
	String		LMK009~011下加密的ZPK密文。
data	byte[]	是	计算MAC的数据，长度：0~4096字节。

名称	类型	是否必须	描述
inputMacLength	int	是	输出MAC长度，输出MAC/TAC值长度的字节数 0x01-0x10。

返回值

返回生成的MAC值

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.145 calculateMAC

ZAK/TAK产生X9.9和和X9.19的文报文MAC（MS）。

```
public java.lang.String calculateMAC(int Flag,
                                      int keyType,
                                      int keyLength,
                                      int dataType,
                                      java.lang.Object Key,
                                      java.lang.String IV,
                                      byte[] data)
                               throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
Flag	int	是	报文标识块。
keyType	int	是	密钥类型： • *0*：TAK • *1*：ZAK
keyLength	int	是	密钥长度，0~8字节，单长度DES密钥。
dataType	int	是	数据类型： • *0*：二进制 • *1*：扩展十六进制
Key	-	是	TAK/ZAK密钥索引或LMK下加密的TAK/ZAK密文。
IV	String	否	用于计算MAC/TAC的初始向量

名称	类型	是否必须	描述
			• 当ZAK密钥标识为*Z*/*X*/*U*时，该域为16H。 • 当ZAK密钥标识为*R*/*P*/*L*时，该域为32H。 **说明**： 仅当**Flag**为*2*/*3*时有此域。
data	byte[]	是	计算MAC的数据，0~4096字节。

返回值

返回生成的MAC值

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.146 transferPinTpkToLmk

将PIN由TPK加密转换为LMK加密（JC）。

```
public java.lang.String transferPinTpkToLmk(
                                java.lang.Object srcTPKKey,
                                java.lang.String srcPinBlockCipher,
                                int pinBlockAlg,
                                java.lang.String userId)
                         throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
srcTPKKey	-	是	源TPK密钥
srcPinBlockCipher	String	是	源PINBLOCK密文
pinBlockAlg	int	是	PINBLOCK格式
userId	String	是	账号

返回值

PIN密文

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.147 transferPinTpkToZpk

将PIN由TPK加密转换为ZPK加密（CA）。

```
public java.lang.String[] transferPinTpkToZpk(
                                java.lang.Object srcTPKKey,
                                java.lang.Object tarZPKKey,
                                int maxPinLength,
                                java.lang.String srcPinBlockCipher,
                                int srcPinBlockAlg,
                                int tarPinBlockAlg,
                                java.lang.String userId)
                          throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
srcTPKKey	-	是	源TPK密钥
tarZPKKey	-	是	目标ZPK密钥
maxPinLength	int	是	最大PIN长度
srcPinBlockCipher	String	是	源PINBLOCK密文
pinBlockAlg	int	是	PINBLOCK格式
tarPinBlockAlg	int	是	目标PINBLOCK格式
userId	String	是	账号

返回值

PIN长度，目标PINBLOCK密文，目标PINBLOCK格式。

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.148 produceIbmPinOffset

产生IBM PIN Offset（DE）。

```
public java.lang.String produceIbmPinOffset(
                            java.lang.Object pvkKey,
                            java.lang.String pinCipher,
                            int pinCheckLength,
                            java.lang.String userId,
                            java.lang.String transferTable,
                            java.lang.String pinCheckData)
                        throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
pvkKey	-	是	PVK密文
pinCipher	String	是	pin密文
pinCheckLength	int	是	PIN校验长度
userId	String	是	账号
transferTable	String	是	十进制转换表
pinCheckData	String	是	PIN校验数据

返回值

pinOffset

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.149 ibmGetPin

使用IBM方式得到一个PIN（EE）。

```
public java.lang.String ibmGetPin(java.lang.Object pvkKey,
                            java.lang.String pinOffset,
                            int pinCheckLength,
```

```
                    java.lang.String userId,
                    java.lang.String transferTable,
                    java.lang.String pinCheckData)
                throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
pvkKey	-	是	PVK密文
pinOffset	String	是	PINOFFSET
pinCheckLength	int	是	PIN校验长度
userId	String	是	账号
transferTable	String	是	十进制转换表
pinCheckData	String	是	PIN校验数据

返回值

PIN

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.150 checkIbmTerminalPin

校验一个用IBM方式的终端PIN（DA）。

```
public java.lang.String checkIbmTerminalPin(
                    java.lang.Object tpkKey,
                    java.lang.Object pvkKey,
                    int maxPinLength,
                    java.lang.String pinCipher,
                    int pinBlockAlg,
                    int pinCheckLength,
                    java.lang.String userId,
                    java.lang.String transferTable,
                    java.lang.String pinCheckData,
                    java.lang.String pinOffset)
                throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
tpkKey	-	是	TPK

名称	类型	是否必须	描述
pvkKey	-	是	PVK密文
maxPinLength	int	是	PIN最大长度，取值为*12*
pinCipher	String	是	PIN密文
pinBlockAlg	int	是	PINBLOCK格式代码
pinCheckLength	int	是	PIN校验长度
userId	String	是	账号
transferTable	String	是	十进制转换表
pinCheckData	String	是	PIN校验数据
pinOffset	String	是	PINOFFSET

返回值

result

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.151 checkIbmExchangePin

校验一个用IBM方式的交换PIN（EA）。

```
public java.lang.String checkIbmExchangePin(
                                java.lang.Object zpkKey,
                                java.lang.Object pvkKey,
                                int maxPinLength,
                                java.lang.String pinCipher,
                                int pinBlockAlg,
                                int pinCheckLength,
                                java.lang.String userId,
                                java.lang.String transferTable,
                                java.lang.String pinCheckData,
                                java.lang.String pinOffset)
                            throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
zpkKey	-	是	ZPK

名称	类型	是否必须	描述
pvkKey	-	是	PVK密文
maxPinLength	int	是	PIN最大长度，取值为*12*
pinCipher	String	是	PIN密文
pinBlockAlg	int	是	PINBLOCK格式代码
pinCheckLength	int	是	PIN校验长度
userId	String	是	账号
transferTable	String	是	十进制转换表
pinCheckData	String	是	PIN校验数据
pinOffset	String	是	PINOFFSET

返回值

result

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.152 produceVisaPvv

产生VISA PVV（DG）。

```
public java.lang.String produceVisaPvv(java.lang.Object pvkKey,
                                       java.lang.String pinCipher,
                                       java.lang.String userId,
                                       java.lang.String pvki)
                                throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
pvkKey	-	是	PVK密文
pinCipher	String	是	PIN密文
userId	String	是	账号
pvki	String	是	PVKI PVK标识

返回值

result（PVV）

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.153 pvvCheckZpkPinBlock

PVV校验ZPK加密的PINBLOCK（EC）。

```
public java.lang.String pvvCheckZpkPinBlock(
                        java.lang.Object zpkKey,
                        java.lang.Object pvkKey,
                        java.lang.String pinBlockCipher,
                        int pinBlockAlg,
                        java.lang.String userId,
                        java.lang.String pvki,
                        java.lang.String pvv)
                    throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
zpkKey	-	是	ZPK密文
pvkKey	-	是	PVK密文
pinBlockCipher	String	是	PINBLOCK密文
pinBlockAlg	int	是	PINBLOCK格式代码
userId	String	是	账号
pvki	String	是	PVKI PVK标识
pvv	String	是	VISA PVV

返回值

resultx（错误码）

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.154 produceOrCheckCsc

生成或者校验美国运通的CSC（RY）。

```
public int[] produceOrCheckCsc(int model,
                               int flag,
                               java.lang.Object cvkKey,
                               java.lang.String userId,
                               java.lang.String expires,
                               java.lang.String fiveCsc,
                               java.lang.String fourCsc,
                               java.lang.String threeCsc)
                        throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
model	int	是	模式
flag	int	是	标识
cvkKey	-	是	CVK密文
userId	String	是	账号
expires	String	是	过期时间
fiveCsc	String	否	5位的CSC
fourCsc	String	否	4位的CSC
threeCsc	String	否	3位的CSC

返回值

resultx

- 当模式为3时，5位的CSC，4位的CSC，3位的CSC
- 当模式为4时，5位CSC的校验结果，4位CSC的校验结果，3位CSC的校验结果

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.155 inDataHash

对一个数据块进行哈希运算（GM）。

```
public byte[] inDataHash(int hashFlag,
                         byte[] inData,
                         byte[] userId,
                         byte[] sm2PublicKey)
                  throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
hashFlag	int	是	哈希标识
inData	byte[]	是	数据块
userId	byte[]	是	用户ID
sm2PublicKey	byte[]	是	SM2算法公钥

返回值

hash值

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.156 lodingRsaKeyOld

装载RSA密钥对，兼容旧版本保留（EK）。

```
public java.lang.String lodingRsaKeyOld(int keyIndex,
                                        byte[] privateData)
```

```
                    throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
keyIndex	int	是	密钥索引
privateData	byte[]	是	私钥数据

返回值

result

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.157 lodingRsaKey

装载RSA密钥对（EJ）。

```
public java.lang.String lodingRsaKey(char flag,
                                     int privateFormat,
                                     java.lang.String privatePassword,
                                     byte[] privateData,
                                     int keyIndex,
                                     java.lang.String keyTag)
                              throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
flag	char	是	扩展标识
privateFormat	int	是	私钥格式
privatePassword	String	是	私钥口令
privateData	byte[]	是	私钥数据
keyIndex	int	是	密钥索引
keyTag	String	是	密钥标签

返回值

result

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.158 rsaExportSymKey

RSA公钥保护导出一条对称密钥，RACAL兼容（GK）。

```
public java.lang.String[] rsaExportSymKey(int encFlag,
                                          int paddingModelFlag,
                                          java.lang.Integer mgf,
                                          java.lang.Integer mgfAlg,
                                          byte[] oaep,
                                          java.lang.String symKeyType,
                                          int symKeyFlag,
                                          java.lang.String symKey,
                                          java.lang.String chack,
                                          byte[] mac,
                                          byte[] publicKey,
                                          java.lang.String proveData)
                                   throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
encFlag	int	是	加密标识
paddingModelFlag	int	是	填充模式标识
mgf	Integer	是	MGF
mgfAlg	Integer	是	MGF杂凑算法
oaep	byte[]	是	OAEP编码参数
symKeyType	String	是	对称密钥类型
symKeyFlag	int	是	对称密钥标记
symKey	String	是	对称密钥（LMK）
chack	String	是	校验值
mac	byte[]	是	MAC
publicKey	byte[]	是	公钥
proveData	String	是	证明数据

返回值

- [0]：初始化值
- [1]：密文密钥（PK）

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.159 rsaImportSymKey

RSA公钥保护导入一条对称密钥，RACAL兼容（GI）。

```
public java.lang.String[] rsaImportSymKey(
                                int encFlag,
                                int paddingModelFlag,
                                java.lang.Integer mgf,
                                java.lang.Integer mgfAlg,
                                byte[] oaep,
                                java.lang.String symKeyType,
                                byte[] keyCipher,
                                int rsaIndex,
                                byte[] privateKey,
                                java.lang.Character keyPlanZmk,
                                java.lang.Character keyPlanLmk,
                                java.lang.Character keyCheckType)
                            throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
encFlag	int	是	加密标识
paddingModelFlag	int	是	填充模式标识
mgf	Integer	是	MGF
mgfAlg	Integer	是	MGF杂凑算法
oaep	byte[]	是	OAEP编码参数
symKeyType	String	是	对称密钥类型
keyCipher	byte[]	是	公钥下加密的密钥
rsaIndex	int	是	RSA密钥索引号
privateKey	byte[]	是	私钥

名称	类型	是否必须	描述
keyPlanZmk	Character	是	密钥方案（ZMK）
keyPlanLmk	Character	是	密钥方案（LMK）
keyCheckType	Character	是	密钥校验值类型

返回值

- [0]：初始化值
- [1]：密钥（LMK）
- [2]：密钥校验值

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.160 encAESGCM256

AES-GCM-256加密。

```
public java.util.List<java.lang.Object> encAESGCM256(
                                int algType,
                                java.lang.String keyType,
                                java.lang.Object key,
                                java.lang.String disperFactor,
                                int sessionKeyType,
                                java.lang.String sessionKeyFactor,
                                byte[] data,
                                java.lang.String iv,
                                java.lang.String aad,
                                int tagInputLength)
                               throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
algType	int	是	算法模式，标识密钥加密数据时的算法模式。取值如下： *00*：AES-GCM
keyType	String	是	密钥类型： • 000：KEK • 00A：DEK

名称	类型	是否必须	描述
key	-	是	密钥，此处只支持AES类型密钥，其他类型报错。
disperFactor	String	是	密钥分散因子
sessionKeyType	int	是	会话密钥模式： • *00*：不产生会话密钥 • *02*：ECB模式加密16字节会话密钥因子，得16字节会话密钥 • *05*：CBC模式加密16字节会话密钥因子，得16字节会话密钥
sessionKeyFactor	String	否	会话密钥因子，仅当**sessionKeyType**为*02*/*05*时存在。 • **sessionKeyType**为*02*时，该域为16字节（32H），适用于产生PBOC规范的双长度会话密钥。 取值：6字节0x00||2字节ATC||6字节0x00||2字节ATC的非 • **sessionKeyType**为*05*时，该域为16字节（32H），适用于产生GP规范SCP02的卡片会话密钥。 取值：2字节密钥类型||2字节卡计数器||12字节0x00
data	byte[]	是	输入待加密的数据。
iv	String	是	长度：0001~0128
aad	String	是	长度：0000~0256
tagInputLength	int	是	TAG输出长度：00~16，最大16字节。

返回值

List0.密文；1.TAG

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.161 decAESGCM256

AES-GCM-256解密。

```
public byte[] decAESGCM256(int algType,
                           java.lang.String keyType,
                           java.lang.Object key,
                           java.lang.String disperFactor,
                           int sessionKeyType,
                           java.lang.String sessionKeyFactor,
                           byte[] data,
                           java.lang.String iv,
                           java.lang.String aad,
                           java.lang.String tag)
                    throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
algType	int	是	算法模式，标识密钥加密数据时的算法模式。取值如下： *00*：AES-GCM
keyType	String	是	密钥类型： • 000：KEK • 00A：DEK
key	-	是	密钥，此处只支持AES类型密钥，其他类型报错。
disperFactor	String	是	密钥分散因子
sessionKeyType	int	是	会话密钥模式： • *00*：不产生会话密钥 • *02*：ECB模式加密16字节会话密钥因子，得16字节会话密钥 • *05*：CBC模式加密16字节会话密钥因子，得16字节会话密钥
sessionKeyFactor	String	否	会话密钥因子，仅当**sessionKeyType**为*02*/*05*时存在。

名称	类型	是否必须	描述
			• **sessionKeyType**为*02*时，该域为16字节（32H），适用于产生PBOC规范的双长度会话密钥。 取值：6字节0x00||2字节ATC||6字节0x00||2字节ATC的非 • **sessionKeyType**为*05*时，该域为16字节（32H），适用于产生GP规范SCP02的卡片会话密钥。 取值：2字节密钥类型||2字节卡计数器||12字节0x00
data	byte[]	是	输入待加密的数据。
iv	String	是	长度：0001~0128
aad	String	是	长度：0000~0256
tag	int	是	TAG

返回值

List0.密文；1.TAG

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.162 calcHMAC

计算数据HMAC-明文密钥。

```
public byte[] calcHMAC(int hashAlg,
                       byte[] key,
                       byte[] data)
                throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
hashAlg			HASH算法标识： • 01：SHA-1

名称	类型	是否必须	描述
			• 02：MD5
key			密钥，用于计算HMAC的明文密钥。 取值范围：1~256字节数
data			用于计算HMAC的输入数据。 取值范围：1~1984字节

返回值

输出的HMAC结果。

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.163 symEncExportSM2Key

保护密钥（对称）加密导出一对SM2密钥（TT）。

```
public java.util.ArrayList<byte[]> symEncExportSM2Key(
                                int encAlgModel,
                                java.lang.String proKeyType,
                                java.lang.Object proKey,
                                java.lang.String proKeyDisperFactor,
                                int SM2KeyIndex,
                                byte[] publicKey,
                                byte[] privateKey)
                         throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
encAlgModel	int	是	加密算法模式： • *00*：ECB • *01*：CBC
proKeyType	String	是	保护密钥类型： • 000：KEK • 109：MDK
proKey	int	是	用于加密导出SM2密钥的保护密钥索引。
	String		用于加密导出SM2密钥的保护密钥密文。

名称	类型	是否必须	描述
proKeyDisperFactor	String	是	保护密钥分散子。 取值：00~08
SM2KeyIndex	int	是	SM2密钥索引号SM2公钥在密码机内存储的索引号。 取值：0001~0064
publicKey	byte[]	否	公钥可选域，SM2密钥索引号为9999时存在
privateKey	byte[]	否	私钥可选域，SM2密钥索引号为9999时存在

返回值

公钥、私钥分量d密文

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.164 symImportSm2KeyPair

保护密钥（对称）加密对导入一对SM2密钥。

```
public byte[] symImportSm2KeyPair(int encAlgModel,
                                  java.lang.String proKeyType,
                                  java.lang.Object proKey,
                                  java.lang.String proKeyDisperFactor,
                                  int storeIndex,
                                  java.lang.String storeLabel,
                                  byte[] publicKey,
                                  byte[] privateKey)
                           throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
encAlgModel	int	是	加密算法模式： • *00*：ECB • *01*：CBC
proKeyType	String	是	保护密钥类型： • 000：KEK

名称	类型	是否必须	描述
			• 109：MDK
proKey	int	是	用于加密导出SM2密钥的保护密钥索引。
	String		用于加密导出SM2密钥的保护密钥密文。
proKeyDisperFactor	String	是	保护密钥分散子。 取值：00~08
storeIndex	int		SM2密钥索引号 • 取值为1~64时表示储存到密码机索引当中 • 取值为9999时，表示不储存到密码机当中
storeLabel	String	否	密钥标签，当且仅当**storeIndex**取值在1-64之间时有效。
publicKey	byte[]	是	公钥
privateKey	byte[]	是	私钥分量

返回值

私钥数据

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.165 getHsmFunctionState

获取密码机运行状态。

```
public java.lang.String getHsmFunctionState()
                        throws cn.tass.exceptions.TAException
```

请求参数

无

返回值

以host开头行下包含7行，分别代表：

- 显示设备主密钥是否OK。
  - 0：异常
  - 1：正常
- 显示设备服务状态是否OK。
  - 0：异常
  - 1：正常
- 支持的最大连接数。
- 已被占用的连接数。
- CPU利用率。
- 内存使用率。
- 开机后业务总数。

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.166 generateZmkLetter

生成ZMK密钥信封。

```
public java.lang.String[] generateZmkLetter(char keyAlg,
                                            java.lang.String charset,
                                            java.lang.String str1,
                                            java.lang.String str2,
                                            java.lang.String str3,
                                            java.lang.String str4,
                                            java.lang.String str5,
                                            java.lang.String str6)
                                     throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
keyAlg	char	是	密钥算法标识： • *Z*：单倍长DES密钥 • *X*：双倍长DES密钥 • *Y*：三倍长DES密钥 • *R*：16字节SM4密钥 • *P*：16字节SM1密钥

名称	类型	是否必须	描述
			• *L*：16字节AES密钥 • *M*：24字节AES密钥 • *N*：32字节AES密钥
charset	String	是	指定字符集编码
str1	String	是	机构号
str2	String	是	终端号
str3	String	是	密钥管理员
str4	String	是	成分员
str5	String	是	打印时间
str6	String	是	成分员

返回值

- 0：主密钥在LMK下加密的密文
- 1：主密钥校验值

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.167 generateZmkLetter

生成ZMK密钥信封（由外部指定编码格式）。

```
public java.lang.String[] generateZmkLetter(java.lang.String charset,
                                            java.lang.String SerilNo,
                                            char keyAlg,
                                            int hashAlgFlag)
                                     throws cn.tass.exceptions.TAException
```

此方法为定制功能，生成一个ZMK并按内置的格式打印两个密钥信封，信封上分别包括ZMK的左右两部分值，一级根据指定SerilNo生成的ZMK目标装入设备的检验值。

- 第一成份为ZMK左半部分,第二成份为ZMK右半部分。
- 设备序列号由用户自定义，长度限制为128位ASCII字符。
- 密钥检验值为ZMK密钥加密一组值为全0数据生成，取左4字节。
- 设备校验值生成过程：

1. **SerilNo**通过Hash运算，得到**SerilNo**的Hash值。
2. 将Hash值左右两部分进行异或。
3. 使用ZMK加密异或结果，取左4字节作为设备检验值，ZMK加密异或结果时采用PBOC2.0数据加解密的填充方式。

请求参数

名称	类型	是否必须	描述
charset	String	是	指定字符集编码格式。
SerilNo	String	是	设备序列号，暂限制长度为1~128。
keyAlg	char	是	密钥算法标识： • *Z*：单倍长DES密钥 • *X*：双倍长DES密钥 • *Y*：三倍长DES密钥 • *R*：16字节SM4密钥 • *P*：16字节SM1密钥 • *L*：16字节AES密钥 • *M*：24字节AES密钥 • *N*：32字节AES密钥
hashAlgFlag	int	是	计算设备检验值时使用的HASH算法标识。 • *1*：SHA_1 • *2*：MD5 • *3*：ISO10118_2 • *5*：SHA_224 • *6*：SHA_256 • *7*：SHA_384 • *8*：SHA_512 • *20*：SM3_256

返回值

- 0号索引下：主密钥在LMK下加密的密文。
- 1号索引下：主密钥校验值。
- 2号索引下：设备检验值。

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.168 eccSignature

ECC私钥签名运算。

```
public byte[] eccSignature(int hashAlg,
                           int curveFlag,
                           byte[] inData,
                           java.lang.Object eccPrivateKey,
                           int codeFormat)
                    throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
hashAlg	int	是	Hash算法标识： • *1*：SHA-1 • *5*：SHA-224 • *6*：SHA-256 • *7*：SHA-384 • *8*：SHA-512
curveFlag	int	是	曲线标识： • *20*：prime192v1 • *21*：prime192v2 • *22*：prime192v3 • *23*：secp192k1 • *33*：secp256k1 • *30*：NISTP256 • *31*：Brainpoolp256r1 • *32*：frp256v1
data	byte[]	是	待签名的数据 长度：0000~1984
eccPrivateKey	-	是	SM2私钥索引或lmk密文
codeFormat	int	是	签名编码格式： • *0*：签名值数据串（r、s序列）

名称	类型	是否必须	描述
			• *1*：DER编码格式（r、s序列编码）

返回值

签名结果

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.169 eccVerify

Ecc公钥验签运算。

```
public boolean eccVerify(int hashAlg,
                         int curveFlag,
                         byte[] srcData,
                         java.lang.Object eccPublicKey,
                         int signCodeFormat,
                         byte[] signature)
                  throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
hashAlg	int	是	Hash算法标识： • *1*：SHA-1 • *5*：SHA-224 • *6*：SHA-256 • *7*：SHA-384 • *8*：SHA-512
curveFlag	int	是	曲线标识： • *20*：prime192v1 • *21*：prime192v2 • *22*：prime192v3 • *23*：secp192k1 • *33*：secp256k1 • *30*：NISTP256 • *31*：Brainpoolp256r1

名称	类型	是否必须	描述
			• *32*：frp256v1
srcData	byte[]	是	待验证的签名的数据 长度：0000~1984
eccPublicKey	-	是	密钥索引或公钥明文
signCodeFormat	int	是	签名编码格式： • *0*：签名值数据串（r、s序列） • *1*：DER编码格式（r、s序列编码）
signature	byte[]	是	待验证的签名

返回值

错误码

- 成功返回true。
- 失败返回false。

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.170 symmTransformCipher

数据转加密（通用）。

```
public byte[] symmTransformCipher(
                        int srcEncFlag,
                        int srcKeyType,
                        java.lang.Object srcKey,
                        java.lang.String srcDeriveFactor,
                        int srcSessionKeyFlag,
                        java.lang.String srcSessionKeyFactor,
                        int srcPaddingFlag,
                        java.lang.String srcIV,
                        int dstEncFlag,
                        int dstKeyType,
                        java.lang.Object dstKey,
                        java.lang.String dstDeriveFactor,
                        int dstSessionKeyFlag,
                        java.lang.String dstSessionKeyFactor,
                        int dstPaddingFlag,
                        java.lang.String dstIV,
                        byte[] data)
```

```
                    throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
srcEncFlag	int	是	源密钥加密算法模式： • *00*：ECB • *01*：CBC • *02*：CFB • *03*：OFB
srcKeyType	int	是	源密钥类型： • 000：KEK • 109：MDK • 309：MK-SMC • 00A：ZEK/DEK • 00B：TEK • 011：KMC
srcKey	int	是	用于加密数据的源密钥索引。
	String		用于加密数据的源密钥密文。
srcDeriveFactor	String	是	源密钥分散因子。
srcSession KeyFlag	int	是	源密钥会话密钥模式： • *00*：不产生会话密钥。 • *01*：ECB模式加密8字节会话密钥因子，得8字节会话密钥。 • *02*：ECB模式加密16字节会话密钥因子，得16字节会话密钥。 • *03*：密钥的左右8字节异或，得8字节会话密钥。 • *04*：取密钥的左8字节做为会话密钥。 • *05*：CBC模式加密16字节会话密钥因子，得16字节会话密钥。
srcSession KeyFactor	String	否	源密钥会话密钥因子，仅当**srcSessionKeyFlag**为*01*/*02*/*05*时存在。

名称	类型	是否必须	描述
			• **srcSessionKeyFlag**为*01*时，该域为8字节（16H），适用于产生PBOC规范的单长度会话密钥。 取值：6字节0x00||2字节ATC • **srcSessionKeyFlag**为*02*时，该域为16字节（32H），适用于产生PBOC规范的双长度会话密钥。 取值：6字节0x00||2字节ATC||6字节0x00||2字节ATC的非 • **srcSessionKeyFlag**为*05*时，该域为16字节（32H），适用于产生GP规范SCP02的卡片会话密钥。 取值：2字节密钥类型||2字节卡计数器||12字节0x00
srcPaddingFlag	int	是	源密钥加密时的数据PAD标识。
srcIV	String	是	源密钥加密时的IV
dstEncFlag	int	是	目标密钥加密算法模式： • *00*：ECB • *01*：CBC • *02*：CFB • *03*：OFB
dstKeyType	int	是	用于加密数据的目标密钥类型： • 000：KEK • 109：MDK • 309：MK-SMC • 00A：ZEK/DEK • 00B：TEK • 011：KMC
dstKey	int	是	用于加密数据的目标密钥索引。
	String		用于加密数据的目标密钥密文。
dstDeriveFactor	String	是	目标密钥分散因子。

名称	类型	是否必须	描述
dstSessionKeyFlag	int	是	目标密钥会话密钥模式： • *00*：不产生会话密钥。 • *01*：ECB模式加密8字节会话密钥因子，得8字节会话密钥。 • *02*：ECB模式加密16字节会话密钥因子，得16字节会话密钥。 • *03*：密钥的左右8字节异或，得8字节会话密钥。 • *04*：取密钥的左8字节做为会话密钥。 • *05*：CBC模式加密16字节会话密钥因子，得16字节会话密钥。
dstSessionKeyFactor	String	否	目标密钥会话密钥因子，仅当**dstSessionKeyFlag**为*01*/*02*/*05*时存在。 • **dstSessionKeyFlag**为*01*时，该域为8字节（16H），适用于产生PBOC规范的单长度会话密钥。 取值：6字节0x00||2字节ATC • **dstSessionKeyFlag**为*02*时，该域为16字节（32H），适用于产生PBOC规范的双长度会话密钥。 取值：6字节0x00||2字节ATC||6字节0x00||2字节ATC的非 • **dstSessionKeyFlag**为*05*时，该域为16字节（32H），适用于产生GP规范SCP02的卡片会话密钥。 取值：2字节密钥类型||2字节卡计数器||12字节0x00
dstPaddingFlag	int	是	目标密钥加密时的数据PAD标识。
dstIV	String	是	目标密钥加密时的IV。
data	byte[]	是	源密钥加密时的数据密文。

返回值

目标密钥加密时的数据密文。

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.171 EccEciesEncrypt

Ecc NISTP256曲线加密运算。

```
public java.util.ArrayList<byte[]> EccEciesEncrypt(
                                java.lang.Object EccPublicKey,
                                int KDFHash,
                                int symmFlag,
                                byte[] Adata,
                                java.lang.String IV,
                                byte[] SharedinfoS1,
                                int HmacFlag,
                                int Hmac,
                                byte[] SharedinfoS2,
                                int Padding,
                                byte[] indata)
                            throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
EccPublicKey	-	是	Peer ECC公钥索引号或密钥密文。 • 使用索引时取值范围：1~64。 • 使用密钥密文时为DER编码公钥密文。
KDFHash	int	是	KDF-HASH算法标识： • *1*：SHA-1 • *5*：SHA-224 • *6*：SHA-256 • *7*：SHA-384 • *8*：SHA-512
symmFlag	int	是	对称密钥加密算法： • *1*：AES-128-ECB • *2*：AES-128-CBC • *3*：AES-256-ECB • *4*：AES-256-CBC • *5*：AES-128-GCM • *6*：AES-256-GCM

名称	类型	是否必须	描述
			• *7*：3DES-128-ECB • *8*：3DES-128-CBC • *9*：3DES-192-ECB • *10*：3DES-192-CBC
Adata	byte[]	否	Adata数据，取值长度：0~128字节。 **说明**： 当且仅当**symmFlag**取值为*5*/*6*时存在。
IV	String	是	初始向量： • 当**symmFlag**取值为*5*/*6*时，取值长度：0~128字节。 • 当**symmFlag**取值为*2*/*4*时，长度为32H。 • 当**symmFlag**取值为*8*/*10*时，长度为32H。
SharedinfoS1	byte[]	是	共享加密信息S1，用于KDF分散密钥，取值范围0~128字节。
HmacFlag	int	是	HAMC计算标识： • *0*：不计算HAMC • *1*：计算HAMC
Hmac	int	否	HAMC算法标识： • *01*：SHA-1 • *06*：SHA-256 • *08*：SHA-512 **说明**： 仅当**HmacFlag**取值为*1*时有效。
SharedinfoS2	byte[]	否	S2长度，取值范围0~128字节。 **说明**： 仅当**HmacFlag**取值为*1*时有效。
Padding	int	是	数据填充规则： • *0*：PBOC2.0填充模式

名称	类型	是否必须	描述
			• *1*：ISO/IEC9797-1的PADDING模式2 • *2*：ISO/IEC9797-1的PADDING模式1 • *3*：ANSIX9.23 • *4*：PKCS#5 • *5*：NoPadding模式 • *10*：PBOC3.0 • *11*：左填充+ISO/IEC9797-1
indata	byte[]	是	待加密数据。

返回值

返回密钥协商数据，其中下标0，1为必返回值，2和3根据协商时的数据参数进行返回。

- 0：随机公钥。
- 1：密钥密文。
- 2：当且仅当HmacFlag取值为1时存在HMAC验证信息。
- 3：当且仅当symmFlag取值为5、6时存在该值。

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.172 EccEciesDecrypt

Ecc NISTP256曲线解密运算。

```
public byte[] EccEciesDecrypt(java.lang.Object EccPublicKey,
                              java.lang.Object oneselfEccPrivateKey,
                              int KDFHash,
                              int symmFlag,
                              byte[] Adata,
                              java.lang.String IV,
                              byte[] TAG,
                              byte[] SharedinfoS1,
                              int HmacFlag,
                              int Hmac,
                              byte[] SharedinfoS2,
                              byte[] D,
                              int Padding,
                              byte[] cipher)
```

```
                    throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
EccPublicKey	-	是	PEcc协商随机公钥，使用密钥密文时为DER编码公钥密文。
oneselfEccPrivateKey	-	是	己方私钥索引或密文。 • 索引取值范围：1~64。 • 密钥密文格式为lmk加密的私钥密文。
KDFHash	int	是	KDF-HASH算法标识： • *1*：SHA-1 • *5*：SHA-224 • *6*：SHA-256 • *7*：SHA-384 • *8*：SHA-512
symmFlag	int	是	对称密钥加密算法： • *1*：AES-128-ECB • *2*：AES-128-CBC • *3*：AES-256-ECB • *4*：AES-256-CBC • *5*：AES-128-GCM • *6*：AES-256-GCM • *7*：3DES-128-ECB • *8*：3DES-128-CBC • *9*：3DES-192-ECB • *10*：3DES-192-CBC
Adata	byte[]	否	Adata数据，取值长度：0~128字节。 **说明**： 当且仅当**symmFlag**取值为*5*/*6*时存在。
IV	String	是	初始向量： • 当**symmFlag**取值为*5*/*6*时，取值长度：0~128字节。

名称	类型	是否必须	描述
			• 当**symmFlag**取值为*2*/*4*时，长度为32H。 • 当**symmFlag**取值为*8*/*10*时，长度为32H。
TAG	byte[]	否	GCM算法验证数据，协商时的返回结果  **说明**： 当且仅当**symmFlag**取值为*5*/*6*时存在。
SharedinfoS1	byte[]	是	共享加密信息S1，用于KDF分散密钥，取值范围0~128字节。
HmacFlag	int	是	HAMC计算标识： • *0*：不计算HAMC • *1*：计算HAMC
Hmac	int	否	HAMC算法标识： • *01*：SHA-1 • *06*：SHA-256 • *08*：SHA-512  **说明**： 仅当**HmacFlag**取值为*1*时有效。
SharedinfoS2	byte[]	否	S2长度，取值范围0~128字节。  **说明**： 仅当**HmacFlag**取值为*1*时有效。
D	byte[]	否	HMAC验证数据。  **说明**： 仅当**HmacFlag**取值为*1*时有效。
Padding	int	是	数据填充规则： • *0*：PBOC2.0填充模式 • *1*：ISO/IEC9797-1的PADDING模式2 • *2*：ISO/IEC9797-1的PADDING模式1 • *3*：ANSIX9.23

名称	类型	是否必须	描述
			• *4*：PKCS#5 • *5*：NoPadding模式 • *10*：PBOC3.0 • *11*：左填充+ISO/IEC9797-1
cipher	byte[]	是	待解密的密文。

返回值

返回解密数据。

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.173 printPINDate

打印PIN/PIN请求数据。

```
public java.lang.String[] printPINDate(java.lang.String DocumentsType,
                                       java.lang.String id,
                                       java.lang.String pin,
                                       java.lang.String... args)
                                throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
DocumentsType	String	是	文档类型： • A： “two-up” 格式中的第一个信封。 • B： “two-up” 格式中的第二个信封。 • C：1步格式。
id	String	是	账号，账号中去除校验位的最右12位。
pin	String	是	PIN，LMK加密pin。
args	String...	是	打印。 **说明**： 不能包含分号（；）字符

返回值

-

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.174 generateLengthPIN

产生指定长度的随机字符PIN。

```
public java.lang.String generateLengthPIN(int pinLength,
                                          java.lang.Object zpkKey,
                                          int pinblockAlg,
                                          java.lang.String id)
                                   throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
pinLength	int	是	待产生的随机PIN的长度。 取值范围：04~16
zpkKey	-	是	ZPK密钥，用于加密PIN的ZPK的密钥索引或密文。
pinblockAlg	int	是	字符PINBLOCK格式： • *00*：PIN与账号分别左对齐，以填充0x00方式扩展为48H（采用64位分组算法）或64H（采用128位分组算法），再异或后得到PIN数据块。 • *01*：ASCII码序列[账号|PIN|填充字符]得到PIN数据块。 填充规则：根据密钥算法的分组长度，按需要填充的字节数填入相应字符，例如缺少6个字节，则填入6个字符'6'；若满足分组长度的倍数则不填充。 • *02*：PIN与账号分别左对齐，以填充0x00方式扩展为32H，再异或后得到PIN数据块。

名称	类型	是否必须	描述
pin	String	是	标识账号长度。 账号位数：01~24

返回值

PIN密文，ZPK下加密的随机字符PIN的密文。

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.175 transferLmkToZpk

将字符PIN由LMK加密转换为ZPK加密。

```
public java.lang.String transferLmkToZpk(java.lang.Object zpkKey,
                                         int pinBlockAlg,
                                         java.lang.String rscId,
                                         java.lang.String pinCipher)
                                  throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
zpkKey	-	是	ZPK密钥，用于加密PIN的ZPK的密钥索引或密文。
pinblockAlg	int	是	字符PINBLOCK格式： • *00*：PIN与账号分别左对齐，以填充0x00方式扩展为48H（采用64位分组算法）或64H（采用128位分组算法），再异或后得到PIN数据块。 • *02*：PIN与账号分别左对齐，以填充0x00方式扩展为32H，再异或后得到PIN数据块。 • *03*：柜员登录密码的加密规范，限定PIN的最大长度为8。柜员号前补0至16或32位作为二进制串；PIN密码BCD扩展后补0至16或32位，然后异或用密钥加密。

名称	类型	是否必须	描述
rscId	String	是	用户有效主帐号或客户号。 当**pinblockAlg**取值为*02*/*03*时，该域不能超过16个数字（长度01~24）。
pinCipher	String	是	PIN密文，LMK下加密的PIN密文。

返回值

目标PINBLOCK密文。

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.176 transferZpkToLmk

将字符PIN由ZPK加密转换为LMK加密。

```
public java.lang.String transferZpkToLmk(java.lang.Object zpkKey,
                                         java.lang.String pinCipher,
                                         int pinBlockAlg,
                                         java.lang.String rscId)
                                  throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
zpkKey	-	是	ZPK密钥，用于加密PIN的ZPK的密钥索引或密文。
pinCipher	String	是	在源ZPK下加密的PINBLOCK密文。
pinblockAlg	int	是	字符PINBLOCK格式： • *00*：PIN与账号分别左对齐，以填充0x00方式扩展为48H（采用64位分组算法）或64H（采用128位分组算法），再异或后得到PIN数据块。 • *02*：PIN与账号分别左对齐，以填充0x00方式扩展为32H，再异或后得到PIN数据块。 • *03*：柜员登录密码的加密规范，限定PIN的最大长度为8。柜员号前

名称	类型	是否必须	描述
			补0至16或32位作为二进制串；PIN密码BCD扩展后补0至16或32位，然后异或用密钥加密。
rscId	String	是	用户有效主帐号或客户号。 当**pinblockAlg**取值为*02*/*03*时，该域不能超过16个数字（长度01~24）。

返回值

PIN密文，LMK下加密的PIN密文。

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.177 printPINBolckDate

打印LMK加密字符PIN。

```
public java.lang.String[] printPINBolckDate(
                                java.lang.String DocumentsType,
                                java.lang.String id,
                                java.lang.String pin,
                                java.lang.String... args)
                         throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
DocumentsType	String	是	文档类型： • *A*：two-up格式中的第一个信封。 • *B*：two-up格式中的第二个信封。 • *C*：1步格式。
id	String	是	账号，账号中去除校验位的最右12位。
pin	String	是	PIN，LMK加密pin。
args	String...	是	打印域。 **说明**：

名称	类型	是否必须	描述
			不能包含分号（;）字符。

返回值

打印LMK加密字符PIN。

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```

3.178 getAsymmKeyPublicKeyAndPrivateKey

返回RSA或ECC算法指定索引下的私钥和DER编码公钥。

```
public java.util.ArrayList<byte[]> getAsymmKeyPublicKeyAndPrivateKey(
                                    java.lang.String keyFlag,
                                    int keyIndex)
                             throws cn.tass.exceptions.TAException
```

请求参数

名称	类型	是否必须	描述
keyFlag	String	是	非对称算法标识： • RSA • ECC
keyIndex	int	是	非对称密钥索引，取值范围为1~64。 **说明**： RSA算法和ECC算法各有64个索引。

返回值

- RSA算法返回：
  - 0：lmk加密的私钥密文。
  - 1：DER编码公钥。
- ECC算法返回：
  - 0：OID标识

- 1：lmk加密的私钥密文。
- 2：DER编码的公钥。

异常处理

程序运行中出错则抛出异常。

```
cn.tass.exceptions.TAException
```