**签名验签服务器**

**Java语言接口使用规范**

北京神州安付科技股份有限公司

2023-03-16

目录

[1. 算法标识和数据结构 5](#_Toc14369)

[1.1. 设备信息 5](#_Toc10381)

[1.2. 证书信任列表 5](#_Toc9825)

[2. 签名服务器接口 5](#_Toc6791)

[2.1. 获取设备 5](#_Toc9311)

[2.2. 关闭设备 5](#_Toc1642)

[3. 设备接口 5](#_Toc2627)

[3.1. 获取设备信息 5](#_Toc18108)

[3.2. 获取随机数 5](#_Toc10097)

[3.3. 导出公钥 5](#_Toc14437)

[3.3.1. 导出SM2签名公钥 5](#_Toc19453)

[3.3.2. 导出SM2加密公钥 5](#_Toc24266)

[3.3.3. 导出RSA签名公钥 5](#_Toc6868)

[3.3.4. 导出RSA加密公钥 5](#_Toc12758)

[3.4. 生成密钥对 5](#_Toc21019)

[3.4.1. 生成SM2密钥对 5](#_Toc11615)

[3.4.2. 生成RSA密钥对 5](#_Toc25486)

[3.5. RSA签名验签 5](#_Toc13436)

[3.5.1. RSA 签名 内部私钥 5](#_Toc17649)

[3.5.2. RSA 签名 外部私钥 6](#_Toc12491)

[3.5.3. RSA 文件签名 内部私钥 6](#_Toc24769)

[3.5.4. RSA 文件签名 外部私钥 6](#_Toc21694)

[3.5.5. RSA 验签 内部公钥 6](#_Toc4563)

[3.5.6. RSA 验签 外部公钥 6](#_Toc24930)

[3.5.7. RSA 文件验签 内部公钥 6](#_Toc10368)

[3.5.8. RSA 文件验签 外部公钥 6](#_Toc4919)

[3.5.9. RSA 证书验签 6](#_Toc11913)

[3.5.10. RSA 证书文件验签 6](#_Toc14113)

[3.6. RSA加解密 6](#_Toc1761)

[3.6.1. RSA加密 内部公钥 6](#_Toc12146)

[3.6.2. RSA加密 外部公钥 6](#_Toc12541)

[3.6.3. RSA 加密 证书公钥 6](#_Toc12949)

[3.6.4. RSA解密 内部私钥 6](#_Toc29524)

[3.6.5. RSA解密 外部私钥 6](#_Toc6130)

[3.7. SM2 签名验签 6](#_Toc3895)

[3.7.1. SM2 签名 内部私钥 6](#_Toc27897)

[3.7.2. SM2 签名 外部私钥 不带z值 6](#_Toc28723)

[3.7.3. SM2 签名 外部私钥 带z值 6](#_Toc7030)

[3.7.4. SM2 签名 外部私钥+证书 带z值 6](#_Toc7051)

[3.7.5. SM2 文件签名 内部密钥 6](#_Toc29599)

[3.7.6. SM2 文件签名 外部私钥 不带Z值 6](#_Toc2202)

[3.7.7. SM2 文件签名 外部私钥 带Z值 7](#_Toc16251)

[3.7.8. SM2 文件签名 外部私钥 +证书 带Z值 7](#_Toc4369)

[3.7.9. SM2 验签 内部公钥 7](#_Toc29329)

[3.7.10. SM2 验签 外部公钥 不带Z值 7](#_Toc24696)

[3.7.11. SM2 验签 一张证书 带Z值 7](#_Toc24577)

[3.7.12. SM2 验签 两张证书 带Z值 7](#_Toc17494)

[3.7.13. SM2 验证文件签名 内部公钥 7](#_Toc29303)

[3.7.14. SM2 验证文件签名 外部公钥 不带Z值 7](#_Toc2790)

[3.7.15. SM2 验证文件签名 一张证书 带Z值 7](#_Toc31890)

[3.7.16. SM2 验证文件签名 两张证书 带Z值 7](#_Toc19537)

[3.8. SM2 加解密 7](#_Toc11707)

[3.8.1. SM2 加密 内部公钥 7](#_Toc6840)

[3.8.2. SM2 加密 外部公钥 7](#_Toc22145)

[3.8.3. SM2 加密 使用证书 7](#_Toc14233)

[3.8.4. SM2 解密 内部私钥 7](#_Toc16620)

[3.8.5. SM2 解密 外部私钥 7](#_Toc8530)

[3.9. 对称加密 7](#_Toc17520)

[3.9.1. SM4 ECB 内部密钥加密 7](#_Toc8661)

[3.9.2. SM4 ECB 外部密钥加密 7](#_Toc14944)

[3.9.3. SM4 ECB 密钥句柄加密 7](#_Toc24340)

[3.9.4. SM4 CBC 内部密钥加密 7](#_Toc15731)

[3.9.5. SM4 CBC 外部密钥加密 7](#_Toc19742)

[3.9.6. SM4 CBC 密钥句柄加密 8](#_Toc11771)

[3.9.7. SM1 ECB 内部密钥加密 8](#_Toc27453)

[3.9.8. SM1 ECB 外部密钥加密 8](#_Toc9629)

[3.9.9. SM1 ECB 密钥句柄加密 8](#_Toc11002)

[3.9.10. SM1 CBC 内部密钥加密 8](#_Toc7319)

[3.9.11. SM1 CBC 外部密钥加密 8](#_Toc9488)

[3.9.12. SM1 CBC 密钥句柄加密 8](#_Toc3176)

[3.10. 对称解密 8](#_Toc18946)

[3.10.1. SM4 ECB 内部密钥解密 8](#_Toc5248)

[3.10.2. SM4 ECB 外部密钥解密 8](#_Toc20631)

[3.10.3. SM4 ECB 密钥句柄解密 8](#_Toc6479)

[3.10.4. SM4 CBC 内部密钥解密 8](#_Toc6425)

[3.10.5. SM4 CBC 外部密钥解密 8](#_Toc22807)

[3.10.6. SM4 CBC 密钥句柄解密 8](#_Toc13181)

[3.10.7. SM1 ECB 内部密钥解密 8](#_Toc3041)

[3.10.8. SM1 ECB 外部密钥解密 8](#_Toc26594)

[3.10.9. SM1 ECB 密钥句柄解密 8](#_Toc4230)

[3.10.10. SM1 CBC 内部密钥解密 8](#_Toc23570)

[3.10.11. SM1 CBC 外部密钥解密 8](#_Toc30390)

[3.10.12. SM1 CBC 密钥句柄解密 8](#_Toc1504)

[3.11. 对称批量加密 8](#_Toc30291)

[3.11.1. SM4 内部批量加密 ECB 8](#_Toc22933)

[3.11.2. SM4外部批量加密 ECB 9](#_Toc9895)

[3.11.3. SM4 密钥句柄批量加密 ECB 9](#_Toc23771)

[3.11.4. SM4 内部批量加密 CBC 9](#_Toc7222)

[3.11.5. SM4 外部批量加密 CBC 9](#_Toc18867)

[3.11.6. SM4 密钥句柄批量加密 CBC 9](#_Toc2622)

[3.11.7. SM1内部批量加密 ECB 9](#_Toc2693)

[3.11.8. SM1外部批量加密 ECB 9](#_Toc11560)

[3.11.9. SM1 密钥句柄批量加密 ECB 9](#_Toc24395)

[3.11.10. SM1 内部批量加密 CBC 9](#_Toc11628)

[3.11.11. SM1 外部批量加密 CBC 9](#_Toc15709)

[3.11.12. SM1 密钥句柄批量加密 CBC 9](#_Toc26301)

[3.12. 对称批量解密 9](#_Toc18036)

[3.13. MAC计算 10](#_Toc7546)

[3.14. Hash计算 10](#_Toc22740)

[3.15. 获取内部密钥句柄 10](#_Toc15539)

[3.16. 证书管理 10](#_Toc2497)

[3.17. PKCS7编码解码数据 11](#_Toc18754)

[4. 附录 11](#_Toc9390)

1. **算法标识和数据结构**
   1. **设备信息**
   2. **证书信任列表**
2. **签名服务器接口**
   1. **获取设备**

|  |  |  |
| --- | --- | --- |
| 方法 | AFSVDevice device = new AFSVDevice.Builder(host, port,passwd) // 服务器IP、端口、设备密钥为必须参数  //以下为可选参数 根据实际情况设置  .connectTimeOut(10000) // 连接超时时间，默认5秒  .responseTimeOut(10000) // 响应超时时间，默认10秒  .retryCount(3) // 重试次数，默认3次  .retryInterval(1000) // 重试间隔，默认1秒  .bufferSize(1024 \* 1024) // 缓冲区大小，默认1M  .channelCount(10) // 通道数，默认10个  .build(); | |
| 作用 | 获取设备 | |
| 参数 | connectTimeOut | 连接超时时间，默认5秒 |
| responseTimeOut | 响应超时时间，默认10秒 |
| retryCount | 重试次数，默认3次 |
| retryInterval | 重试间隔，默认1秒 |
| bufferSize | 缓冲区大小，默认1M |

* 1. **关闭设备**

|  |  |  |
| --- | --- | --- |
| 方法 | default void close(NettyClient client) | |
| 作用 | 与设备断开连接,释放连接资源 | |
| 参数 | client | 设备的连接客户端 |
| 调用 | device.close(AFSVDevice.getClient()); | device :设备 |

1. **设备接口**
   1. **获取设备信息**

|  |  |  |
| --- | --- | --- |
| 方法 | public DeviceInfo getDeviceInfo() throws AFCryptoException; | |
| 作用 | 获取设备信息，用户获取密码机的设备信息,查看设备信息时，可使用DeviceInfo中的toString()方法。 | |
| 例如：System.out.println(device.getDeviceInfo().toString()); | |
| 私有参数 | IssuerName | 设备生产厂商名称 |
| DeviceName | 设备型号 |
| DeviceSerial | 设备编号  包含：日期（8字符）、批次号（3字符）、流水号（5字符） |
| DeviceVersion |  |
| StandardVersion | 密码设备内部软件的版本号 |
| AsymAlgAbility | 密码设备支持的接口规范版本号 |
| SymAlgAbility | 所有支持的对称算法，表示方法为对称算法标识按位或运算结果 |
| HashAlgAbility | 所有支持的杂凑算法，表示方法为杂凑算法标识按位或运算结果 |
| BufferSize | 支持的最大文件储存空间 |
| 返回值 | 返回签名验签服务器设备信息 | |

* 1. **获取随机数**

|  |  |  |
| --- | --- | --- |
| 方法 | public byte[] getRandom(int length) throws AFCryptoException; | |
| 作用 | 获取随机数，用户获取指定长度的随机数数据 | |
| 调用 | byte[] random = device.getRandom(5); | device说明：  AFSVDevice device =new AFSVDevice.Builder(host, port, passwd)  .responseTimeOut(100000)  .build(); |
| 参数 | length | 待取得随机数的长度 |
| 返回值 | 返回取得的随机数数据,Base64编码格式,需要解码后为length长度 | |

* 1. **导出公钥**
     1. **导出SM2签名公钥**

|  |  |  |
| --- | --- | --- |
| 方法 | public byte[] getSM2SignPublicKey(int index) throws AFCryptoException; | |
| 作用 | 导出SM2公钥，导出密码机内部对应索引和用途的SM2公钥信息 | |
| 参数 | index | 密码设备内部存储的SM2索引号 |
| 返回值 | 返回Base64编码的公钥数据 | |

* + 1. **导出SM2加密公钥**

|  |  |  |
| --- | --- | --- |
| 方法 | public byte[] getSM2EncryptPublicKey(int index) throws AFCryptoException; | |
| 作用 | 导出SM2公钥，导出密码机内部对应索引和用途的SM2公钥信息 | |
| 参数 | index | 密码设备内部存储的SM2索引号 |
| 返回值 | 返回Base64编码的公钥数据 | |

* + 1. **导出RSA签名公钥**

|  |  |  |
| --- | --- | --- |
| 方法 | public byte[] getRSASignPublicKey(int index) throws AFCryptoException; | |
| 作用 | 导出RSA公钥，导出密码机内部对应索引和用途的RSA公钥信息 | |
| 参数 | index | 密码设备内部存储的RSA索引号 |
| 返回值 | 返回Base64编码的公钥数据,符合ASN.1 DER编码格式 | |

* + 1. **导出RSA加密公钥**

|  |  |  |
| --- | --- | --- |
| 方法 | public byte[] getRSAEncPublicKey(int index) throws AFCryptoException; | |
| 作用 | 导出RSA公钥，导出密码机内部对应索引和用途的RSA公钥信息 | |
| 参数 | index | 密码设备内部存储的RSA索引号 |
| 返回值 | 返回Base64编码的公钥数据 | |

* 1. **生成密钥对**
     1. **生成SM2密钥对**

|  |  |  |
| --- | --- | --- |
| 方法 | public SM2KeyPairStructure generateSM2KeyPair(int keyType) throws AFCryptoException; | |
| 作用 | 生成SM2 密钥对 | |
| 参数 | keyType | 密钥类型 0:签名密钥对 1:加密密钥对 2:密钥交换密钥对 3:默认密钥对 |
| 返回值 | SM2KeyPairStructure 公钥私钥ASN.1 DER编码格式 然后使用Base64编码 | public class SM2KeyPairStructure {  private byte[] pubKey;  private byte[] priKey;  } |

* + 1. **生成RSA密钥对**

|  |  |  |
| --- | --- | --- |
| 方法 | public RSAKeyPairStructure generateRSAKeyPair(ModulusLength length) throws AFCryptoException; | |
| 作用 | 生成RSA 密钥对 | |
| 参数 | keyType | 密钥类型 0:签名密钥对 1:加密密钥对 2:密钥交换密钥对 3:默认密钥对 |
| 返回值 | RSAKeyPairStructure 公钥私钥ASN.1 DER编码格式 然后使用Base64编码 | public class RSAKeyPairStructure {  private byte[] pubKey;  private byte[] priKey;  } |

* 1. **RSA签名验签**
     1. **RSA 签名 内部私钥**

|  |  |  |
| --- | --- | --- |
| 方法 | public byte[] rsaSignature(int keyIndex, byte[] inData) throws AFCryptoException ; | |
| 作用 | RSA签名，使用RSA内部密钥进行签名运算 | |
| 参数 | keyIndex | 密码设备内部存储的RSA索引号 |
| inData | 待签名的原始数据 |
| 返回值 | 返回Base64编码的签名数据 | |

* + 1. **RSA 签名 外部私钥**

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | public byte[] rsaSignature(byte[] privateKey, byte[] inData) throws AFCryptoException; | | |
| 作用 | RSA签名，使用RSA外部密钥进行签名运算 | | |
| 参数 | privateKey | base64编码的RSA私钥数据，其结构应满足PKCS#1中的RSA结构定义 | 说明：  RSAPrivateKey ::= SEQUENCE {  version Version,  modulus INTEGER, --- n  publicExponent INTEGER, --- e  privateExponent INTEGER, --- d  prime1 INTEGER, --- p  prime2 INTEGER, --- q  exponent1 INTEGER, --- d mod (p-1)  exponent2 INTEGER, --- d mod (q-1)  coefficient INTEGER, --- (inverse of q) mod p  otherPrimeInfos OtherPrimeInfos OPTIONAL  } |
| inData | 待签名的原始数据 | |
| 返回值 | 返回Base64编码的签名数据 | | |

* + 1. **RSA 文件签名 内部私钥**

|  |  |  |
| --- | --- | --- |
| 方法 | public byte[] rsaSignFile(int keyIndex, byte[] filePath) throws AFCryptoException; | |
| 作用 | 对文件进行RSA签名运算，使用RSA内部密钥对文件内容进行签名运算 | |
| 参数 | keyIndex | 密码设备内部存储的RSA索引号 |
| filePath | 待签名的文件路径 |
| 返回值 | 返回Base64编码的签名数据 | |

* + 1. **RSA 文件签名 外部私钥**

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | public byte[] rsaSignFile(byte[] privateKey, byte[] filePath) throws AFCryptoException; | | |
| 作用 | 对文件进行RSA签名运算，使用外部RSA密钥对文件内容进行签名运算 | | |
| 参数 | privateKey | base64编码的RSA私钥数据，其结构应满足PKCS#1中的RSA结构定义 | 说明：  RSAPrivateKey ::= SEQUENCE {  version Version,  modulus INTEGER, --- n  publicExponent INTEGER, --- e  privateExponent INTEGER, --- d  prime1 INTEGER, --- p  prime2 INTEGER, --- q  exponent1 INTEGER, --- d mod (p-1)  exponent2 INTEGER, --- d mod (q-1)  coefficient INTEGER, --- (inverse of q) mod p  otherPrimeInfos OtherPrimeInfos OPTIONAL  } |
| filePath | 待签名的文件路径 | |
| 返回值 | 返回Base64编码的签名数据 | | |

* + 1. **RSA 验签 内部公钥**

|  |  |  |
| --- | --- | --- |
| 方法 | public boolean rsaVerify(int keyIndex, byte[] inData, byte[] signatureData) throws AFCryptoException; | |
| 作用 | 对文件进行RSA签名运算，使用内部RSA密钥对数据进行验证签名运算 | |
| 参数 | keyIndex | 密码设备内部存储的RSA索引号 |
| inData | 原始数据 |
| signatureData | Base64编码的签名数据 |
| 返回值 | true | 验证成功 |
| false | 验证失败 |

* + 1. **RSA 验签 外部公钥**

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | public boolean rsaVerify(byte[] publicKey, byte[] rawData, byte[] signatureData) throws AFCryptoException; | | |
| 作用 | 对文件进行RSA签名运算，使用外部RSA密钥对数据进行验证签名运算 | | |
| 参数 | publicKey | base64编码的RSA公钥数据，其结构应满足PKCS#1中的RSA结构定义 | 说明：  RSAPublicKey ::= SEQUENCE {  modulus INTEGER, --- n  publicExponent INTEGER, --- e  } |
| inData | 原始数据 | |
| signatureData | Base64编码的签名数据 | |
| 返回值 | true | 验证成功 | |
| false | 验证失败 | |

* + 1. **RSA 文件验签 内部公钥**

|  |  |  |
| --- | --- | --- |
| 方法 | public boolean rsaVerifyFile(int keyIndex, byte[] filePath, byte[] signatureData) throws AFCryptoException ; | |
| 作用 | 对文件进行RSA验证签名，使用内部密钥对文件签名值进行验证 | |
| 参数 | keyIndex | 密码设备内部存储的RSA索引号 |
| filePath | 文件路径 |
| signatureData | Base64编码的签名数据 |
| 返回值 | true | 验证成功 |
| false | 验证失败 |

* + 1. **RSA 文件验签 外部公钥**

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | public boolean rsaVerifyFile(byte[] publicKey, byte[] filePath, byte[] signatureData) throws AFCryptoException; | | |
| 作用 | 对文件进行RSA验证签名，使用外部密钥对文件签名值进行验证 | | |
| 参数 | publicKey | base64编码的RSA公钥数据，其结构应满足PKCS#1中的RSA结构定义 | 说明：  RSAPublicKey ::= SEQUENCE {  modulus INTEGER, --- n  publicExponent INTEGER, --- e  } |
| filePath | 文件路径 | |
| signatureData | Base64编码的签名数据 | |
| 返回值 | true | 验证成功 | |
| false | 验证失败 | |

* + 1. **RSA 证书验签**

|  |  |  |
| --- | --- | --- |
| 方法 | public boolean rsaVerifyByCertificate(byte[] certificate, byte[] inData, byte[] signatureData) throws AFCryptoException; | |
| 作用 | RSA验证签名，使用证书对数据进行验证签名运算 | |
| 参数 | certificatePath | 证书 Base64编码 DER格式 |
| inData | 原始数据 |
| signatureData | Base64编码的签名数据 |
| 返回值 | true | 验证成功 |
| false | 验证失败 |

* + 1. **RSA 证书文件验签**

|  |  |  |
| --- | --- | --- |
| 方法 | public boolean rsaVerifyFileByCertificate(byte[] certificate, byte[] filePath, byte[] signatureData) throws AFCryptoException ; | |
| 作用 | RSA验证签名，使用证书对文件签名数据进行验证签名运算 | |
| 参数 | certificatePath | base64编码的RSA数字证书 |
| fileName | 文件名称 |
| signatureData | Base64编码的签名数据 |
| 返回值 | true | 验证成功 |
| false | 验证失败 |

* 1. **RSA加解密**
     1. **RSA加密 内部公钥**

|  |  |  |
| --- | --- | --- |
| 方法 | public byte[] rsaEncrypt(int keyIndex, byte[] data) throws AFCryptoException; | |
| 作用 | RSA加密，使用内部密钥进行RSA加密 | |
| 参数 | keyIndex | 密码设备内部存储的RSA索引号 |
| data | 待加密的原始数据 |
| 返回值 | Base64编码的加密数据 | |

* + 1. **RSA加密 外部公钥**

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | public byte[] rsaEncrypt(byte[] publicKey, byte[] data) throws AFCryptoException; | | |
| 作用 | RSA加密，使用外部密钥进行RSA加密 | | |
| 参数 | publicKey | base64编码的RSA公钥数据，其结构应满足PKCS#1中的RSA结构定义 | 说明：  RSAPublicKey ::= SEQUENCE {  modulus INTEGER, --- n  publicExponent INTEGER, --- e  } |
| data | 待加密的原始数据 | |
| 返回值 | Base64编码的加密数据 | | |

* + 1. **RSA 加密 证书公钥**

|  |  |  |
| --- | --- | --- |
| 方法 | public byte[] rsaEncryptByCertificate(byte[] certificatePath, byte[] data) throws AFCryptoException; | |
| 作用 | RSA加密，使用RSA数字证书对数据进行加密 | |
| 参数 | certificatePath | 证书路径 |
| data | 待加密的原始数据 |
| 返回值 | Base64编码的加密数据 | |

* + 1. **RSA解密 内部私钥**

|  |  |  |
| --- | --- | --- |
| 方法 | public byte[] rsaDecrypt(int keyIndex, byte[] encData) throws AFCryptoException; | |
| 作用 | RSA解密，使用内部密钥进行RSA解密 | |
| 参数 | keyIndex | 密码设备内部存储的RSA索引号 |
| encData | Base64编码的加密数据 |
| 返回值 | 原始数据 | |

* + 1. **RSA解密 外部私钥**

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | public byte[] rsaDecrypt(byte[] privateKey, byte[] encData) throws AFCryptoException; | | |
| 作用 | RSA解密，使用外部密钥进行RSA解密 | | |
| 参数 | privateKey | base64编码的RSA私钥数据，其结构应满足PKCS#1中的RSA结构定义 | 说明：  RSAPrivateKey ::= SEQUENCE {  version Version,  modulus INTEGER, --- n  publicExponent INTEGER, --- e  privateExponent INTEGER, --- d  prime1 INTEGER, --- p  prime2 INTEGER, --- q  exponent1 INTEGER, --- d mod (p-1)  exponent2 INTEGER, --- d mod (q-1)  coefficient INTEGER, --- (inverse of q) mod p  otherPrimeInfos OtherPrimeInfos OPTIONAL  } |
| encData | Base64编码的加密数据 | |
| 返回值 | 原始数据 | | |

* 1. **SM2 签名验签**
     1. **SM2 签名 内部私钥**

|  |  |  |
| --- | --- | --- |
| 方法 | public byte[] sm2Signature(int index, byte[] data) throws AFCryptoException; | |
| 作用 | SM2内部密钥签名，使用签名服务器内部密钥进行 SM2签名运算 | |
| 参数 | index | 待签名的签名服务器内部密钥索引 |
| data | 待签名的数据 |
| 返回值 | base64编码的签名数据 | |

* + 1. **SM2 签名 外部私钥 不带z值**

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | public byte[] sm2Signature(byte[] privateKey, byte[] data) throws AFCryptoException; | | |
| 作用 | SM2外部密钥签名 不带Z值 | | |
| 参数 | data | 待签名的数据 | |
| privateKey | base64编码的SM2私钥数据, 其结构应满足 GM/T 0009-2012中关于SM2私钥结构的数据定义 | 说明：  SM2PrivateKey ::= INTEGER |
| 返回值 | base64编码的签名数据 | | |

* + 1. **SM2 签名 外部私钥 带z值**

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | byte[] sm2SignatureByCertificate(byte[] data, byte[] privateKey, byte[] base64Certificate) throws AFCryptoException; | | |
| 作用 | SM2外部密钥签名 带Z值 | | |
| 参数 | data | 待签名的数据 | |
| privateKey | base64编码的SM2私钥数据, 其结构应满足 GM/T 0009-2012中关于SM2私钥结构的数据定义 | 说明：  SM2PrivateKey ::= INTEGER |
| 返回值 | base64编码的签名数据 | | |

* + 1. **SM2 签名 外部私钥+证书 带z值**

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | public byte[] sm2SignatureByCertificate(byte[] privateKey, byte[] data, byte[] base64Certificate) throws AFCryptoException; | | |
| 作用 | 基于一张证书的SM2签名 带Z值 | | |
| 参数 | data | 待签名的数据 | |
| base64Certificate | 证书 | |
| privateKey | base64编码的SM2私钥数据, 其结构应满足 GM/T 0009-2012中关于SM2私钥结构的数据定义 | 说明：  SM2PrivateKey ::= INTEGER |
| 返回值 | base64编码的签名数据 | | |

* + 1. **SM2 文件签名 内部密钥**

|  |  |  |
| --- | --- | --- |
| 方法 | public byte[] sm2SignFile(int index, byte[] filePath) throws AFCryptoException; | |
| 作用 | SM2文件签名，使用签名服务器内部密钥对文件进行 SM2签名运算 | |
| 参数 | index | 待签名的签名服务器内部密钥索引 |
| fileNafilePathme | 待签名的文件路径 |
| 返回值 | base64编码的签名数据 | |

* + 1. **SM2 文件签名 外部私钥 不带Z值**

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | public byte[] sm2SignFile(byte[] privateKey, byte[] filePath) throws AFCryptoException; | | |
| 作用 | SM2文件签名，使用外部密钥对文件进行 SM2签名运算 | | |
| 参数 | filePath | 待签名的文件路径 | |
| privateKey | base64编码的SM2私钥数据, 其结构应满足 GM/T 0009-2012中关于SM2私钥结构的数据定义 | 说明：  SM2PrivateKey ::= INTEGER |
| 返回值 | base64编码的签名数据 | | |

* + 1. **SM2 文件签名 外部私钥 带Z值**

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | public byte[] sm2SignFileByPrivateKey(byte[] privateKey, byte[] filePath) throws AFCryptoException; | | |
| 作用 | 基于证书的SM2文件签名 | | |
| 参数 | filePath | 待签名的文件路径 | |
| privateKey | base64编码的SM2私钥数据, 其结构应满足 GM/T 0009-2012中关于SM2私钥结构的数据定义 | 说明：  SM2PrivateKey ::= INTEGER |
| 返回值 | base64编码的签名数据 | | |

* + 1. **SM2 文件签名 外部私钥 +证书 带Z值**

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | public byte[] sm2SignFileByCertificate(byte[] privateKey, byte[] filePath, byte[] base64CertificatePath) throws AFCryptoException; | | |
| 作用 | 基于证书的SM2文件签名 | | |
| 参数 | filePath | 待签名的文件路径 | |
| privateKey | base64编码的SM2私钥数据, 其结构应满足 GM/T 0009-2012中关于SM2私钥结构的数据定义 | 说明：  SM2PrivateKey ::= INTEGER |
| base64CertificatePath | 签名的外部证书路径 | |
| 返回值 | base64编码的签名数据 | | |

* + 1. **SM2 验签 内部公钥**

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | public boolean sm2Verify(int keyIndex, byte[] data, byte[] signature) throws AFCryptoException; | | |
| 作用 | SM2内部密钥验证签名，使用签名服务器内部密钥进行 SM2验证签名运算 | | |
| 参数 | keyIndex | 待验证签名的签名服务器内部密钥索引 | |
| data | 待验证签名的原始数据 | |
| signature | 待验证签名的签名数据---BASE64编码格式, 其结构应满足 GM/T 0009-2012中关于SM2签名数据结构的定义 | 说明：  SM2Signature ::= {  R INTEGER, --签名值的第一部分  S INTEGER --签名值的第二部分  } |
| 返回值 | true | 验证签名成功 | |
| false | 验证签名失败 | |

* + 1. **SM2 验签 外部公钥 不带Z值**

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | public boolean sm2Verify(byte[] publicKey, byte[] data, byte[] signature) throws AFCryptoException; | | |
| 作用 | SM2 外部公钥验签 | | |
| 参数 | publicKey | 待验证签名的签名服务器内部密钥索引 | |
| data | 待验证签名的原始数据 | |
| signature | 待验证签名的签名数据---BASE64编码格式, 其结构应满足 GM/T 0009-2012中关于SM2签名数据结构的定义 | 说明：  SM2Signature ::= {  R INTEGER, --签名值的第一部分  S INTEGER --签名值的第二部分  } |
| 返回值 | true | 验证签名成功 | |
| false | 验证签名失败 | |

* + 1. **SM2 验签 一张证书 带Z值**

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | public boolean sm2VerifyByCertificate(byte[] cert, byte[] data, byte[] signature) throws AFCryptoException; | | |
| 作用 | SM2 外部公钥验签 携带Z值 | | |
| 参数 | cert | 证书 | |
| data | 待验证签名的原始数据 | |
| signature | 待验证签名的签名数据---BASE64编码格式, 其结构应满足 GM/T 0009-2012中关于SM2签名数据结构的定义 | 说明：  SM2Signature ::= {  R INTEGER, --签名值的第一部分  S INTEGER --签名值的第二部分  } |
| 返回值 | true | 验证签名成功 | |
| false | 验证签名失败 | |

* + 1. **SM2 验签 两张证书 带Z值**

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | public boolean sm2VerifyByCertificate(byte[] signCert, byte[] hashCert, byte[] data, byte[] signData) throws AFCryptoException; | | |
| 作用 | SM2 外部公钥验签 携带Z值,需要两张证书,签名证书和hash证书 | | |
| 参数 | signCert | 签名证书 Base64编码 DER格式 | |
| hashCert | 杂凑证书 Base64编码 DER格式 | |
| data | 原始数据 | |
| signature | 待验证签名的签名数据---BASE64编码格式, 其结构应满足 GM/T 0009-2012中关于SM2签名数据结构的定义 | 说明：  SM2Signature ::= {  R INTEGER, --签名值的第一部分  S INTEGER --签名值的第二部分  } |
| 返回值 | true | 验证签名成功 | |
| false | 验证签名失败 | |

* + 1. **SM2 验证文件签名 内部公钥**

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | public boolean sm2VerifyFile(int keyIndex, byte[] filePath, byte[] signature) throws AFCryptoException; | | |
| 作用 | SM2 内部密钥 验证文件签名 | | |
| 参数 | keyIndex | 内部密钥索引 | |
| filePath | 文件路径 | |
| signature | 待验证签名的签名数据---BASE64编码格式, 其结构应满足 GM/T 0009-2012中关于SM2签名数据结构的定义 | 说明：  SM2Signature ::= {  R INTEGER, --签名值的第一部分  S INTEGER --签名值的第二部分  } |
| 返回值 | true | 验证签名成功 | |
| false | 验证签名失败 | |

* + 1. **SM2 验证文件签名 外部公钥 不带Z值**

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | public boolean sm2VerifyFile(byte[] sm2PublicKey, byte[] filePath, byte[] signature) throws AFCryptoException; | | |
| 作用 | SM2 内部密钥 验证文件签名 | | |
| 参数 | sm2PublicKey | 外部公钥 | |
| filePath | 文件路径 | |
| signature | 待验证签名的签名数据---BASE64编码格式, 其结构应满足 GM/T 0009-2012中关于SM2签名数据结构的定义 | 说明：  SM2Signature ::= {  R INTEGER, --签名值的第一部分  S INTEGER --签名值的第二部分  } |
| 返回值 | true | 验证签名成功 | |
| false | 验证签名失败 | |

* + 1. **SM2 验证文件签名 一张证书 带Z值**

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | public boolean sm2VerifyFileByCertificate(byte[] base64Certificate, byte[] filePath, byte[] signature) throws AFCryptoException; | | |
| 作用 | SM2 内部密钥 验证文件签名 | | |
| 参数 | base64Certificate | 证书 Base64编码 DER格式 | |
| filePath | 文件路径 | |
| signature | 待验证签名的签名数据---BASE64编码格式, 其结构应满足 GM/T 0009-2012中关于SM2签名数据结构的定义 | 说明：  SM2Signature ::= {  R INTEGER, --签名值的第一部分  S INTEGER --签名值的第二部分  } |
| 返回值 | true | 验证签名成功 | |
| false | 验证签名失败 | |

* + 1. **SM2 验证文件签名 两张证书 带Z值**

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | public boolean sm2VerifyFileByCertificate(byte[] signCert, byte[] hashCert, byte[] filePath, byte[] signature) throws AFCryptoException; | | |
| 作用 | SM2 内部密钥 验证文件签名 | | |
| 参数 | signCert | 签名证书 Base64编码 DER格式 | |
| hashCert | 摘要证书 Base64编码 DER格式 | |
| filePath | 文件路径 | |
| signature | 待验证签名的签名数据---BASE64编码格式, 其结构应满足 GM/T 0009-2012中关于SM2签名数据结构的定义 | 说明：  SM2Signature ::= {  R INTEGER, --签名值的第一部分  S INTEGER --签名值的第二部分  } |
| 返回值 | true | 验证签名成功 | |
| false | 验证签名失败 | |

* 1. **SM2 加解密**
     1. **SM2 加密 内部公钥**

|  |  |  |
| --- | --- | --- |
| 方法 | public byte[] sm2Encrypt(int keyIndex, byte[] inData) throws AFCryptoException; | |
| 作用 | sm2加密，使用内部密钥进行SM2加密 | |
| 参数 | keyIndex | 密码设备内部密钥索引 |
| inData | 待加密的数据原文 |
| 返回值 | Base64编码的密文数据 | |

* + 1. **SM2 加密 外部公钥**

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | public byte[] sm2Encrypt(byte[] publicKey, byte[] inData) throws AFCryptoException; | | |
| 作用 | sm2加密，使用外部钥进行SM2加密 | | |
| 参数 | publicKey | base64编码的SM2公钥数据, 其结构应满足 GM/T 0009-2012中关于SM2公钥结构的数据定义 | 说明：  SM2PublicKey ::= BIT STRING  其结构为 04||X||Y |
| inData | 待加密的数据原文 | |
| 返回值 | Base64编码的密文数据 | | |

* + 1. **SM2 加密 使用证书**

|  |  |  |
| --- | --- | --- |
| 方法 | public byte[] sm2EncryptByCertificate(byte[] certificate, byte[] inData) throws AFCryptoException; | |
| 作用 | sm2加密，使用SM2证书对数据进行加密 | |
| 参数 | certificate | 证书 Base64编码 DER格式 |
| inData | 待加密的数据原文 |
| 返回值 | Base64编码的密文数据 | |

* + 1. **SM2 解密 内部私钥**

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | public byte[] sm2Decrypt(int keyIndex, byte[] encData) throws AFCryptoException; | | |
| 作用 | sm2解密，使用内部密钥进行SM2解密 | | |
| 参数 | keyIndex | 密码设备内部密钥索引 | |
| encData | Base64编码的加密数据, 其结构应满足 GM/T 0009-2012中关于SM2公钥结构的数据定义 | 说明：  SM2Cipher ::= SEQUENCE {  XCoordinate INTEGER, --x分量  YCoordinate INTEGER, --y分量  HASH OCTET STRING SIZE(32), --杂凑值  CipherText OCTET STRING, --密文  } |
| 返回值 | 原文数据 | | |

* + 1. **SM2 解密 外部私钥**

|  |  |  |  |
| --- | --- | --- | --- |
| 方法 | public byte[] sm2Decrypt(byte[] privateKey, byte[] encData) throws AFCryptoException; | | |
| 作用 | sm2解密，使用外部钥进行SM2解密 | | |
| 参数 | privateKey | base64编码的SM2私钥数据, 其结构应满足 GM/T 0009-2012中关于SM2私钥结构的数据定义 | 说明：  SM2PrivateKey ::= INTEGER |
| encData | Base64编码的加密数据, 其结构应满足 GM/T 0009-2012中关于SM2公钥结构的数据定义 | 说明：  SM2Cipher ::= SEQUENCE {  XCoordinate INTEGER, --x分量  YCoordinate INTEGER, --y分量  HASH OCTET STRING SIZE(32), --杂凑值  CipherText OCTET STRING, --密文  } |
| 返回值 | 原文数据 | | |

* 1. **对称加密**
     1. **SM4 ECB 内部密钥加密**

|  |  |  |
| --- | --- | --- |
| 方法 | public byte[] sm4InternalEncryptECB(int keyIndex, byte[] plain) throws AFCryptoException; | |
| 作用 | SM4 ECB 内部密钥加密 | |
| 参数 | keyIndex | 密钥索引 |
| plain | 原始数据 |
| 返回值 | 加密数据 | |

* + 1. **SM4 ECB 外部密钥加密**

|  |  |  |
| --- | --- | --- |
| 方法 | public byte[] sm4ExternalEncryptECB(byte[] key, byte[] plain) throws AFCryptoException; | |
| 作用 | SM4 ECB 外部密钥加密 | |
| 参数 | key | 密钥 |
| plain | 原始数据 |
| 返回值 | 加密数据 | |

* + 1. **SM4 ECB 密钥句柄加密**

|  |  |  |
| --- | --- | --- |
| 方法 | public byte[] sm4HandleEncryptECB(int keyHandle, byte[] plain) throws AFCryptoException; | |
| 作用 | SM4 ECB 密钥句柄加密 | |
| keyHandle | 密钥句柄 |
| plain | 原始数据 |
| 返回值 | 加密数据 | |

* + 1. **SM4 CBC 内部密钥加密**

|  |  |  |
| --- | --- | --- |
| 方法 | public byte[] sm4InternalEncryptCBC(int keyIndex, byte[] iv, byte[] plain) throws AFCryptoException; | |
| 作用 | SM4 CBC 内部密钥加密 | |
| keyIndex | 密钥索引 |
| iv | 初始向量 |
| plain | 原始数据 |
| 返回值 | 加密数据 | |

* + 1. **SM4 CBC 外部密钥加密**

|  |  |  |
| --- | --- | --- |
| 方法 | public byte[] sm4ExternalEncryptCBC(byte[] key, byte[] iv, byte[] plain) throws AFCryptoException; | |
| 作用 | SM4 CBC 外部密钥加密 | |
| key | 密钥索 |
| iv | 初始向量 |
| plain | 原始数据 |
| 返回值 | 加密数据 | |

* + 1. **SM4 CBC 密钥句柄加密**

|  |  |  |
| --- | --- | --- |
| 方法 | public byte[] sm4HandleEncryptCBC(int keyHandle, byte[] iv, byte[] plain) throws AFCryptoException; | |
| 作用 | SM4 CBC 密钥句柄加密 | |
| keyHandle | 密钥句柄 |
| iv | 初始向量 |
| plain | 原始数据 |
| 返回值 | 加密数据 | |

* + 1. **SM1 ECB 内部密钥加密**

|  |  |  |
| --- | --- | --- |
| 方法 | public byte[] sm1InternalEncryptECB(int keyIndex, byte[] plain) throws AFCryptoException; | |
| 作用 | SM1 ECB 内部密钥加密 | |
| keyIndex | 密钥索引 |
| plain | 原始数据 |
| 返回值 | 加密数据 | |

* + 1. **SM1 ECB 外部密钥加密**

|  |  |  |
| --- | --- | --- |
| 方法 | public byte[] sm1ExternalEncryptECB(byte[] key, byte[] plain) throws AFCryptoException; | |
| 作用 | SM1 ECB 外部密钥加密 | |
| key | 密钥 |
| plain | 原始数据 |
| 返回值 | 加密数据 | |

* + 1. **SM1 ECB 密钥句柄加密**

|  |  |  |
| --- | --- | --- |
| 方法 | public byte[] sm1HandleEncryptECB(int keyHandle, byte[] plain) throws AFCryptoException; | |
| 作用 | SM1 ECB 密钥句柄加密 | |
| keyHandle | 密钥句柄 |
| plain | 原始数据 |
| 返回值 | 加密数据 | |

* + 1. **SM1 CBC 内部密钥加密**

|  |  |  |
| --- | --- | --- |
| 方法 | public byte[] sm1InternalEncryptCBC(int keyIndex, byte[] iv, byte[] plain) throws AFCryptoException; | |
| 作用 | SM1 CBC 内部密钥加密 | |
| keyIndex | 密钥索引 |
| Iv | 初始向量 |
| plain | 原始数据 |
| 返回值 | 加密数据 | |

* + 1. **SM1 CBC 外部密钥加密**

|  |  |  |
| --- | --- | --- |
| 方法 | public byte[] sm1ExternalEncryptCBC(byte[] key, byte[] iv, byte[] plain) throws AFCryptoException; | |
| 作用 | SM1 CBC 外部密钥加密 | |
| key | 密钥 |
| Iv | 初始向量 |
| plain | 原始数据 |
| 返回值 | 加密数据 | |

* + 1. **SM1 CBC 密钥句柄加密**

|  |  |  |
| --- | --- | --- |
| 方法 | public byte[] sm1ExternalEncryptCBC(byte[] key, byte[] iv, byte[] plain) throws AFCryptoException; | |
| 作用 | SM1 CBC 密钥句柄加密 | |
| key | 密钥 |
| Iv | 初始向量 |
| plain | 原始数据 |
| 返回值 | 加密数据 | |

* 1. **对称解密**
     1. **SM4 ECB 内部密钥解密**

|  |  |  |
| --- | --- | --- |
| 方法 | public byte[] sm4InternalDecryptECB(int keyIndex, byte[] cipher) throws AFCryptoException; | |
| 作用 | SM4 ECB 内部密钥解密 | |
| 参数 | keyIndex | 密钥索引 |
| cipher | 密文数据 |
| 返回值 | 明文数据 | |

* + 1. **SM4 ECB 外部密钥解密**

|  |  |  |
| --- | --- | --- |
| 方法 | public byte[] sm4ExternalDecryptECB(byte[] key, byte[] cipher) throws AFCryptoException; | |
| 作用 | SM4 ECB 外部密钥解密 | |
| key | 密钥 |
| cipher | 密文数据 |
| 返回值 | 明文数据 | |

* + 1. **SM4 ECB 密钥句柄解密**

|  |  |  |
| --- | --- | --- |
| 方法 | public byte[] sm4HandleDecryptECB(int keyHandle, byte[] cipher) throws AFCryptoException; | |
| 作用 | SM4 ECB 密钥句柄解密 | |
| keyHandle | 密钥句柄 |
| cipher | 密文数据 |
| 返回值 | 明文数据 | |

* + 1. **SM4 CBC 内部密钥解密**

|  |  |  |
| --- | --- | --- |
| 方法 | public byte[] sm4InternalDecryptCBC(int keyIndex, byte[] iv, byte[] cipher) throws AFCryptoException; | |
| 作用 | SM4 CBC 内部密钥解密 | |
| keyIndex | 密钥索引 |
| Iv | 初始向量 |
| cipher | 密文数据 |
| 返回值 | 明文数据 | |

* + 1. **SM4 CBC 外部密钥解密**

|  |  |  |
| --- | --- | --- |
| 方法 | public byte[] sm4ExternalDecryptCBC(byte[] key, byte[] iv, byte[] cipher) throws AFCryptoException; | |
| 作用 | SM4 CBC 外部密钥解密 | |
| key | 密钥 |
| Iv | 初始向量 |
| cipher | 密文数据 |
| 返回值 | 明文数据 | |

* + 1. **SM4 CBC 密钥句柄解密**

|  |  |  |
| --- | --- | --- |
| 方法 | public byte[] sm4HandleDecryptCBC(int keyHandle, byte[] iv, byte[] cipher) throws AFCryptoException; | |
| 作用 | SM4 CBC 内部密钥解密 | |
| 参数 | keyHandle | 密钥句柄 |
| Iv | 初始向量 |
| cipher | 密文数据 |
| 返回值 | 明文数据 | |

* + 1. **SM1 ECB 内部密钥解密**

|  |  |  |
| --- | --- | --- |
| 方法 | public byte[] sm1InternalDecryptECB(int keyIndex, byte[] cipher) throws AFCryptoException; | |
| 作用 | SM1 ECB 内部密钥解密 | |
| 参数 | keyIndex | 密钥索引 |
| cipher | 密文数据 |
| 返回值 | 明文数据 | |

* + 1. **SM1 ECB 外部密钥解密**

|  |  |  |
| --- | --- | --- |
| 方法 | public byte[] sm1ExternalDecryptECB(byte[] key, byte[] cipher) throws AFCryptoException; | |
| 作用 | SM1 ECB 外部密钥解密 | |
| 参数 | key | 密钥 |
| cipher | 密文数据 |
| 返回值 | 明文数据 | |

* + 1. **SM1 ECB 密钥句柄解密**

|  |  |  |
| --- | --- | --- |
| 方法 | public byte[] sm1HandleDecryptECB(int keyHandle, byte[] cipher) throws AFCryptoException; | |
| 作用 | SM1 ECB 密钥句柄解密 | |
| 参数 | keyHandle | 密钥句柄 |
| cipher | 密文数据 |
| 返回值 | 明文数据 | |

* + 1. **SM1 CBC 内部密钥解密**

|  |  |  |
| --- | --- | --- |
| 方法 | public byte[] sm1InternalDecryptCBC(int keyIndex, byte[] iv, byte[] plain) throws AFCryptoException; | |
| 作用 | SM1 CBC内部密钥解密 | |
| 参数 | keyIndex | 密钥索引 |
| iv | 初始向量 |
| cipher | 密文数据 |
| 返回值 | 明文数据 | |

* + 1. **SM1 CBC 外部密钥解密**

|  |  |  |
| --- | --- | --- |
| 方法 | public byte[] sm1ExternalDecryptCBC(byte[] key, byte[] iv, byte[] cipher) throws AFCryptoException; | |
| 作用 | SM1 CBC外部密钥解密 | |
| 参数 | key | 密钥 |
| iv | 初始向量 |
| cipher | 密文数据 |
| 返回值 | 明文数据 | |

* + 1. **SM1 CBC 密钥句柄解密**

|  |  |  |
| --- | --- | --- |
| 方法 | public byte[] sm1HandleDecryptCBC(int keyHandle, byte[] iv, byte[] cipher) throws AFCryptoException; | |
| 作用 | SM1 CBC密钥句柄解密 | |
| 参数 | keyHandle | 密钥句柄 |
| iv | 初始向量 |
| cipher | 密文数据 |
| 返回值 | 明文数据 | |

* 1. **对称批量加密**
     1. **SM4 内部批量加密 ECB**

|  |  |  |
| --- | --- | --- |
| 方法 | public List<byte[]> sm4InternalBatchEncryptECB(int keyIndex, List<byte[]> plainList) throws AFCryptoException; | |
| 作用 | SM4 内部密钥批量加密 ECB模式 | |
| 参数 | keyIndex | 密钥索引 |
| plainList | 明文数据列表 |
| 返回值 | 密文数据列表 | |

* + 1. **SM4外部批量加密 ECB**

|  |  |  |
| --- | --- | --- |
| 方法 | public List<byte[]> sm4ExternalBatchEncryptECB(byte[] key, List<byte[]> plainList) throws AFCryptoException; | |
| 作用 | SM4 外部密钥批量加密 ECB模式 | |
| 参数 | key | 密钥 |
| plainList | 明文数据列表 |
| 返回值 | 密文数据列表 | |

* + 1. **SM4 密钥句柄批量加密 ECB**

|  |  |  |
| --- | --- | --- |
| 方法 | public List<byte[]> sm4HandleBatchEncryptECB(int keyHandle, List<byte[]> plainList) throws AFCryptoException; | |
| 作用 | SM4 密钥句柄批量加密 ECB模式 | |
| 参数 | keyHandle | 密钥句柄 |
| plainList | 明文数据列表 |
| 返回值 | 密文数据列表 | |

* + 1. **SM4 内部批量加密 CBC**

|  |  |  |
| --- | --- | --- |
| 方法 | public List<byte[]> sm4InternalBatchEncryptCBC(int keyIndex, byte[] iv, List<byte[]> plainList) throws AFCryptoException; | |
| 作用 | SM4 内部密钥批量加密 CBC模式 | |
| 参数 | keyIndex | 密钥句柄 |
| iv | 初始向量 |
| plainList | 明文数据列表 |
| 返回值 | 密文数据列表 | |

* + 1. **SM4 外部批量加密 CBC**

|  |  |  |
| --- | --- | --- |
| 方法 | public List<byte[]> sm4ExternalBatchEncryptCBC(byte[] key, byte[] iv, List<byte[]> plainList) throws AFCryptoException; | |
| 作用 | SM4 外部密钥批量加密 CBC模式 | |
| 参数 | key | 密钥 |
| iv | 初始向量 |
| plainList | 明文数据列表 |
| 返回值 | 密文数据列表 | |

* + 1. **SM4 密钥句柄批量加密 CBC**

|  |  |  |
| --- | --- | --- |
| 方法 | public List<byte[]> sm4HandleBatchEncryptCBC(int keyHandle, byte[] iv, List<byte[]> plainList) throws AFCryptoException; | |
| 作用 | SM4 密钥句柄批量加密 CBC模式 | |
| 参数 | key | 密钥 |
| iv | 初始向量 |
| plainList | 明文数据列表 |
| 返回值 | 密文数据列表 | |

* + 1. **SM1内部批量加密 ECB**

|  |  |  |
| --- | --- | --- |
| 方法 | public List<byte[]> sm1InternalBatchEncryptECB(int keyIndex, List<byte[]> plainList) throws AFCryptoException; | |
| 作用 | SM1 内部密钥批量加密 ECB模式 | |
| 参数 | keyIndex | 密钥索引 |
| plainList | 明文数据列表 |
| 返回值 | 密文数据列表 | |

* + 1. **SM1外部批量加密 ECB**

|  |  |  |
| --- | --- | --- |
| 方法 | public List<byte[]> sm1ExternalBatchEncryptECB(byte[] key, List<byte[]> plainList) throws AFCryptoException; | |
| 作用 | SM1 外部密钥批量加密 ECB模式 | |
| 参数 | key | 密钥 |
| plainList | 明文数据列表 |
| 返回值 | 密文数据列表 | |

* + 1. **SM1 密钥句柄批量加密 ECB**

|  |  |  |
| --- | --- | --- |
| 方法 | public List<byte[]> sm1HandleBatchEncryptECB(int keyHandle, List<byte[]> plainList) throws AFCryptoException; | |
| 作用 | SM1 密钥句柄批量加密 ECB模式 | |
| 参数 | keyHandle | 密钥句柄 |
| plainList | 明文数据列表 |
| 返回值 | 密文数据列表 | |

* + 1. **SM1 内部批量加密 CBC**

|  |  |  |
| --- | --- | --- |
| 方法 | public List<byte[]> sm1InternalBatchEncryptCBC(int keyIndex, byte[] iv, List<byte[]> plainList) throws AFCryptoException; | |
| 作用 | SM1内部密钥批量加密 CBC模式 | |
| 参数 | key | 密钥 |
| iv | 初始向量 |
| plainList | 明文数据列表 |
| 返回值 | 密文数据列表 | |

* + 1. **SM1 外部批量加密 CBC**

|  |  |  |
| --- | --- | --- |
| 方法 | public List<byte[]> sm1ExternalBatchEncryptCBC(byte[] key, byte[] iv, List<byte[]> plainList) throws AFCryptoException; | |
| 作用 | SM1外部密钥批量加密 CBC模式 | |
| 参数 | key | 密钥 |
| iv | 初始向量 |
| plainList | 明文数据列表 |
| 返回值 | 密文数据列表 | |

* + 1. **SM1 密钥句柄批量加密 CBC**

|  |  |  |
| --- | --- | --- |
| 方法 | public List<byte[]> sm1HandleBatchEncryptCBC(int keyHandle, byte[] iv, List<byte[]> plainList) throws AFCryptoException; | |
| 作用 | SM1密钥句柄批量加密 CBC模式 | |
| 参数 | keyHandle | 密钥句柄 |
| iv | 初始向量 |
| plainList | 明文数据列表 |
| 返回值 | 密文数据列表 | |

* 1. **对称批量解密**
     1. **SM4 内部批量解密 ECB**

|  |  |  |
| --- | --- | --- |
| 方法 | public List<byte[]> sm4InternalBatchDecryptECB(int keyIndex, List<byte[]> cipherList) throws AFCryptoException; | |
| 作用 | SM4 内部密钥批量解密 ECB模式 | |
| 参数 | keyIndex | 密钥索引 |
| cipherList | 密文数据列表 |
| 返回值 | 明文数据列表 | |

* + 1. **SM4外部批量解密 ECB**

|  |  |  |
| --- | --- | --- |
| 方法 | public List<byte[]> sm4ExternalBatchDecryptECB(byte[] key, List<byte[]> cipherList) throws AFCryptoException; | |
| 作用 | SM4 外部密钥批量解密 ECB模式 | |
| 参数 | key | 密钥 |
| cipherList | 密文数据列表 |
| 返回值 | 明文数据列表 | |

* + 1. **SM4 密钥句柄批量解密 ECB**

|  |  |  |
| --- | --- | --- |
| 方法 | public List<byte[]> sm4HandleBatchDecryptECB(int keyHandle, List<byte[]> cipherList) throws AFCryptoException; | |
| 作用 | SM4 外部密钥批量解密 ECB模式 | |
| 参数 | keyHandle | 密钥句柄 |
| cipherList | 密文数据列表 |
| 返回值 | 明文数据列表 | |

* + 1. **SM4 内部批量解密 CBC**

|  |  |  |
| --- | --- | --- |
| 方法 | public List<byte[]> sm4InternalBatchDecryptCBC(int keyIndex, byte[] iv, List<byte[]> cipherList) throws AFCryptoException; | |
| 作用 | SM4 内部密钥批量解密 CBC模式 | |
| 参数 | keyIndex | 密钥索引 |
| iv | 初始向量 |
| cipherList | 密文数据列表 |
| 返回值 | 明文数据列表 | |

* + 1. **SM4 外部批量解密 CBC**

|  |  |  |
| --- | --- | --- |
| 方法 | public List<byte[]> sm4ExternalBatchDecryptCBC(byte[] key, byte[] iv, List<byte[]> cipherList) throws AFCryptoException; | |
| 作用 | SM4 外部密钥批量解密 CBC模式 | |
| 参数 | key | 密钥 |
| iv | 初始向量 |
| cipherList | 密文数据列表 |
| 返回值 | 明文数据列表 | |

* + 1. **SM4 密钥句柄批量解密 CBC**

|  |  |  |
| --- | --- | --- |
| 方法 | public List<byte[]> sm4HandleBatchDecryptCBC(int keyHandle, byte[] iv, List<byte[]> cipherList) throws AFCryptoException; | |
| 作用 | SM4 密钥句柄批量解密 CBC模式 | |
| 参数 | keyHandle | 密钥句柄 |
| iv | 初始向量 |
| cipherList | 密文数据列表 |
| 返回值 | 明文数据列表 | |

* + 1. **SM1内部批量解密 ECB**

|  |  |  |
| --- | --- | --- |
| 方法 | public List<byte[]> sm1InternalBatchDecryptECB(int keyIndex, List<byte[]> cipherList) throws AFCryptoException; | |
| 作用 | SM1 内部密钥批量解密 ECB模式 | |
| 参数 | keyIndex | 密钥索引 |
| cipherList | 密文数据列表 |
| 返回值 | 明文数据列表 | |

* + 1. **SM1外部批量解密 ECB**

|  |  |  |
| --- | --- | --- |
| 方法 | public List<byte[]> sm1ExternalBatchDecryptECB(byte[] key, List<byte[]> cipherList) throws AFCryptoException; | |
| 作用 | SM1 外部密钥批量解密 ECB模式 | |
| 参数 | key | 密钥 |
| cipherList | 密文数据列表 |
| 返回值 | 明文数据列表 | |

* + 1. **SM1 密钥句柄批量解密 ECB**

|  |  |  |
| --- | --- | --- |
| 方法 | public List<byte[]> sm1HandleBatchDecryptECB(int keyHandle, List<byte[]> cipherList) throws AFCryptoException; | |
| 作用 | SM1 密钥句柄批量解密 ECB模式 | |
| 参数 | keyHandle | 密钥句柄 |
| cipherList | 密文数据列表 |
| 返回值 | 明文数据列表 | |

* + 1. **SM1 内部批量解密 CBC**

|  |  |  |
| --- | --- | --- |
| 方法 | public List<byte[]> sm1InternalBatchDecryptCBC(int keyIndex, byte[] iv, List<byte[]> cipherList) throws AFCryptoException; | |
| 作用 | SM1 内部密钥批量解密 CBC模式 | |
| 参数 | keyIndex | 密钥索引 |
| iv | 初始向量 |
| cipherList | 密文数据列表 |
| 返回值 | 明文数据列表 | |

* + 1. **SM1 外部批量解密 CBC**

|  |  |  |
| --- | --- | --- |
| 方法 | public List<byte[]> sm1ExternalBatchDecryptCBC(byte[] key, byte[] iv, List<byte[]> cipherList) throws AFCryptoException; | |
| 作用 | SM1 外部密钥批量解密 CBC模式 | |
| 参数 | key | 密钥 |
| iv | 初始向量 |
| cipherList | 密文数据列表 |
| 返回值 | 明文数据列表 | |

* + 1. **SM1 密钥句柄批量解密 CBC**

|  |  |  |
| --- | --- | --- |
| 方法 | public List<byte[]> sm1HandleBatchDecryptCBC(int keyHandle, byte[] iv, List<byte[]> cipherList) throws AFCryptoException; | |
| 作用 | SM1 密钥句柄批量解密 CBC模式 | |
| 参数 | keyHandle | 密钥句柄 |
| iv | 初始向量 |
| cipherList | 密文数据列表 |
| 返回值 | 明文数据列表 | |

* 1. **MAC计算**
     1. **SM4 计算MAC 内部密钥**

|  |  |  |
| --- | --- | --- |
| 方法 | public byte[] sm4InternalMac(int keyIndex, byte[] iv, byte[] data) throws AFCryptoException; | |
| 作用 | SM4 算法计算MAC 内部密钥 | |
| 参数 | keyIndex | 密钥索引 |
| iv | 初始向量 |
| data | 计算数据 |
| 返回值 | MAC值 | |

* + 1. **SM4 计算MAC 外部密钥**

|  |  |  |
| --- | --- | --- |
| 方法 | public byte[] sm4ExternalMac(byte[] key, byte[] iv, byte[] data) throws AFCryptoException; | |
| 作用 | SM4 算法计算MAC 外部密钥 | |
| 参数 | key | 密钥 |
| iv | 初始向量 |
| data | 计算数据 |
| 返回值 | MAC值 | |

* + 1. **SM4 计算MAC 密钥句柄**

|  |  |  |
| --- | --- | --- |
| 方法 | public byte[] sm4HandleMac(int keyHandle, byte[] iv, byte[] data) throws AFCryptoException; | |
| 作用 | SM4 算法计算MAC 外部密钥 | |
| 参数 | keyHandle | 密钥句柄 |
| iv | 初始向量 |
| data | 计算数据 |
| 返回值 | MAC值 | |

* + 1. **SM1 计算MAC 内部密钥**

|  |  |  |
| --- | --- | --- |
| 方法 | public byte[] sm1InternalMac(int keyIndex, byte[] iv, byte[] data) throws AFCryptoException; | |
| 作用 | SM1 算法计算MAC 内部密钥 | |
| 参数 | keyIndex | 密钥索引 |
| iv | 初始向量 |
| data | 计算数据 |
| 返回值 | MAC值 | |

* + 1. **SM1 计算MAC 外部密钥**

|  |  |  |
| --- | --- | --- |
| 方法 | public byte[] sm1ExternalMac(byte[] key, byte[] iv, byte[] data) throws AFCryptoException; | |
| 作用 | SM1 算法计算MAC 外部密钥 | |
| 参数 | key | 密钥 |
| iv | 初始向量 |
| data | 计算数据 |
| 返回值 | MAC值 | |

* + 1. **SM1 计算MAC 密钥句柄**

|  |  |  |
| --- | --- | --- |
| 方法 | public byte[] sm1HandleMac(int keyHandle, byte[] iv, byte[] data) throws AFCryptoException; | |
| 作用 | SM1 算法计算MAC 外部密钥 | |
| 参数 | keyHandle | 密钥句柄 |
| iv | 初始向量 |
| data | 计算数据 |
| 返回值 | MAC值 | |

* + 1. **SM3-HMAC**

|  |  |  |
| --- | --- | --- |
| 方法 | public byte[] sm3Hmac(byte[] key, byte[] data) throws AFCryptoException; | |
| 作用 | SM3-HMAC 计算 | |
| 参数 | key | 密钥 |
| data | 计算数据 |
| 返回值 | HMAC值 | |

* 1. **Hash计算**
     1. **Init**

|  |  |
| --- | --- |
| 方法 | public void sm3HashInit() throws AFCryptoException; |
| 作用 | Hash初始化 需要与Hash update Hash doFinal 共同完成一次摘要操作 |

* + 1. **Init 带公钥**

|  |  |  |
| --- | --- | --- |
| 方法 | public void sm3HashInitWithPubKey(SM2PublicKey publicKey, byte[] userId) throws AFCryptoException; | |
| 作用 | 携带公钥的Hash初始化 需要与Hash update Hash doFinal 共同完成一次摘要计算 | |
| 参数 | publicKey | 公钥 |
| userId | 用户id |

* + 1. **Update**

|  |  |  |
| --- | --- | --- |
| 方法 | public void sm3HashUpdate(byte[] data) throws AFCryptoException; | |
| 作用 | 追加需要Hash的数据 需要与init doFinal 共同完成一次摘要计算 | |
| 参数 | data | 追加的数据 |

* + 1. **doFinal**

|  |  |
| --- | --- |
| 方法 | public byte[] sm3HashFinal() throws AFCryptoException; |
| 作用 | 返回所有追加数据的Hash值 需要init update 共同完成一次摘要计算 |
| 返回值 | Hash值 |

* + 1. **Hash**

|  |  |  |
| --- | --- | --- |
| 方法 | public byte[] sm3Hash(byte[] data) throws AFCryptoException; | |
| 作用 | Hash运算 不带公钥 | |
| 参数 | data | 计算数据 |
| 返回值 | Hash值 | |

* + 1. **Hash 带公钥**

|  |  |  |
| --- | --- | --- |
| 方法 | public byte[] sm3HashWithPubKey(SM2PublicKey publicKey, byte[] userId, byte[] data) throws AFCryptoException; | |
| 作用 | Hash运算 携带公钥 | |
| 参数 | publicKey | 公钥 |
| data | 计算数据 |
| userId | 用户id |
| 返回值 | Hash值 | |

* 1. **获取内部密钥句柄**

|  |  |  |
| --- | --- | --- |
| 方法 | public int getSymKeyHandle(int keyIndex) throws AFCryptoException; | |
| 作用 | 根据内部密钥索引获取该密钥的密钥句柄 | |
| 参数 | keyIndex | 密钥索引 |
| 返回值 | 密钥句柄 | |

* 1. **证书管理**
     1. **获取证书的个数**

|  |  |  |
| --- | --- | --- |
| 方法 | public int getCertCountByAltName(byte[] altName) throws AFCryptoException; | |
| 作用 | 获取存储在设备的证书个数 | |
| 参数 | altName | 证书别名 |
| 返回值 | 证书个数 | |

* + 1. **根据别名获取单个证书**

|  |  |  |
| --- | --- | --- |
| 方法 | public byte[] getCertByAltName(byte[] altName, int certIndex) throws AFCryptoException; | |
| 作用 | 根据证书名字获取证书 | |
| 参数 | altName | 证书别名 |
| certIndex | 证书索引号(与函数getCertCountByAltName中获取到的值相匹配) |
| 返回值 | 证书 Base64编码 | |

* + 1. **获取所有 CA 证书的别名**

|  |  |
| --- | --- |
| 方法 | public CertAltNameTrustList getCertTrustListAltName() throws AFCryptoException ; |
| 作用 | 获取所有CA证书的别名 |
| 返回值 | 信任列表别名组合，如： CA001|CA002|CA003 |

* + 1. **验证证书有效性**

|  |  |  |
| --- | --- | --- |
| 方法 | public int validateCertificate(byte[] cert) throws AFCryptoException; | |
| 作用 | 验证证书有效性 | |
| 参数 | cert | 证书 Base64编码 DER格式 |
| 返回值 | 0：验证成功，其他：验证失败 | |

* + 1. **验证证书是否被吊销**

|  |  |  |
| --- | --- | --- |
| 方法 | public boolean isCertificateRevoked(byte[] base64Certificate, byte[] crlData) throws CertificateException, AFCryptoException; | |
| 作用 | 验证证书是否被吊销 | |
| 参数 | base64Certificate | 待验证的证书--BASE64编码 DER格式 |
| crlData | 待验证证书的CRL文件数据 --BASE64编码 |
| 返回值 | 返回证书验证结果，true ：当前证书已被吊销, false ：当前证书未被吊销 | |

* + 1. **获取证书信息**

|  |  |  |
| --- | --- | --- |
| 方法 | public byte[] getCertInfo(byte[] base64Certificate, int certInfoType) throws AFCryptoException; | |
| 作用 | 获取证书信息 | |
| 参数 | base64Certificate | 待验证的证书--BASE64编码 DER格式 |
| certInfoType | 用户待获取的证书内容类型 : 类型定义在类{@link com.af.constant.CertParseInfoType} |
| 返回值 | 证书信息 | |

* + 1. **获取证书扩展信息**

|  |  |  |
| --- | --- | --- |
| 方法 | public byte[] getCertInfoByOid(byte[] base64Certificate, byte[] certInfoOid) throws AFCryptoException; | |
| 作用 | 根据OID获取证书拓展信息 | |
| 参数 | base64Certificate | 待验证的证书--BASE64编码 DER格式 |
| certInfoOid | 用户待获取的证书内容类型OID值 : OID值定义在类 {@link com.af.constant.CertParseInfoType} |
| 返回值 | 证书拓展信息 | |

* + 1. **获取设备证书**
       1. **获取设备证书,优先获取签名证书**

|  |  |
| --- | --- |
| 方法 | public byte[] getServerCert() throws AFCryptoException; |
| 作用 | 读取当前应用的服务器的签名证书，如果有签名证书则得到签名证书，否则得到加密证书 |
| 返回值 | Base64编码的服务器证书 |

* + - 1. **获取指定类型的设备证书**

|  |  |  |
| --- | --- | --- |
| 方法 | public byte[] getServerCertByUsage(int usage) throws AFCryptoException; | |
| 作用 | 根据OID获取证书拓展信息 | |
| 参数 | usage | 证书用途 1：加密证书 | 2：签名证书 |
| 返回值 | Base64编码的服务器证书 | |

* + 1. **获取应用实体信息**

|  |  |  |
| --- | --- | --- |
| 方法 | public AFSvCryptoInstance getInstance(String policyName) throws AFCryptoException ; | |
| 作用 | 根据策略名称获取应用策略，此应用策略为用户在管理程序中创建。用户获取应用策略后，签名服务器会根据用户设定的策略内容进行相关的服务操作 | |
| 参数 | policyName | 策略名称 |
| 返回 | AFSvCryptoInstance | public class AFSvCryptoInstance {  public String policyName; //应用实体名称  public int keyIndex; //密钥索引  public int keyType; //密钥类型 3:SM2 4:RSA  public int policy; //策略 该参数无用，兼容以前版本  } |

* + 1. **根据证书的 DN 信息获取 CA 证书**

|  |  |  |
| --- | --- | --- |
| 方法 | public byte[] getCaCertByDn(byte[] dn) throws AFCryptoException ; | |
| 作用 | 根据颁发者信息获取CA证书 | |
| 参数 | dn | 颁发者信息 |
| 返回 | CA证书 Base64编码 | |

* + 1. **获取应用实体 签名证书**

|  |  |  |
| --- | --- | --- |
| 方法 | public byte[] getSignCertByPolicyName(String policyName) throws AFCryptoException; | |
| 作用 | 获取应用实体签名证书 | |
| 参数 | policyName | 应用实体名称 |
| 返回 | 签名证书 Base64编码 | |

* + 1. **获取应用实体 加密证书**

|  |  |  |
| --- | --- | --- |
| 方法 | public byte[] getEncCertByPolicyName(String policyName) throws AFCryptoException; | |
| 作用 | 获取应用实体加密证书 | |
| 参数 | policyName | 应用实体名称 |
| 返回 | 加密证书 Base64编码 | |

* + 1. **获取证书的OCSP地址**

|  |  |  |
| --- | --- | --- |
| 方法 | public byte[] getOcspUrl(byte[] base64Certificate) throws AFCryptoException; | |
| 作用 | 获取证书的OCSP地址 | |
| 参数 | base64Certificate | 证书 Base64编码 |
| 返回 | OCSP地址 | |

* 1. **PKCS7编码解码数据**
     1. **PKCS7 签名信息编码 不带原文**

|  |  |  |
| --- | --- | --- |
| 方法 | public byte[] encodeSignedDataForSM2(byte[] priKey, byte[] base64Certificate, byte[] data) throws AFCryptoException; | |
| 作用 | 编码签名信息 不带原文 | |
| 参数 | priKey | 私钥 Base64编码 DER格式 |
| base64Certificate | 证书 Base64编码 |
| data | 待签名数据 |
| 返回 | 名编码信息数据 Base64编码 DER格式 | |

* + 1. **PKCS7 签名信息编码**

|  |  |  |
| --- | --- | --- |
| 方法 | public byte[] encodeSignedDataForSM2(boolean ifCarryText, byte[] privateKey, byte[] signerCertificate, byte[] data) throws AFCryptoException; | |
| 作用 | 编码签名信息 | |
| 参数 | ifCarryText | 是否携带原文 true 携带原文 false 不携带原文 |
| priKey | 私钥 Base64编码 DER格式 |
| base64Certificate | 证书 Base64编码 |
| data | 待签名数据 |
| 返回 | 签名编码信息数据 Base64编码 DER格式 | |

* + 1. **PKCS7 签名信息解码**

|  |  |  |
| --- | --- | --- |
| 方法 | public AFSM2DecodeSignedData decodeSignedDataForSM2(byte[] signedData) throws AFCryptoException ; | |
| 作用 | 解码签名信息 | |
| 参数 | signedData | Base64编码的签名数据 |
| 返回值 | AFSM2DecodeSignedData | 解码后的数据，包括签名者证书，HASH算法标识，被签名的数据以及签名值 |

* + 1. **PKCS7 签名信息验证**

|  |  |  |
| --- | --- | --- |
| 方法 | public boolean verifySignedDataForSM2(byte[] signedData, byte[] rawData) throws AFCryptoException ; | |
| 作用 | 解码签名信息 | |
| 参数 | signedData | Base64编码的签名数据 |
| rawData | 原文 |
| 返回值 | true 验证成功 false 验证失败 | |

* + 1. **PKCS7 带签名信息的数字信封编码**

|  |  |  |
| --- | --- | --- |
| 方法 | public byte[] encodeEnvelopedDataForSM2(byte[] priKey, byte[] symKey, byte[] signCert, byte[] encCert, byte[] data) throws AFCryptoException ; | |
| 作用 | 解码签名信息 | |
| 参数 | priKey | 私钥 Base64编码DER格式 |
| symKey | 对称密钥 |
| signCert | 签名证书 |
| encCert | 加密证书 |
| rawData | 原文 |
| 返回值 | 编码数据 | |

* + 1. **PKCS7 带签名信息的数字信封解码**

|  |  |  |
| --- | --- | --- |
| 方法 | public AFPkcs7DecodeData decodeEnvelopedDataForSM2(byte[] priKey, byte[] encodeData) throws AFCryptoException; | |
| 作用 | 解码签名信息 | |
| 参数 | priKey | 私钥 Base64编码 DER格式 |
| encodeData | 编码数据 |
| 返回值 | 解码数据 | |

* 1. **获取私钥访问权限**

|  |  |  |
| --- | --- | --- |
| 方法 | public void getPrivateAccess(int index, int keyType, String psw) throws AFCryptoException | |
| 作用 | 使用内部私钥进行计算之前,需要获取私钥访问权限 | |
| 参数 | index | 内部密钥索引 |
| keyType | 密钥类型 3:SM2 4:RSA |
| psw | 私钥授权码,在管理平台密钥管理-私钥授权码 处查看 |
| 返回值 | Void | |

* 1. **P10 请求**
     1. **根据密钥索引产生证书请求(CSR)**

|  |  |  |
| --- | --- | --- |
| 方法 | public String getCSRByIndex(int keyIndex, CsrRequest csrRequest) throws AFCryptoException | |
| 作用 | 根据密钥索引获取一个证书请求文件 | |
| 参数 | keyIndex | 密钥索引,该索引不能是已经使用的密钥 |
| csrRequest | CSR请求信息 每个字段都不能为空字符串或者null |
| 返回值 | CSR请求文件 Base64编码 | |

* + 1. **根据密钥索引导入证书**

|  |  |  |
| --- | --- | --- |
| 方法 | public void importCertByIndex(int keyIndex, String signCert, String encCert, String encPriKey) throws AFCryptoException | |
| 作用 | 根据密钥id 导入证书 证书需要用CSR文件去签发 | |
| 参数 | keyIndex | 生成CSR文件的密钥索引 |
| signCert | 签名证书 可选,如果没有可传 空字符串 |
| encCert | 加密证书 可选,如果没有可传 空字符串 |
| encPriKey | 加密私钥 可选,如果没有可传 空字符串 |
| 返回值 | Void | |

* + 1. **根据密钥索引获取证书**

|  |  |  |
| --- | --- | --- |
| 方法 | public Map<String, String> getCertByIndex(int keyIndex) throws AFCryptoException | |
| 作用 | 根据密钥索引获取导入的证书 | |
| 参数 | keyIndex | 导入证书的密钥索引 |
| 返回值 | 证书Map<String, String>  key:证书类型( encCert | signCert )  value:如果存在则为证书内容，否则为null | |

* + 1. **删除密钥**

|  |  |  |
| --- | --- | --- |
| 方法 | public void deleteKey(int keyIndex) throws AFCryptoException | |
| 作用 | 根据密钥id 删除密钥以及导入的证书 | |
| 参数 | keyIndex | 导入证书的密钥索引 |
| 返回值 | Void | |

1. **附录**