



密码机

JAVA 接口说明文档 V1.01

单位：杭州信雅达科技有限公司

地址：浙江省杭州市滨江区江南大道 3888 号信雅达科技大厦

邮编：310053

电话：0571-56686999

网站域名：<http://www.sydtech.com.cn>

目录

JAVA 接口说明文档 V1.01	1
一、一般说明	3
1.1 说明	3
1.2 类继承说明	3
1.2.1 继承关系	3
1.2.2 类方法说明	3
1.2.3 限定符说明	4
二、接口说明	4
2.1 连接管理	4
2.1.1 连接到加密机集群	4
2.1.2 disconnect(断开连接)	5
2.2 签名验签接口	6
2.2.1 PKCS#1 签名 (DN 获取私钥)	6
2.2.2 PKCS#1 验签 (DN 获取公钥)	6
2.2.3 PKCS#1 签名 (序列号获取私钥)	6
2.2.4 PKCS#1 验签 (序列号获取公钥)	7
2.2.5 PKCS#7 签名 Detach (DN 获取私钥)	7
2.2.6 PKCS#7 验签 Detach	7
2.2.7 PKCS#7 签名 attach (DN 获取私钥)	8
2.2.8 PKCS#7 验签 attach	8
2.3 证书管理	8
2.3.1 通过 DN 获取公钥	8
2.3.2 通过 DN 获取序列号	9
2.3.3 解析 X509 证书	9
2.3.4 上传指定证书到加密设备	9
2.3.5 删除指定证书	10
2.4 加密解密接口	10
2.4.1 生成数字信封(遵循 PKCS#7)	10
2.4.2 解密数字信封(遵循 PKCS#7)	10
2.5 哈希接口	11
2.5.1 SM3 哈希 (带公钥)	11
2.5.2 SM3 哈希 (不带公钥)	11
2.6 非对称加密解密接口	11
2.6.1 公钥加密	11
2.6.2 私钥解密(通过 ID 获取私钥)	12
附录一 常见错误码表	13

一、一般说明

1.1 说明

信雅达公司提供的国密软算法以 jar 包的形式提供，JVM 版本为 1.7，jar 包内部已包含针对不同的操作平台的支持。

本文仅包含的接口

sydapi.zip 压缩文件解压后，jar 目录下为 jar 包，sample 目录下为调用示例。

本文档的方法都是 sydapi 类的实例的方法。

1.2 类继承说明

1.2.1 继承关系

X509 继承自 Cert 继承自 CertInfo 继承自 Info.

1.2.2 类方法说明

Info 拥有方法：

<code>java.lang.String</code>	<code>getIssuer()</code> 获取证书颁发者 DN
<code>java.lang.String</code>	<code>getNotAfter()</code> 获取证书生效日期
<code>java.lang.String</code>	<code>getNotBefore()</code> 获取证书失效日期
<code>java.lang.String</code>	<code>getSerialNumber()</code> 获取证书序列号
<code>java.lang.String</code>	<code>getSubject()</code> 获取证书 DN

CertInfo 继承 **Info** 上述方法，并增加如下方法：

限定符和类型	方法和说明
<code>java.lang.String</code>	<code>getCert()</code> 获取 X509 证书内容（PEM 格式）

Cert 继承 **Info**、**CertInfo** 上述方法，并增加如下方法：

`byte[]` `getPubKey()` 获取证书中的公钥，DER 编码

X509 继承 **Info**、**CertInfo**、**Cert** 方法

1.2.3 限定符说明

若部分示例使用了限定符可参考如下：

<i>public static final int</i>	DATA_HASH 签名数据为 SM3-HASH 值	0
<i>public static final int</i>	DATA_ORIGIN 签名数据为原数据	1
<i>public static final int</i>	FOR_SIGN	0
<i>public static final byte</i>	KCV_S	83
<i>public static final java.lang.String</i>	RET_PRIVATE_KEY	"PrivateKey"
<i>public static final java.lang.String</i>	RET_PUBLIC_KEY	"PublicKey"

二、接口说明

2.1 连接管理

2.1.1 连接到加密机集群

短连接的建立

使用 `SydApiClusterBuilder` 类的静态方法 `SYD_Short_Connect_Ex`。

接口原型：

```
public static SydApi SYD_Short_Connect_Ex(String[] pclpList, int[] iPortList, int iConnectTimeOut, int allDealTimeOut)
```

功能：建立一个连接到集群的短连接

参数：

pclpList 加密机 ip 列表
iPortList 加密机 port 列表
iConnectTimeOut 连接超时时间
allDealTimeOut 交易超时时间

长连接的建立

使用 **SydApiPoolBuilder** 类辅助创建，**SydApiPoolBuilder** 通常只用创建一个实例，实例对象 **builder** 可以在线程间共享。

接口原型：

```
public SydApiPoolBuilder(  
    String[] pclpList,  
    int[] iPortList,  
    int iConnectTimeOut,  
    int iDealTimeOut  
)
```

功能：**SydApiPoolBuilder** 的构造函数

参数：

pclpList 加密机 ip 列表
iPortList 加密机 port 列表
iConnectTimeOut 连接超时时间
allDealTimeOut 交易超时时间

返回：**SydApiPoolBuilder** 的实例

获取 **SydApi** 对象

接口原型：

```
public SydApi build()
```

功能：从 **builder** 获取一个本线程内的 **API** 实例。

参数：无

返回：**SydApi** 对象的实例

2.1.2 disconnect(断开连接)

接口原型：

```
int disconnect()
```

功能：

断开连接

返回：

0 表示成功；非 0 表示失败；

2.2 签名验签接口

2.2.1 PKCS#1 签名（DN 获取私钥）

接口原型：

```
byte[] SYD_NakedSign(byte[] orgData,java.lang.String sCertDN);
```

功能：

使用指定的证书 DN，从加密设备获取对应的私钥证书，对指定的原始数据编制裸签名（遵循 PKCS#1）。

参数：

orgData - 待签名的原始数据。

sCertDN - 签名者证书 DN。

返回：

签名数据。

2.2.2 PKCS#1 验签（DN 获取公钥）

接口原型：

```
boolean SYD_NakedVerify(byte[] orgData,  
                        byte[] sign,  
                        java.lang.String sCertDN);
```

功能：

使用指定的证书 DN，对指定的原始数据编制裸验签（遵循 PKCS#1）。

参数：

orgData - 待签名的原始数据。

sign - 签名数据。

sCertDN - 签名者证书 DN。

返回：

验签是否通过

2.2.3 PKCS1 签名（序列号获取私钥）

接口原型：

```
java.lang.String netSign(java.lang.String pCertSerial,  
                        byte[] pOrgData)
```

功能：

用指定的证书私钥对指定的数据进行数字签名。

参数：

pCertSerial - 输入。签名 X509 证书的序列号。

pOrgData - 输入。待签名的原始数据。

返回：

签名数据(BASE64 编码)。

2.2.4 PKCS1 验签（序列号获取公钥）

接口原型：

```
boolean netVerify(java.lang.String pCertSerial,  
                  byte[] pOrgData,  
                  java.lang.String pSignData)
```

功能：

用指定证书对指定的原始数据进行数字签名验证。

参数：

pCertSerial - 输入。签名 X509 证书的序列号。

pOrgData - 输入。待签名的原始数据。

pSignData - 输入。签名数据(BASE64 编码)。

返回：

true 表示验证正确；false 表示失败；

2.2.5 PKCS#7 签名 Detach（DN 获取私钥）

接口原型：

```
java.lang.String sm3AndDetachedSign(byte[] orgData,  
                                     java.lang.String publicKey,  
                                     java.lang.String sCertDN);
```

功能：

2.2.7 章节接口仅支持 4K 以内数据的 P7 签名，本接口不限制长度，但是内部封装了带公钥的 SM3 和 P7 签名接口，效率略低。

参数：

orgData - 待签名的原始数据

publicKey - sm3 hash 所用的公钥

sCertDN - 签名者证书 DN

返回：

签名数据

2.2.6 PKCS#7 验签 Detach

接口原型：

```
boolean sm3AndDetachedVerify(byte[] orgData,  
                              java.lang.String publicKey,  
                              java.lang.String sign);
```

功能：

2.2.8 章节接口仅支持 4K 以内数据的 P7 验签，本接口不限制长度，但是内部封装了带公钥的 SM3 和 P7 验签接口，效率略低。

参数：

orgData - 待签名的原始数据

publicKey - sm3 hash 所用的公钥

sign - 签名数据

返回:

验签是否通过

2.2.7 PKCS#7 签名 attach (DN 获取私钥)

接口原型:

```
String attachedSign(byte[] orgData, String sCertDN);
```

功能:

使用指定的证书 DN，从加密设备获取对应的私钥证书，对指定的原始数据编制带公钥证书的数字签名（遵循 PKCS#7）。

参数:

orgData: 待签名的原始数据

sCertDN: 证书 DN。

返回:

String: 符合 PKCS7 标准的签名结果。

2.2.8 PKCS#7 验签 attach

接口原型:

```
RetWrap attachedVerifyAndGetAll (byte[] sign);
```

功能:

对指定的原始数据核验其带公钥证书签名，并返回签名的原始数据。

参数:

sign: 已签名数据。

返回:

RetWrap 继承自 HashMap 类型,键值包括 x509, oData。

x509: X509 格式 保存了 x509 证书

oData: byte[] 格式 保存了签名的原数据。

2.3 证书管理

2.3.1 通过 DN 获取公钥

接口原型:

```
java.lang.String SYD_SM2_Get_Dn_PublicKey_HA(byte[] Dn,  
int nKeyType);
```

功能:

通过证书 DN 获得机构的公钥(DER 编码)，联调前需要导入公钥证书。

参数:

Dn - 证书的 DN，每个证书有唯一的 DN。

nKeyType - 密钥类型，0: 签名公钥，1:加密公钥，如无加解密业务，传入 0 即可。

返回:

输出的公钥串，DER 编码，HEX 格式，此公钥为他方机构的公钥。

2.3.2 通过 DN 获取序列号

接口原型:

```
java.lang.String getX509SN(int nCertType,  
                             byte[] certDN)
```

功能:

通过证书的 DN，获取证书的序号。

参数:

nCertType - 输入 证书类型，0x00:签名证书,0x01:加密证书。

certDN - 输入证书的 DN。

返回:

CertSerial 证书序列号。

2.3.3 解析 X509 证书

接口原型:

```
X509 GetX509CertInfo(java.lang.String pCert)
```

功能:

输入 X509 格式的证书，获取证书信息。

参数:

pCert - base64 编码的证书，PEM 格式或 DER 编码格式。

返回:

证书信息类，包含：颁发者 DN、证书序列号、证书主题、证书有效期的起始时间、证书有效期的终止时间、DER 编码的公钥。

类的方法参考 1.2.2:

2.3.4 上传指定证书到加密设备

接口原型:

```
X509 SYD_Get_X509_CertInfo_HA(byte[] orgData,String sign);
```

功能:

导入证书到密码设备。集群模式下，只要有一个节点同步失败则失败。

参数:

orgData: 待验签的原始数据（包含加密证书）

sign: P7 格式签名数据。

返回:

X509: 证书信息（包含公钥字段）。

2.3.5 删除指定证书

接口原型:

```
public boolean deleteCert(String dn)
```

功能:

从集群中删除证书,只要有一个节点同步失败则失败。

参数:

dn: 证书 DN。

返回:

是否删除成功。

2.4 加密解密接口

2.4.1 生成数字信封(遵循 PKCS#7)

接口原型:

```
RetWrap generateDEByDn(String dn, byte[] oData);
```

功能:

产生对称密钥,通过公钥加密生成数字信封。

参数:

dn: 证书 DN,例如: C=CN,O=CFCA,支持顺序反转。

oData: 明文数据。外部处理编码。

返回:

RetWrap 继承自 HashMap 类型,键值包括 CipherKey, SessionKey, KCV。

byte[] CipherKey: 数字信封。Base64 编码后的字节。

String SessionKey: 本地 LMK 加密的会话密钥, HEX 格式,本地 LMK 加密的会话密钥长度, HEX 格式(不包括第 1 个字符),一般为 33 个 Bytes。默认输入值为缓冲区的大小。

String KCV: 会话密钥的检验值, HEX 格式,长度为 32H。

2.4.2 解密数字信封(遵循 PKCS#7)

接口原型:

```
RetWrap decryptDEByDN (String dn, byte[] pCipherKey);
```

功能:

接收用私钥解密数字信封,还原原文。

参数:

dn: 证书 DN,例如: C=CN,O=CFCA,支持顺序反转。

pCipherKey: 数字信封。Base64 编码后的字节。

返回:

RetWrap 继承自 HashMap 类型,键值包括 oData。

byte[] oData 原数据。

2.5 哈希接口

2.5.1 SM3 哈希（带公钥）

接口原型：

```
byte[] SM3Hash(java.lang.String publicKey,  
                byte[] data)
```

功能：

输入对指定的原始数据和公钥，通过 SM3 算法计算数据的散列值。

参数：

publicKey - 公钥(默认 ID 为“1234567812345678”).
data - 原始数据。

返回：

HASH 值，32 个 Bytes。

2.5.2 SM3 哈希（不带公钥）

接口原型：

```
byte[] SM3Hash(byte[] data);
```

功能：

输入对指定的原始数据 (无公钥)，通过 SM3 算法计算数据的散列值。

参数：

data - 原始数据。

返回：

HASH 值，32 个 Bytes。

2.6 非对称加密解密接口

2.6.1 公钥加密

接口原型：

```
String SM2Encrypt(byte[] publicKey, byte[] pOrgData);
```

指令：77

功能：

用指定的 SM2 公钥对指定的密文数据进行加密，默认算法标识为 SM2。

参数:

publicKey: 公钥串, DER 编码, HEX 格式。
pOrgData: 加密的数据。

返回:

String:加密的数据 base64 编码。

2.6.2 私钥解密(通过 ID 获取私钥)

接口原型:

```
byte[] SM2DecryptC(String userId, String pEnData);
```

指令: 78

功能:

通过使用者 ID 所对应的私钥对指定的密文数据进行解密, 默认算法标识为 SM2。

参数:

userId: 使用者的 ID, 最大长度为 16 位数字, 右对齐。
pEnData: 加密的数据, base64 编码。

返回:

byte[]:明文数据。

附录一 常见错误码表

错误码（2N）	含义
00	无错误。
15	输入数据错。
80	数据长度错误。
错误码（2H）	含义说明
0x8001	签名错误，一般是数据或者密钥不合法。
0x8002	验签错误，一般是数据、密钥、签名值不对应或不完整。
0x8004	证书过期。
0x8005	证书不可信，未相应导入根证书。
0x8006	证书格式错误，一般可能是数据不完整。
0x8008	非用户证书，可能把根证书勿做用户证书。
0x8009	生成证书请求错误
0x8010	证书已被吊销
0x7007	实体白名单未添加
0x9002	待导入证书数据已存在
0x9005	硬件错误