



JAVA 接口说明文档

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

目录

一、 接口说明	3
1.1 SM2 私钥签名	3
1.2 SM2 签名验签	4
1.3 生成数字信封(遵循 PKCS#7)	5
1.4 解密数字信封(遵循 PKCS#7)	5
1.5 带公钥证书与原始数据的数字签名（遵循 PKCS#7）	6
1.6 带公钥证书与原始数据的数字签名验签（遵循 PKCS#7）	6
1.7 带公钥证书的数字签名（SM2_DETACH 签名）	8
1.8 带公钥证书的数字签名验签（SM2_DETACH 验签）	8
1.9 计算 SM3 散列值	9
1.10 获取公钥	9
二、 连接管理	9
2.1 BUILDER 方式	9
2.2 CONNECT(连接加密机)	10
2.3 DISCONNECT(断开连接)	10

一、接口说明

1.1 SM2 私钥签名

1.1.1 SM2 签名（缺省 ID）

接口原型：

```
byte[] SM2SignC(int flag, String privateKey, byte[] publicKey, byte[] orgData);
```

指令：74

功能：

sm2 私钥签名

用户 ID 默认为 1234567812345678

签名结果 R+S。

参数：

flag：私钥标志位，当为 1 时第二个参数传入存储私钥的索引，当为 0 时第二个参数传入私钥密文

privateKey：当私钥标志位为 1 时传入存储私钥的索引，当为 0 时传入私钥密文

publicKey：公钥的 DER 编码

orgData：要签名的信息。

返回：

byte[] 签名结果（签名结果 R+S）。

1.1.2 SM2 签名（自定义 ID）

接口原型：

```
byte[] SM2SignC(int flag, String uid, String privateKey, byte[] publicKey,  
byte[] orgData);
```

指令：74

功能：

sm2 私钥签名

签名结果 R+S。

参数：

flag：私钥标志位，当为 1 时第二个参数传入存储私钥的索引，当为 0 时第二个参数传入私钥密文

uid：用户 ID

privateKey: 当私钥标志位为 1 时传入存储私钥的索引, 当为 0 时传入私钥
密文

publicKey: 公钥的 DER 编码

orgData: 要签名的信息。

返回:

byte[]: 签名数据 (R+S)。

1.2 SM2 签名验签

1.2.1 SM2 验签 (缺省 ID)

接口原型:

```
boolean SYD_SM2_Verify(int nOrgDataType, byte[] publicKey, byte[] orgData, byte[]  
sign);
```

指令: 75

功能:

采用 SM2 算法, 用指定的公钥对指定的原始数据进行数字签名验证,
默认用户 id 为 1234567812345678。

参数:

nOrgDataType: 数据类型, SydApi.DATA_ORIGIN 原始数据
SydApi.DATA_HASH: hash 后的类型

publicKey: 公钥串, DER 编码, HEX 格式

orgData: 待签名的原始数据

sign: 签名数据 (R+S)。

返回:

boolean: true 验签通过 false 验签失败。

1.2.2 SM2 验签 (自定义 ID)

接口原型:

```
boolean SM2Verify(String pUserID, int nOrgDataType, byte[] publicKey, byte[]  
orgData, byte[] sign);
```

指令: 75

功能:

采用 SM2 算法, 用指定的公钥对指定的原始数据进行数字签名验证,

参数:

pUserID: 用户 ID

nOrgDataType: 签名数据类型, 1: 原始数据 0: hash 后的类型

publicKey: 公钥串, DER 编码, HEX 格式

orgData: 待签名的原始数据, “签名数据类型”为 0 时: HASH 值。“签名数据类型”为 1 时: 要签名的信息

sign: 签名数据 (R+S)。

返回:

boolean: true 验签通过 false 验签失败。

1.3 生成数字信封(遵循 PKCS#7)

接口原型:

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

指令: 6A

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。

1.4 解密数字信封(遵循 PKCS#7)

接口原型:

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

指令: 6B

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。

1.5 带公钥证书与原始数据的数字签名（遵循 PKCS#7）

接口原型：

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

指令：0x8018

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。

1.6 带公钥证书与原始数据的数字签名验签（遵循 PKCS#7）

1.6.1 验签

接口原型：

boolean attachedVerify(String sign);

指令：0x8019

功能：

对指定的原始数据核验其带公钥证书签名。

参数：

sign: 已签名数据。

返回：

boolean: true 验签通过 false 验签失败。

1.6.2 验签，返回公钥证书信息

接口原型：

X509 attachedVerifyAndGetX509 (String sign);

指令：0x8019

功能：

对指定的原始数据核验其带公钥证书签名。如签名有效（含证书无效）且需返回签名

者公钥证书信息时，同时返回签名者公钥证书信息。

参数：

sign: 已签名数据。

返回：

X509: CERT_INFO 结构的证书数据。

1.6.3 验签，返回原始数据

接口原型：

byte[] attachedVerifyRD (String sign);

指令：0x8019

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：

byte[]: 成功返回签名的原数据，失败则报错。

1.6.4 验签，返回原始数据和证书信息

接口原型：

RetWrap attachedVerifyAndGetAll (String sign);

指令：0x8019

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。

1.7 带公钥证书的数字签名（SM2_Detach 签名）

接口原型：

```
String detachedSign(int dataType, byte[] orgData, String sCertDN);
```

指令：0x8005

功能：

通过指定的私钥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：

dataType：数据模式，0:输入的数据为 HASH 后的数据，1:输入的数据为原始数据库

orgData：待签名的原始数据

sCertDN：签名者证书 DN。

返回：

String：签名数据。

1.8 带公钥证书的数字签名验签（SM2_Detach 验签）

接口原型：

```
boolean detachedVerify(int dataType, byte[] orgData, String sign);
```

指令：0x8006

功能：

对指定的原始数据核验其带公钥证书签名。如签名有效（含证书无效）且需返回签名者公钥证书信息时，同时返回签名者公钥证书信息，

默认标记为 0000 不返回。

参数：

dataType：数据模式，0:输入的数据为 HASH 后的数据，1:输入的数据为原始数据库

orgData：待签名的原始数据

sign：签名数据。

返回：

boolean：验签是否通过。

1.9 计算 SM3 散列值

接口原型:

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

指令: 76

功能:

输入对指定的原始数据, 通过 SM3 算法计算数据的散列值。

参数:

msg: 原始数据, 二进制的消息块。

返回:

byte[]: HASH 值, 32 个 Bytes。

1.10 获取公钥

接口原型:

```
String SM2GetPublicKeyC(String userId, int nKeyType);
```

指令: QQ

功能:

通过索引获取公钥。

参数:

userId: 使用者的 ID, 最大长度为 16 位数字, 右对齐。

nKeyType: 密钥类型, 0x00: 签名公钥, 0x01: 加密公钥。

返回:

String: 公钥。

二、连接管理

2.1 Builder 方式

使用 SydApiBuilder 类辅助创建, 可以链式调用。常用方法:

- SydApiBuilder setIp(String ip) 设置加密机链接 ip
- SydApiBuilder setPort(int port) 设置加密机链接 port
- SydApiBuilder setTimeout(int timeout) 设置加密机链接超时参数。

- SydApi build() 构造一个新的 SydApi 实例。

示例：

```
SydApi api = new SydApiBuilder()  
    .setIp("192.168.1.123")  
    .setPort(8889)  
    .setTimeout( 3000 )  
    .build();
```

2.2 connect(连接加密机)

接口原型：

```
SydApi connect(String ip, int port, String pwd, int connectTimeout, int readTimeout , int  
retryWhenNetError);
```

功能：

建立到指定地址和端口号的加密设备的 TCP/IP 连接，返回 TCP/IP 连接句柄。如果采用短连接，交易结束，要调用 disconnect() 进行释放连接。

参数：

ip: 加密设备的 IP 地址。
port: 加密设备的端口号。
pwd: 保留。
connectTimeout: 链接超时时间。
readTimeout: 通信超时时间。
retryWhenNetError: 网络失败时重试次数

返回：

设备操作对象

2.3 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；

