网证通 CA 证书

功能列表

修改记录

时间	修改人	修改说明
2019-01-02	周锐，齐奇	初版

目 录

修改记录 1

1. 应用典型流程 7

1.1. 证书登录（电子认证用户身份认证）实现流程 7

1.1.1. 功能描述 7

1.1.2. 流程图 7

1.2. 业务系统管理员绑定用户证书流程 9

1.2.1. 功能描述 9

1.2.2. 流程图 9

1.3. 证书更新后的自动绑定 9

1.3.1. 新旧姆印法 9

1.3.2. 实体唯一标识符法 10

1.4. 表单签名 11

1.4.1. 功能描述 11

1.4.2. 流程图 11

1.5. 报价加密 12

1.5.1. 功能描述 12

1.5.2. 流程图 13

1.6. 报价解密 13

1.6.1. 功能描述 13

1.6.2. 流程图 14

1.7. 电子合同 PDF 签署实现 14

1.7.1. 前台电子合同签章 14
1.7.2. 后台批量合同 15
2. 实现接口 16
2.1. 工具类接口 16
2.1.1. convertByte 16
2.1.2. convertString 16
2.1.3. convertHex 16
2.1.4. base64Encode 16
2.1.5. base64Decode 17
2.1.6. getRandom 17
2.1.7. hashDataHex 17
2.1.8. hashDataBase64 18
2.1.9. hashDataMD5 18
2.1.10. readFile 19
2.1.11. writeFile 19
2.2. 证书操作类接口 19
2.2.1. getX509Certificates 19
2.2.2. getX509Certificate【重载】 20
2.2.3. getServerCertificate 22
2.2.4. getX509CertificateInfo【重载】 22
2.2.5. getX509CertificateThumbprint 23
2.3. 签名类接口 23

2.3.1. signNETCA 23
2.3.2. signedDataByPwd【重载】 24
2.3.3. signedDataByCertificate【重载】 24
2.3.4. signedDataWithTSA【重载】 25
2.3.5. verifySignedDataWithTSA 26
2.3.6. verifySignedData【重载】 26
2.3.7. getSourceFromSignedData 27
2.3.8. signByCertificate【重载】 27
2.3.9. verify【重载】 28
2.4. 加密类接口 29
2.4.1. envelopedData【重载】 29
2.4.2. developedData 29
2.4.3. developedDataBySoft 30
2.4.4. asymmetriEncrypt 30
2.4.5. asymmetriDecrypt 31
2.5. 时间戳接口 31
2.5.1. getTimeStamp 31
2.5.2. verifyTimeStampToken 31
2.6. 硬件类接口 32
2.6.1. getKeySerial 32
2.6.2. getDeviceSet 32
2.6.3. getCertWithDeviceSN 32

2.7. 证书验证类接口 33

2.7.1. 网关验证类接口 33

2.7.2. 安全中间件证书验证类接口 36

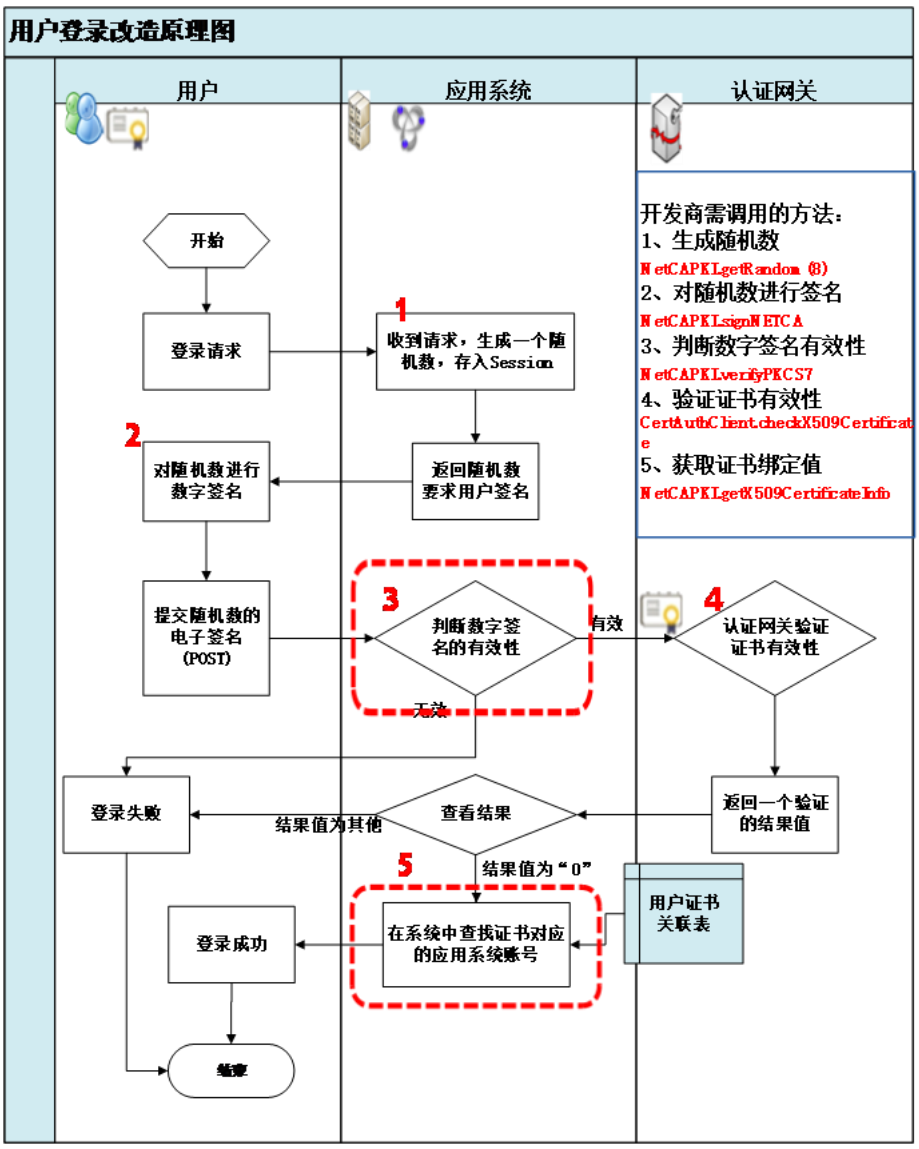
1.应用典型流程

1.1. 证书登录（电子认证用户身份认证）实现流程

1.1.1.功能描述

使用证书登录，替换原有系统账号名、密码方式登录，能提供系统的安全性，防范网络攻击。证书登录采用双因子认证，登录人需持有证书介质，同时需要输入证书使用者密码（PIN 码），是系统通过安全等级评测的基本条件之一。

1.1.2.流程图



证书登录及底层数据交互

1. 客户端发出登录请求（开发商实现）。

2. 服务端响应请求，生成随机数，并存入 session（即证书登录接口及底层数据交互图中“1”标示处）。

3. 把第二步生成的随机数，返回给用户，要求用户签名。

4. 调用签名的方法对随机数进行签名（即证书登录接口及底层数据交互图中“2”标示处），并 post 到服务端。

5. 判断签名的有效性（即证书登录接口及底层数据交互图中“3”标示处）。

6. 调用认证网关接口进行网关认证（即证书登录接口及底层数据交互图中“4”标示处）。

7. 服务端根据认证网关回传的响应码和证书状态码，判断证书验签结果，验签不通过则直接返回状态码对应的信息，验签通过则获取证书绑定值，用于对比应用系统中的用户证书关联表，获取对应用户名信息，允许登录。

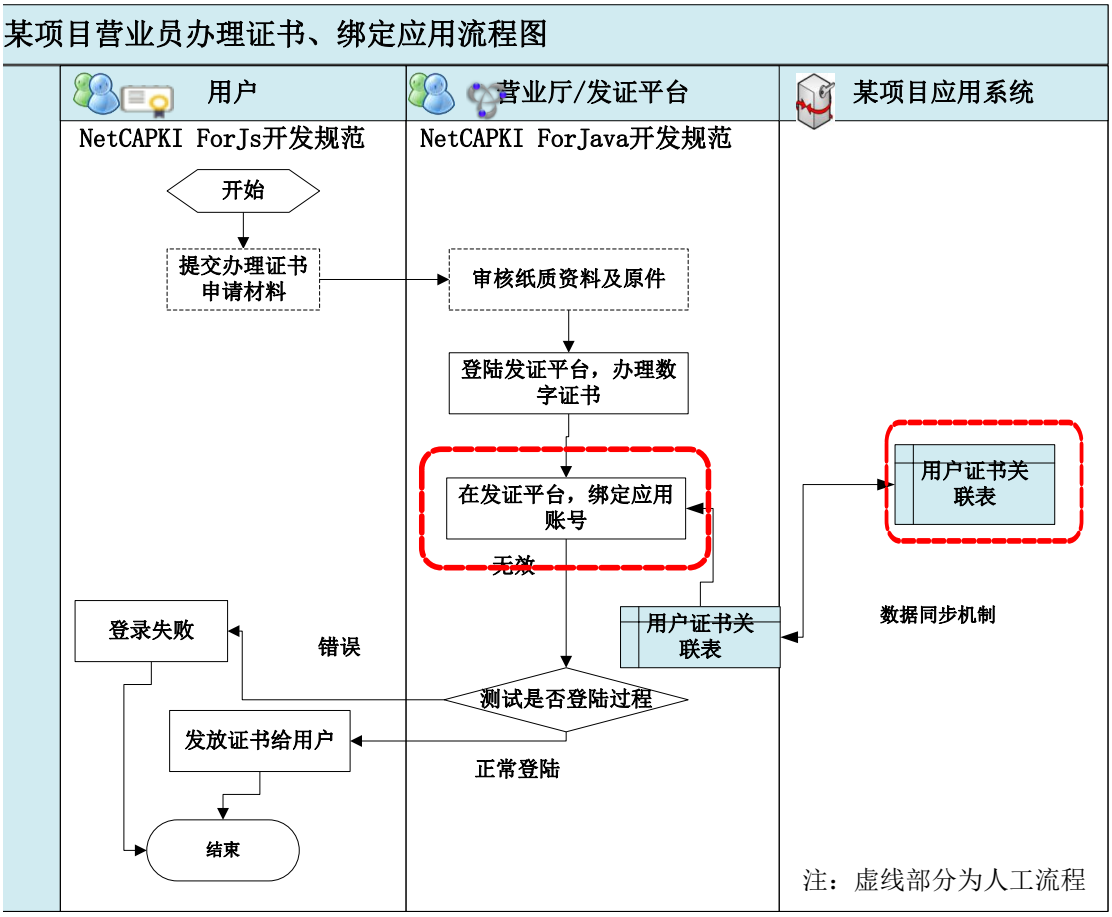
（即证书登录接口及底层数据交互图中“5”标示处）。

1.2. 业务系统管理员绑定用户证书流程

1.2.1.功能描述

绑定用户账号与证书，将其关联起来。

1.2.2.流程图



绑定用户和证书 UsrCertID（用户证书绑定值）。

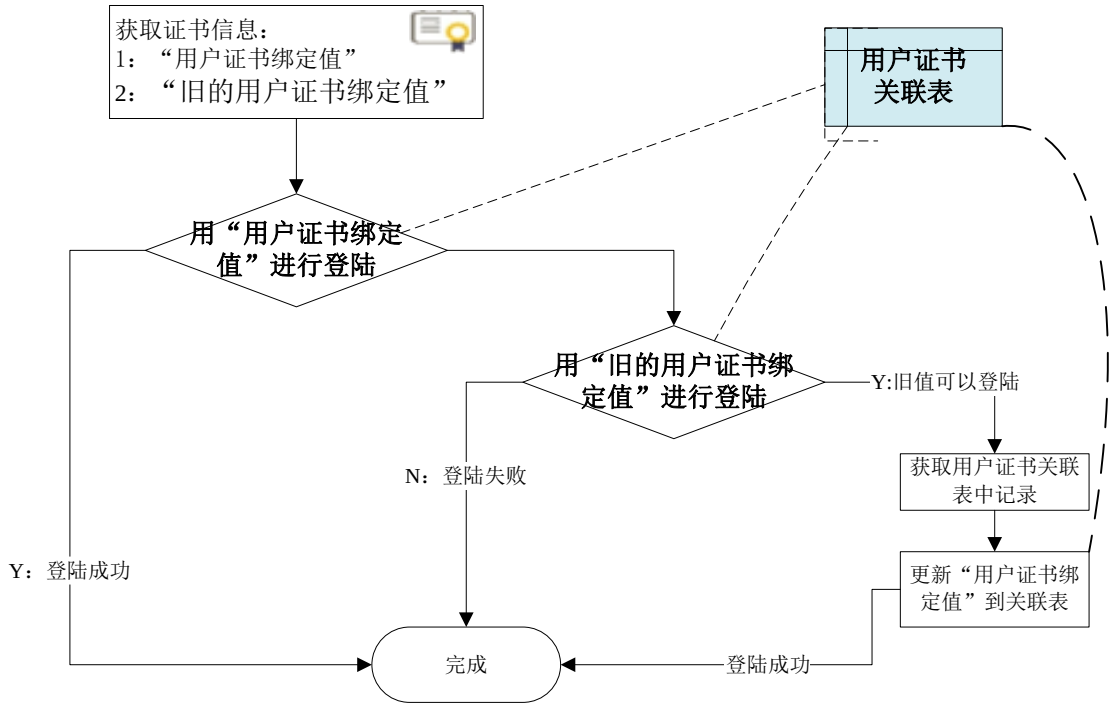
1.3. 证书更新后的自动绑定

1.3.1.新旧姆印法

1.3.1.1. 功能描述

由于证书更新后,证书中信息变更,会造成应用系统需重新进行证书绑定,加大了应用系统的维护工作。

1.3.1.2. 流程图



1. 用证书属性“ OldUsrCertID:旧用户证书绑定值”进行登陆，如成功，进入下一步操作（与原有蓝色部分相同）；
2. 步骤 1 如失败，获取证书属性值“旧的用户证书绑定值”，用“旧的用户证书绑定值”进行登陆；
3. 步骤 2 如成攻，则修改更新“用户证书关联表”中证书信息为步骤 1 中的“用户证书绑定值”，用户给予登陆；
4. 步骤 3 如失败，代表用户证书未在系统中做过绑定，不予登陆，显示“用户未绑定证书”的错误信息；

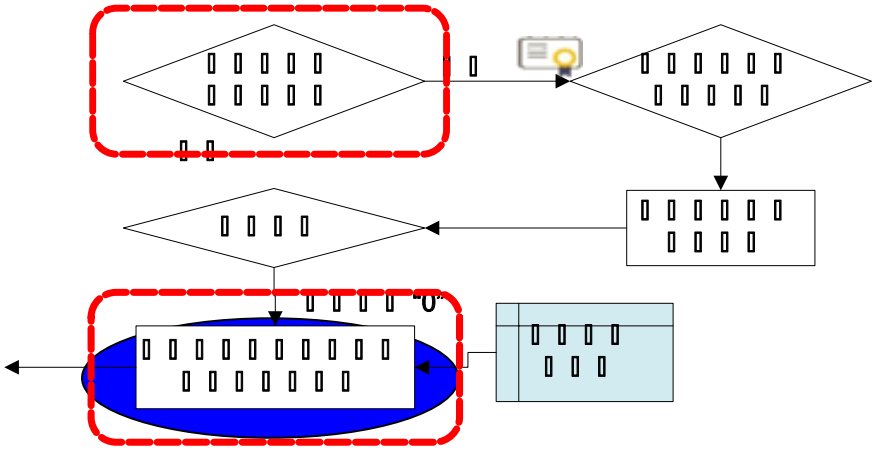
1.3.2. 实体唯一标识符法

1.3.2.1. 功能描述

由于证书更新后,证书中信息变更,会造成应用系统需重新进行证书绑定,加大了应用系统的维护

工作。

1.3.2.2. 流程图



1. 改造需用到证书属性值“证书绑定值”（UsrCertID:用户证书绑定值）,该值在证书更新后保持不变；

2. 各家 CA,应能签发出具有该属性的证书：

   NETCA 需签发“证件号码信息扩展域值” （证书实体唯一标识）扩展域的证书；

   GDCA 需签发“具信任号 TrustID” （GDCA 的信任号 TrustID）扩展域的证书；

3. 注意事项：

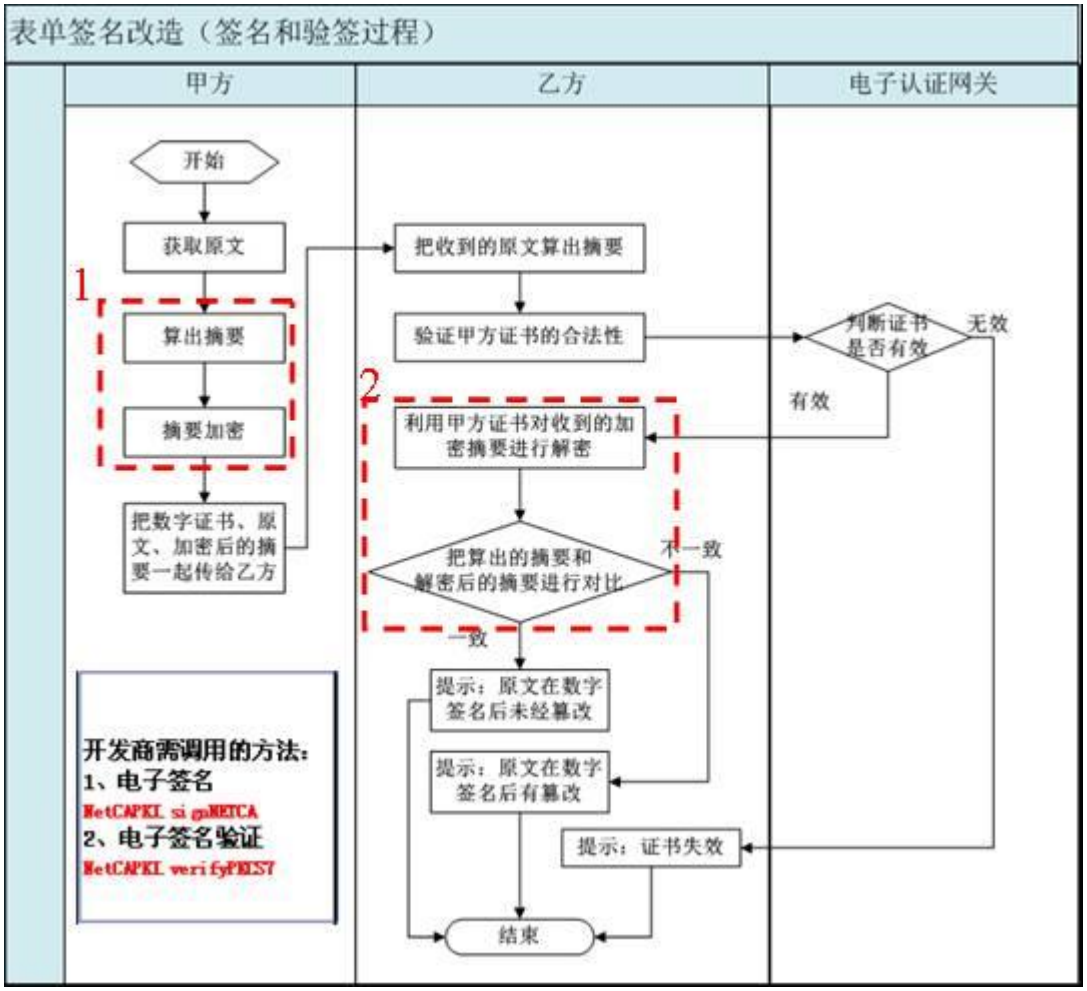
   NETCA/GDCA 更新后证书，如“证书绑定值”与原有“证书绑定值”不一至(一般为 CA 制证原因或个人原因)，只能进行人工绑定。

1.4. 表单签名

1.4.1. 功能描述

对招投标报价进行签名。

1.4.2. 流程图



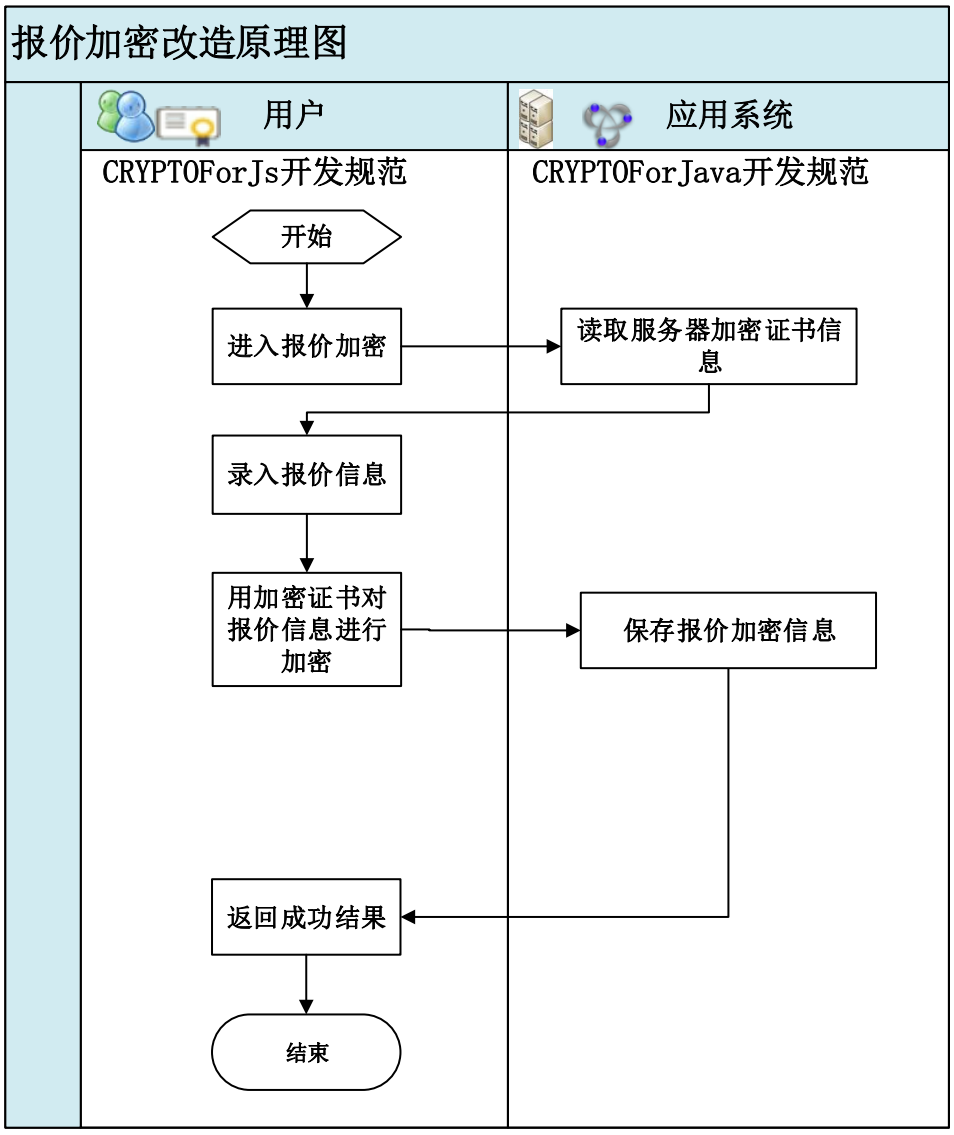
1. 客户端获取原文（注意浮点位数，空格等因素）。
2. 调用签名的方法对原文进行签名（即电子签名接口及底层数据交互图中“1”标注处），并将原文、电子签名数据一起 post 到服务端。
3. 电子签名验证（即电子签名接口及底层数据交互图中“2”标注处）。
4. 根据第三步获取到的证书绑定值（UsrCertID，用户证书绑定值），对比登录时的用户证书绑定值是否登录人证书，若是，则直接验签通过，若不是，则直接返回验签失败。
5. 将签名值保存到数据库。

1.5. 报价加密

1.5.1. 功能描述

招投标报价加密。

1.5.2.流程图

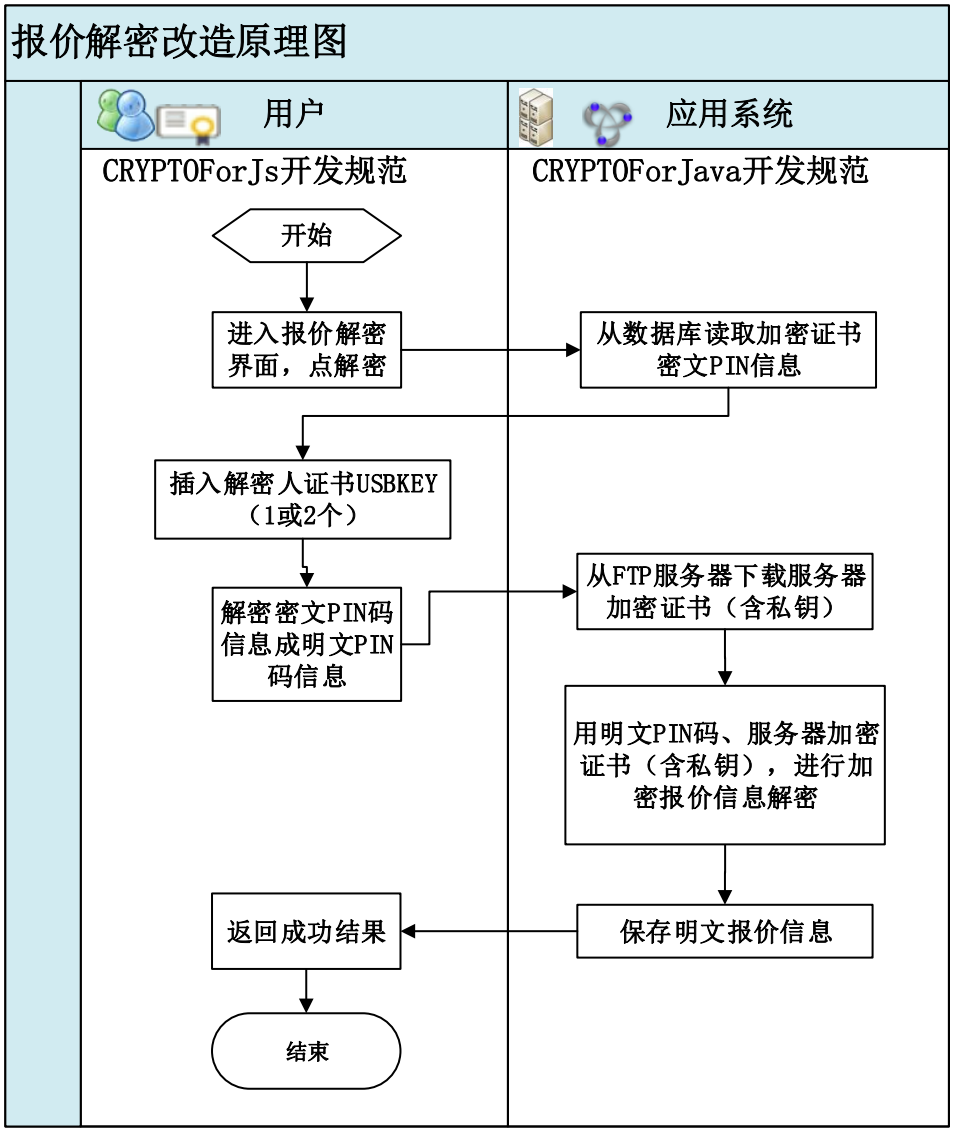


1.6. 报价解密

1.6.1.功能描述

招投标报价解密。

1.6.2.流程图



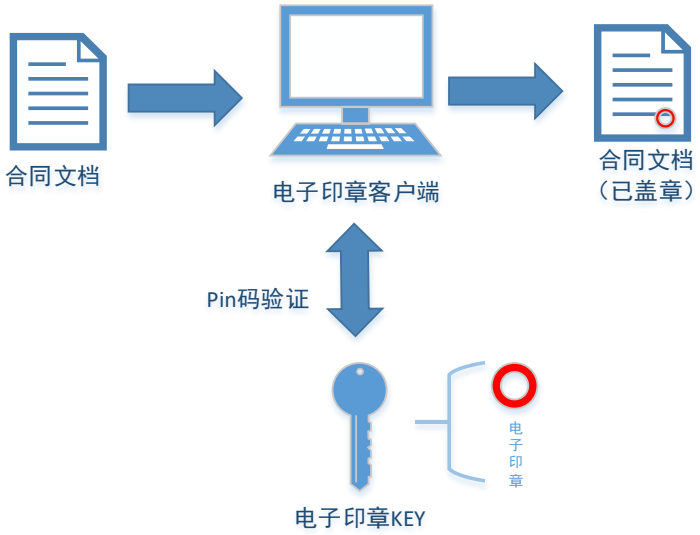
1.7. 电子合同 PDF 签署实现

1.7.1.前台电子合同签章

1.7.1.1. 功能描述

前台合同电子印章的使用和传统印章的使用方式基本相同，需要有一台专用的电子印章客户端系统，该系统安装在特定的电脑终端上。

1.7.1.2. 流程图



1. 将存有电子印章的实体（如 USBKey）插入电脑终端的 USB 接口。
2. 启动电子印章客户端系统。
3. 读入需要加盖电子印章的电子文书。
4. 在电子文书中需要盖电子印章的地方点击‘盖章’。
5. 输入正确的电子印章使用 pin 码，则该文书就被盖上电子印章了。

1.7.2.后台批量合同

1. 登录电子交易系统。
2. 系统鉴别身份。
3. 获取需要加盖电子印章的电子文书。
4. 调用批量签章接口，获取数字证书与电子印章。
5. 批量盖章完成。

2.实现接口

2.1. 工具类接口

2.1.1. convertByte

功能：字符串转字节数组

原型	byte[] convertByte(String data)
参数	data：字符串数据
返回值	转换的字节数组
说明	字符串转 byte 数组。 **注意：采用的编码方式在项目定制常量“NETCAPKI_CP”中定义。**

2.1.2. convertString

功能：字节数组转字符串

原型	String convertString (byte[] data)
参数	data: 字节数组
返回值	转换的字符串
说明	byte 数组转字符串。 **注意：采用的编码方式在项目定制常量“NETCAPKI_CP”中定义。**

2.1.3. convertHex

功能：将字节数组转 Hex 编码字符串。大写字母

原型	String convertHex (byte[] data)
参数	data: 字节数组
返回值	大写的 Hex 编码字符串
说明	将字节数组转 Hex 编码字符串。大写字母

2.1.4. base64Encode

功能：将字节数组进行 Base64 编码，得到 Base64 字符串。

原型	String base64Encode(byte[] data)
参数	data: 字节数组
返回值	Base64 编码的字符串
说明	将字节数组进行 Base64 编码，得到 Base64 字符串。

2.1.5. base64Decode

功能：将 Base64 字符串解码为字节数组。

原型	byte[] base64Decode(String data)
参数	data: Base64 字符串
返回值	字节数组
说明	将 Base64 字符串解码为字节数组。

2.1.6. getRandom

功能：获取随机数。

原型	String getRandom(int length)
参数	length: 随机数的字节数长度
返回值	经过 Hex 编码的随机数，即：十六进制大写的字符串
说明	

2.1.7. hashDataHex

功能：获取信息摘要码（Hex 编码方式）。

重载一:

原型	String hashDataHex(String data)
参数	Data：信息数据（字符串）
返回值	信息摘要码（Hex 编码方式）
说明	注：算法参见项目定制常量“NETCAPKI_ALGORITHM_HASH”中定义

重载二:

原型	String hashDataHex(byte[] data)
参数	Data：信息数据（字节数组）
返回值	信息摘要码（Hex 编码方式）
说明	注：算法参见项目定制常量“NETCAPKI_ALGORITHM_HASH”中定义

2.1.8. hashDataBase64

功能：获取信息摘要码（Base64 编码方式）。

重载一:

原型	String hashDataBase64 (String data)
参数	Data：信息数据（字符串）
返回值	信息摘要码（Base64 编码方式）
说明	注：算法参见项目定制常量“NETCAPKI_ALGORITHM_HASH”中定义

重载二:

原型	String hashDataBase64 (byte[] data)
参数	Data：信息数据（字节数组）
返回值	信息摘要码（Base64 编码方式）
说明	注：算法参见项目定制常量“NETCAPKI_ALGORITHM_HASH”中定义

2.1.9. hashDataMD5

功能：进行 hash 计算，获得摘要码（早期编码方式:MD5 哈希算法;Hex 制编码）。

重载一:

原型	String hashDataMD5 (String data)
参数	Data：信息数据（字符串）
返回值	信息摘要码（Hex 编码方式）
说明	注：算法为 ***MD5*** 计算的摘要值(BASE64 编码)

重载二:

原型	String hashDataMD5 (byte[] data)
参数	Data：信息数据（字节数组）
返回值	信息摘要码（Hex 编码方式）
说明	注：算法为 ***MD5*** 计算的摘要值(BASE64 编码)

2.1.10. readFile

功能：二进制读文件。

原型	byte[] readFile(String filePath)
参数	filePath: 文件的绝对路径
返回值	文件的字节数组
说明	注：文件大小不能超过 50M

2.1.11. writeFile

功能：二进制写文件。

原型	boolean writeFile(String filePath, byte[] content)
参数	filePath：读取文件的路径 content：字节数组，需要写入文件的内容
返回值	二进制写文件的操作，返回是否写入成功，如果成功，则返回 true，反之为 false
说明	

2.2. 证书操作类接口

2.2.1. getX509Certificates

功能：获得 X509 证书集合，返回一个证书对象集合

原型	List<Certificate> getX509Certificates(int purpose)
参数	purpose：证书用途（加密、签名）
返回值	一个证书对象集合
说明	只适用于 Window 系统。 证书用途（purpose）： 所有证书， 0 加密证书， 1，对应常量为 Certificate.PURPOSE_ENCRYPT

	签名证书， 2，对应常量为 Certificate.PURPOSE_SIGN 通过证书库、或者直接通过弹出框的形式选择证书 使用频率：较少用到； 使用场景： 1）证书登陆时，网页列出插入证书下拉框，需选择对应证书进行登陆； 2）证书绑定时，绑定相应证书； **注意：最后不使用 Certificate 对象时，要调用 free 方法释放 Certificate 对象资源**

2.2.2. getX509Certificate【重载】

功能：获取证书集（通过证书库、或者直接通过弹出框）

重载 1：

原型	Certificate getX509Certificate(int purpose)
参数	purpose: 证书用途（加密、签名）
返回值	一个证书对象
说明	只适用于 Window 系统。 证书用途（purpose）： 所有证书， 0 加密证书， 1，对应常量为 Certificate.PURPOSE_ENCRYPT 签名证书， 2，对应常量为 Certificate.PURPOSE_SIGN 使用频率：较常用； 使用场景： 选择证书通常采用此函数。1）证书绑定时，2）证书登录时； 根据全局变量定制项 2、3，可通过该函数支持多 CA 支持； **注意：最后不使用 Certificate 对象时，要调用 free 方法释放 Certificate 对象资源**

重载 2：

原型	Certificate getX509Certificate(String base64Cert)
参数	base64Cert：证书的 Base64 编码值
返回值	一个证书对象
说明	base64Cert 编码值中不允许有回车符 **注意：最后不使用 Certificate 对象时，要调用 free 方法释放 Certificate 对象资源**

重载 3:

原型	Certificate getX509Certificate(byte[] data)
参数	base64Cert：证书的 Base64 编码值
返回值	一个证书对象
说明	base64Cert 编码值中不允许有回车符 从 HTTPS 通信中获取证书对象（从 HTTPS 通信中） 使用频率：SSL 登录时使用，频率较少 使用场景 1）SSL 证书登陆时，用 https 通道中获取客户端证书 **注意：最后不使用 Certificate 对象时，要调用 free 方法释放 Certificate 对象资源**

重载 4:

原型	Certificate getX509Certificate(**int** purpose, **int** infoType, String certInfoValue)
参数	purpose ：证书用途，参见 Certificate 中关于 PURPOSE 常量的定义； infoType：证书信息特定域类别，证书属性值编号 certInfoValue：证书信息特定域值
返回值	一个证书对象
说明	获取证书对象（根据证书特定域的值） 使用频率：较少用，有特定需求时，才用到； 应用场景: 1）证书登陆后，会将部分证书属性信息放到 session 中； 2）签名时，会用登陆证书进行签名，此时，需根据 session 中的属性值获取证书 **注意：最后不使用 Certificate 对象时，要调用 free 方法释放 Certificate 对象资源**

重载 5:

原型	Certificate getX509Certificate(String type,String expr)
参数	type：证书类型，也可以是获取条件的 JSON。可以参考《NETCA PKI API 参考手册》 expr：表达式条件，见 match 方法说明
返回值	一个证书对象
说明	只适用于 Window 系统。 当满足获取条件的证书为多张时,自动弹出选择框,由用户选择; 举个例子: 从设备上获取，KEY 上的签名证书 type: {"UIFlag":"default","InValidity":true,"Type":"signature","Method":"device","Value":"any"} expr: InValidity='True'&&(IssuerCN~'NETCA'||IssuerCN~'GDCA'||IssuerCN~'SZCA')&&CertType='Signature'&&CheckPrivKey='True'

	从设备上获取，KEY 上的加密证书 type: {"UIFlag":"default","InValidity":true,"Type":" encrypt","Method":"device","Value":"any"} expr: InValidity='True'&&(IssuerCN~'NETCA'||IssuerCN~'GDCA'||IssuerCN~'SZCA')&&CertType='Encrypt'&&CheckPrivKey='True' **注意：最后不使用 Certificate 对象时，要调用 free 方法释放 Certificate 对象资源**

2.2.3. getServerCertificate

功能：获得服务器证书，返回一个证书对象

原型	String getServerCertificate(String CN)
参数	CN：证书主题中 CN 项值（通用名）
返回值	一个证书对象的 Base64 编码
说明	获取一个服务器证书（从证书库，根据证书主题 CN 项） 使用频率：较少用 使用场景： 服务器端使用，一般用来做服务器端证书加密解密用; 服务器证书，安装到服务器上，在 MMC 本地计算机上能看到; 获取服务器证书，发送到前台，供客户用来加密;

2.2.4. getX509CertificateInfo【重载】

功能：获得 X509 证书相关的信息内容，返回一个字符串

重载一:

原型	String getX509CertificateInfo(Certificate certificate,int iInfoType)
参数	certificate：一个证书对象 iInfoType：证书信息特定域类别
返回值	证书信息特定域对应的证书信息
说明	iInfoType：证书信息特定域类别，其中的参数选择参考 5.1.2 章节的证书中相关信息的类型编码 量取值 [常用]获取证书属性信息 使用频率：较常用，多 CA 支持时，需定制; 注：第 9 项值一般作为证书的绑定值，该项值为一个复合值;

重载二:

原型	String getX509CertificateInfo(Certificate certificate, String OID)
参数	certificate：一个证书对象 OID：证书拓展域信息
返回值	证书信息特定域对应的证书信息
说明	iInfoType：证书信息特定域类别，其中的参数选择参考 5.1.2 ，证书中相关信息的类型编码 [常用]获取证书属性信息 使用频率：较常用，多 CA 支持时，需定制； 注：第 9 项值一般作为证书的绑定值，该项值为一个复合值；

2.2.5. getX509CertificateThumbprint

功能：获得 X509 证书姆印值，返回一个证书对象缩略图

原型	String getX509CertificateThumbprint(Certificate certificate)
参数	Certificate：一个证书对象
返回值	证书的姆印值
说明	注：算法参见项目定制常量“NETCAPKI_ALGORITHM_HASH”中定义

2.3. 签名类接口

2.3.1. signNETCA

功能：PKCS#7 签名（通用,兼容以前模式）

原型	String signNETCA (String source,boolean hasDetached)
参数	source: 原文 hasDetached：是否带原文（false：带，true：不带）
返回值	PKCS#7Base64 编码的签名值
说明	只适用于 Window 系统。 没有证书与密码的参数，通过弹出框的方式 select 一个证书，并通过输入密码完成签名过程 使用场景：较常用，兼容以前代码改造

2.3.2. signedDataByPwd【重载】

功能：带 PIN 码 PKCS7 签名

重载 1：

原型	String signedDataByPwd (String source,boolean hasDetached,String password)
参数	source: 原文 hasDetached：是否带原文（false：带，true：不带） password：证书中 key 的密码
返回值	PKCS#7Base64 编码的签名值
说明	只适用于 Window 系统。 通过弹出框的方式 select 一个证书 使用频率：少用，一般用 signNETCA

重载 2：

原型	String signedDataByPwd (byte[] source,boolean hasDetached,String password)
参数	source:原文（字节数组） hasDetached：是否带原文（false：带，true：不带） password：证书中 key 的密码
返回值	PKCS#7Base64 编码的签名值
说明	只适用于 Window 系统。 通过弹出框的方式 select 一个证书 使用频率：少用，一般用 signNETCA

2.3.3. signedDataByCertificate【重载】

功能：使用证书进行 PKCS7 签名

原型	String signedDataByCertificate (Certificate certificate,String source,boolean hasDetached)
参数	certificate：需要用来进行签名的证书对象 source: 原文 hasDetached：是否带原文（false：带，true：不带）
返回值	PKCS#7Base64 编码的签名值
说明	通过弹出框的方式输入密码完成签名过程

重载 2：

原型	String signedDataByCertificate (Certificate certificate,String source,boolean hasDetached,String password)
参数	certificate：需要用来进行签名的证书对象 source: 原文 hasDetached：是否带原文（false：带，true：不带） password：证书中 key 的密码
返回值	PKCS#7Base64 编码的签名值
说明	

重载 3：

原型	String signedDataByCertificate (Certificate certificate,byte[] source,boolean hasDetached,String password)
参数	certificate：需要用来进行签名的证书对象 source: 原文（字节数组） hasDetached：是否带原文（false：带，true：不带） password：证书中 key 的密码
返回值	PKCS#7Base64 编码的签名值
说明	使用证书进行 PKCS7 签名[具体代码实现]，password 可以为 null

2.3.4. signedDataWithTSA【重载】

功能：获得 PKCS#7 时间戳签名的签名值

重载 1：

原型	String signedDataWithTSA(String source,String url,boolean hasDetached)
参数	source：原文，即签名内容 url：时间戳服务器的 URL hasDetached：是否带原文（false：带，true：不带）
返回值	时间戳的签名值
说明	只适用于 Window 系统。 通过弹出框的方式 select 一个证书 举个例子：url：http://tsa.cnca.net/NETCATimeStampServer/TSAServer.jsp

重载 2：

原型	String signedDataWithTSA(Certificate certificate,String source,String url,boolean hasDetached)
参数	Certificate：一个证书对象 source：原文，即签名内容 url：时间戳服务器的 URL hasDetached：是否带原文（false：带，true：不带）
返回值	时间戳的签名值
说明	举个例子：url：http://tsa.cnca.net/NETCATimeStampServer/TSAServer.jsp

2.3.5. verifySignedDataWithTSA

功能：验证 PKCS#7 时间戳的签名值，返回验证的时间戳的值

原型	String verifyDataWithTSA(String source, String signValue)
参数	source：原文 signValue：签名值
返回值	验证的时间戳的值
说明	

2.3.6. verifySignedData【重载】

功能：验证 PKCS#7 签名值，返回签名证书

重载 1：

原型	Certificate verifySignedData (String source, String signValue)
参数	Source：原文 signValue：签名值
返回值	一个签名证书
说明	**注意：最后不使用 Certificate 对象时，要调用 free 方法释放 Certificate 对象资源**

重载 2：

原型	Certificate verifySignedData (String sSource,byte[] signValue)

参数	Source：原文 signValue：签名值（Base64 编码转字节数组）
返回值	一个签名证书
说明	PKCS7 时间戳签名验证并获取证书（字节数组签名值） **注意：最后不使用 Certificate 对象时，要调用 free 方法释放 Certificate 对象资源**

2.3.7. getSourceFromSignedData

功能：带原文 PKCS#7 签名,验证并获取原文

原型	String getSourceFromSignedData(String signValue)
参数	signValue：签名值
返回值	原文
说明	

2.3.8. signByCertificate【重载】

功能：PKCS#1 签名，获得签名值

重载 1：

原型	String signByCertificate (String source)
参数	source：原文
返回值	签名值
说明	只适用于 Window 系统。 通过弹出框的方式 select 一个证书

重载 2：

原型	String signByCertificate (Certificate certificate, String source)
参数	certificate：一个证书对象 source：原文
返回值	签名值

说明	

2.3.9. verify【重载】

功能：验证 PKCS#1 签名值，返回是否验证成功

重载 1：

原型	boolean verify(String strCertificate, String source, String signValue)
参数	strCertificate： 一个验证签名的证书对象的 Base64 编码 source：字符串原文 signValue：签名值
返回值	是否验证成功
说明	

重载 2：

原型	boolean verify(Certificate certificate, String source, String signValue)
参数	certificate： 一个验证签名的证书对象 source：字符串原文 signValue：签名值
返回值	是否验证成功
说明	

重载 3：

原型	boolean verify(Certificate certificate,byte[] source, String signValue)
参数	certificate： 一个验证签名的证书对象 source：字节数组原文 signValue：签名值
返回值	是否验证成功
说明	

2.4. 加密类接口

2.4.1. envelopedData【重载】

功能：通过数字信封对原文进行非对称加密，得到加密值

重载 1：

原型	String envelopedData(byte[] source)
参数	source：原文，即需要进行加密的值（字节数组）
返回值	加密值
说明	只适用于 Window 系统。 通过弹出框的方式，select 一个加密证书 [较常用]数字信封加密 使用自己的加密证书加密信息

重载 2：

原型	String envelopedData(Certificate certificate, byte[] source)
参数	certificate：加密证书，用来对数据进行加密 source：原文，即需要进行加密的值
返回值	加密值
说明	

重载 3：

原型	String envelopedData(String strCertificate, byte[] source)
参数	strCertificate：数字信封使用到的证书 Base64 编码值。 source：原文，即需要进行加密的值
返回值	加密值
说明	用服务器证书加密

2.4.2. developedData

功能：通过数字信封对原文进行非对称解密，得到解密值

原型	byte[] developedData(String envelopedValue)
参数	envelopedValue：Base64 编码的加密值，即需要数字信封进行解密的值
返回值	解密值
说明	只适用于 Window 系统。 通过弹出框的方式，select 一个加密证书

2.4.3. developedDataBySoft

功能：通过数字信封对原文进行非对称解密，得到解密值

原型	byte[] developedDataBySoft (String fileUrlPfx, String keyStoreType,String certPath, Strin password,String envelopedValue)
参数	certKeyStorePath：软证书 jks 或者 pfx 路径 keyStoreType： keyStore 类型 certPath：证书路径 password：证书 key 的密码 envelopedValue：Base64 编码的加密值，即需要数字信封进行解密的值
返回值	解密值（字节数组）
说明	通过软证书来实现数字信封解密，需要用到 PFX 或者 jks 软证书 数字信封解密，使用 pfx 或者 jks 软证书 使用场景：客户端用 cer 文件进行加密，服务器端使用 PFX 软证书进行解密 1).在负载均衡情况下，证书应放到公共位置（如 FTP 服务器），解密时下载到本服务器，进行解密； 2).为防止 PIN 码泄露，解密后应变更密码；

2.4.4. asymmetriEncrypt

功能：对原文进行非对称加密，得到加密值

原型	byte[] asymmetriEncrypt(String source)
参数	source：原文，即需要进行加密的值
返回值	加密值（字节数组）
说明	只适用于 Window 系统。 通过弹出框的方式 select 一个加密证书 [较常用]数字信封加密 使用自己的加密证书加密信息

2.4.5. asymmetriDecrypt

功能：对原文进行非对称解密，得到解密值

原型	String asymmetriDecrypt(byte[] encryptValue)
参数	encryptValue：加密值，即需要进行解密的值（字节数组）
返回值	解密值
说明	只适用于 Window 系统。 通过弹出框的方式 select 一个加密证书，如果该证书与加密时候选择的证书不同，那么返回值 乱码。

2.5. 时间戳接口

2.5.1. getTimeStamp

功能：获得一个时间戳

原型	String[] getTimeStamp(String source, String url)
参数	Source:原文 url：时间戳获取地址
返回值	返回时间戳相对应值，字符串数组
说明	获得时间戳 设置时间戳的 URL：http://tsa.cnca.net/NETCATimeStampServer/TSAServer.jsp 设置时间戳的 hash 算法：Hash.*SHA256* 如果时间戳服务器响应失败，则返回 TSA 响应失败的状态值 第一位：时间； 第二位：时间戳 Token; 第三位：时间戳证书；

2.5.2. verifyTimeStampToken

功能：时间戳验证

原型	String verifyTimeStampToken(String source, String token)
参数	Source： 原文

	Token：时间戳 Token
返回值	若验证成功即返回时间戳的时间，否则为 null
说明	说明：时间格式为 yyyy-MM-dd HH:mm:ss

2.6. 硬件类接口

2.6.1. getKeySerial

功能：获取所有设备中的序列号

原型	List<String> getKeySerial()
参数	无
返回值	一个包含设备序列号的集合
说明	

2.6.2. getDeviceSet

功能：获取介质设备

原型	DeviceSet getDeviceSet()
参数	无
返回值	返回当前系统中所有的设施的集合
说明	只适用于 Windows 平台 **注意：最后不使用 DeviceSet 对象时，要调用 free 方法释放 DeviceSet 对象资源**

2.6.3. getCertWithDeviceSN

功能：根据 key 序列号，选择用户证书

原型	Certificate getCertWithDeviceSN(String keySN)
参数	keySN：key 序列号
返回值	对应的证书对象

说明	**注意：最后不使用 Certificate 对象时，要调用 free 方法释放 Certificate 对象资源**

2.7. 证书验证类接口

该类接口全部封装在 CertAuthClient.java 类中；

2.7.1. 网关验证类接口

在项目采购电子认证网关产品的情况下，请参考本章节进行证书验证操作。

2.7.1.1. verifyCertEx（网关验证证书方式一：服务接口【推荐使用】）

功能：验证证书有效性（验证某个时间点当前证书的有效性）。

原型	HashMap<String, String> verifyCertEx(Certificate certificat, String verifytime,int ku)
参数	certificate：需要验证的证书 verifytime：证书验证时间 ku：即 keyusage，证书密钥用法
返回值	返回一个存储证书状态码、签名值、摘要等信息的 hashMap
说明	网关验证证书有效性(某个时间) Verifytime，可为 null，表示当前时间 Ku，一般为 0xc0 strUrl：验证网关接口地址，即 URL，通过 netca_config.properties 配置文件来设置 serverCertificate：服务器证书，通过 netca_config.properties 配置文件来设置

2.7.1.2. checkCert（网关验证证书方式二：HTTP 接口）

功能：验证证书有效性（验证某个时间点当前证书的有效性）。

原型	int[] checkCert(String strCertificate64)
参数	strCertificate64：需要验证的证书的 Base64 编码
返回值	返回证书状态码等相关信息；

说明	网关相关的参数通过 netca_config.properties 配置文件设置。包括： 验证网关接口地址：sMethodTwoServerUrl 服务器证书：serverCertificate 响应信息的格式：请求流水号 | 响应码 | 验证证书姆印 Hex 编码 | 证书状态码 | 对前面内容的签名 Base64 编码 配置文件说明，见下面章节的“配置文件说明（netca_config.properties）” 备注：一共包含两个值，其中可以用数组的第一位表示响应吗，第二位表示证书状 态码，其中响应吗可以通过 pringEchoCode(certCode)方法进行文字解析，证书状态码 可以通过 parseCertCode(certCode)方法进行文字解析。

2.7.1.3. printEchoCode

功能：解析网关验证通信响应码

原型	String printEchoCode(int certCode)
参数	certCode：响应码
返回值	响应码文字解析
说明	值 对应文字解析 0 "响应成功"; 1 "请求数据包有误"; 2 "证书验证服务器内部出错"; 3 "证书验证服务器压力过重，请以后再试"; 4 "客户端签名有误"; 5 "请求数据需要签名"; 6 "请求未被授权";

2.7.1.4. verifyCertByWebService (通过调用 WebService，实现服务接口的网关验证证书方式一)

功能：通过 WebService 服务进行网关验证证书有效性(某个时间)。

原型	HashMap<String, String> verifyCertByWebService(Certificate certificate, String verifytime,int ku)
参数	certificate：被验证的证书

	verifytime：证书验证时间 ku：即 keyusage，证书密钥用法
返回值	回一个存储了数据的证书状态码、签名值、摘要等信息的 hashMap
说明	**WebService URL**，通过 netca_config.properties 配置文件来设置，参考 netca_config.properties 配 文件中，键 WebServiceURL 对应的值 基本格式：http://网关 IP:端口/NetcaCertAA/NetcaCertAAWS?wsdl Web 返回的是一个 xml，进行解析后，返回的是 version、signature、verifytime、digest、certNam status 等，如下所示： 

2.7.1.5. checkCertExByWebService (通过调用 WebService，实现网关 HTTP 接口的网关验证证书)

功能：通过 WebService 服务进行网关验证证书有效性(当前时间)

原型	String checkCertExByWebService (String cert)
参数	Cert：证书的 Base64 编码
返回值	返回由“|”间隔数据的证书状态码、签名值、摘要等信息
说明	**WebService URL**，通过 netca_config.properties 配置文件来设置，参考 netca_config.properties 配 文件中，键 WebServiceURL 对应的值 基本格式：http://网关 IP:端口/NetcaCertAA/NetcaCertAAWS?wsdl 响应信息的格式：请求流水号	响应码	验证证书姆印 Hex 编码	证书状态码 | 对前面内容的签名 Base64 编码 1506657565592|0|18f76ebac352a843b90b83657a07a2c1749b7850|0|GyK0QHsyZZKFOl/O7vqB3ZM6Z4E2c0mZkTaJP

2.7.1.6. parseCertCode

功能：解析证书验证状态码

原型	String parseCertCode (int certCode)

参数	certCode：证书状态码
返回值	证书状态码对应的文字解析
说明	值 | 对应文字解析
	0 | "证书有效";
	1 | "验证处理失败";
	2 | "证书格式有误";
	3 | "证书不在有效期内";
	4 | "密钥用途不合(KU 和 EKU)";
	5 | "证书名字不合(DN 和 SAN)";
	6 | "证书策略不合(取决于 CVS 的配置)";
	7 | "证书扩展不合(取决于 CVS 的配置)";
	8 | "证书链验证失败(包括了签名检查；基本约束检查；名字约束检查；策略约束检查；ACL 检查等)";
	9 | "证书被注销";
	10 | "注销状态不能确定(如果设置了已过期的 CRL 或 OCSP 无法连接)";
	11 | "证书不受信任/未被授权";
	12 | "证书被暂时锁定/未激活";

2.7.2. 安全中间件证书验证类接口

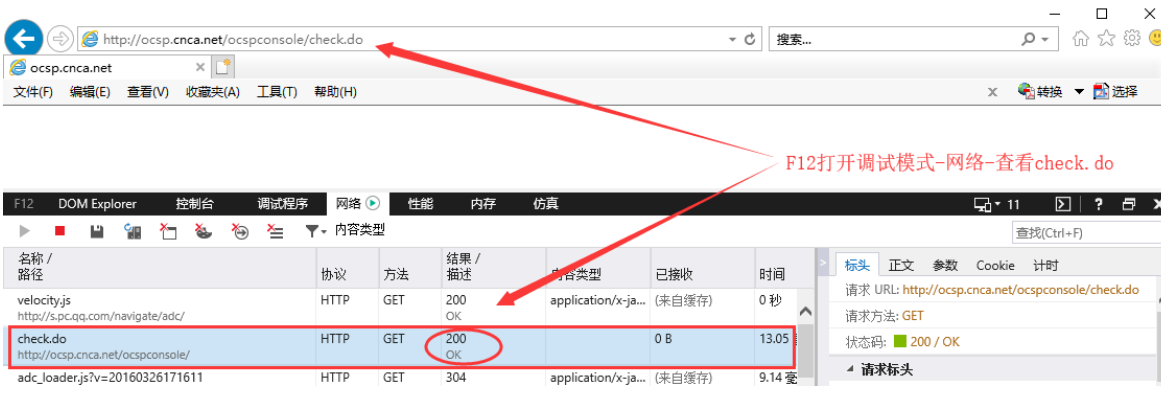
本接口用于应用端证书验证操作

2.7.2.1. checkCertWtihOCSP

功能：OCSP 在线验证用户证书的注销状态。（应用端需连接互联网）

目前在没网关产品的情况下，推荐使用此方式来进行证书验证；

原型	**int** checkCertWtihOCSP(Certificate certificate)
参数	certificate:被验证的用户证书
返回值	证书的注销状态码，经过 Base64 编码的字符串
说明	OCSP 服务器地址一般为：http://ocsp.cnca.net/ocspconsole/check.do OCSP 服务器地址，通过 netca_config.properties 配置文件来设置，参考键 OCSPUrl 对应的值； OCSP 服务器证书，通过 netca_config.properties 配置文件来设置，参考键 OCSPCERTRSA 对应的值

	备注：可以通过 parseCertRevokedState(status)进行文字解析。 说明：如果不能进行域名解析，需要使用到 ip 的方式访问，请联系网证通获取 ip 地址。 Linux 域名访问的命令：curl -o /dev/null -s -w %{http_code} http://ocsp.cncat/ocspconsole/check.d 如果网络畅通，能够访问 ocsp，则返回状态码 200，事例如下：  Window 域名访问测试（打开 IE）： 

2.7.2.2. checkCertWtihCRL

功能：CRL 验证证书注销状态

CRL 会定期进行更新发布，需管理员定期手动下载最新的 CRL，更新到应用中，进行验证；

在离线情况下推荐采用此方式进行证书验证；

原型	int checkCertWtihCRL(Certificate certificate, String crl)
参数	certificate:被验证的用户证书 crl：一个 crl 文件的字符串格式编码值
返回值	证书的注销状态码，经过 Base64 编码的字符串
说明	网证通 CRL 下载地址：http://crl.cnca.net/crl.html **注意：**CRL 的作用范围期限是很短的，大概是一个星期，所以测试案例中的 crl 文件很有可能已经过期，请及时在官网下载最新对应的 CRL 文件

备注：可以通过 parseCertRevokedState(status)进行文字解析。

2.7.2.3. parseCertRevokedState

功能：证书注销状态的解析说明

原型	String parseCertRevokedState (int status)	
参数	status：状态值	
返回值	对应证书注销状态码解析	
说明	值	对应的文字解析
	1	"证书有效";
	-1	"证书的状态不能确定";
	0	"证书已作废";
	-2	"上级 CA 证书已作废";