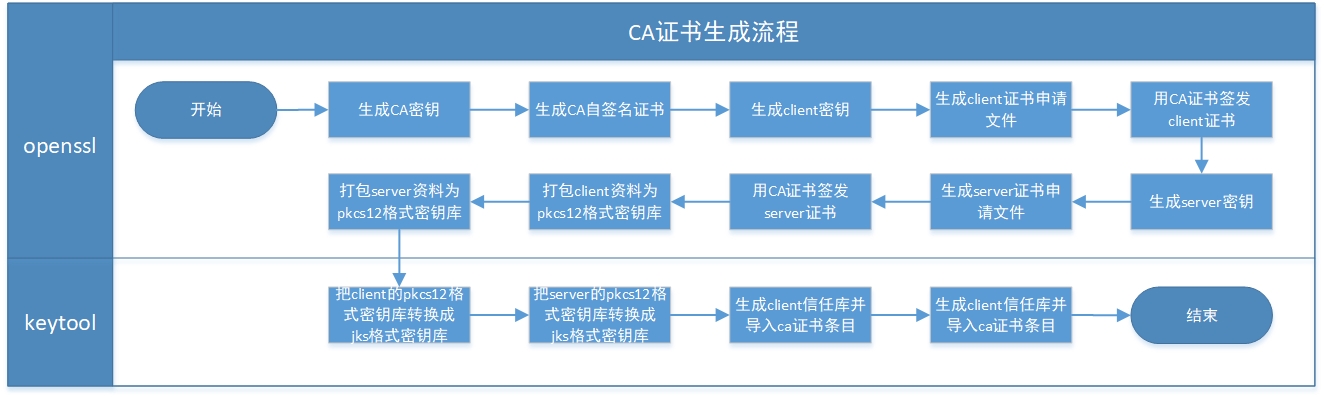
# SSl双向认证中CA证书生成

## 工具

本次例子中使用了OpenSSL和keytool两个工具。OpenSSl主要用于生成CA根证书（用于给客户端、服务端证书签名）、客户端证书、服务端证书；而keytool主要用于把OpenSSL生成的pkcs12类型的密钥库导入到jks类型的密钥库，并且生成保存ca证书的信任库。

## 流程

### 流程图



### 流程文件映射表

|  |  |
| --- | --- |
| 流程 | 生成文件 |
| 生成CA密钥 | cakey.pem |
| 生成CA自签名证书 | ca.cer |
| 生成client密钥 | clientkey.pem |
| 生成client证书申请文件 | client.csr |
| 用CA证书签发client证书 | clinet.crt |
| 生成server密钥 | serverkey.pem |
| 生成server证书申请文件 | server.csr |
| 用CA证书签发server证书 | server.crt |
| 打包client资料为pkcs12格式密钥库 | client.pkcs12 |
| 打包server资料为pkcs12格式密钥库 | server.pkcs12 |
| 把client的pkcs12格式密钥库转换成jks格式密钥库 | client.jks |
| 把server的pkcs12格式密钥库转换成jks格式密钥库 | server.jks |
| 生成client信任库并导入ca证书条目 | clienttrust.jks |
| 生成server信任库并导入ca证书条目 | servertrust.jks |

## OpenSSL

以下代码中统一用caPath代替存放生成文件的文件夹路径。使用时需要替换成自己实际的文件夹路径

1、生成CA密钥

**openssl genrsa –out caPath\ca.pem 2048**

2、生成CA自签名证书，注意此步骤和4、7步不一样 ,并且需要输入机构信息

**openssl req -x509 -new -key caPath\ca.pem -out ca.crt**

3、生成client密钥

**openssl genrsa -out caPath\clientkey.pem 2048**

4、生成client证书申请文件，需要定义机构信息 、证书请求密码

**openssl req -new -key caPath\clientkey.pem -out caPath\client.csr**

5、用CA证书签发client证书

**openssl x509 -req -in caPath\client.csr -CA caPath\ca.crt -CAkey caPath\cakey.pem -CAcreateserial -days 3650 -out caPath\client.crt**

6、生成server密钥

**openssl genrsa -out caPath\serverkey.pem 2048**

7、生成server证书申请文件，需要定义机构信息 、证书请求密码

**openssl req -new -key caPath\serverkey.pem -out caPath\server.csr**

8、用CA证书签发server证书

**openssl x509 -req -in caPath\server.csr -CA caPath\ca.crt -CAkey caPath\cakey.pem -CAcreateserial -days 3650 -out caPath\server.crt**

9、打包client资料为pkcs12格式密钥库，需要定义pkcs12密钥库密码

**openssl pkcs12 -export -in caPath\client.crt -inkey caPath\clientkey.pem -out caPath\client.pkcs12**

10、打包server资料为pkcs12格式密钥库，需要定义pkcs12密钥库密码

**openssl pkcs12 -export -in caPath\server.crt -inkey caPath\serverkey.pem -out caPath\server.pkcs12**

## KeyTool

11、把client的pkcs12格式密钥库转换成jks格式密钥库，需要输入pkcs12密钥库密码，定义jks密钥库密码

**keytool -importkeystore -srckeystore caPath\client.pkcs12 -destkeystore caPath\client.jks -srcstoretype pkcs12**

12、把server的pkcs12格式密钥库转换成jks格式密钥库，需要输入pkcs12密钥库密码，定义jks密钥库密码

**keytool -importkeystore -srckeystore caPath\server.pkcs12 -destkeystore caPath\server.jks -srcstoretype pkcs12**

13、生成client信任库并导入ca证书条目

**keytool -importcert -alias ca -file caPath\ca.crt -keystore caPath\clienttrust.jks**

14、生成server信任库并导入ca证书条目

**keytool -importcert -alias ca -file caPath\ca.crt -keystore caPath\servertrust.jks**

PS：最终代码中需要使用的是client.jks、clienttrust.jks、server.jks、servertrust.jks四个密钥库，查看密钥库中信息可以用以下代码密钥库信息，其中可以查看到密钥库的类型和条目的的类型(PrivateKeyEntry为私钥即密钥库中条目，trustedCertEntry为证书即信任库中条目)

**keytool -list -v -keysrore caPath\client.jks**