keyTool证书管理

1. 生成密钥对

Keytool -genkeypair -keyalg RSA -keysize 2048 -sigalg SHA1withRSA -validity 36000 -alias [www.wcl.org](http://www.wcl.org) -keystore d:/wcl.keystore -dname “CN=www.wcl.org, OU=wcl, O=w ,L=YT ,ST=XA,C=CN”

各参数的含义如下：

-genkeypair 生成密钥对

-keyalg 指定密钥算法，这里指定为RSA算法。

-keysize 指定密钥长度，默认1024位，这里指定为2048

-sigalg 指定数子签名算法，

-alias 指定别名

-keystore 指定密钥库存储位置。

-dname 指定用户信息，这里的用户 ，并不是指代现实意义中的用户个体，而是指代网络环境中的用户个体。其中， CN ： 个体的名字与姓氏。OU：组织单位名称。 O： 组织名称。 L：城市或区域名称。ST：　州或省份名称。CN：国家代码。

1. 导出证书

通过上述操作后，密钥库中己经创建了数字证书。虽然这时的数字证书并没有经过CA认证，但并不影响我们使用。

Keytool –exportcert -alias [www.wcl.org](http://www.wcl.org) -keystore d:/aa.keystore -file wcl.cer -rfc -storepass 12345

-export 证书导出操作

-alias 指定导出别名

-keystore 指定密钥库文件

-file 指定导出文件路径

-rfc 指定以Base64编码格式输出。

1. 三、查看证书信息

Keytool -printcert -file wcl.cer

1. 构建Ca签发证书

如果要获取CA机构认证的数字证书，需要生成数字证签 发申请（CSR），经由CA机构证并颁发，同时将认证后的证书导入本地密钥库和信任库

keytool通过 -certreq 命令生成证书请求，

keytool -certreq -alias [www.wcl.org](http://www.wcl.org) -keystore d:/wcl.store -file wcl.csr -v

-certreq 生成数字证书签发申请操作

-alias 指定别名，这是为[www.zlex.org](http://www.zlex.org).

-keystore 指定密钥库文件。

-file 指定导出文件路径。

-v 详细信息

读者朋友可以提交CSR文件内容，获得相应的签发数字证书。

获得签发的数字证书后，需要将其导入信任库，keytool 通过 -importcert 命令导入证书。

1. 将证书导入信任库

Keytool –importcert -trustcacerts -alias [www.wcl.org](http://www.wcl.org) -file wcl.cer -keystore d:/wcl.keystore -storepass 123456

各参数的含义如下：

-importcert 导入数字证书。

-trustcacerts 将数字证书导入信任库

-alias 指定导别名

-file 指定导入数字证书文件路径，这是为wcl.cer

-keystore 指定密钥库文件。

六、查看导入证书

导入证书后，我们可以通过，相关命令查看该证书。KeyTool通过 -list 命令列出密钥库中的条目。

Keytool -list -alias [www.wcl.org](http://www.wcl.org) -keystore d:/wcl.keystore

可以加入参数 -v 或 -rfc 查看该证书的详细信息。

OpenSSL 证书管理

OpenSSL（http://www.openssl.org）是一个开放源代友软件包，由Eric A .Youn 和 Tim J.Hudson等人编写，实现了SSL及相关加密技术，是最常用的证书管理工具。OpenSSL功能远胜KeyTool ， 可用于根证书，服务器证书和客户证书的管理。相关操作可以参考官方文档（<http://www.openssl.org/docs/apps/poenssl.html>）。

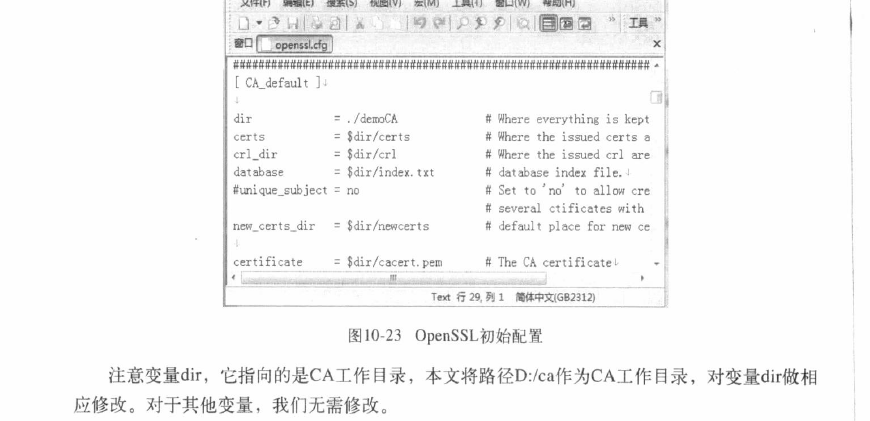
1、准备工作

Windows版OpenSSL分为 Win32 和 Win64共两个平台版本，请读者注意选择合适的发行版本。

下载后，需要将%OpenSSL\_Home%/bin;加入到系统的变量Path中。

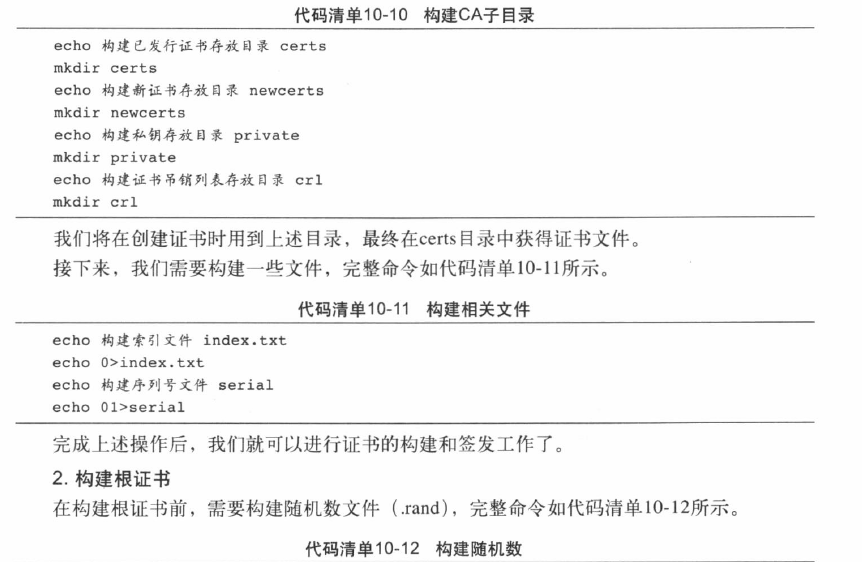
2、工作目录

打开OpenSSL 配置文件，openssl.cfg(%OpenSSL\_Home%\bin\openssl.cfg)

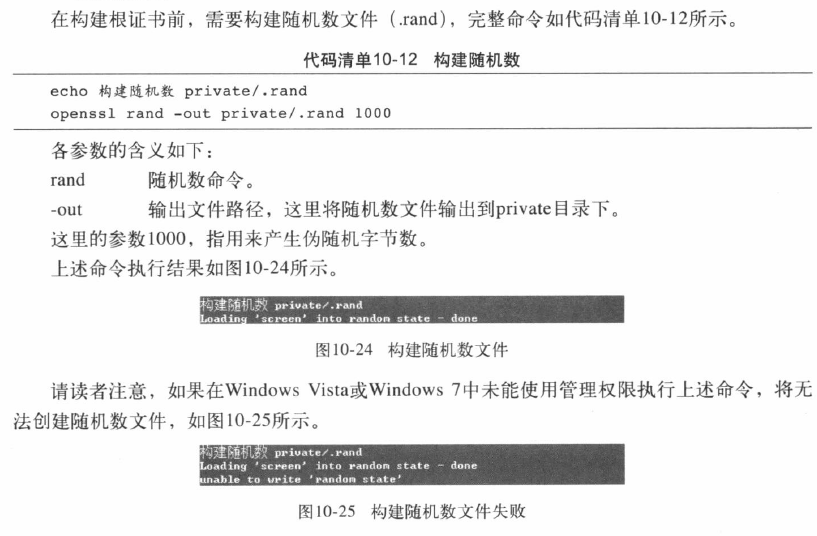


可以通过设置变量， “OPENSSL\_CONF”指定配置文件路径。

创建CA工作目录后，我们需要构建一些子目录，用于存放证书，密钥等。完整命令如下：



3、构建根证书



OpenSSL 通常使用PEM（Privacy Enbanced Mail，隐私增强邮件）编码格式保存私钥。

接下来，我们需要构建根证书密钥（ca.key.pem）,完整命令如下代码清单10-13。

//echo 构建根证书私钥 private/ca.key.per

Openssl genrsa -aes256 -out private/ca.key.pem 2048

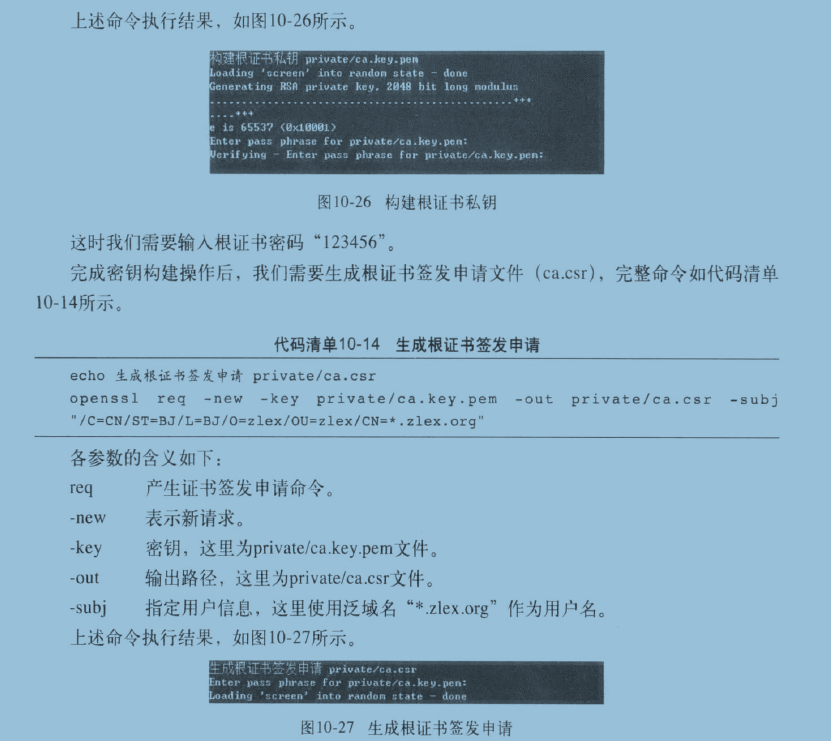
各参数的含义如下：

genrsa 产生RSA密钥命令。

-aes256 使用AES算法（256位密钥）对产生的私钥加密。可选算法包括DES ， DESede , IDEA , 和AES

-out 输出路径。

这里的参数2048，指RSA密钥长度位数，默认长度为512位。



openssl req -new -key private/ca.key.pem -out private /ca.csr -subj “/C=CN/ST=BJ/L=BJ/O=wcl/OU=jxhk/CN=www.wcl.org”

这是我们需要输入根证书密码“12345”

得到根证书签发申请文件后，我们可以将其发送给CA机构签发。当然，我们也可以自行签发根证书。签发根证书完整命令如代码清下：

openssl x509 -req -days 10000 -sha1 -extensiions v3\_ca -signkey private/ca.key.pem -in private/ca.csr -out certs/ca.cer

各参数的含义如下：

x509 签发X.509格式证书命令

-req 证书输入请求

-days 有效天数，

-sha1 证书摘要算法，这里是SHA1算法。

-extensions 按OpenSSL配置文件v3\_ca项添加扩展。

-signkey 自签名密钥，这里为private/ca.key.prm.

-in 输入文件

-out 输出文件

这时我们需要输入根证书密码“12345”。

OpenSSL 产生的数字证书不能在java语言环境中直接使用，需要将其转化为PKCS#12编码格式。完整命令如下：

openssl pkcs12 -export –cacerts -inkey private/ca.key.pem -in certs/ca.cert -out certs/ca.p12

各参数的含义如下：

Pkcs12 PKCS#12编码格式证书命令

-export 导出证书

-cacerts 仅导出CA证书

-inkey 输入密钥

-in 输入文件

-out 输出文件

个人信息交换文件（PKCS#12）可以作为密钥或信任库使用，我们可以通过KeyTool查看该密钥库的详细信息。完整命令如下：

keytool -list -keystore certs/ca.p12 -storetype pkcs12 -v storepass 12334

注意,这里的参数 –storetype值为 “pkcs12”

构建了根证书（ca.cer）,我们可以使用根证书签发服务器证书和客户证书。

**4、构建服务器证书**

服务器证书的构建与根证书的构建相似，首先需要构建私钥。完整命令如下：

opens genrsa -aes256 -out private/server.key.pem 2048

各参数的含义如下：

genrsa 产生RSA密钥的命令

-aes256 使用AES算法（256位密钥 ）对产生的私钥加密。可选DES ，DESede , IDEA 和AES

-out 输出路径， 这是指private/server.key.pem

这里的参数2048，指RSA密钥长度位数，默认长度为512

完成服务器证书密钥构建后，我们需要产生服器证书签发申请。命令如下：

openssl req -new -key private/server.key.pem -out private/server.csr -subj “/C=CN/ST=BJ/L=BJ/O=wcl/OU=jxhk/CN=www.wcl.org”

各参数的含义如下：

Req 生成证书签发申请命令。

-new 新请求。

-key 密钥，

-out 输出路径

-subj 指定用户信息。

用根证书签发服务器证书

openssl x509 -req -days 3650 -sha1 –extensions v3\_req -CA certs/ca.cer -Cakey

private/ca.key.pem -CAserial ca.sr1 -CAcreateserial -in private/server.csr -out certs/server.cer

各参数的含义如下：

X509 签发X.509格式证书命令。

-req 证书输入请求。

-days 有效天数，

-sha1 证书摘要算法

-extensions 按OpenSSL配置文件V3\_req项添加扩展。

-CA CA证书

-CAKey 证书密钥

-CAcreateserial 创建证书序列号。

-in 输入文件，这里为private/server.csr

-out 输出文件

我们同样需要将OpenSSL产生的数字证转化为PKCS#12编码格试。

Openssl pkcs12 -export -clcerts -inkey privte/server.key.pem -in certs/server.cer -out certs/server.p12

pkcs12 PKCS#12编码格式证书命令。

-export 导出证书

-clcerts 仅导出客户证书。

-inkey 输入密钥文件路径

-in 输入文件路径

-out 输出文件路径

**5、构建客户端证书**

客户证书的构建与服务器证书构建基本一致，首先要构建私钥，

openssl genrsa -aes256 -out private/client.key.pem 2048

完成客户证书密钥构建后，我们需要产生客户证书签发申请。

openssl req -new -key private/client.key.pem -out private/client.csr -subj

“/C=CN/ST=BJ/L=BJ/O=JX/OU=JXHK/CN=JXHK”

签发客户证书

openssl ca –days 3650 -in private/client.csr -out certs/client.cer -cert certs/ca.cer -keyfile private/ca.key.pem

ca 签发证书命令

-days 证书有效期

-in 输入文件

-out 输出文件

-cert 证书文件

-keyfile 根证书密钥文件

最后也需要将获得的客户证书转化java语言可以识别的PKCS#12编码格式

Opens pkcs12 -export -inkey private/client.key.pem -in certs/client.cer -out certs/client.p12