

目录

目录	1
一、 关于 keytool	2
1. keytool 简介	2
2. keytool 下载及安装	2
二、 生成证书请求文件 CSR	3
1. 生成私钥	3
2. 生成 CSR 证书请求文件	5
三、 下载服务器证书	7
1. 准备下载证书所需信息	7
2. 下载证书	7
3. 关于证书的格式转换	10
四、 导入根证书和服务器证书	13
1. 下载根证书及 CNNIC 中级根证书	13
2. 开始导入证书	13
1) 将根证书 root.cer 导入 keystore 文件	13
2) 将中级 CA 证书 cnic.cer 导入 keystore 文件	13
3) 将服务器证书 WebCert.cer 导入 keystore 文件	14
4) 查看 keystore 中证书列表	14
五、 修改配置文件	16
1. 找到 Jetty 的配置文件	16
2. 准备密钥库文件	16
3. 修改配置文件	16

一、关于 keytool

1. keytool 简介

keytool 是用于管理密钥和证书的工具，使用户和管理员能管理自己的公/私钥对以及相关的证书。keytool 将密钥和证书储存到一个 keystore (JKS) 类型的文件，该文件使用一个密码保护密钥。

2. keytool 下载及安装

请登录 Oracle 的网站：

<http://www.oracle.com/technetwork/java/javase/downloads/>

下载 java 开发包 (JDK)。JDK 中默认安装有 keytool。安装完成后，请配置系统环境变量 JAVA_HOME，指明 JDK 的安装位置。

二、生成证书请求文件 CSR

请确保 `JAVA_HOME\bin` 或者 `JRE_HOME\bin` 目录存在于 `PATH` 变量中或直接使用绝对路径调用 `keytool` 命令。直接使用 `keytool` 创建证书请求文件需要以下两个步骤：

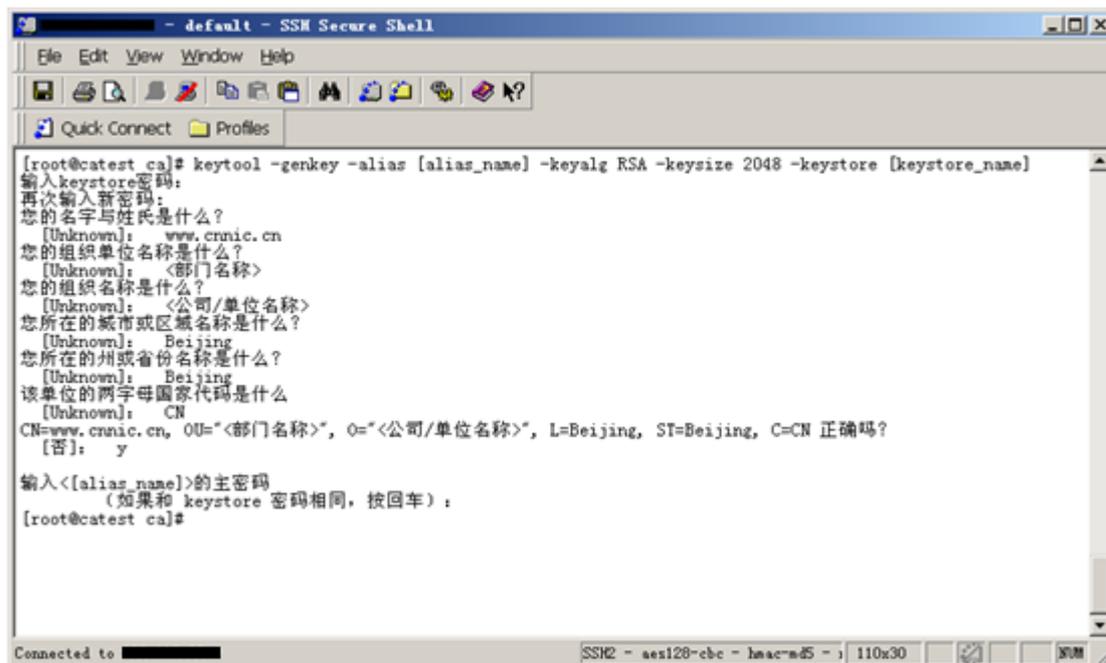
1. 生成私钥

命令格式：`keytool -genkey -alias [alias_name] -keyalg RSA -keysize 2048 -keystore [keystore_name]`

注：[]中的内容为需要输入的参数

- `alias_name`：表示证书的别名，在 `keystore` 中，成对的公/私钥应具有相同的别名，在后续证书导入环节会用到，如忘记此别名，可用：
`keytool -list -keystore [keystore_name]`命令查看
- `keystore_name`：表示证书密钥库的文件名，扩展名一般为 `keystore` 或 `jks`

以申请域名 `www.cnnic.cn` 的证书请求文件为例，运行情况如下图所示：



图表一 创建私钥

系统提示输入 keystore 密码，如不输入密码直接回车则缺省密码为：changeit。也可以指定一个新的密码，但一定保存好该密码。

系统提示输入“您的名字与姓氏？”，请输入您要申请域名证书的域名，而不是您的真实名称与姓氏，例如：如果需要为 www.cnnic.cn 申请域名证书就必须输入 www.cnnic.cn 而不能输入 cnnic.cn。如果申请通配域名证书，则输入通配域名的形式，通配符为“*”，如：*.cnnic.cn；如果申请多域名证书，则输入多域名中第一个域名即可。

关于组织单位名称、组织名称、所在城市、所在省份和国家缩写(中国填：CN，其他国家填其缩写)，除国家缩写必须填 CN 外，其余信息均可以是英文或中文。最后，系统要求输入主密码，可以直接回车，使主密码保持与 keystore 密码一致。

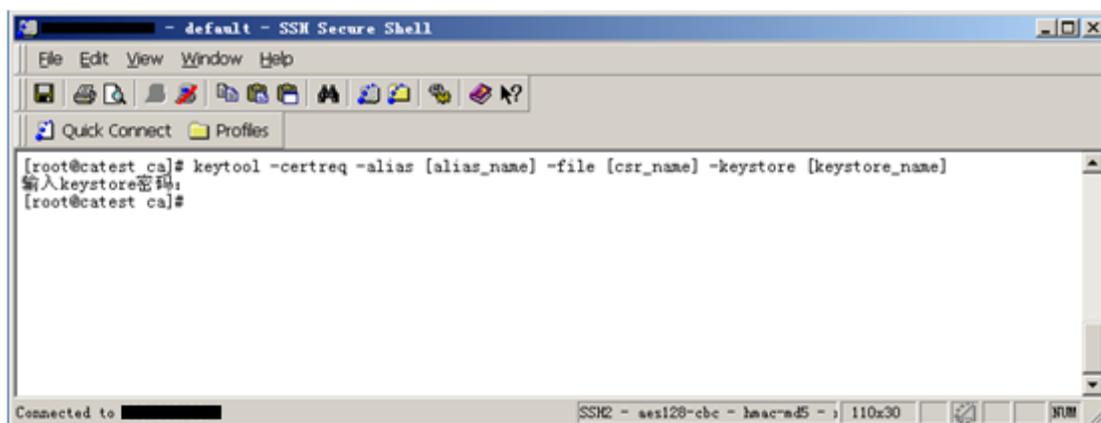
2. 生成 CSR 证书请求文件

命令格式: `keytool -certreq -alias [alias_name] -file [csr_name]`
`-keystore [keystore_name]`

注: []中的内容为需要输入的参数

- `alias_name`: 表示证书的别名
- `csr_name`: 表示证书请求文件的名称, 扩展名一般为 `csr`
- `keystore_name`: 表示证书的密钥库文件名, 扩展名一般为 `keystore` 或 `jks`

使用上例生成的 `keystore` 文件, 运行情况如下图所示:



图表二 生成 csr 文件

系统要求输入第一步骤中填写的 `keystore` 密码。

生成的 `csr` 文件为文本文件, 可以使用记事本等文本查看工具打开刚刚生成的证书请求文件, 如下图所示:



图表三 查看 csr 文件

三、 下载服务器证书

1. 准备下载证书所需信息

参考号与授权码：参考号与授权码是下载证书的密码凭证。当申请的证书通过审核时，用户将接收到由 CNNIC 发送的通过审批的电子邮件通知，该邮件中含有 16 位的参考号与授权码信息，其中参考号与授权码的前 13 位为明文显示，后 3 位为密文显示。审核员会以邮件通知的方式发送后三位的明文显示。

2. 下载证书

登录 CNNIC 官网，进入 CNNIC 服务器证书下载中心页面：

<http://www.cnnic.net.cn/jczyfw/fwqzs/fwqzsxzzx/>

点击相应的链接进入到证书下载页面，如下图所示：

可信服务器证书下载	
点击这里进行在线CSR校验	
参考号：	<input type="text"/>
授权码：	<input type="text"/>
证书请求文件（CSR）：	<p>请把整个CSR文件中 -----BEGIN CERTIFICATE REQUEST----- 和 -----END CERTIFICATE REQUEST----- 之间的内容复制到下边的输入框中</p> <div style="border: 1px solid #ccc; height: 200px; width: 100%;"></div>
<input type="button" value="提交"/> <input type="button" value="重置"/>	

图表四 可信服务器证书下载页面

根据网页上的提示输入“参考号”和“授权码”，将证书请求文件中除去头尾“-----BEGIN NEW CERTIFICATE REQUEST-----”和“-----END NEW CERTIFICATE REQUEST-----”的中间部分内容复制到 CSR 文本框中。结果如下所示：

可信服务器证书下载	
点击这里进行在线CSR校验	
参考号：	<input type="text" value="MV4K646JDDHAF8W5"/>
授权码：	<input type="text" value="CJQLNDB7FQSVEJA3"/>
证书请求文件（CSR）：	<p>请把整个CSR文件中 -----BEGIN CERTIFICATE REQUEST----- 和 -----END CERTIFICATE REQUEST----- 之间的内容复制到下边的输入框中</p> <pre>MIICrDCCAZQCAQAwZzELMAkGA1UEBhMCQ04xEDAOBgNVBAGTB2JlaWppbmcxEDAOBgNVBAcTB2JlaWppbmcxDjAMBgNVBAoTBWNum1jMQ4wDAYDVQQLEwVjbm5pYzEU MBIGAlUEAxMLbTEuY25uaWMuY24wgGEiMAOGCSqGSIb3DQEBAQUAA4IBDwAwggEKAoIBAQcwZKe5sIA8Vv7uYleWQMUvos7K/dagHhyb9DYKouOSQqJkHsFzAMUZzyjLkvE2tUTNtMqbPaxV8TGSg+AcC7zNABYdQpAUWw91dGoLqGtktOsQ/tWd0Bbi1Oj8amCi/yRxpPSPJPrLisBFCxwt+8wGj8WICj3vP8jOqnpUtkfx3S9AMfaAveGretlUF/80DBboVwJXCTKwcc+dHykjsiswAOiWYlgnArdexn1gR4Ym59IjiFmOfiiBSKbGwVlNXJ21f6DsLFKf8JvZq9Yfdjc135QQPOpzGhR98TKzStv/6/c+ocG2yexgFtMZac/Z4lJh9iUmNkp69nbs1sHU5FagMBAAGGADANBgkqhkiG9wOBAQUFAAOCAQEAqGbSXekMJTPsS7VHuP1YzpkOaXN3D3AAyOoT7MC3pEDnlk49e779Vxr2B13nFbh1</pre>
<input type="button" value="提交"/> <input type="button" value="重置"/>	

图表五 填入收到的参考号和授权码以及生成的 CSR

点击“下载”，如果参考号、授权码和证书请求文件均无问题，则显示页面如下所示。

| 证书下载-证书生成

证书文件：	<pre> -----BEGIN CERTIFICATE----- MIIEGzCCAwOgAwIBAgIQEMCXznvJBxWzSSX3sUEd6DANBgkqhkiG9w0BAQUFADAyMQswCQYDVQQG EwJjbjEOMAwGA1UEChMFY25uaWMxEzARBgNVBAMTCmNubmljIHJvb3QwHhcNMTAxMjA3MDkzOTAw WhcNMTEwMjA3MDkzOTAwWjBhMQswCQYDVQQGEwJDTjENMAsGA1UECB4EUXdOrDENMAsGA1UEBx4E UxdOrDENMAsGA1UEChMFY25uaWMxEzARBgNVBAsTBWNubmljMRQwEgYDVQQDEwttMS5jbm5pYy5j bjCCASIwdQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBALBkp7mwigDxW/u5iV5ZAxRU5Lsr91qAe HJv0Ngo645JComQewXMxRnPKMuS8Ta1RM2Oyps8DFXxMzIb4BwLvMOAHJ1CkBRbD3VOaguoa2R2 06xD+1Z3QFuLU6PxqYKL/JHGSk9I8k+suKwEULHC37zAaPxYgKPe8/yM6qe1S2R/HdLOAx9oC94a t62VQX/zQMFuhXAlcJMrDBz50fKSOyKzAA6JZiWCCcT17GfWBHhibn0iOIWY5+KIF IpsbBWWU1cnb V/oOwsUp/wm9mr1h92NyXf1BA86nMaFH3xMrNJO//r9z6hwbbJ7GAWOxlpz9niUmH2JSY2Snr2du </pre>
-------	---

Web服务器证书请将证书编码框中的内容拷贝，并粘贴到文本中，保存成Web服务器能够识别的格式。

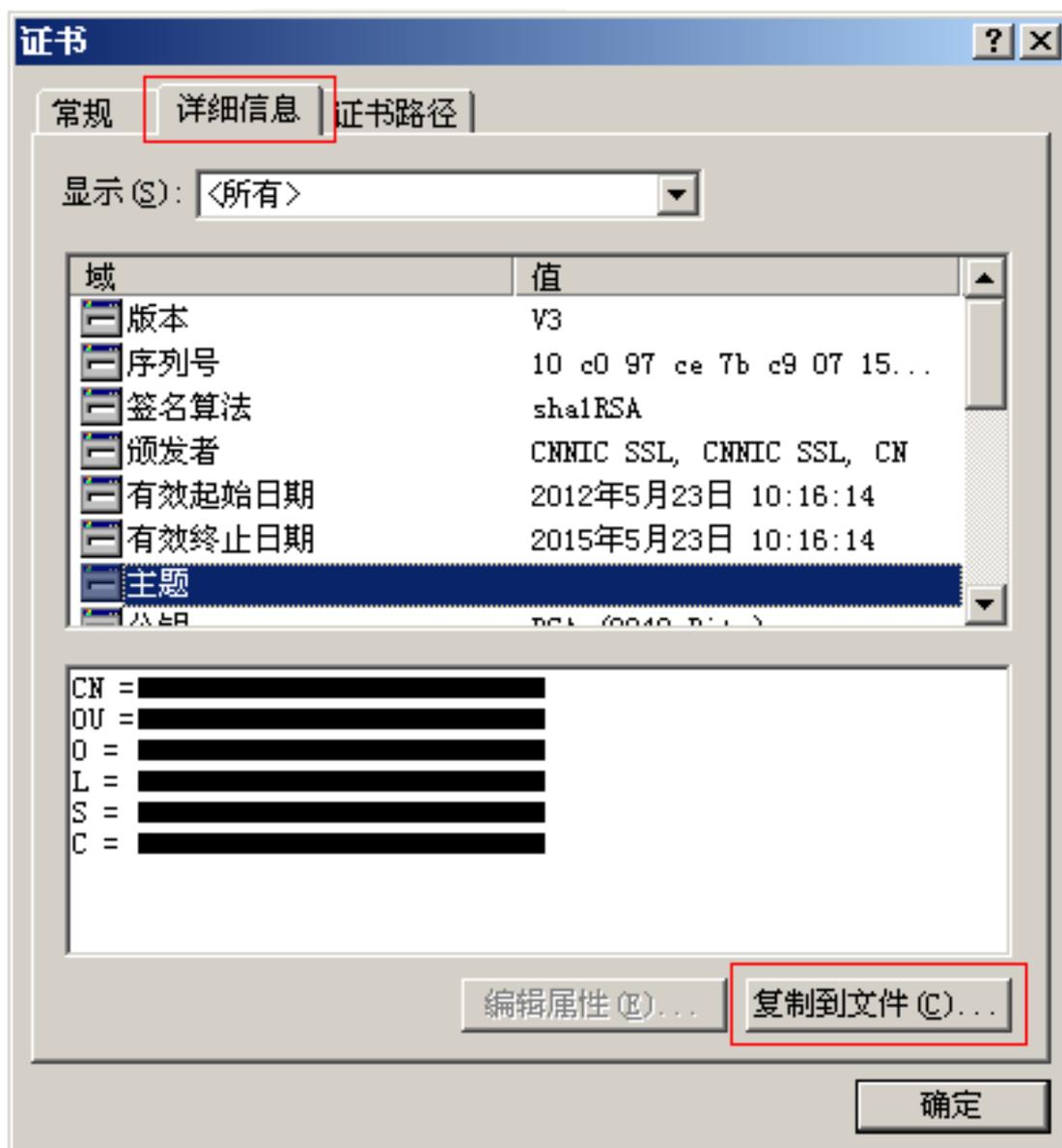
| 保存

图表六 生成证书

请按页面提示将文本框中的内容拷贝下来，粘贴到一个文本文档中保存，为文件起一个方便记忆的名字，以.cer为后缀。您也可以直接点击保存，自动下载一个名为WebCert.cer的文件，该文件即为申请的证书。**请妥善保存该文件，如果该证书丢失，就必须进行证书补发操作，此操作可能会有相应费用产生。**

3. 关于证书的格式转换

从CNNIC获得的证书格式为X.509格式。该将证书文件的扩展名改为cer或crt后，可在windows中双击打开查看证书的相关信息。显示信息类似下图所示：



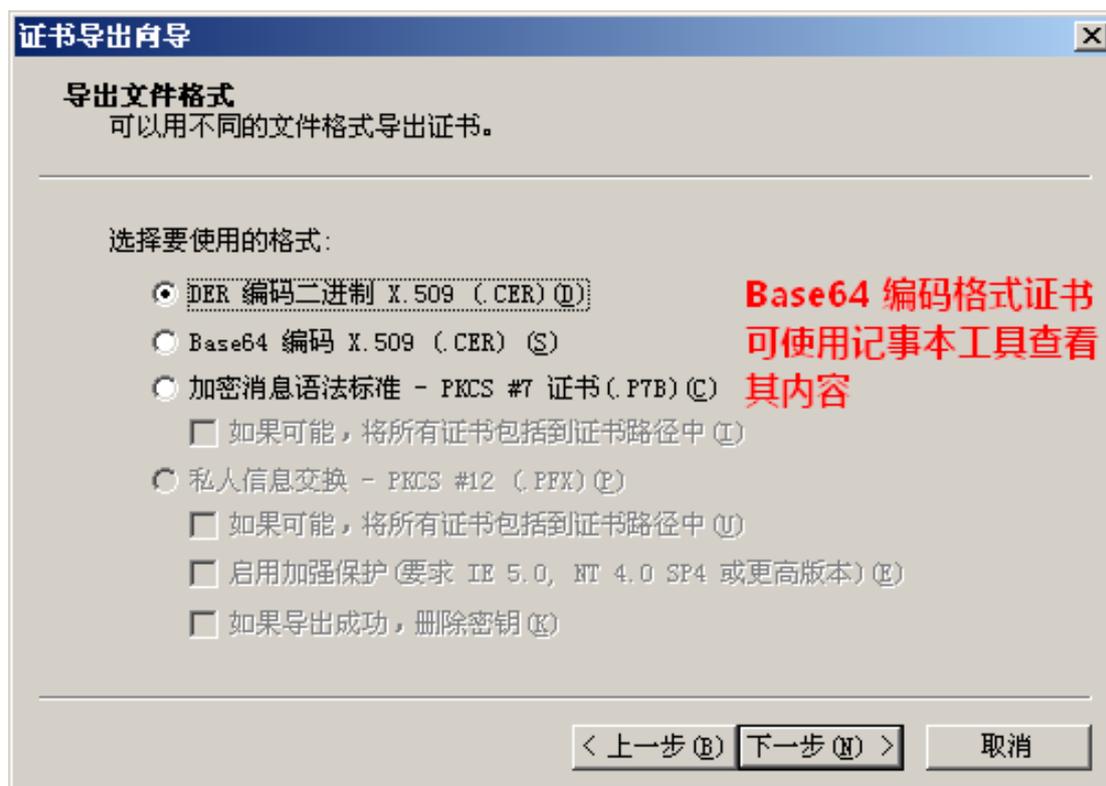
图表七 格式转换

X509 格式的证书利用 windows 提供的图形界面操作工具可以另存为以下两种编码格式：

- BASE64 编码格式：该格式的证书可以用记事本打开
- DER 编码格式：二进制格式

在上图中，点击“详细信息”->“复制到文件”后，即可以根据提示点击

“下一步”利用证书导出向导导出需要格式的证书，如下图所示：



图表八 证书导出向导

四、 导入根证书和服务器证书

1. 下载根证书及 CNNIC 中级根证书

根证书及 CNNIC 中级根证书下载地址：

<http://www.cnnic.net.cn/jczyfw/fwqzs/fwqzsxzzx/>

根据购买产品类型，点击相应的链接下载根证书以及中级根证书，将 CNNIC 中级根证书文件名保存为“cnnic.cer”，将根证书文件名保存为“root.cer”。

2. 开始导入证书

1) 将根证书 root.cer 导入 keystore 文件

命令格式：`keytool -import -trustcacerts -alias root -file root.cer -keystore [keystore_name]`

注：[]中的内容为需要输入的参数

- `keystore_name`：表示证书密钥库的文件名，扩展名一般为 `keystore` 或 `jks`

2) 将中级 CA 证书 cnnic.cer 导入 keystore 文件

命令格式：`keytool -import -trustcacerts -alias cnnic -file cnnic.cer -keystore [keystore_name]`

注：[]中的内容为需要输入的参数

- `keystore_name`：表示证书密钥库的文件名，扩展名一般为 `keystore` 或 `jks`

3) 将服务器证书 WebCert.cer 导入 keystore 文件

命令格式: `keytool -import -trustcacerts -alias [alias_name] -file [server_cert] -keystore [keystore_name]`

注: []中的内容为需要输入的参数

- `server_cert`: 表示服务器证书文件名, 下载服务器证书时默认命名为 WebCert.cer
- `alias_name`: 表示服务器证书的别名, **需要注意的是, 这里的别名要与生成密钥库环节私钥的别名相同, 否则无法完成公/私钥配对**
- `keystore_name`: 表示证书密钥库的文件名, 扩展名一般为 keystore 或 jks

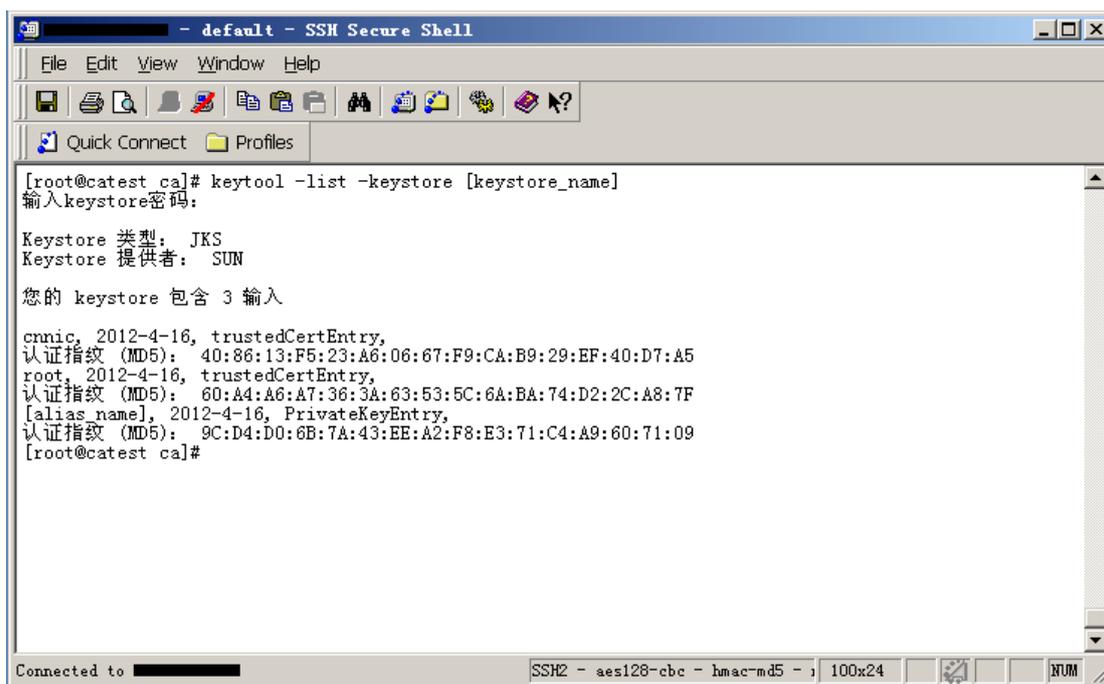
4) 查看 keystore 中证书列表

命令格式: `keytool -list -keystore [keystore_name]`

注: []中的内容为需要输入的参数

- `keystore_name`: 表示证书密钥库的文件名, 扩展名一般为 keystore 或 jks

如果根证书、中级 CA 证书以及服务器证书均成功导入 keystore 中, 用命令查看 keystore 中的证书列表所显示信息应如下图所示, 共含有三级证书链。



```
[root@catest ca]# keytool -list -keystore [keystore_name]
输入keystore密码:

Keystore 类型: JKS
Keystore 提供者: SUN

您的 keystore 包含 3 输入

cnmic, 2012-4-16, trustedCertEntry,
认证指纹 (MD5): 40:86:13:F5:23:A6:06:67:F9:CA:B9:29:EF:40:D7:A5
root, 2012-4-16, trustedCertEntry,
认证指纹 (MD5): 60:A4:A6:A7:36:3A:63:53:5C:6A:BA:74:D2:2C:A8:7F
[alias_name], 2012-4-16, PrivateKeyEntry,
认证指纹 (MD5): 9C:D4:D0:6B:7A:43:EE:A2:F8:E3:71:C4:A9:60:71:09
[root@catest ca]#
```

图表九 查看 keystore 证书

五、修改配置文件

1. 找到 Jetty 的配置文件

首先确认您的 Jetty 安装目录所在位置，打开该安装目录下的 etc 目录，并在 etc 目录下找到 jetty-ssl.xml 文件，这个文件就是 Jetty 的配置文件，您可以以文本方式打开该文件并进行编辑。

2. 准备密钥库文件

将之前步骤生成并已导入各级证书的密钥库 (keystore) 文件准备好，建议复制到 Jetty 安装目录下的 etc 文件夹内。在确认了密钥库文件所在位置的存储路径后就可以开始修改配置文件了。

3. 修改配置文件

打开 jetty-ssl.xml 文件找到如下段落即为配置您的服务器证书所相关的配置。

```
<Configure id="Server" class="org.eclipse.jetty.server.Server">

  <!-- if NIO is not available, use org.eclipse.jetty.server.ssl.SslSocketConnector -->

  <New id="sslContextFactory" class="org.eclipse.jetty.http.ssl.SslContextFactory">
    <Set name="KeyStore"><Property name="jetty.home" default="." />/etc/keystore</Set>
    <Set name="KeyStorePassword">OBF:1vny1zlo1x8e1vnw1vn61x8g1zlu1vn4</Set>
    <Set name="KeyManagerPassword">OBF:1u2u1wml1z7s1z7a1wnl1u2g</Set>
    <Set name="TrustStore"><Property name="jetty.home" default="." />/etc/keystore</Set>
    <Set name="TrustStorePassword">OBF:1vny1zlo1x8e1vnw1vn61x8g1zlu1vn4</Set>
  </New>

  <Call name="addConnector">
    <Arg>
      <New class="org.eclipse.jetty.server.ssl.SslSelectChannelConnector">
        <Arg><Ref id="sslContextFactory" /></Arg>
        <Set name="Port">8443</Set>
        <Set name="maxIdleTime">30000</Set>
```

```
<Set name="Acceptors">2</Set>
<Set name="AcceptQueueSize">100</Set>
</New>
</Arg>
</Call>
</Configure>
```

找到该段落后，请参考如下所示内容修改这段配置文件。

```
<Configure id="Server" class="org.eclipse.jetty.server.Server">

<!-- if NIO is not available, use org.eclipse.jetty.server.ssl.SslSocketConnector -->

<New id="sslContextFactory" class="org.eclipse.jetty.http.ssl.SslContextFactory">
  <Set name="KeyStore"><Property name="jetty.home" default="." />/etc/[keystore_name]</Set>
  <Set name="KeyStorePassword"> password </Set>
  <Set name="KeyManagerPassword"> password </Set>
  <Set name="TrustStore"><Property name="jetty.home" default="." />/etc/[keystore_name]</Set>
  <Set name="TrustStorePassword"> password </Set>
</New>

<Call name="addConnector">
  <Arg>
    <New class="org.eclipse.jetty.server.ssl.SslSelectChannelConnector">
      <Arg><Ref id="sslContextFactory" /></Arg>
      <Set name="Port">443</Set>
      <Set name="maxIdleTime">30000</Set>
      <Set name="Acceptors">2</Set>
      <Set name="AcceptQueueSize">100</Set>
    </New>
  </Arg>
</Call>
</Configure>
```

修改完毕保存退出后，您可以尝试启动 Jetty 服务，测试是否可以正常通过 https 方式访问您的域名。测试成功后请务必妥善备份您的密钥库 (keystore) 文件。