结果如图 22-31 所示，表示证书导入到密钥库中。

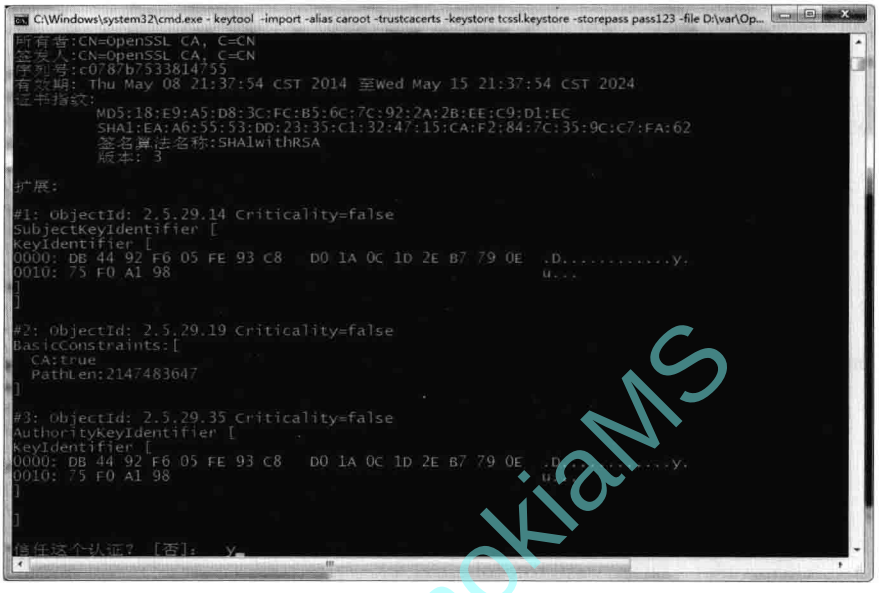

图 22-30　导入 CA 证书

图 22-31　将服务器证书导入到密钥库中

导入服务器证书时，服务器证书的别名必须和私钥别名一致。请留意导入 CA 证书和导入服务器证书时的提示信息，如果在导入服务器证书时使用的别名与私钥别名不一致，系统将提示“认证已添加至 keystore 中”而不是应有的“认证回复已安装在 keystore 中”。

证书导入完成后再次查看 keystore 文件内容，执行命令：

```
keytool -list -keystore tcssl.keystore -storepass pass123
```

结果如图 22-32 所示，表示 tcserver 有对应的证书和私钥。

图 22-32　密钥库中存在证书

2．配置 SSL 连接器

打开 conf 目录下的 server.xml 文件，找到并修改以下内容：

```
<!--
<Connector port= " 8443 "  protocol= " org.apache.coyote.http11.Http11Protocol "
           maxThreads= " 150 "  SSLEnabled= " true "  scheme= " https "  secure= " true "
          clientAuth= " false "  sslProtocol= " TLS "  />
-->
```

修改为

```
<Connector port= " 8443 "  protocol= " org.apache.coyote.http11.Http11Protocol "
          maxThreads= " 150 "  SSLEnabled= " true "  scheme= " https "  secure= " true "
          keystoreFile= " conf\tcssl.keystore "  keystorePass= " pass123 "
          keystoreType= " JKS "
          keyAlias= " tcserver "  keyPass= " test123 "
          clientAuth= " false "  sslProtocol= " TLS "  />
```

可见增加了密钥库文件位置、访问口令、类型，并指定了服务器密钥别名、访问口令参数。其中：

keystoreFile：指定密钥库的位置；

keystorePass：指定密钥库的口令；

keystoreType：指定密钥库的类型，支持 JKS、PKCS#12；

keyAlias：使用的密钥别名；

keyPass：密钥保护口令。

22.3.2 配置 SSL 策略

在 Tomcat 的 SSL 配置中，有 3 个选项与 SSL 握手有关，分别是 sslProtocol、ciphers、clientAuth。

① sslProtocol 指定使用的握手协议，在前面的配置中使用了 TLS 协议，也可以使用 SSLv2、SSLv3、TLSv1、TLSv1.1、TLSv1.2 协议。

② ciphers 指定允许使用的密码算法，本参数使用 JSSE 密码算法名称，多个算法之间以逗号分隔。

③ clientAuth 指定是否验证客户端证书。true 表示客户端必须提供有效证书；want 表示最好提供客户端证书，如果不提供也不算错；false 表示不要求客户端提供证书。

22.3.3 访问 Web Server

在完成上述配置后，重新启动 Tomcat，然后访问 https://127.0.0.1:8443，结果如图 22-33 所示界面，表示 SSL 配置成功，在图中显示了连接的信息。

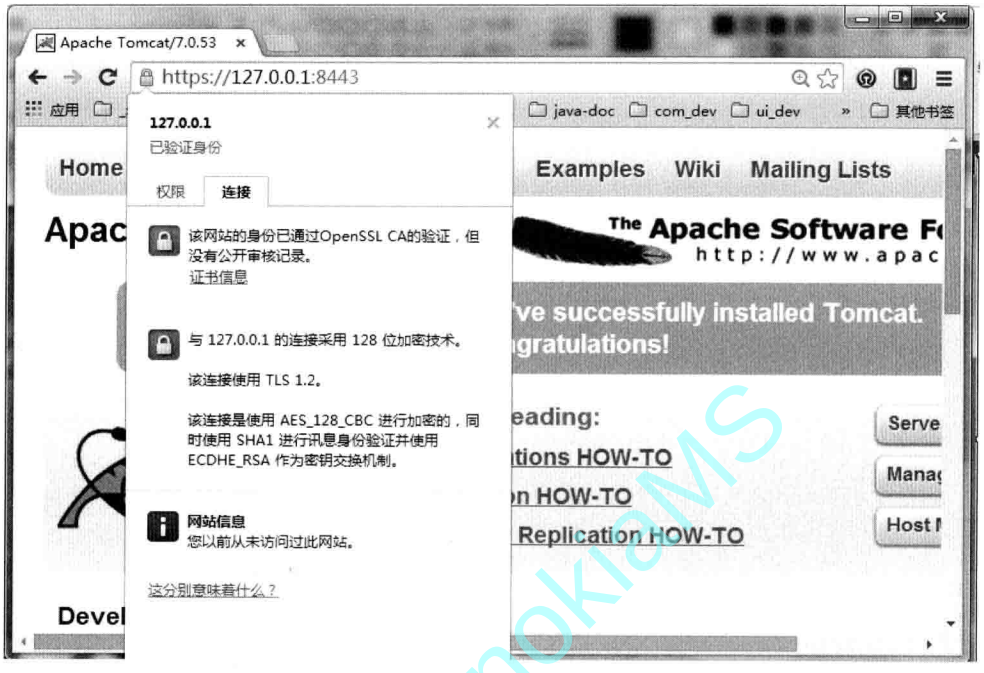

图 22-33　SSL 访问成功界面

第六部分

PKI之运营：CA中心