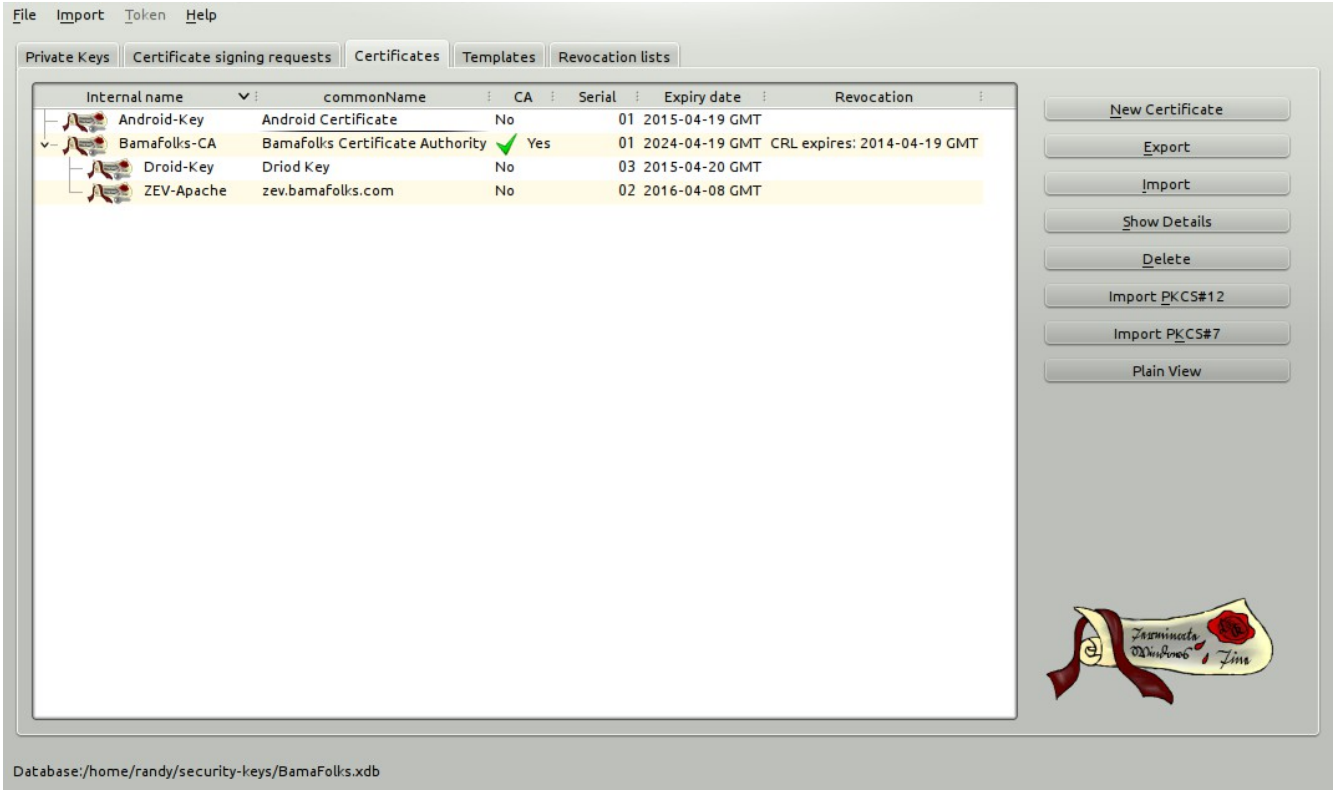
Implement an SSL session with validation.

This example is based on the article and example by Nikolay Elenkov describe here:

http://nelenkov.blogspot.com/2011/12/using-custom-certificate-trust-store-on.html

First, you will need to generate a few SSL certificates. I find the XCA tool (http://xca.sourceforge.net) useful to manage SSL certificates, so I used it to generate the following:



First, the Bamafolks-CA certificate was created. The Android-Key is a self-signed certificate, while the others are examples of a client and server certificates signed by the Bamafolks CA..

Once you have generated the certificates, you need to export them so they can be used with other tools.

To setup Apache, edit the httpd.conf files as shown below. Use XCA to export the certificate and key to the files shown below also:

```
SSLCertificateFile    /etc/ssl/certs/ZEV-Apache.crt
SSLCertificateKeyFile /etc/ssl/private/ZEV-Apache-Key.pem
```

You also have to export the Bamafolks-CA certificate and add it to Apache (or the OS) also. The httpd.conf file has a couple of options as shown below:

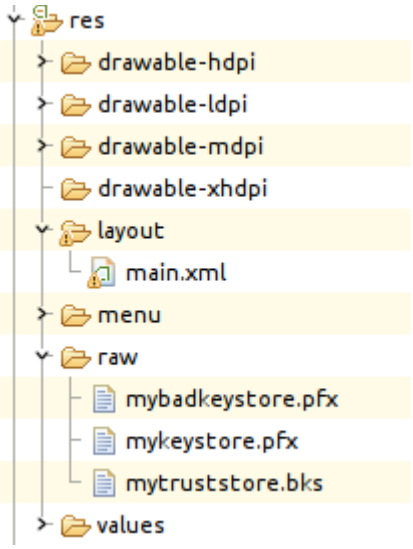
```
#SSLCertificateChainFile /etc/apache2/ssl.crt/server-ca.cr
SSLCACertificatePath /etc/ssl/certs/
```

In order to test client authentication using certificates, I also added this section to the Apache

configuration:

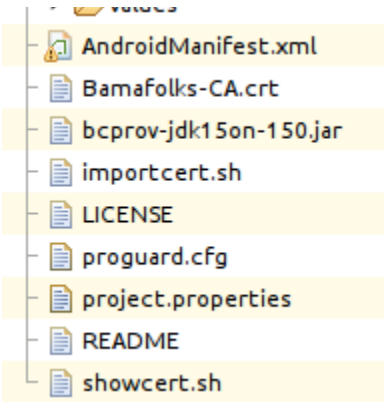
```
<Location /clientauth/>
        SSLVerifyClient require
        SSLVerifyDepth 10
</Location>
```

Next, the custom-cert-https project was downloaded from https://github.com/nelenkov/custom-cert-https and imported into Eclipse. Next, we need to bring the SSL certificates into the project as shown here:



The .pfx files are just the self-signed Android-Key certificate and CA-signed Droid-Key certificates exported to PKCS #12 format. The .bks file is the Bamafolks-CA, stored in the special Bouncy Castle format. That's the format Android uses for its own CA trust store management, so we have to use it also.

To do so, first copy the latest bcprov-*.jar file from https://www.bouncycastle.org/ and edit the importcert.sh file in the project as needed. Latest, export the Bamafolks-CA certificate to a .crt file in the project's main folder:



Finally, you can run this command to import the certificate using the Bouncy Castle provider into the mytruststore.bks file:

```
$ ./importcert.sh Bamafolks-CA.crt
```

Alternately, you can run these commands:

```
$ openssl x509 -inform PEM -subject_hash -noout -in Bamafolks-CA.crt
709a5981

$ keytool -import -v -trustcacerts -alias 709a5981 -file Bamafolks-CA.crt
-keystore res/raw/mytruststore.bks -storetype BKS -providerclass
org.bouncycastle.jce.provider.BouncyCastleProvider -providerpath bcprov-XX-XX.jar
-secret secret
```

You can now restart the Apache server and run the Android project to test things out.