Exercise 5

Creating Keystores for WS-Security

Prior Knowledge

Understand Private Key Crypto and Certificates (at a high level)

Objectives

Create the keystores we will use for the WS-Security Exercise

Software Requirements

- Java Development Kit 7

1. Check that the keytool command is working

   On a command line type keytool

   You should see
   ```
   keytool usage:
   … [LOTS MORE]
   ```

2. Create a directory (e.g. ~/keys/) and change to that directory

3. Now let's create a client key (for Signing)
   Type:
   ```
   keytool -genkey -alias client -keyalg RSA –keystore\
   clientkeystore.jks -storepass clientpass
   ```

   You will be prompted as follows:
   ```
    What is your first and last name?
      [Unknown]:  Paul Fremantle
    What is the name of your organizational unit?
      [Unknown]:  WSO2
    What is the name of your organization?
      [Unknown]:  WSO2
    What is the name of your City or Locality?
      [Unknown]:  Emsworth
    What is the name of your State or Province?
      [Unknown]:  Hampshire
    What is the two-letter country code for this unit?
      [Unknown]:  GB
    Is CN=Paul Fremantle, OU=WSO2, O=WSO2, L=Emsworth,
    ST=Hampshire, C=GB correct?
      [no]:  yes

    Enter key password for <client>
          (RETURN if same as keystore password):
   ```

   You don't have to use my details!

4. Now let's create a server keystore (for encryption):

   ```
   keytool -genkey -alias server -keyalg RSA \
   -keystore serverkeystore.jks \
   -storepass serverpass
   ```

5. Once again fill in the details (this time in a more "server-ish" way perhaps?)

6. Now we need to get these two keystores to trust each other (since there is no uber-CA). Export the client certificate.

```
keytool -export -alias client -keystore clientkeystore.jks \
-file client.cert
Enter keystore password:  [clientpass]
Certificate stored in file <client.cert>
```

7. Now import into the server keystore:

```
keytool -import -file client.cert -keystore serverkeystore.jks \
 -alias client
Enter keystore password:  [serverpass]
Owner: CN=Paul Fremantle, OU=WSO2, O=WSO2, L=Emsworth,
ST=Hampshire, C=GB
Issuer: CN=Paul Fremantle, OU=WSO2, O=WSO2, L=Emsworth,
ST=Hampshire, C=GB
Serial number: 50c484aa
Valid from: Sun Dec 09 12:31:38 GMT 2012 until: Sat Mar 09
12:31:38 GMT 2013
Certificate fingerprints:
       MD5:  50:CC:6D:0F:9F:CC:05:43:F3:A8:A7:DC:AB:F3:58:0F
       SHA1:
90:1B:13:6E:A9:11:02:61:60:80:FB:ED:3E:10:35:31:E3:37:92:1A
       Signature algorithm name: SHA1withRSA
       Version: 3
Trust this certificate? [no]:  yes
Certificate was added to keystore
```

8. Do the opposite – export the server's certificate and import into the client's keystore

9. Validate you have successfully done everything by listing the contents of each keystore. For example:

```
keytool -list -keystore serverkeystore.jks
Enter keystore password:

Keystore type: JKS
Keystore provider: SUN

Your keystore contains 2 entries

client, Dec 9, 2012, trustedCertEntry,
Certificate fingerprint (MD5):
50:CC:6D:0F:9F:CC:05:43:F3:A8:A7:DC:AB:F3:58:0F
server, Dec 9, 2012, PrivateKeyEntry,
Certificate fingerprint (MD5):
0A:B3:EA:C0:09:9D:C2:8F:2A:40:DF:9A:81:AB:55:5B
```

That's all!