How to Configure Mutual Authentication using X.509 Certificate in SMP
SAP Mobile Platform (3.X)

Author: Ali Chalhoub
Global Support Architect Engineer

Date: July 2, 2015
Document History:

<table>
<thead>
<tr>
<th>Document Version</th>
<th>Authored By</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>Ali Chalhoub</td>
<td>First release of this guide</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Document Version</th>
<th>Reviewer</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>Kiran Kola</td>
<td></td>
</tr>
</tbody>
</table>
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>ABSTRACT</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Preparation</td>
<td>4</td>
</tr>
<tr>
<td>2. Getting Started</td>
<td>4</td>
</tr>
<tr>
<td>3. Configuring SMP 3.0</td>
<td>4</td>
</tr>
<tr>
<td>4. Troubleshooting</td>
<td>4</td>
</tr>
</tbody>
</table>

## PREPARATION

<table>
<thead>
<tr>
<th>Definitions</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Environment</td>
<td>5</td>
</tr>
<tr>
<td>Agenda</td>
<td>7</td>
</tr>
<tr>
<td>How Mutual Authentication Works in SMP 3.0</td>
<td>8</td>
</tr>
</tbody>
</table>

## GETTING STARTED

<table>
<thead>
<tr>
<th>Generating x.509 certificates and configuring the SAP Backend</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Mapping</td>
<td>10</td>
</tr>
<tr>
<td>Adjust Profile Parameters</td>
<td>12</td>
</tr>
<tr>
<td>Converting PFX certificate to p.12</td>
<td>13</td>
</tr>
</tbody>
</table>

## CONFIGURING SMP 3.0

<table>
<thead>
<tr>
<th>Import Certificates into SMP Key Store</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>Verifying the certificates in the keystore</td>
<td>23</td>
</tr>
<tr>
<td>Configuring SMP 3.0 OData Endpoint</td>
<td>25</td>
</tr>
<tr>
<td>Pinging the Endpoint</td>
<td>34</td>
</tr>
<tr>
<td>Installing the user certificate on the client browser</td>
<td>35</td>
</tr>
<tr>
<td>Generating SMP public certificate</td>
<td>42</td>
</tr>
<tr>
<td>Install the SMP 3.0 Public Certificate</td>
<td>50</td>
</tr>
<tr>
<td>Installing POSTMAN</td>
<td>54</td>
</tr>
<tr>
<td>Registering with SMP using X.509 user certificate</td>
<td>56</td>
</tr>
<tr>
<td>Verifying the Registration in SMP Cockpit</td>
<td>63</td>
</tr>
</tbody>
</table>

## TROUBLESHOOTING

<table>
<thead>
<tr>
<th>Clearing the Cache</th>
<th>64</th>
</tr>
</thead>
<tbody>
<tr>
<td>Response Code 404</td>
<td>64</td>
</tr>
<tr>
<td>Troubleshooting X.509 Client Certificate communication issues</td>
<td>64</td>
</tr>
</tbody>
</table>
ABSTRACT
This white paper provides the following:

1. Preparation
   1.1. Definitions
   1.2. Environment
   1.3. Agenda
   1.4. How Mutual Authentication Works between SMP 3.0 and SAP Gateway

2. Getting Started
   2.1. Generating x.509 certificates and configuring the SAP Backend
   2.2. User Mapping
   2.3. Adjust Profile Parameters

3. Configuring SMP 3.0
   3.1. Import Certificates into SMP Keystore
   3.2. Verifying the certificates in the Keystore
   3.3. Configuring SMP 3.0 OData endpoint
   3.4. Pinging the Endpoint
   3.5. Installing the user certificate on the client browser
   3.6. Install the SMP 3.0 Public Certificate
   3.7. Generating SMP public certificate
   3.8. Installing POSTMAN
   3.9. Registering with SMP using X.509 user certificate
   3.10. Verifying the Registration in SMP Cockpit

4. Troubleshooting
   4.1. Clearing the Cache
   4.2. Response Code 404
   4.3. Troubleshooting X.509 Client Certificate communication issues
Definitions

Before getting started, let’s explain few terms that we will be using in this white paper.

**SSL Handshake:** SSL is used to encrypt information between client(s) and server(s).

**X.509 Certificate Authentication:** It is basically used to verify the identity of the server when using SSL.

**Mutual Authentication:** It is a method of which a client must prove its identity when it communicates with the server, as well the server must prove its identity to the client before any traffic is sent between the client and the server. It is called two way identifications.

**Technical User:** An identity that represents a system not a person. The objective is to configure the backend to trust the SMP server that the user certificate is passing are valid certificates. It is about establishing the chain of trust.
Figure 1 - SSL Handshake Process

**SSL HANDSHAKE**

Client browser request a secure site https://www.domain.com

1. Client browser requests a secure website
2. It asks the server to identify its identity

<table>
<thead>
<tr>
<th>1</th>
<th>a. Client browser requests a secure website</th>
<th>b. It asks the server to identify its identity</th>
<th><a href="https://www.domain.com">https://www.domain.com</a></th>
</tr>
</thead>
</table>

| 2 | a. The server gets the requests and it knows the request for a secure website because it starts with https | b. The server will respond to the client and sends back a PUBLIC KEY and a CERTIFICATE that identify its identity |
|---|---|---|---|

2. The server gets the requests and it knows the request for a secure website because it starts with https
3. The server will respond to the client and sends back a PUBLIC KEY and a CERTIFICATE that identify its identity
How to Configure Mutual Authentication using X.509 Certificate in SMP

| 3 | a. The browser now receives the PUBLIC KEY and CERTIFICATE from the server  
   b. The client browser will inspect the certificate root against a list of trusted CAs (Certificate Authority) to confirm it was signed or issued by a trusted entity and to confirm the server identity. It verifies if the certificate expired or not and if the it belongs to the website that was requested  
   c. After everything is checked and everything is OK, the browser will generate a random symmetric key which is used to encrypted the actual data  
   d. Then the browser uses the server PUBLIC KEY to encrypt the payload |
|---|
| 4 | a. Now the server uses its own PRIVAT KEY to decrypt the payload  
   b. It fetches the symmetric session key  
   c. Server sends back the acknowledgement to the client that the server is ready to use the symmetric session key to start encryption of the data |
| 5 | The browser and the server will start encrypting and decrypting the data using the symmetric session key |

**Note:** Two way identifications: In Step 2 the server sends the CA Lists as well requesting the client certificate + Nonce (random bytes)

---

**Environment**

In this white paper we are using

1. SAP Mobile Platform 3.0 with SP07  
2. Open SSL  
3. KeyStore Explorer 4.1.1  
4. SAP Gateway OData Endpoint  
5. Windows 2008 Enterprise R2  
6. VMware Instance  
7. Chrome  
8. Firefox  
9. Postman
Agenda

In this white paper we are going to do the following

1. Generate a technical user certificate and sign it
2. Generate a user certificate that represents the client and sign it
3. Import the Root certificate and technical user into SMP keystore
4. Configure an Application in SMP to access the SAP OData Endpoint
5. Import the user certificate into browser personal certificates
6. Import the Root certificate into the Trusted Root Certification Authorities
7. Register with SMP using Postman

How Mutual Authentication Works in SMP 3.0

In this section, we will go through the steps of how Mutual Authentication works in SMP 3.0 and SAP Gateway

To do Mutual Authentication with SMP 3.0

1. Client connects on https 8082

2. SMP says: Here is my server certificate; I need to see your certificate and here is what are accepted as a CA. The client will validate the server certificate first – signed by a CA that is trusted by the device truststore. CA certs on the device are provisioned by the device manufacturer for BYOD, or can be managed by MDM. If the server certificate is signed by a trusted CA, is within its validity
period, and the Subject CN matches the host.domain of the https request URL, then the certificate is accepted.

3. Client picks a certificate from the local store that thinks is going to be acceptable based on the CA that the server can accept the client certificate that goes with its private key to decrypt the content. Standard HTTPS mutual certificate handshake ensues. During this handshake the server will verify the client’s certificate is signed by a trusted CA, within its validity period and (optional) check for certificate revocation via CRL or OCSP. If the certificate passes validation, we establish an initial authenticated JAAS Subject with a Principal name derived from the certificate Subject DN, and we create a Credential containing the X.509 client cert which is also attached to the Subject.

4. Now we got through the authentication portion. Cached the x.509 certificate in memory

5. Now SMP talking to Back-end (i.e SAP backend)

6. SMP is not a browser, the Administrator configure the certificate alias and uses that to complete the handshake. This is a “technical user” certificate that has been generated and signed by a CA the SAP backend server will trust, and has been imported into the SMP keystore (using keytool or whatever tools works for you) with a specific alias name that you provide in the Cockpit when you configure the endpoint connection to SAP backend. *NOTE* the password for this private key entry must be the same as the keystore password. JKS keystores allow for each private key entry to have its own password, but SMP doesn’t support this feature.

7. Backend sees SMP as a technical user and not the end user client. Now we completed the handshake

8. SMP is ready to forward the client request to the backend, but it knows that it should *try* to do SSO so the backend knows who the ultimate client user is. SMP knows how to do SSO using {X.509 user certificate, MYSAPSSO2 token, username/password}. It looks for user credentials (attached to the JAAS Subject) in that order. In this case we find the X.509 credential and will use that for SSO. The way that SSO is performed using X.509 is to add the SSL_CLIENT_CERT http header to the request, with the value as the base64 encoded client cert

**Note:** Netweaver Gateway and several other SAP server systems are aware of this SSL_CLIENT_CERT mechanism in cases where the client’s SSL has been terminated up-stream
GETTING STARTED

Generating x.509 certificates and configuring the SAP Backend

To digitally identify a particular individual client certificates are used. In general, certificates are issued by company's PKI (X.509 Public Key Infrastructure). In our case, to test our scenario we will create end user certificates using OpenSSL.

Installing and Configuring OpenSSL

First thing we need to do is to generate the SAP server side technical user certificate by doing the following

1. Download OpenSSL from the following URL:
   
   http://slproweb.com/download/Win64OpenSSL-1_0_0n.exe

2. Install OpenSSL, by default it gets installed under C:\OpenSSL-Win64\bin

3. Set the environment variable:
   
   set OPENSSL_CONF=c:\OpenSSL-Win64\bin\openssl.cfg

Creating the CSR file

1. Open the command prompt in Administration mode
2. Navigate to C:\OpenSSL-Win64\bin
3. Issue the following command:
   
   openssl req -sha256 -out SUPUSER.csr -new -newkey rsa:2048 -nodes -keyout server.key

4. Sign the certificate using your internal CA. In our test case, we are using our SAP internal CA
5. Once it is signed, save it to your local drive. In our test, we called the CSR file SUPUSER.CRT
6. After receiving the signed certificate, we need to convert it to pfx format by issuing:
   
   openssl pkcs12 -export -out SUPUSER.pfx -inkey server.key -in SUPUSER.crt

   Very Important: Enter password same as your SMP Keystore. In our test, we are using the default s3pAdmin

7. Repeat the above procedure to create the user certificate SSLUSER.pfx

User Mapping
How to Configure Mutual Authentication using X.509 Certificate in SMP

Using the SAP backend TCode do the following:

1. Go to SE11
2. Database Table: VUSERXTID and hit Display Click on Contents (CTRL+SHIFT+F10)
   
   **Dictionary: Display View**
   
   Maintenance View: VUSERXTID  Active
   Short Description: Assignment of External ID to Users

3. External ID Type: DN of certificate (DN) and clicked on continue

4. Now click on change

5. Now click on New entries

6. External ID should be the **Subject DN** Assign the user (Prior to this activity make user is already created).

7. Check Activated example:
   
   **Important:** Subject DN should match with the VUSERXTID entry as shown in the following screens below:
8. Repeat above steps for SSLUSER

Adjust Profile Parameters

Adding profile parameters for client verification and establishing trusted relationship between SAP Mobile Platform and ICM. In TCode enter:

1. RZ10
2. Set the AS ABAP profile parameter icm/HTTPS/verify_client to the value 1 (accept certificates) or 2 (require certificates) to support the use of client certificates
3. TCode: RZ10 For X.509-based logon to NW AS, you need following parameters to create a trusted relationship between the SMP Server and ICM

\[
\text{icm/HTTPS/trust_client_with_issuer} = \text{<Root Subject DN of the SMP Technical User>}
\]
How to Configure Mutual Authentication using X.509 Certificate in SMP

icm/HTTPS/trust_client_with_subject = <Subject DN of the SMP Technical User>

Example:

icm/HTTPS/trust_client_with_issuer = EMAIL=email@sap.com, CN=SAPNetCA, OU=SAPNet, O=SAPAG, C=DE

icm/HTTPS/trust_client_with_subject = CN=HOSTNAME, OU=COE, O=SAP-AG, C=DE

Please refer to the following KBA for more details: 2008296

Note: For any issues experienced in the backend, change the trace levels and check the logs. Please refer to this URL:
http://help.sap.com/saphelp_nw70ehp2/helpdata/en/48/3a062c902131c3e1000000a42189d/content.htm

Converting PFX certificate to p.12

In order to import the technical user into SMP keystore, we must convert the technical user to p.12 format. In our case we need to convert SUPUSER.PFX to SUPUSER.p12

One simple method is to use a browser that allows you to backup the certificate. In our example we are using Firefox:

1. Open the Certificate Manager in Firefox as shown below:

![Certificate Manager](image)

2. Click on Import
3. Browse to the technical user, in our case it is called SUPUSER.PFX
4. Enter the password of the certificate

5. Successful import message is displayed
6. Now export the certificate

7. Type the name which is in our case `SUPUSER.p12` and select the save as type (`*.p12`)

8. Repeat the same steps for `SSLUSER.pfx`
How to Configure Mutual Authentication using X.509 Certificate in SMP

CONFIGURING SMP 3.0

Import Certificates into SMP Key Store

Prerequisites:
1. Make sure you have generated a technical user certificate with p.12 format and signed by a trusted CA
2. Make sure you have generated a user certificate with p.12 format and signed by a trusted CA
3. Make sure the Technical user certificate has the same password as the SMP keystore password

Preparations and Importing Technical User

There are many ways to import the certificates into SMP keystore. In our white paper, I am using a KeyStore explorer 4.1.1

1. Backup the SMP 3.0 key store which is located in the following default location:
   C:\SAP\MobilePlatform3\Server\configuration\smp_keystore.jks

2. Install KeyStore Explorer 4.1.1 or higher on the machine where SMP 3.0 is installed
   Note: Keystore Explorer requires the following:
   a. Java 1.6 runtime
   b. Java Cryptography Strength. Follow the instructions on how to download it and install it

3. After keyStore Explorer is configured, you should see the following screen when it is run

   ![KeyStore Explorer Screenshot](image-url)
4. Click on Open an existing Keystore
5. Select `smp_keystore.jks` located under, 
C:\SAP\MobilePlatform3\Server\configuration, as shown below:

   ![Open KeyStore](image)

   Look-in: configuration
   
   Tree view:
   - com.sap.mobile.platform.server.agentry
   - com.sap.mobile.platform.server.cluster
   - com.sap.mobile.platform.server.introscope
   - com.sap.mobile.platform.server.launcher
   - com.sap.mobile.platform.server.mobiliser.core
   - com.sap.mobile.platform.server.security
   - etc
   - org.eclipse.core.runtime
   - org.eclipse.equinox.app
   - org.eclipse.equinox.simpleconfigurator
   - org.eclipse.osgi
   - org.eclipse.update
   - smp_keystore - backup

   File name: smp_keystore.jks
   Files of type: Keystore Files (* .ks, *.jks; *.pfx; *.p12)

6. Then click Open
7. You will be asked to type the Keystore password. In our example, it is `s3pAdmin` and click OK.

8. Once the keystore is opened, select from Tools, Import Key Pair (CTRL+K) as shown below:
How to Configure Mutual Authentication using X.509 Certificate in SMP

9. Select PKCS #12 as the key type to import

10. Click on OK to go to the import screen

11. Under Decryption password, type the password of the Technical user, in our example it is s3pAdmin and browse to the Technical user p.12 certificate. In our example it is called SUPUSER.p12, the
How to Configure Mutual Authentication using X.509 Certificate in SMP

screen should look like this below:

12. Click on Import

13. A giving alias will be used, in our case, we are giving the alias name as the certificate name, see below:

14. Click OK

15. You will be asked the type the password of the certificate. In our case it is the same as the keystore password which is s3pAdmin

16. You should see the following if everything was successful:

17. Click on OK
Importing CA or ROOT certificate into SMP key store

Same process that we used to import the technical user, but this time we are going to import the Root certificate that signed that Technical user certificate. To do that, do the following:

1. If the KeyStore Explorer is running from the previous step, go to Tools and select Import Trusted Certificate as shown below:

   ![KeyStore Explorer](image)

   ![Import Trusted Certificate](image)

2. Browse to the Root Certificate as shown below:

   ![Import Trusted Certificate](image)

3. Click on Import
4. You may get the following warning shown below, ignore it

5. Click OK

6. You will see the certificate details being displayed as shown below:

7. Once everything is verified, click on OK

8. You will be asked if you want to import the certificate, click Yes
9. You will be asked to give an alias, we are using the same alias as the certificate name as shown below:

![Certificate Entry Alias](image)

10. Click on OK

11. If everything goes well, you will see a success message indicating the import was successful

![Certificate Import Successful](image)

12. **IMPORTANT**: Keep the KeyStore Explorer open and do not close it, because we need to save our changes later

**Verifying the certificates in the keystore**

After successfully importing the Trusted Root certificate and the Technical user certificate into the keystore we need to verify if the technical user is signed by the Root certificate that got imported and both certificates are in the keystore, todo that, do the following:

1. If the keyStore Explorer is still opened from the previous step, verify if you see the two certificates aliases as shown below:

![KeyStore Explorer](image)
2. Next Verify if the Technical user certificate is signed by the ROOT certificate by double clicking on supuser, you should see the following:

   ![Certificate Details for Entry 'supuser']

   - Certificate Hierarchy:
     - SAPNetCA
     - SUPUSER

   - Version: 3
   - Subject: [Certificate Details]
   - Issuer: [Certificate Details]
   - Serial Number: 0x:1000000
   - Valid From: 04/May/1990 13:56:34 CEST
   - Valid Until: 18/Jul/2015 14:00:00 CEST
   - Public Key: RSA 1024 bits
   - Signature Algorithm: SHA-1 with RSA

   ![OK]

3. Notice SUPUSER is signed by SAPNetCA ROOT certificate
4. Click on OK
5. Now click on SAVE icon to save all the changes as shown below:

   ![smp_keystore.jks - KeyStore Explorer 4.1.1]

6. Once everything is saved to the Keystore, close KeyStore Explorer and it is time now to configure SMP 3.0 OData Endpoint
Configuring SMP 3.0 OData Endpoint

Before doing any configuration to the SMP 3.0, we need to restart the SMP 3.0 server, not the physical machine or VM, for the changes we have done to the keystore take effect.

1. After restarting the SMP server, open the Cockpit by going to the following URL:
   https://localhost:8083/Admin using Chrome

2. You may see this screen below:

   ![SMP Cockpit Connection Error]
   
   **Your connection is not private**

   Attackers might be trying to steal your information from localhost (for example, passwords, messages, or credit cards). NET:ERR_CERT_AUTHORITY_INVALID

   ![SMP Cockpit Connection Error]
   
   This server could not prove that it is localhost; its security certificate is not trusted by your computer's operating system. This may be caused by a misconfiguration or an attacker intercepting your connection.

   ![SMP Cockpit Connection Error]

   **Proceed to localhost (unsafe)**

3. Click on Proceed to localhost (unsafe)
4. Login to the Cockpit using `smpAdmin` user and password, by default password is `s3pAdmin`.

![Login Screen](image)

5. After login successfully to SMP Cockpit, you should see the following screen below:

![SMP Cockpit Screen](image)

6. To create an application, click on the Applications Rectangle box.
How to Configure Mutual Authentication using X.509 Certificate in SMP

7. Click on New button

8. Type the ID of the Endpoint application and the name, see the sample below:

   **New Application**
   
   - **ID**: com.sap.flight
   - **Name**: SAP Flight Sample
   - **Vendor**: SAP AOS
   - **Type**: Native
   - **Description**: Sample used to do Mutual Authentication

9. Click Save, and you should see the following:

10. First step here we need to provide the OData backend Endpoint. In our example we are using the Flight OData endpoint: Please refer to the following configuration below:
11. Enter the Endpoint as shown below and the alias:

```
APPLICATION
com.sap.flight

ABOUT
Name: SF Flight Sample
Description: Example used to do Mutual Authentication
Type: Native
Version: SAP ABAP
Created on: 25/06/2019
Push URL: 
Change

STATUS
1 Security profile is empty or invalid
2 Endpoint is empty or invalid
3 At least one SSO Mechanism required

OVERVIEW | BACK END | AUTHENTICATION | CLIENT POLICY | PUSH | CLIENT RESOURCES | OFFLINE
--- | --- | --- | --- | --- | --- | ---
Endpoint: https://<domain>:<port>/sap/opu/odata/IWBEP/RMTSAMPLE/FLIGHT_2/?$format=xml

- Use System Proxy
- Allow anonymous access

Maximum Connections: 500
Certificate alias: SUPUSER
Rewrite Mode: Rewrite URL in SMP
Relative Path:

SSO Mechanisms
Add
Type
Delete

No data

Back-end Connections: (0)

12. Next click on Add under SSO Mechanisms and select X.509
13. Click Save, you should see the following:

**SSO Mechanism Changes**

![Alert icon]

SSO Mechanism successfully added. Please press Save to update your changes.

14. You should have the following configuration so far:
15. Next we need to add the Authentication Security Profile, click on AUTHENTICATION tab.

SECURITY PROFILE

*Profile Name: Enter New Profile

- Check Impersonation

AUTHENTICATION PROVIDERS

<table>
<thead>
<tr>
<th>Add</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
</tr>
<tr>
<td>X.509</td>
</tr>
</tbody>
</table>

No data
16. Under profile name give a name to the new security profile. Example: authflightmu as shown below and click on Add

17. After clicking on Add, you should see the following screen:

18. Select x.509 User Certificate and click Save as shown below:
19. If everything goes well, you should see the following:

**Authentication Provider Changes**

- Authentication Provider successfully added. Save your Security Profile to Update your changes.

20. Click on OK
21. Now click on Save

22. Once you clicked on Save, you are asked to confirm the Update. Click on Yes
23. Now we should have something like this screen below:

![APPLICATIONS screen](image)

<table>
<thead>
<tr>
<th>Application ID</th>
<th>Name</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>com.sap_flight</td>
<td>SAP Flight Sample</td>
<td>Native</td>
</tr>
</tbody>
</table>

**Pinging the Endpoint**

Next step is to test the OData endpoint and make sure SMP server can ping it. To ping the end point do the following:

1. Select the Application name that got created in the previous step as shown below:

![APPLICATIONS screen](image)

<table>
<thead>
<tr>
<th>Application ID</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>com.sap_flight</td>
<td>SAP Flight Sample</td>
</tr>
</tbody>
</table>

2. Once the application is selected, click on Ping button as shown below:

![APPLICATIONS screen](image)

3. If everything is configured correctly and the Endpoint hostname is reachable, you should see a success message, like the one below:

![Ping status](image)

<table>
<thead>
<tr>
<th>URL</th>
<th>Profile</th>
<th>Status</th>
<th>Status Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>https://&lt;endpointhostname&gt;</td>
<td>Backend profile</td>
<td>✔️</td>
<td>Backend system reached successfully</td>
</tr>
</tbody>
</table>

4. Close the Ping status dialog message by clicking on the X on the top right corner
Installing the user certificate on the client browser

After configuring SMP server and getting the Endpoint to ping successfully, next step now is to add the user certificate to the browser.

1. On the VM or client machine where you are going to test the Mutual Authentication, navigate to the location where the user p.12 certificate is located.

2. Open Internet Option and select the Content Tab as shown below:
3. Click on Certificates
4. Make sure the Personal tab is selected as shown below:

![Certificates dialog box]

5. Under the Personal Tab, click on **Import**...
6. Click Next on the first screen, until you go to the File to Import screen as shown below:

File to Import

Specify the file you want to import.

File name: [ ]

Browse...

Note: More than one certificate can be stored in a single file in the following formats:

- Personal Information Exchange- PKCS #12 (.PFX, .P12)
- Cryptographic Message Syntax Standard- PKCS #7 Certificates (.P7B)
- Microsoft: Serialized Certificate Store (.SST)

Learn more about certificate file formats

< Back  Next >  Cancel

7. Browse to the User p.12 certificate as shown below:
8. You should see the following:

![Certificate Import Wizard](image1)

Note: More than one certificate can be stored in a single file in the following formats:

- Personal Information Exchange - PKCS #12 (.PFX, .P12)
- Cryptographic Message Syntax Standard - PKCS #7 Certificates (.P7B)
- Microsoft Serialized Certificate Store (.SST)

Learn more about [certificate file formats](#).

9. Click Next

![Certificate Import Wizard](image2)

Type the password for the private key.

Password:

```
********
```

- [ ] Enable strong private key protection. You will be prompted every time the private key is used by an application if you enable this option.
- [ ] Mark this key as exportable. This will allow you to back up or transport your keys at a later time.
- [x] Include all extended properties.

Learn more about [protecting private keys](#).
10. Type the User certificate password, and click **Next**
11. You should see the following screen:

![Certificate Import Wizard](image)

- **Certificate Store**: Certificate stores are system areas where certificates are kept.

  - Windows can automatically select a certificate store, or you can specify a location for the certificate.
  - Automatically select the certificate store based on the type of certificate
  - Place all certificates in the following store

  - **Certificate store**: Personal

  - Learn more about certificate stores

12. Make sure the Place all certificates in the following store is set for **Personal**, and click **Next**
13. Once it is done, click Finish as shown below
14. You should see the following confirmation, click OK:

The import was successful.

15. Now verify if the certificate did get imported as shown below:
16. As you can see SSLUSER certificate got imported successfully. Click on Close
17. Click on OK to close the Internet Option properties

Generating SMP public certificate

SMP 3.0 has a self-signed certificate that gets generated during the installation of the server. In production, you need to create CSR for the SMP server and sign it. Usually same Root Certificate that signed by the the
technical user. In our test, we are going to use the self-signed certificate. To generate the public certificate do the following:

1. Using Chrome, access the Cockpit by typing https://localhost:8083/Admin
2. You will get the following screen:

![Log On Screen](image)

3. Now click on the https exception as shown below
How to Configure Mutual Authentication using X.509 Certificate in SMP

This site uses a weak security configuration (SHA-1 signatures), so your connection may not be private.

The identity of this website has been verified by site owner's private key but does not have public audit records.

The site is using outdated security settings that may prevent future versions of Chrome from being able to safely access it.

Certificate information

Your connection to dewdhwsbs5505.emea.global.corp.sap is encrypted with obsolete cryptography.

The connection uses TLS 1.2.

The connection is encrypted using AES_128_CBC, with SHA1 for message authentication and DHE_RSA as the key exchange mechanism.

Site information

You first visited this site on Jun 25, 2015.

What do these mean?
4. Click on Certificate Information

This certificate is intended for the following purpose(s):
- All issuance policies
- All application policies

- Issued to: [Redacted]
- Issued by: [Redacted]
- Valid from 6/24/2015 to 6/23/2017

5. Click on Details tab
6. Click on Copy to File… as shown below

7. Click Next on the Welcome screen
8. Select Base-64 Encoded X.509 (CER)

Select the format you want to use:

- DER encoded binary X.509 (.CER)
- Base-64 encoded X.509 (.CER)
- Cryptographic Message Syntax Standard - PKCS #7 Certificates (.P7B)
  - Include all certificates in the certification path if possible
- Personal Information Exchange - PKCS #12 (.PFX)
  - Include all certificates in the certification path if possible
- Delete the private key if the export is successful
- Export all extended properties
- Microsoft Signed Certificates Store (.SST)

Learn more about certificate file formats

9. Click Next
10. Provide a file name of the certificate by clicking on browse and select a location.
11. Once done click Finish

12. If the export is successful, you will see the following screen:

   The export was successful.

13. Click on OK
Install the SMP 3.0 Public Certificate

After generating the Base 64 encoded certificate, now we need to install it to the browser into the Trusted Root Certification Authorities. To do that, follow the following steps:

1. Navigate to the location where you exported the public certificate
2. If you saved it with extension ".cer", double click on it, you will see the following:

   ![Certificate Information](image)

   - **Issued to:** [Redacted]
   - **Issued by:** [Redacted]
   - **Valid from:** 6/24/2015 to 6/23/2017

3. Click on Install Certificate…
4. Click Next on the Welcome screen
5. Select Place all certificates in the following store and click Browse

6. Select Trusted Root Certification Authorities

7. Click OK
8. Click Next

- Certificate Import Wizard

**Certificate Store**

Certificate stores are system areas where certificates are kept.

Windows can automatically select a certificate store, or you can specify a location for the certificate.

- Automatically select the certificate store based on the type of certificate
- Place all certificates in the following store

Certificate store:

[Text field: Trusted Root Certification Authorities]

Learn more about certificate stores.

[Buttons: < Back, Next >, Cancel]
9. Now click Finish

10. Public self-signed certificate is now installed to the browser

11. Click OK to close confirmation message

12. Click OK to close the Certificate window
**Installing POSTMAN**

In order to use POSTMAN, we need to use Chrome.

1. Open Chrome browser and navigate to this URL:  
   https://chrome.google.com/webstore/detail/postman-rest-client/fdmmgilgnpjgdojojpjoooidkmcomcm?hl=en

2. Click on ADD TO CHROME

3. You should see the following screen below:

   Confirm New App

   Add "Postman - REST Client"

   It can:
   - Read and change all your data on the websites you visit

   Add  Cancel

4. Click on Add
5. After Postman is added, you will be able to access it, by typing the following in the URL:

```
chrome://apps/
```

6. Now click on Postman – REST

7. You should see the following:
Registering with SMP using X.509 user certificate

After successfully installing POSTMAN, now we are ready to use it and test our Mutual Authentication configuration and register with SMP. To register with SMP using POSTMAN, do the following:

1. Under POSTMAN, make sure Normal tab is selected

2. In order to register with SMP, we need to provide the URL for SMP and application Endpoint as well, we need to use port 8082 for Mutual Authentication. Enter the following configuration below:

   URL: https://<SMP-Fully-Qualified-Domain>:8082/odata/applications/latest/com.sap.flight/Connections

   8082: Used to do Mutual Authentication with SMP
   com.sap.flight: is the application name we created in SMP. It MUST match what we created in SMP

3. POSTMAN configuration should look like this

4. Next we need to change the method from GET to POST, do the following as shown below
5. Once POST is selected, now we need to change the submission from Form to RAW, see below:

6. You should see the following:

7. Enter the following XML code in the body as shown below and change

```xml
<?xml version="1.0" encoding="UTF-8"?>
xmlns="http://www.w3.org/2005/Atom"
  <content type="application/xml">
    <m:properties>
      <d:DeviceType>Windows</d:DeviceType>
    </m:properties>
  </content>
</entry>

Note: Make sure the xml:base matches your URL where it says <your-domain-URL>
8. After typing this in the body, change the Text type to XML as shown below:

```
8. After typing this in the body, change the Text type to XML as shown below:
```

9. Finally we should have the following:

```
9. Finally we should have the following:
```

10. Next we need to configure the Basic Auth Tab, click on Basic Auth as shown below:

```
10. Next we need to configure the Basic Auth Tab, click on Basic Auth as shown below:
```

11. Click on Refresh Headers

12. You should see this screen below:
13. Change Authorization to the content type as shown below:

<table>
<thead>
<tr>
<th>Normal</th>
<th>Basic Auth</th>
<th>Digest Auth</th>
<th>OAuth 1.0</th>
<th>No environment</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

```
https://BENIGNNODE.Excelglobal.corp:8002/odata/application: POST

Authorization
Basic Og==
Header
Value
```

```
https://BENIGNNODE.Excelglobal.corp:8002/odata/application: POST

Content-Type application/atom+xml; charset=utf-8
Header
Value
Header
Value
```

```
<?xml version="1.0" encoding="UTF-8"?>
      content-type="application/atom+xml; charset=utf-8"
      m:properties>
  <d:DeviceType>Windows</d:DeviceType>
</entry>
```
14. Now we are ready to register, click on the Send Button a pop up window should be displayed asking you to select the user x.509 Certificate as shown below:

```
<xml version="1.0" encoding="UTF-8"?>
 xmlns:d="http://schema.org">
<properties>
<DeviceType>Windows</DeviceType>
</content>
</entry>
```
15. Select the User Certificate, in my case it is SSLUSER which is already selected. Click on OK

16. As you can see we are able to register with SMP using Mutual Authentication with x.509 user certificate. Now to access the end point, use the following URL and change the method to GET and use the Application ID that is highlighted. So your new configuration should look like this:

URL: https://<YOUR-DOMAIN>:8082/com.sap.flight
Method: GET
Header: X-SMP-APPCID and value, the d03b9056-eb27-40f0-990b-79e15d42cc17

17. Now click Send to fetch the Flight entities as shown below:
18. As you can see now, this indicates, we were able to access the Flight OData endpoint using Mutual Authentication in SMP 3.0
Verifying the Registration in SMP Cockpit

To verify and check the application registration in SMP 3.0 Cockpit, do the following:

1. Login to the SMP Cockpit
2. Click on Registrations
   
   ![SMP Cockpit Registration](image)

3. You should see the following X-SMP-APPCID which is the Registration ID value below:

   **Registrations:** (1)

<table>
<thead>
<tr>
<th>Filter by</th>
<th>Registration ID</th>
<th>Application ID</th>
<th>Device Type</th>
<th>User Name</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>00369516-eb27-4000-920b-79e135426c17</td>
<td>com.sap.flight</td>
<td>Windows</td>
<td></td>
</tr>
</tbody>
</table>

4. This will show we have completed the registration and we were able to fetch all the OData Endpoint Entities
**Clearing the Cache**

When working with POSTMAN, if you started to experience unexplainable behavior, clear the cache and redo your task.

**Response Code 404**

If you received from the backend a response code 404, but you are able to access the backend OData Endpoint directly from the browser, that means you missed adding the profile parameters. Please refer to the following KBA: [2008296](http://help.sap.com/saphelp_nw70ehp2/helpdata/en/48/3a062c902131c3e10000000a42189d/content.htm)

**Troubleshooting X.509 Client Certificate communication issues**

Symptoms: SAP Gateway and SAP Mobile Platform is configured to use X.509 certificates to authenticate backend gateway. If you receive certificate authentication issues, then increase your backend Internet Communication Manager (ICM) trace to level 2 for retrieving detailed SSL debug information.

For more information refer following link:

[http://help.sap.com/saphelp_nw70ehp2/helpdata/en/48/3a062c902131c3e10000000a42189d/content.htm](http://help.sap.com/saphelp_nw70ehp2/helpdata/en/48/3a062c902131c3e10000000a42189d/content.htm)