Use of Certificates in Interactions between a Browser, Web Application, and Web Service

Purpose

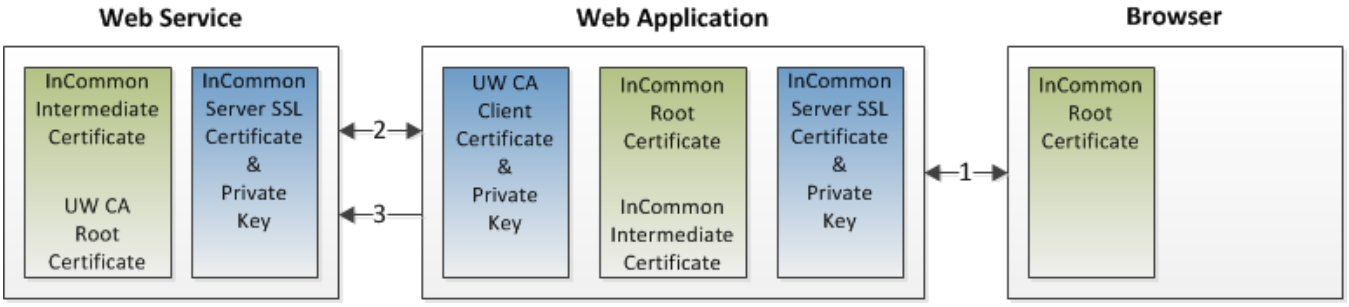
This page describes the most common scenario for use of certificates where a web application has browser clients and also relies on a UW enterprise web service for data integration. Multiple certificates are at play in this scenario and the role of each certificate is often a point of confusion. The goal of this page is to provide clarification on this topic.

Scenario

There are three major components in this scenario (see **Figure 1**):

- **Web Service** - A UW Enterprise Web Service (e.g. Person Web Service, IDCard Web Service, Financial Web Service) that provides data or other services to a Web Application via supported APIs. The Web Service requires secure connections over SSL and also requires relying applications to authenticate themselves to the Web Service with a client certificate.
- **Web Application** - Software on a web server that provides functionality to the Browser client and also relies on a UW Enterprise Web Service to provide data or other services required for the application. The Web Application communicates with Browser clients and a Web Service securely over SSL.
- **Browser** - The end-user's web browser which is used to interact with the Web Application.

Figure 1. Certificates used and interactions between a Web Service, Web Application, and Browser. Blue boxes enclose certificates and corresponding private keys specifically issued to a component for its own use. Green boxes enclose other certificates that must also be installed in the certificate store of a component in order to enable interactions with other components. See text for description of interactions represented as numbered arrows in the diagram.



Interactions

In this scenario each component must have one or more certificates installed to support secure communications over SSL and/or client authentication using a certificate. There are three primary interactions (see numbered arrows in **Figure 1**).

1. Secure communication between the Browser and Web Application using SSL
    a. The Web Application uses its own InCommon Server SSL certificate, corresponding private key, and the InCommon Intermediate Certificate when establishing secure communications with the Browser.
    b. The Browser will only trust the Web Application if it can find the corresponding InCommon Root Certificate in its local certificate store and can verify the trust chain from InCommon Server SSL Certificate -> InCommon Intermediate Certificate -> InCommon Root Certificate.
    c. All current browsers ship with the InCommon Root Certificate, so there should be no issues establishing the trust chain.
    d. If the Web Application uses a server SSL certificate from another CA (not InCommon) then the browser must have the root certificate for that CA installed.
2. Secure communication between the Web Application and Web Service using SSL
    a. The Web Service uses its own InCommon Server SSL certificate, corresponding private key, and the InCommon intermediate certificate when establishing secure communications with the Web Application.
    b. The Web Application will only trust the Web Service if it can find the corresponding InCommon Root Certificate in its local certificate store and can verify the trust chain from InCommon Server SSL Certificate -> InCommon Intermediate Certificate -> InCommon Root Certificate.
    c. Depending on the host OS and development environment, the InCommon Root Certificate might or might not be available to the Web Application in order to verify the trust chain. Some examples:
        i. .Net applications running on Internet Information Server rely on CA root certificates in the Windows Certificate store. This store is automatically updated by Windows Update and it includes the InCommon Root Certificate.
        ii. PHP and Python applications rely on the CA certificates file provided by OpenSSL. These files must be updated manually if the file version that came with the installation did not already contain the InCommon Root Certificate. See the documentation for your version of PHP or Python.
        iii. Java applications rely on the Java keystore to provide CA root certificates. If the keystore that shipped with the Java installation did not include the InCommon Root Certificate then it will need to be updated using Java's "keytool" utility. See your Java documentation.
    d. If the Web Service uses a server SSL certificate from another CA (not InCommon), then the Web Application will need to have the corresponding CA root certificate available in its certificate store. For example, the Groups Web Service (GWS) uses the UW CA so a Web Application that uses GWS will need to have the UW CA root certificate installed.
3. Client certificate authentication of the Web Application to the Web Service

a. UW Enterprise web services require client certificate authentication based on certificates issued by the UW CA.
b. The Web Application will use its own UW CA client certificate and private key while authenticating to the Web Service.
c. The Web Service will use its local copy of the UWCA Root Certificate and local authorization rules to determine if the Web Application will be granted access.

Certificate Names and Download Locations

The certificates enclosed in green boxes in **Figure 1** have the following Common Names (CN):

- InCommon Root Certificate = "AddTrust External CA Root"
  - Download from Comodo
- InCommon Intermediate Certificate = "InCommon Server CA"
  - Download along with your InCommon Server SSL certificate from UW Certificate Services
- UWCA Root Certificate = "UW Services CA"
  - Download from the UW Certificate Authority