Understanding InCommon SSL certificate chains

[Version 1.1 – 10/5/2014 updated serial number for InCommon RSA Server CA which was reverted by InCommon/Comodo from sha512 to sha384 due to compatibility concerns.

Version 1.0 – 9/23/2014 initial publication]

In most SSL implementations:

- the client (e.g. web browser) automatically trusts the Root CA certificate (considered a "well-known root")
- the server hosting the SSL certificate (the End-Entity Certificate) presents both the Intermediate (InCommon Server CA) and the End-Entity Certificate
- the browser then "chains" the End-Entity Certificate to the Intermediate and then to the Root.

In the above scenario, it's always a good thing to have both the Root CA certificate **and** the Intermediate Certificate located on the web server's local certificate store.

Current chain (for SHA-1 certificates):
AddTrust External CA Root [this is referred to as a "Root CA" Certificate]
InCommon Server CA [this is referred to as an "Intermediate CA Certificate"]
End-Entity Certificate (i.e., for an xyz.colostate.edu domain name)

In simple terms, "AddTrust External CA Root" signs "InCommon Server CA" which signs "End-Entity Certificate".

New chain (for SHA-2 certificates):

The new chain for SHA-2 certificates is more complicated, due to legacy/compatibility issues. Three new certificates, unseen before, are now involved:

1) **USERTrust RSA Certification Authority** [this is a "Root CA" Certificate, issued by USERTrust RSA Certification Authority – i.e., self-signed by the CA itself. Serial = 01 fd 6d 30 fc a3 ca 51 a8 1b bc 64 0e 35 03 2d]
2) **USERTrust RSA Certification Authority** [this is an "Intermediate CA Certificate" and is issued by AddTrust External CA Root. Serial = 13 ea 28 70 5b f4 ec ed 0c 36 63 09 80 61 43 36]
3) **InCommon RSA Server CA** [this is a new "Intermediate CA Certificate". Serial = 47 20 d0 fa 85 46 1a 7e 17 a1 64 02 91 84 63 74]

Note that #1 and #2 above are **named the same**, but are for different purposes.

Scenarios:

1) For a client that trusts the new SHA-2 root, the server must provide one Intermediate CA Certificate necessary for the client to build a chain to the new root.

   USERTrust RSA Certification Authority [this is the Root CA, already trusted by the browser]
   Serial = 01 fd 6d 30 fc a3 ca 51 a8 1b bc 64 0e 35 03 2d

InCommon RSA Server CA [this is the Intermediate CA]
Serial = 47 20 d0 fa 85 46 1a 7e 17 a1 64 02 91 84 63 74

(The chain is End-Entity-Certificate→ InCommon RSA Server CA→ USERTrust RSA Certification Authority)

2) For a client that does NOT yet trust the new SHA-2 root, the server must provide two Intermediate CA Certificates necessary for the client to build a chain to the old (SHA-1) root.

AddTrust External CA Root [this is the Root CA, already trusted by nearly all browsers]
Serial = 01
USERTrust RSA Certification Authority [in this scenario, this is Intermediate CA #1]
Serial = 13 ea 28 70 5b f4 ec ed 0c 36 63 09 80 61 43 36
InCommon RSA Server CA [this is Intermediate CA #2]
Serial = 47 20 d0 fa 85 46 1a 7e 17 a1 64 02 91 84 63 74

(The chain is End-Entity-Certificate→ InCommon RSA Server CA→ USERTrust RSA Certification Authority→AddTrust External CA Root)

This is referred to as cross-certification because it is a signing of one root certificate by another. It's needed to build the chain up to the older trusted root for older clients/servers that don't automatically trust the "USERTrust RSA Certification Authority signed by USERTrust RSA Certification Authority" Root Certificate.

Guidance from Comodo/InCommon:

For a client running IE or Chrome on a version of MS Windows newer than XP SP2, the "USERTrust RSA Certification Authority" root will be pulled into the root store on first use and the shorter chain (Scenario 1 above) will be what the browser builds. Clients running Chrome on linux or Safari or FireFox will not yet trust the "USERTrust RSA Certification Authority" root, and so will build the longer chain (Scenario 2 above).

Comodo expects the new root to be present in FireFox, Chrome on Linux, and Safari by the end of 2014, although they recommend to keep using the chain back to the legacy root (Scenario 2 above) for some years to come to catch the long-tail of older clients.

In Sum:

If you completely control the client environment in which you are utilizing InCommon SSL certificates, you can likely get away with only providing the InCommon RSA Server CA Intermediate CA Certificate from your server (Scenario 1).

If you don't completely control the client environment in which you are utilizing InCommon SSL certificates, you should provide both the USERTrust RSA Certification Authority (Serial = 13 ea 28 70 5b f4 ec ed 0c 36 63 09 80 61 43 36) **and** the InCommon RSA Server CA Intermediate CA Certifficate (Scenario 2).