KeysAndCertificates

The most confusing part of installing Shibboleth and making it work is the use of public key technology to secure the system. Many Web SSO packages use symmetric/shared keys because they are simpler to understand and use. But Shibboleth is designed to work in more varied environments with less trust between parties, and only public key cryptography can easily support this.

The problem is partly just the volume and variety of keys and certificates involved in a typical deployment. Some don't even have anything to do with Shibboleth directly. Many people trying the software have never even installed a real SSL certificate before. This is like learning to ride a bike by riding a motorcycle; it won't go well.

There are lots of places to find help on SSL and PKI in general. Some example links:

- Apache SSL/TLS FAQ
- OASIS PKI Resources
- Guideline for Implementing Cryptography in the Federal Government

---

So, let's focus on Shibboleth itself. By keypair, we mean a private key and public key certificate as a unit. Here are all the potential keypairs in a typical setup:

Identity Provider

Part of Shibboleth

- XML Signing Keypair for SAML Services

This is used to apply a signature to XML messages. The IdP normally requires this when using the BrowserPOST profile, which is the default authentication profile in current versions. This keypair must be trusted by SPs and may need to be signed by a federation-approved CA. It could be the same as the SSL keypair described next, but is configured very differently.

In current versions, signing keypairs are configured in *idp.xml* in the `<Credentials>` element. Each potential keypair is assigned an `Id` . The `<RelyingParty>` element(s) point at the keypair to use when signing XML for those SPs. One keypair might be used for everything, or specific keypairs might be used with particular SPs or groups.

- SSL Server Keypair for SAML Services

This is used to authenticate the SAML server "endpoint" that handles HTTP connections and SOAP messages from SPs. Because of limitations in Apache and Tomcat support for SSL, these services usually run on a different host address or port from the parts of Shibboleth that browsers use. We usually suggest an alternate IP address or using port 8443. This keypair must also be trusted by SPs and may sometimes need to be signed by a federation-approved Certificate Authority.

In current versions, Apache is usually used as the SSL "front-end" for the SOAP endpoint. SSL keypair information is controlled by Apache mod_ssl commands. As of ShibOnedotThree, the Apache configuration no longer controls authentication of clients and this is done by the Java code itself. This means that Apache no longer needs to have a master list of acceptable CAs.

Separate from Shibboleth

- SSL Server Keypair for Browser-Facing Services

Shibboleth expects users to login somehow to prove their identity before issuing assertions to SPs. This normally takes place using SSL to encrypt any secret information like a password sent to the SingleSignOnService and authenticate the server to prevent phishing. This use of SSL is not part of Shibboleth and you are free to use any keypair you wish. Often a commercial certificate is used to prevent browser warnings. Since this keypair may not be trusted by SPs, the browser-facing and SOAP endpoints of the IdP are often run with different keypairs using a different address or port.

Service Provider

Part of Shibboleth

- SSL Client Keypair for SAML Services

Shibboleth is different than a lot of software because it uses SSL for both client and server authentication. Client authentication has always been part of SSL but is almost never used and is not well-documented or understood. In current versions, SOAP messages are often sent from an SP to an IdP using SSL to authenticate both ends. The SP needs a keypair trusted by the IdP to prove its identity, sometimes signed by a federation-approved CA. It could be the same as the SSL keypair described next, but may be configured very differently.

In current versions, SSL client keypairs are configured in ShibbolethXml_ in the `<Credentials>` element. Each potential keypair is assigned an `Id` . The `<CredentialUse>` element points at the default TLS keypair to use and might contain `<RelyingParty>` elements that override the keypair to use for particular IdPs.

- XML Signing Keypair for SAML Requests

A less used feature of ShibOnedotThree is the ability for a SP to sign SAML messages just as IdPs often do. To support this, a keypair trusted by IdPs would be used, possibly also signed by a federation-approved CA. It could be and usually is the same as the TLS/SSL client keypair described earlier. It is configured in a similar manner in ShibbolethXml

The main reason this feature is less used is because SSL can be much faster than signing due to the session caching provided by most SSL libraries. It's also not fully supported yet by the IdP for all the possible uses of client signing.

Separate from Shibboleth

- SSL Server Keypair for Browser-Facing Services

The SAML profiles that Shibboleth is based on assume that SPs use SSL on at least some of the web server that is being secured. We strongly recommend use of SSL across all of a web site because Shibboleth sessions are ultimately based on HTTP cookies. Stealing cookies is much easier if they travel in the clear. The strongest authentication technology in the world is reduced to a simple cookie. This is all you've got left. Consider why you're bothering to install Shibboleth at all before going further.

If you insist on not using SSL everywhere the session is used, you should still use SSL to protect the AssertionConsumerService locations that the Shibboleth software exposes and uses to accept incoming sessions.

This use of SSL is a browser-facing keypair, and you are free to use any keypair you wish. Often a commercial certificate is used to prevent browser warnings. Configuration of this keypair depends on your web server and you should use its documentation to help you.