Key Usage

Uses of Keys in Federation Metadata

Public keys are bound to X.509 certificates in metadata. These keys are used for message-level signing and encryption, and for back channel exchanges over TLS.

> **Use of TLS Certificates**
>
> In addition to message-level signing and encryption, X.509 certificates in metadata are used for TLS back-channel SOAP exchanges, typically on a nonstandard port such as 8443. These certificates are **not** the same as and have nothing to do with TLS server certificates used for browser-facing transactions over port 443. The latter type of TLS certificates are **not** contained in metadata.

Terminology

Definition. A *role descriptor* is a metadata element whose type is based on the SAML **md:RoleDescriptorType** type.

Examples of role descriptors familiar to site administrators include `<md:IDPSSODescriptor>`, `<md:AttributeAuthorityDescriptor>`, `<md:SPSSODescriptor>`, and so forth.

We will use the following standard key usage terminology:

- A *signing key* generally refers to a key pair used in conjunction with XML Signature. The private key is used to sign an XML node (such as a SAML response) while the corresponding public key is used to verify the signature. The latter is sometimes referred to as a *verification key*.
- An *TLS key* generally refers to a key pair used during a back-channel exchange, usually a SOAP exchange (such as artifact resolution or attribute query). Such a key may be used for TLS client-server authentication.
- An *encryption key* generally refers to a key pair used in conjunction with XML Encryption. The public key is used to encrypt an XML node (such as a SAML assertion) while the corresponding private key is used to decrypt the ciphertext. (That's an over-simplification of XML Encryption, but it will suffice in what follows.) The latter is sometimes referred to as a *decryption key*.

A single key may be used for multiple purposes as we shall see.

Types of Keys in Metadata

Recall that there are three types of key descriptors in SAML metadata:

1. `<md:KeyDescriptor use="signing">`
2. `<md:KeyDescriptor use="encryption">`
3. `<md:KeyDescriptor>`

A type 1 key is used for both signing and TLS. That is, the key is used to provide authenticity and integrity but not necessarily confidentiality.

> **The Actual Use of a Type 1 Key**
>
> When used for signing, a type 1 key in metadata is actually a verification key, not a signing key. The private signing key is held securely by the signing entity.

A type 2 key is used for encryption only, that is, the key is used to provide confidentiality.

Since the `use` XML attribute is missing on a type 3 key descriptor, such a key may be used for all of the above, that is, for signing, TLS, and encryption.

> **Recognizing a TLS Key in Metadata**
>
> Any `<md:KeyDescriptor>` element in metadata that has either a `use="signing"` attribute or no `use` attribute whatsoever is intended for use with TLS.

Keys in IdP Metadata

In the InCommon Federation, IdP metadata typically contains two role descriptors: an `<md:IDPSSODescriptor>` element and an `<md:AttributeAuthorityDescriptor>` element. Normally, each role descriptor contains a single type 1 key descriptor (with `use="signing"` XML attribute). Although not required, the two key descriptors almost always contain the very same key.

For an IdP, certificate migration is the controlled phasing in of a new type 1 key descriptor, no more or less.

Keys in SP Metadata

There is just one role descriptor for SPs in Federation metadata, namely, the `<md:SPSSODescriptor>` element. Under normal circumstances, this role descriptor contains a single type 3 key descriptor (with no `use` XML attribute).

For an SP, certificate migration is more complicated than for an IdP. This is because two decryption keys need to be configured in the SP software at one time.

Most SAML V2.0 IdPs are configured to encrypt assertions sent to the SP. It's important, therefore, that an encryption key always be available to a SAML V2.0 IdP, and so SP metadata must always contain an encryption key.

> If an SP supports SAML V2.0, there MUST be **at least one encryption key in metadata at all times**; that is, there MUST be at least one `<md:KeyDescriptor>` element with no `use` XML attribute in Federation metadata.

The only exception to the previous rule is an SP that supports SAML V1.1 **only**. Since such SP deployments are declining rapidly, and since it doesn't hurt to have an unused encryption key in metadata, it is RECOMMENDED that **all** SPs follow the above rule.

To avoid complications with non-conforming IdPs, it is further RECOMMENDED that there be **exactly one encryption key** in SP metadata at all times. To facilitate this practice, the administrative interface permits an existing certificate in SP metadata to be modified such that the parent key descriptor has an `use="signing"` XML attribute if and only if there is another key descriptor with no `use` XML attribute. See the SP Certificate Migration topic for details.