Key rollover on Shibboleth IdP

This page describes the process of certificate rollover for Shibboleth Identity Providers. The procedure described below allows replacing certificates without any service disruptions.

In SWAMID default installation we have both an Encryption and a Signing certificate.

The IdP defaults to the use of separate keypairs for signing and decryption, the latter being a capability that is rarely used at present and primarily comes into play only when processing `<LogoutRequest>` messages that contain an encrypted subject ID.

Warning regarding SPs based on implementations other than Shibboleth, ADFS or SimpleSAMLphp!

Be aware that many SP implementations other than Shibboleth, ADFS or SimpleSAMLphp are not properly capable of dealing with certificate rollover at the IdP!

1. The SP either ignores additional certificates in metadata and only uses the first or last certificate.
2. The SP is not able to regularly load metadata and will therefore never learn about a new IdP certificate.

In both cases, such SPs will fail to verify the IdP's signature on the SAML assertion and interoperation with that IdP will fail.

In case 1) it will work again once the rollover is complete and the expiring certificate is removed from metadata in step 4).

In case 2) it will only work again once the SP's configuration gets manually updated with the new IdP certificate after step 3).

Note:

- Interfederated IdPs should wait more than 24h before applying step 3).
- However, if you have important SPs using an implementation other than Shibboleth, ADFS or SimpleSAMLphp, do not wait longer than 2h before applying step 3) and best apply the certificate rollover outside of office hours when only few users are expected to access such SPs!

Step 0 : Create new certificate

To generate a new keypair and self-signed certificate for the IdP, run the following commands as root user:

```
sudo -s

cd /opt/shibboleth-idp/credentials

/opt/shibboleth-idp/bin/keygen.sh \
  --lifetime 10 \
  --size 4096 \
  --certfile idp-encryption.crt.new \
  --keyfile idp-encryption.key.new \
  --hostname idp.example.org

/opt/shibboleth-idp/bin/keygen.sh \
  --lifetime 10 \
  --size 4096 \
  --certfile idp-signing.crt.new \
  --keyfile idp-signing.key.new \
  --hostname idp.example.org

chmod 600 idp-encryption.key.new idp-signing.key.new
chown --reference idp-encryption.key idp-encryption.key.new
chown --reference idp-signing.key idp-signing.key.new
```

With the above commands a new certificate and private key are generated inside the `/opt/shibboleth-idp/credentials/` directory.

In case the `JAVA_HOME` environment variable is not set, you can set it for the `keygen.sh` command like this `JAVA_HOME=/usr/lib/jvm/jre /opt/shibboleth-idp/bin/keygen.sh ...`

Step 1 : Add key to Shibboleth

Edit /opt/shibboleth-idp/conf/idp.properties an uncomment

```
idp.encryption.key.2 = %{idp.home}/credentials/idp-encryption-old.key
idp.encryption.cert.2 = %{idp.home}/credentials/idp-encryption-old.crt
```

Edit /opt/shibboleth-idp/conf/credentials.xml and uncomment (remove <!-- and --> around this block)

```
<bean class="net.shibboleth.idp.profile.spring.factory.BasicX509CredentialFactoryBean"
    p:privateKeyResource="%{idp.encryption.key.2}"
    p:certificateResource="%{idp.encryption.cert.2}"
    p:entityId-ref="entityID" />
```

Rearrange keys and reload config

```
sudo -s

cd /opt/shibboleth-idp/credentials

# Backup old key
mv idp-encryption.crt idp-encryption-old.crt
mv idp-encryption.key idp-encryption-old.key

# Put new key in place
mv idp-encryption.crt.new idp-encryption.crt
mv idp-encryption.key.new idp-encryption.key

# The rest could be done as a normal user
exit

# To trigger the IdP to start using the changed credentials, reload the RelyingParty service that also reloads
the conf/credentials.xml file and its referenced credential files:
curl -k https://127.0.0.1/idp/profile/admin/reload-service?id=shibboleth.RelyingPartyResolverService
```

Step 2 : Upload new Metadata

metadata/idp-metadata.xml is NOT automaticaly updated

Note that the metadata is generated as a one-time operation during installation. It does not result from an in-depth analysis of the IdP configuration and does not change when the configuration changes. It's a starter example, not a real metadata source.

First we need to update our XML and replace the encryption certificate and add the new signing certificate.

Either download the XML from metadata.swamid.se OR edit the "original" file /opt/shibboleth-idp/idp-metadata.xml

Replace	With

`<md:KeyDescriptor use="encryption">`
`  <ds:KeyInfo>`
`  <ds:X509Data>`
`  <ds:X509Certificate>Old cert</ds:X509Certificate>`
`  </ds:X509Data>`
`  </ds:KeyInfo>`
`</md:KeyDescriptor>`
`<md:KeyDescriptor use="signing">`
`  <ds:KeyInfo>`
`  <ds:X509Data>`
`  <ds:X509Certificate>Old cert</ds:X509Certificate>`
`  </ds:X509Data>`
`  </ds:KeyInfo>`
`</md:KeyDescriptor>`
	`<md:KeyDescriptor use="encryption">`
	`  <ds:KeyInfo>`
	`  <ds:X509Data>`
	`  <ds:X509Certificate>New cert</ds:X509Certificate>`
	`  </ds:X509Data>`
	`  </ds:KeyInfo>`
	`</md:KeyDescriptor>`
	`<md:KeyDescriptor use="signing">`
	`  <ds:KeyInfo>`
	`  <ds:X509Data>`
	`  <ds:X509Certificate>New cert</ds:X509Certificate>`
	`  </ds:X509Data>`
	`  </ds:KeyInfo>`
	`</md:KeyDescriptor>`
	`<md:KeyDescriptor use="signing">`
	`  <ds:KeyInfo>`
	`  <ds:X509Data>`
	`  <ds:X509Certificate>Old cert</ds:X509Certificate>`
	`  </ds:X509Data>`
	`  </ds:KeyInfo>`
	`</md:KeyDescriptor>`

- Upload the XML to metadata.swamid.se/admin.
- Use "Merge missing from published" to copy over all EntityCategory's and MDUI information from the old Entity if not already in the XML-file.
- Request publication.

- Wait until you get confirmation of publication and then for at least 8 h more (recommended 24 h if in SWAMID and 48 h in eduGAIN) for all entities to pick up the new cert/key.

Step 3 : Switch signing cert

Rearrange keys and reload config

```
sudo -s

cd /opt/shibboleth-idp/credentials

# Backup old key
mv idp-signing.crt idp-signing-old.crt
mv idp-signing.key idp-signing-old.key

# Put new key in place
mv idp-signing.crt.new idp-signing.crt
mv idp-signing.key.new idp-signing.key

# The rest could be done as a normal user
exit

# To trigger the IdP to start using the changed credentials, reload the RelyingParty service that also reloads
the conf/credentials.xml file and its referenced credential files:
curl -k https://127.0.0.1/idp/profile/admin/reload-service?id=shibboleth.RelyingPartyResolverService
```

Step 4 : Upload new Metadata again

Now we need update our XML and remove the old signing certificate.

Replace	With

`<md:KeyDescriptor use="encryption">`
`  <ds:KeyInfo>`
`  <ds:X509Data>`
`  <ds:X509Certificate>New cert</ds:X509Certificate>`
`  </ds:X509Data>`
`  </ds:KeyInfo>`
`</md:KeyDescriptor>`
`<md:KeyDescriptor use="signing">`
`  <ds:KeyInfo>`
`  <ds:X509Data>`
`  <ds:X509Certificate>New cert</ds:X509Certificate>`
`  </ds:X509Data>`
`  </ds:KeyInfo>`
`</md:KeyDescriptor>`
`<md:KeyDescriptor use="signing">`
`  <ds:KeyInfo>`
`  <ds:X509Data>`
`  <ds:X509Certificate>Old cert</ds:X509Certificate>`
`  </ds:X509Data>`
`  </ds:KeyInfo>`
`</md:KeyDescriptor>`
	`<md:KeyDescriptor use="encryption">`
	`  <ds:KeyInfo>`
	`  <ds:X509Data>`
	`  <ds:X509Certificate>New cert</ds:X509Certificate>`
	`  </ds:X509Data>`
	`  </ds:KeyInfo>`
	`</md:KeyDescriptor>`
	`<md:KeyDescriptor use="signing">`
	`  <ds:KeyInfo>`
	`  <ds:X509Data>`
	`  <ds:X509Certificate>New cert</ds:X509Certificate>`
	`  </ds:X509Data>`
	`  </ds:KeyInfo>`
	`</md:KeyDescriptor>`

- Upload the XML to metadata.swamid.se/admin.
- Use "Merge missing from published" to copy over all EntityCategory's and MDUI information from the old Entity if not already in the XML-file.
- Request publication.

Step 5 : Disable / remove key from software.

Edit /opt/shibboleth-idp/conf/credentials.xml and comment (add <!-- and --> around this block)

```
<bean class="net.shibboleth.idp.profile.spring.factory.BasicX509CredentialFactoryBean"
    p:privateKeyResource="%{idp.encryption.key.2}"
    p:certificateResource="%{idp.encryption.cert.2}"
    p:entityId-ref="entityID" />
```

Reload the config to stop accepting encryption with the old keys.