Replacing back-channel certificate and key on an SP

If the back-channel certificates on an existing SP need to be replaced (e.g., due to the private key being possibly compromised or for any other reason), this sequence of steps does this key replacement without any impact on the users of the SP (i.e., login works through all the steps).

These instructions are strongly based on the Shibboleth Project SP Key Rollover instructions - but elaborate some of the generic steps into easy-to-follow guidelines.

In these instructions we assume the same certificate is used for both encryption and signing and we are replacing both of them. Replacing the signing certificate is rather straightforward: in a similar way as when replacing the certificate for an IdP, publish multiple signing certificates in the federation metadata, and after sufficient delay for propagation, switch over the certificate used by the SP. The difficulty is in handling the encryption key - the SP must be able to decrypt assertions encrypted with the new certificate already before the switch-over to the new key, while the new certificate is being propagated through the federation.

Therefore, the sequence of steps is slightly more complex:

1. Generate a new certificate and key-pair, under a different file name (more specifically, in a different directory):

```
mkdir /etc/shibboleth/newcert
cd /etc/shibboleth/newcert
/etc/shibboleth/keygen.sh -u shibd -g shibd -h `hostname` -e https://`hostname`/shibboleth
```

   - Adjust the command-line accordingly if your hostname is different from the user-facing DNS name - or if you need a different entityID for some other reason.

2. Add this newly generated key into your SP configuration explicitly as a decryption key (`use="encryption"`).
   - In `/etc/shibboleth/shibboleth2.xml`, below the existing `<CredentialResolver>` element, add:

```
<CredentialResolver type="File" key="newcert/sp-key.pem" certificate="newcert/sp-cert.pem" use="
encryption" />
```

3. Add the new certificate into the federation metadata for both **signing** (both old and new certificates are trusted) and **encryption** (an IdP can encrypt the assertion sent to this SP using either old or new certificate). The second part is now OK as the SP would be able to decrypt the assertion.
   - Log into the Federation Registry (at https://registry.tuakiri.ac.nz/federationregistry/)
   - Navigate to the details page for your SP
   - On the Certificates tab (under SAML in FR2 used in Tuakiri-TEST), add the new certificate.
     - Use the default settings, adding the certificate for both **signing** and **encryption** use.

4. Wait for the updated metadata to propagate through the federation. To be sure all IdPs have loaded the new metadata, wait for 4 hours.

5. Switch the SP to use the new key for general use but still keep the old key in for decrypting messages - change the `<CredentialResolver>` entries in `/etc/shibboleth/shibboleth2.xml` to (moving the `use="encryption"` attribute from the new certificate to the old certificate):

```
        <CredentialResolver type="File" key="sp-key.pem" certificate="sp-cert.pem" use="encryption" />
        <CredentialResolver type="File" key="newcert/sp-key.pem" certificate="newcert/sp-cert.pem"/>
```

6. Remove the old certificate from the federation metadata:
   - Again, log into the Federation Registry
   - Navigate to the details page for your SP
   - On the Certificates tab (under SAML in FR2 used in Tuakiri-TEST), remove the old certificate (make sure this is the old one and not the new one).
     - Note that the certificate is listed twice (separately for the **signing** and **encryption** roles), so it is necessary to delete the certificate twice.

7. Wait for propagation again (again for four hours).
   - At this point, the old certificate is no longer trusted and cannot be abused within the federation (no IdP would be trusting the old SP certificate).

8. Change the SP configuration to only use the new certificate and key. While you could just remove the `<CredentialResolver>` element for the old certificate, you'd end up with the new certificate and key being loaded from a non-standard location. So instead do (in this order! - as touching `/etc/shibboleth/shibboleth2.xml` triggers `shibd` to reload the configuration):
   - Rename the old certificate and key to a different name and move the new certificate and key into the standard location:

```
cd /etc/shibboleth
mv sp-cert.pem sp-cert.pem.old
mv sp-key.pem sp-key.pem.old
mv newcert/sp-cert.pem sp-cert.pem
mv newcert/sp-key.pem sp-key.pem
```

- Edit `shibboleth.xml`, reverting it to the original configuration - only one `CredentialResolver` entry of this form:

```
<CredentialResolver type="File" key="sp-key.pem" certificate="sp-cert.pem" />
```

9. This completes the certificate replacement process.