

Renewal of client certificate using soap web service of certificate service

This guide describes one example way to renew SSL client certificate that has been granted by Finnish Tax Administration. Renewal process can be implemented to the client application based on this guide.

Renewal process of test certificates in this guide can be also applied to renewal process of production certificates. Real organizational information and production web service endpoints are used with production certificate renewal process.

Prerequisites

- Understanding of PKI concepts for creating key pair and certificate signing request
- Software development skills, compiling and running the signing application example
- Certificate service web service message schemas, WSDL:
https://vero.fi/globalassets/tulorekisteri/kuvat/varmennepalvelu-rajapinta_v1.01.zip
- New key pair for certificate signing request (CSR)
- Current client certificate and related private key for signing the renewal xml message as pfx file and password for the pfx file if pfx is protected
- Business id and name of the test company related to the current client certificate

Required tools

- OpenSSL: <https://wiki.openssl.org/index.php/Binaries>
- .NET Core 3.1 or later: <https://dotnet.microsoft.com/download/dotnet/3.1>
- Visual Studio Code: <https://code.visualstudio.com/>
- Example implementation for signing the xml message (SignXmlNew.exe), can be downloaded from Vero.fi: https://www.vero.fi/globalassets/tietoa-verohallinnosta/ohjelmistokehittajille/swaggerui/verohallinto_program.zip
- Client for consuming SOAP web service: SoapUI
<https://www.soapui.org/downloads/soapui/> or Curl <https://curl.se/download.html>

More information:

- [Tax Administration's certificate service](#)
- Vero API Slack-channel: <https://vero-api.slack.com>
 - Join to the channel using feedback form on Vero API page

Step by step guide for renewal of client certificate

1. Create new private key for new certificate

Create new private key with OpenSSL using command line:
`openssl genrsa -out newprivate.key 2048`

New private key is created to a file called newprivate.key

2. Create new certificate signing request for renewal

Create new certificate signing request file (CSR) using the new private key created in step 1 with OpenSSL:
`openssl req -new -key newprivate.key -out certificaterequest.csr`

Enter following information for OpenSSL from current client certificate that is being renewed:

Country Name = *FI*

Organization Name = *Name of your test company name*

Common Name = *Business id of your test company*

New certificate signing request is created to a file called certificaterequest.csr

3. Create renewal xml message that is going to be signed

Create content part of the renewal xml message. Only this content part will be signed in later steps. Use following template and fill needed information. **Note!** Remove all line breaks before signing the message:

```
<cer:RenewCertificateRequest xmlns:cer="http://certificates.vero.fi/2017/10/certificateservices"
xmlns:xd="http://www.w3.org/2000/09/xmldsig#">
  <Environment>TEST</Environment>
  <CustomerId>Business id of your test company</CustomerId>
  <CustomerName>Name of your test company</CustomerName>
  <CertificateRequest>Certificate signing request content that was created at step 2 as
base64 string without --- begin certificate request -- and --- end certificate request ---
headers</CertificateRequest>
</cer:RenewCertificateRequest>
```

Fill needed information to the template: business id of your test company, name of you test company and CSR that was created in step 2. CSR must be a base64 encoded string without begin header ("--- begin certificate request---") and end header. Save the template to a file without line breaks for signing.

4. Sign the xml message

Create signature for the content part of the renewal message that was created in step 3. You can sign the content with any available method, or you can use example implementation from Tax Administration. This guide is based on example implementation of Tax Administration and it requires the current client certificate and related private key as pfx file.

Example signing application is available here: https://www.vero.fi/globalassets/tietoa-verohallinnosta/ohjelmistokehittajille/swaggerui/verohallinto_program.zip

.NET core 3.1 or later and Visual Studio Code are required to run the example. Visual Studio Code is available here: <https://code.visualstudio.com/>

4.1 Create pfx file for the signing application using Open SSL

New pfx file is not needed if you have already one that contains current client certificate (the certificate that is being renewed) and related private key. In this case you can skip this step and to to step 4.2

Run the following command that generates the pfx file. The current certificate and the private key with which it was generated are exported to the file.

If the pfx file does not already exist, use the following command to build the pfx file and export the private key and the current certificate (stored in the cert.cer file in base64 format):
`openssl.exe pkcs12 -export -out test.pfx -inkey private.key -in cert.cer`

The private key of the certificate currently in use is in the command input in the private.key file in base64 format, unprotected, and the public part of the certificate (= signed public key) is in the cert.cer file in base64 format.

```
<cer:RenewCertificateRequest xmlns:cer="http://certificates.vero.fi/2017/10/certificateservices"
xmlns:xd="http://www.w3.org/2000/09/xmldsig#">
  <Environment>TEST</Environment>
  <CustomerId>0123456-7</CustomerId>
  <CustomerName>Ab PKI Developer Company Oy</CustomerName>
  <CertificateRequest>MIICjTCCAXUCAQAwSDELMAGGA1UEBhMCRkxkEzARBgNVBAGMCINvbWUuU3RhdGUx
JDAiBgNVBAAoMG0FIbFBLSSBEZXLbG9wZXIlgQ29tcGFueSBPeTCCASlwdQYJKoZI
hvcNAQEBBQADggEPADCCAQoCggEBAJkBP88eLdbxbJfPluDI/rNP0EUpluRohxgx
MnfuYVv9kXQdQpZCsV/QjwZFpWBSFy6PDJlKyvAqe83XSfoGpt9apy3QaUJuXR
4/P5H6VT+eZpt1TCf5CEaKb0aW4bZ1kN9BLerrJ81HsR6cutpE/t0bzArc4kna/l
rzYb3tlU34YoHyx9bXNwKSPsUdL7N32vluSO8Me/3NjFzA9CBYRrP56qnXlyTmm
0x5GJXGBJqJm2xBRCmpMWg5WGUOF8mAGxkPDxyEfZpaHXbSLaBQ1nJyDPg0+n/Ak
rcweyDEOBKMH3rSITH/M5D726yKGAHEWERG1Nz06ei+a+KJUcCAWEAAaAAMA0G
CSqGSIb3DQEBCwJAA4IBAQBslqCulgyrfU+DVZxS60Hvu4d8GcKKRGCTfBt508BM
c+NSnevgakWZXXMMWKOJStsDHsOPnwfalvmlFLWRKAsqxt2dlGgWMzFh9NaX0Anwm
CbiUruot9C8zguP7Y/67AFSeageNYrHmgIBHoZyNle+tpR4Y5DxcQBI6HtyzJ/q
Nej5mp2zSIW5P1QoEKs3MU8Gm0mpCBylyAvCzeYHOop6caZMQctVCmpTo+OPYx0T
qEmO15vGj/rIN4btjEKSyYfNj56MMN8slc/6vqdkKKmMwTLRjXq73liOYyJ11s
9433VK1J/Umvay3y2jYKVDUuW567HD8C3IsT+A+ifkCo</CertificateRequest>
  <Signature xmlns="http://www.w3.org/2000/09/xmldsig#"><SignedInfo><CanonicalizationMethod
Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" /><SignatureMethod Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-
sha256" /><Reference URI=""><Transforms><Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"
/></Transforms><DigestMethod Algorithm="http://www.w3.org/2001/04/xmldsig#sha256"
/><DigestValue>i13a6CV9yr+uqy/qx4yhvyysDvcKnoiNijUdj7Arr1A=</DigestValue></Reference></SignedInfo><SignatureValue>VEja4
6Y17IaMXHmJfcZMRM+3pZTLSEpvzWeR2JLMMCz3nWldJynhs1MjGmbqJ3glsebmke3UX10T0Z0LobtbeACFYz78dDKbWHTc4cU
11WKZU3DpXQ5svgJWNk1L+B2SDH7V+ethFNqBmwLCgsE2D78pt7XwsBOnZe/Rt30fIEMd5sSWYYJeb1FzMXAcaFVloVs31T9HcoCF
upgMH9YWsgzpknQHTSTKfjBZbhsjBnvnDlwSceFhxxNpcmY/zvJRVB56WeC2qhQgZFN7PsnCJ6KnNOTkYr2w7CVCFNwofCMU3eXUI
+n5khTJmNQV+SZ2S0qPzBSp6TD/reCVJJA==</SignatureValue><KeyInfo><X509Data><X509Certificate>MIIFqzCCA5OgAwIBAgIIG
ZoeTGyXo3lwdQYJKoZIhvcNAQELBQAwSjEKMCIgA1UEAAwBUeUjIFNlcnZpY2UyRGV2ZWxvcGVyIENBIHlyMRUwEwYDVQwKDAA
wWZXJvaGFsbG9wZXIlgQ29tcGFueSBPeTCCASlwdQYJKoZIhvcNAQEBBQADggEPADCCAQoCggEBAmrF+WUX2nu
YBOeGK3PxqzleMmMVRwIqmbBTHjdV0AamRZ34cuh+Do/76U0mqg9G4IVsj8WaM8fmh7tdCQ3xcCPnpQrkeGuWQV4nlhok74kDnQb1
2FrOkK3LIOMONHS2+9E8HKwS8giFzKp8UUNJK8PmptJQo+E6jIey+yzSsHouf0UMCgp9MutN9RIAtjqS6lyHtpq8BLn2hdEM1srIqCX
BRigkAH5w1mqbB8SiVkgCaYJ+I5AY201ZTUlB138SYbYk9gflS1aY1gEf+667Bmrys0aJk4JRLHujMqfEuHrFRWo1ps739H+8UPqkRm
JfNybGFUPoJEWcfGlkXdyGpUCAwEAAaOCASwswgFnMAwGA1UdEwEB/wQCMAAwHwYDVR0jBBGwFoAUT1PJe8BCr9h+uQE8W6
CNC7/QfeYwUwYIKwYBBQUHAQEERzBFMEMGCGCSGAQUFBzAChjdodHRwOi8vY3JsLXRlc3RlLnZlcm8uZmkvY2EvUEUyZDZydmJlZ
URldmVsb3BickNbdjEuy3J0MBMGA1UdJQMMMAoGCCSGAQUFBzMCMIgCgBGNVHR8EgZJwCgZEWgY26gPKA6hjhodHRwOi8vY3JsL
XRlc3RlLnZlcm8uZmkvY3JsL1BLSVNIcnZpY2EYZEZXLbG9wZXIlgQ29tcGFueSBPeTCCASlwdQYJKoZIhvcNAQELBQADggIBADZKk4T+rVIAe9a53/9zrWLUJqe+WePxIoEk5ozX
WDb2FeR0uEyUS2Ba0gJ/JwPm9Go6CAia3J9nFGyVUUNCM2of2dQGEx4JkrRc7c08JPamY74tJR9wwj8R5sshAXPDVMMwH9MI8LHG
6hqz20ic0IK9cSsAHBGI3GBGL3/6y+SPWGMK9f0Qim5of63qA8la95y04aUJL7td2Yxiu6jKUFp4haLOBVJFM//o6Ge5LxT3nFPZxESblb
LE21D0ksyO+JfZljlfxleQk9rWY7Zyq/Go9+ElvEILXE2aDjqQrwoNIQHmqLg0DuDKpJKzSi7nRvDVHAB5YIdtLDJ4PXZITkib8QBOZWm
HCw58lVfEdLOWfUrpzJmIcF8oyzLWRagtnEQhwnWnkXOtPqivRq3Rh35M4mQPNVPikduzYlhvQzwCAVkgzspEZVT5hQITEXBiZZQ8jC
8Mb6U1u7G/NndHGwdWOnWtNYDMrhqEZGoHxglTLwaU4d5suHkzv0glxkreR4fnVdiVWd4zCNQk6tr9Jo3pOyLFGM49G3kszHPcYxxB
mzqrSBoBKX5S9n+jOF39fxE6LNCmBiZ49WhSOTSLjXk4LB0T4NB0CtzeEdhQK0lz1JC5GvNuVnmKeZEIlt3qLvx4ktc6QxIH2zZ48
BR+m/cXyvcylZ2fgyAIW</X509Certificate></X509Data></KeyInfo></Signature></cer:RenewCertificateRequest>
```

You will notice that the signing program has added a Signature block to the end of the RenewCertificateRequest block.

Importantly, the contents of the file must not be altered in any way before it is sent to the certificate service interface, so as not to change the signed content. The change is also caused by line breaks or other formatting, in which case the signature is no longer valid and the certificate service responds with error code PKI010.

5. Send a signed message to the certificate service endpoint

Send the signed test certificate renewal message using for example SoapUI to the test address of the certificate service:

<https://pkiws-testi.vero.fi/2017/10/CertificateServices>

In SoapUI, you can use the templates of the message structures based on the WSDL description. Download the certificate service interface package (https://vero.fi/globalassets/tulorekisteri/kuvat/varmennepalvelu-rajapinta_v1.01.zip) and open the WSDL file with SoapUI.

Generate the outgoing message so that the signed content is unchanged inside the body element of the soap envelope. Do not format the signed content in any way.

Example of soap envelope and where to export the content:

```
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/"  
xmlns:xs="http://www.w3.org/2001/XMLSchema" ><env:Body>  
the content signed here as it is, formed in section 4 of this guide</env:Body></env:Envelope>
```

Check the URL and send a message. The answer is OK and retrievalID, which can be used to retrieve the renewed certificate using the GetCertificate operation.

Instead of SoapUI, you can use curl. In the Windows environment, you need to be careful that line breaks must not be present in the content to be signed, and all content must be on a single line. Be sure to add soap envelope to the signed file. Use the CURL command below to submit a renewal request:

```
curl -i -v -d @template_signed_env.xml --header "SOAPAction:renewCertificate" -H "Content-Type:  
text/xml; charset=UTF-8" -H "Accept-Encoding: gzip, deflate" https://pkiws-  
testi.vero.fi/2017/10/CertificateServices
```

6. Retrieve a renewed certificate using the GetCertificate operation

When picking up, use the instructions of the Tax Administration's certificate service.

Store the PFX file and private key with care.