

---

# HSM Proxy Web Service Specifications

Version 0.2.0

Frank Cornelis

Copyright © 2013 FedICT

## Abstract

This document details on the HSM Proxy web service specifications.

1. Introduction .....	1
2. Signatures .....	2
2.1. Request .....	2
2.2. Response .....	2
3. Aliases .....	3
3.1. Request .....	3
3.2. Response .....	3
4. Get Certificate Chain .....	4
4.1. Request .....	4
4.2. Response .....	4
5. Security .....	5
A. HSM Proxy Web Service Specifications License .....	6
B. HSM Proxy Project License .....	7

## 1. Introduction

The HSM Proxy web service is implemented according to [OASIS DSS Core 1.0](http://docs.oasis-open.org/dss/v1.0/oasis-dss-core-spec-v1.0-os.html) [http://docs.oasis-open.org/dss/v1.0/oasis-dss-core-spec-v1.0-os.html] and [OASIS WS-Security 1.1](https://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf) [https://www.oasis-open.org/committees/download.php/16790/wss-v1.1-spec-os-SOAPMessageSecurity.pdf] standards.

The messages are transported using [W3C SOAP 1.2](http://www.w3.org/TR/soap12-part1/) [http://www.w3.org/TR/soap12-part1/] over [TLS 1.0](http://www.ietf.org/rfc/rfc2246.txt) [http://www.ietf.org/rfc/rfc2246.txt] .

The XML namespaces used in the following sections are described in [Table 1, "XML Namespaces"](#)

.

**Table 1. XML Namespaces**

Prefix	Namespace
dss	urn:oasis:names:tc:dss:1.0:core:schema
ds	http://www.w3.org/2000/09/xmldsig#

Prefix	Namespace
soap12	http://www.w3.org/2003/05/soap-envelope
wsse	http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd
wsu	http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd
ec	http://www.w3.org/2001/10/xml-exc-c14n#
hsm	urn:be:fedict:hsm-proxy:ws:dss:profiles:hsm-proxy:1.0

## 2. Signatures

### 2.1. Request

An example signature request message without SOAP envelope:

```
<dss:SignRequest RequestID="request-id">
  <dss:OptionalInputs>
    <dss:KeySelector>
      <ds:KeyInfo>
        <ds:KeyName>alias</ds:KeyName>
      </ds:KeyInfo>
    </dss:KeySelector>
  </dss:OptionalInputs>
  <dss:InputDocuments>
    <dss:DocumentHash>
      <ds:DigestMethod
        Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
      <ds:DigestValue>Kq5sNclPz7QV2+lfQIuc6R7oRu0=</ds:DigestValue>
    </dss:DocumentHash>
  </dss:InputDocuments>
</dss:SignRequest>
```

The supported digest algorithms are listed in [Table 2, "Digest Algorithms"](#).

**Table 2. Digest Algorithms**

URI
http://www.w3.org/2000/09/xmldsig#sha1
http://www.w3.org/2001/04/xmlenc#sha256

### 2.2. Response

The corresponding signature response message without SOAP envelope:

```
<dss:SignResponse RequestID="request-id">
  <dss:Result>
    <dss:ResultMajor>
      urn:oasis:names:tc:dss:1.0:resultmajor:Success
    </dss:ResultMajor>
    <dss:ResultMinor>
      urn:oasis:names:tc:dss:1.0:resultminor:valid:signature:OnAllDocuments
    </dss:ResultMinor>
  </dss:Result>
  <dss:SignatureObject>
    <Base64Signature>...</Base64Signature>
  </dss:SignatureObject>
</dss:SignResponse>
```

The `<Base64Signature>` element contains the base64 encoded PKCS#1 RSA signature value.

## 3. Aliases

Via the HSM Proxy web service you can also retrieve a list of available aliases.

### 3.1. Request

An example request message without SOAP envelope:

```
<hsm:GetAliasesRequest RequestID="request-id"/>
```

Notice here that the `<hsm:GetAliasesRequest>` element is part of an HSM Proxy specific OASIS DSS profile. The corresponding XML namespace and profile URI are:

```
urn:be:fedict:hsm-proxy:ws:dss:profiles:hsm-proxy:1.0
```

### 3.2. Response

An example response message without SOAP envelope:

```
<dss:Response
  Profile="urn:be:fedict:hsm-proxy:ws:dss:profiles:hsm-proxy:1.0"
  RequestID="request-id">
  <dss:Result>
    <dss:ResultMajor>
      urn:oasis:names:tc:dss:1.0:resultmajor:Success
    </dss:ResultMajor>
  </dss:Result>
```

```
<dss:OptionalOutputs>
  <hsm:Aliases>
    <hsm:Alias>alias 1</hsm:Alias>
    <hsm:Alias>alias 2</hsm:Alias>
    ...
    <hsm:Alias>alias n</hsm:Alias>
  </hsm:Aliases>
</dss:OptionalOutputs>
</dss:Response>
```

## 4. Get Certificate Chain

Via the HSM Proxy web service you can also retrieve a certificate chain for a given alias.

### 4.1. Request

An example request message without SOAP envelope:

```
<hsm:GetCertificateChainRequest RequestID="request-id">
  <dss:OptionalInputs>
    <hsm:Alias>alias</hsm:Alias>
  </dss:OptionalInputs>
</hsm:GetCertificateChainRequest>
```

Notice here that the `<hsm:GetCertificateChainRequest>` element is part of an HSM Proxy specific OASIS DSS profile. The corresponding XML namespace and profile URI are:

```
urn:be:fedict:hsm-proxy:ws:dss:profiles:hsm-proxy:1.0
```

### 4.2. Response

An example response message without SOAP envelope:

```
<dss:Response Profile="urn:be:fedict:hsm-proxy:ws:dss:profiles:hsm-proxy:1.0"
  RequestID="request-id">
  <dss:Result>
    <dss:ResultMajor>
      urn:oasis:names:tc:dss:1.0:resultmajor:Success
    </dss:ResultMajor>
  </dss:Result>
  <dss:OptionalOutputs>
    <ds:KeyInfo>
      <ds:X509Data>
```

```

        <ds:X509Certificate>...</ds:X509Certificate>
        ...
        <ds:X509Certificate>...</ds:X509Certificate>
    </ds:X509Data>
</ds:KeyInfo>
</dss:OptionalOutputs>
</dss:Response>

```

## 5. Security

The web service client should add a WS-Security SOAP header according to [OASIS WS-Security X.509 Certificate Token Profile 1.1](https://www.oasis-open.org/committees/download.php/16785/wss-v1.1-spec-os-x509TokenProfile.pdf) [https://www.oasis-open.org/committees/download.php/16785/wss-v1.1-spec-os-x509TokenProfile.pdf] . Example SOAP message with WS-Security SOAP header:

```

<?xml version="1.0"?>
<soap12:Envelope>
  <soap12:Header>
    <wsse:Security soap12:mustUnderstand="true">
      <wsu:Timestamp wsu:Id="TS">
        <wsu:Created>2013-05-15T09:21:45.483Z</wsu:Created>
        <wsu:Expires>2013-05-15T09:26:45.483Z</wsu:Expires>
      </wsu:Timestamp>
      <wsse:BinarySecurityToken
        EncodingType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-soap-message-security-1.0#Base64Binary"
        ValueType="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-x509-token-profile-1.0#X509v3"
        wsu:Id="X509">
        ...
      </wsse:BinarySecurityToken>
      <ds:Signature Id="SIG">
        <ds:SignedInfo>
          <ds:CanonicalizationMethod
            Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
            <ec:InclusiveNamespaces PrefixList="S"/>
          </ds:CanonicalizationMethod>
          <ds:SignatureMethod
            Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-
sha256"/>
          <ds:Reference URI="#TS">
            <ds:Transforms>
              <ds:Transform
                Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
                <ec:InclusiveNamespaces PrefixList="wsse S"/>
              </ds:Transform>
            </ds:Transforms>

```

```

        <ds:DigestMethod
            Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
        <ds:DigestValue>...</ds:DigestValue>
    </ds:Reference>
    <ds:Reference URI="#id-body">
        <ds:Transforms>
            <ds:Transform
                Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"
                <ec:InclusiveNamespaces PrefixList=""/>
            </ds:Transform>
        </ds:Transforms>
        <ds:DigestMethod
            Algorithm="http://www.w3.org/2001/04/xmlenc#sha256"/>
        <ds:DigestValue>...</ds:DigestValue>
    </ds:Reference>
</ds:SignedInfo>
<ds:SignatureValue>...</ds:SignatureValue>
<ds:KeyInfo Id="KI">
    <wsse:SecurityTokenReference wsu:Id="STR">
        <wsse:Reference URI="#X509"
            ValueType="http://docs.oasis-open.org/wss/2004/01/
oasis-200401-wss-x509-token-profile-1.0#X509v3"/>
    </wsse:SecurityTokenReference>
</ds:KeyInfo>
</ds:Signature>
</wsse:Security>
</soap12:Header>
<soap12:Body wsu:Id="id-body">
    ...
</soap12:Body>
</soap12:Envelope>

```

Each request SHOULD include a WS-Security SOAP header as above. A `wsu:Timestamp` timestamp element MUST be included. The signature MUST digest both the timestamp and the SOAP body element. The signature digest algorithm MUST be SHA-256. The signature algorithm must be SHA-256 with RSA.

## A. HSM Proxy Web Service Specifications License



This document has been released under the [Creative Commons 3.0](http://creativecommons.org/licenses/by-nc-nd/3.0/) [http://creativecommons.org/licenses/by-nc-nd/3.0/] license.

## B. HSM Proxy Project License

The HSM Proxy Project source code has been released under the GNU LGPL version 3.0.

This is free software; you can redistribute it and/or modify it under the terms of the GNU Lesser General Public License version 3.0 as published by the Free Software Foundation.

This software is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the GNU Lesser General Public License for more details.

You should have received a copy of the GNU Lesser General Public License along with this software; if not, see <http://www.gnu.org/licenses/>.

