EKMI SOAP Profile Version 1.0

Committee Specification Draft 01

19 July 2011

Specification URIs:

This version:

http://docs.oasis-open.org/ekmi/ekmi-soap-profile/v1.0/csd01/ekmi-soap-profile-v1.0-csd01.odt (Authoritative)

http://docs.oasis-open.org/ekmi/ekmi-soap-profile/v1.0/csd01/ekmi-soap-profile-v1.0-csd01.html

http://docs.oasis-open.org/ekmi/ekmi-soap-profile/v1.0/csd01/ekmi-soap-profile-v1.0-csd01.pdf

Previous version:

N/A

Latest version:

http://docs.oasis-open.org/ekmi/ekmi-soap-profile/v1.0/ekmi-soap-profile-v1.0.odt (Authoritative)

http://docs.oasis-open.org/ekmi/ekmi-soap-profile/v1.0/ekmi-soap-profile-v1.0.html

http://docs.oasis-open.org/ekmi/ekmi-soap-profile/v1.0/ekmi-soap-profile-v1.0.pdf

Technical Committee:

OASIS Enterprise Key Management Infrastructure (EKMI) TC

Chairs:

Anil Saldhana, Red Hat

Tim Bruce, CA Technologies

Editor:

Tomas Gustavsson, PrimeKey Solutions AB

Related work:

This specification is related to:

- Symmetric Key Services Markup Language (SKSML) Version 1.0

Abstract:

This profile specifies how Enterprise Key Management Infrastructure (EKMI) Symmetric Key Services Markup Language (SKSML) messages are transferred securely using a SOAP Version 1.2 envelope with Web Services Security (WSS) security features.

Status:

This document was last revised or approved by the OASIS Enterprise Key Management Infrastructure (EKMI) TC on the above date. The level of approval is also listed above. Check the “Latest version” location noted above for possible later revisions of this document.

Technical Committee members should send comments on this specification to the Technical Committee’s email list. Others should send comments to the Technical Committee by using the “Send A Comment” button on the Technical Committee’s web page at http://www.oasis-open.org/committees/ekmi/.

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the Technical Committee web page (http://www.oasis-open.org/committees/ekmi/ipr.php).

Citation format:

When referencing this specification the following citation format should be used:

[EKMI-SOAP-Profile]

EKMI SOAP Profile Version 1.0. 19 July 2011. OASIS Committee Specification Draft 01. http://docs.oasis-open.org/ekmi/ekmi-soap-profile/v1.0/csd01/ekmi-soap-profile-v1.0-csd01.html.

Notices

Table of Contents

1 Introduction

The EKMI SOAP v1.2 profile specifies how the SKSML messages defined in EKMI v1.0 is transferred over SOAP messaging using WSS for authenticity and integrity protection.

1.1 Terminology

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this specification are to be interpreted as described in IETF RFC 2119.

1.2 Normative References

[RFC 2119] S. Bradner. *Key words for use in RFCs to Indicate Requirement Levels*. IETF RFC 2119, March 1997. http://www.ietf.org/rfc/rfc2119.txt.

[AES] Advanced Encryption Standard.NIST FIPS 197. http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[RFC 2119] S. Bradner. *Key words for use in RFCs to Indicate Requirement Levels*. IETF RFC 2119, March 1997.

[SOAP] SOAP v1.2 Specification. W3C Recommendation. 27 April 2007. http://www.w3.org/TR/soap12

[**XMLEncryption]** XML Encryption Syntax and Processing. W3C Recommendation. 10 Dec 2002.

http://www.w3.org/TR/xmlenc-core/

[XMLSignature] XML Signature Syntax and Processing. W3C Recommendation. 10 June 2008.

http://www.w3.org/TR/xmldsig-core/

[WSS] OASIS Standard, “Web Services Security – SOAP Message Security 1.0”, March 2004. http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf

1.3 Non-normative References

[Reference] [reference citation]

2 SOAP Profile

When SKSML messages are transferred using SOAP the SOAP message is secured using WSS. The actual SKSML content is placed in the SOAP body. The rest of the message is constructed of SOAP envelope and WSS elements. The SOAP profile relies on the SOAP standard v1.2 and the Web Services Security (WSS) standard 1.0, which in turn supports the use of XML Signature and XML Encryption within the SOAP Header. Relying only the on the WSS profile that uses RSA cryptographic key-pairs and digital certificates, SKSML uses the digital signatures for authenticity and message-integrity, while using RSA-encryption for confidentiality.

The general structure of the SOAP message is constructed of the following elements.

```
<SOAP Envelope>
      <SOAP Header>
            <WSSE Security Attributes/>
      </SOAP Header>
      <SOAP Body>
            <SKSML Request/Response/>
      </SOAP Body>
</SOAP Envelope>
```

The message below shows a SKSML SymkeyRequest. While the SymkeyRequest element is very simple, the Web Service Security (WSS) envelope – which provides security for the SKSML messages in the SOAP profile – expands the size of the message. The SymkeyRequest, is displayed below in its entirety, with its WSS envelope. Please note that some content – such as Base64 encoded binary content - has been reformatted for aesthetics and clarity of the XML elements.

```
<?xml version='1.0' ?>
<env:Envelope xmlns:env="http://www.w3.org/2003/05/soap-envelope">
   <env:Header>
      <wsse:Security
         xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/
            oasis-200401-wss-wssecurity-secext-1.0.xsd"
            env:mustUnderstand="true">
         <wsse:BinarySecurityToken
            xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/
            oasis-200401-wss-wssecurity-utility-1.0.xsd"
            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="XWSSGID-1172790302111-1738806553">
MIIDfDCCAmSgAwIBAgIIAe/AvliGc3AwDQYJKoZIhvcNAQELBQAwZzEmMCQGA1UEAxMdU3Ryb25nab
S2V5IERFTU8gU3Vib3JkaW5hdGUgQ0ExJDAiBgNVBAsTG0ZvciBTdHJvbmdLZXkgREVNTyBVc2Ug1d
T25seTEXMBUGA1UEChMOU3Ryb25nQXV0aCBJbmMwHhcNMDYwNzI1MTcxMDMwWhcNMDcwNzIa64dd3k
A1UECxMbRm9yIFN0cm9uZ0tleSBERU1PIFVzZSBPbmx5MRcwFQYDVQQKEw5TdHJvbmdBdXRoIEl2da
S2V5IERFTU8gU3Vib3JkaW5hdGUgQ0ExJDAiBgNVBAsTG0ZvciBTdHJvbmdLZXkgREVNTyBVc2Ugia
T25seTEXMBUGA1UEChMOU3Ryb25nQXV0aCBJbmMwHhcNMDYwNzI1MTY0NjEwWhcNMDcwNzI1s34wdd
NjEwWjBpMREwDwYKCZImiZPyLGQBARMBOTEVMBMGA1UEAxMMU0tTIFNlcnZlci0xMSQwIgYDVQsdw2
ExtGb3IgU3Ryb25nS2V5IERFTU8gVXNlIE9ubHkxFzAVBgNVBAoTDlN0cm9uZ0F1dGggSW5jMIIBd2
NBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAztppqRoU5A8plxx1Rz1QEUnlAAM1D5g9+isIr3wxa
hbwjtFSMYilnY4iV77xU/nsMOnMZ7RxsLYKdCzQ1ODVYqQwqmAvaJ5Z6SVy34gZ51YG+rSWE3NjFsd
bOXW8RJYA/Tn6Lmht/qngrcaqqmtP0cAAiMRZOWtCTmC2K/LEqDabXSyU6Hh8ySNE3njybvmWpresf
zsYokTdvnWQqT6tKo1OwJsdJ1+hxM7DrnMLvMNq5reINfsKhDdX17wzhrBUx+hiYA/qo8tMXkL6wsd
4PN5dYugtzpSzIdUO5tIg58Avhzwo7hy5oofBlKFY22CeljQ36u0bMjuyGj6UYHs3rdfdfsds32rda
YzCBnzANBgkqhkiG9w0BAQEFAAOBjQAwgYkCgYEAyAmxMZhYA8wHJ4UE4b61s51JVWe4Fygj4MCf3a
hvcNAQELBQADggEBACK05PtvZD4WPgl0e=
         </wsse:BinarySecurityToken>
         <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
            <ds:SignedInfo>
               <ds:CanonicalizationMethod
                     Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
                  <InclusiveNamespaces
                     xmlns="http://www.w3.org/2001/10/xml-exc-c14n#"
                     PrefixList="wsse env"/>
               </ds:CanonicalizationMethod>
               <ds:SignatureMethod
```

```
[b41]                      Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
[b42]                   <ds:Reference URI="#XWSSGID-1172790300636-653454040">
[b43]                      <ds:DigestMethod
[b44]                          Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
[b45]                      <ds:DigestValue>
[b46]                          lU4m+rp4oebgl9g+t3nRaZYqUlE=
[b47]                      </ds:DigestValue>
[b48]                   </ds:Reference>
[b49]               </ds:SignedInfo>
[b50]               <ds:SignatureValue>
[b51]  svStAvBRRrF+g2biPl7uWHkJTQPIl8t4phMbOZQsZlQcn36tcMSj/a4+4LPNfOB3Y8yO2lr1Oa1
[b52]  fGqCPAWZNuEH34VQEM196rRwV258mgp8uwpXEYJIgPJqg89w8+/NdaODccLQ2Bizu7QM/HSM2ab
[b53]  ogNJwqmbSyIazOsnOcU=
[b54]               </ds:SignatureValue>
[b55]               <ds:KeyInfo>
[b56]                   <wsse:SecurityTokenReference
[b57]                          xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/
[b58]                          oasis-200401-wss-wssecurity-utility-1.0.xsd"
[b59]                          wsu:Id="XWSSGID-1172790300633-442423344">
[b60]                      <wsse:Reference
[b61]                          URI="#XWSSGID-1172790302111-1738806553"
[b62]                          ValueType="http://docs.oasis-open.org/wss/2004/01/
[b63]                          oasis-200401-wss-x509-token-profile-1.0#X509v3"/>
[b64]                   </wsse:SecurityTokenReference>
[b65]               </ds:KeyInfo>
[b66]            </ds:Signature>
[b67]            <wsu:Timestamp
[b68]                   xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/
[b69]                   oasis-200401-wss-wssecurity-utility-1.0.xsd"
[b70]                   wsu:Id="XWSSGID-11727903006377088718O5">
[b71]               <wsu:Created>
[b72]                   2007-03-01T23:05:00Z
[b73]               </wsu:Created>
[b74]               <wsu:Expires>
[b75]                   2007-03-01T23:05:05Z
[b76]               </wsu:Expires>
[b77]            </wsu:Timestamp>
[b78]         </wsse:Security>
[b79]     </env:Header>
[b80]     <env:Body
[b81]            xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/
[b82]            oasis-200401-wss-wssecurity-utility-1.0.xsd"
[b83]            wsu:Id="XWSSGID-1172790300636-653454040">
[b84]         <ekmi:SymkeyRequest
[b85]               xmlns:ekmi="http://docs.oasis-open.org/ekmi/2008/01">
[b86]            <ekmi:GlobalKeyID>10514-0-0</ekmi:GlobalKeyID>
[b87]         </ekmi:SymkeyRequest>
[b88]     </env:Body>
[b89]  </env:Envelope>
```

[b01] is the XML declaration which specifies the version of XML being used.
[b02] is the start of the SOAP envelope and identifies the namespaces to which this XML conforms, and the location of their XML Schema Definitions (XSD).
[b03] is the start of the SOAP header where the XML signature fields are located.
[b04] to [b07] is the start of the WSS part and identifies the namespaces for WSS.
[b08] to [b31] is the WSS security token. This is the X.509 certificate the SKSML client uses to authenticate the SOAP message with the SKSML server.
[b32] is the start of the XMLSignature digital signature that ensures the integrity of the SOAP message.
[b33] is the start of the XMLSignature SignedInfo part, i.e. the part that is actually signed. We have added indentation to this part, while in fact it is better practice to form the `<SignedInfo>` element with no whitespace before the elements and just a single newline after each line, in order to avoid some canonicalization issues.
[b34] to [b39] specifies the canonicalization method used, in this case *Exclusive* XML Canonicalization.
[b40] and [b41] specifies the signature algorithm used, in this case SHA1WithRSA.
[b42] to [b48] specifies the digest algorithm used and the digest value of the input string, i.e. the data to be signed, i.e. the contents of the Body element.

[b50] to [b54] is the signature value, i.e. the RSA encrypted digest from the DigestValue element.
[b55] to [b65] is an XMLSignature KeyInfo element that contains a WSS SecurityTokenReference. This is merely a reference to the X.509 certificate specified in the BinarySecurityToken above.
[b67] to [b77] is an WSS TimeStamp token. This is an optional element that specifies a creation time and an expiration time of the enclosing context.
[b80] is the start of the SOAP body that contains the actual SKSML message.
[b84] to [b87] is the SKSML message, which this SOAP message encapsulated the protects using WSS.

# Conformance

Conforming implementations MUST support SOAP v1.2 and WSS v1.0.
Conforming implementations MUST support XML Signature for authenticity and integrity protection of SOAP messages.
Conforming implementations MUST support RSAWithSHA1 and RSAWithSHA256 digital signature algorithms.
Conforming implementations MUST support SHA1 and SHA256 digest algorithms.
Implementations MAY support XML Encryption for confidentiality of SOAP messages.
Implementations MAY support other digital signature and digest algorithms.

A. Acknowledgments

The following individuals have participated in the creation of this specification and are gratefully acknowledged

Participants:

Ezer Farhi, Associate
Benjamin Tomhave, Btplc
Tim Bruce, CA
June Leung, Associate
Shaheen N Abdul Jabbar, Individual
Ken Adler, Individual
Stefan Drees. Individual
Marc Massar, Individual
Michael Nelson, Individual
Davi Ottenheimer, Individual
Allen Schaaf, Individual
Harry Haury, NuParadigm Government Systems, Inc.
Tomas Gustavsson, PrimeKey Solutions AB
Anil Saldhana, Red Hat
Arshad Noor, Associate
Sandi Roddy, US Department of Defense (DoD)
Thomas Hardjono, Associate
Upendra Mardikar, Associate
Eric Lengvenis, Wells Fargo

B. Non-Normative Text

C. Revision History

Version	Date	Author	Notes
DRAFT 1	June 19, 2011	Tomas Gustavsson	Initial version