Web Services Security Rights Expression Language (REL) Token Profile Version 1.1.1

Candidate OASIS Standard 01

15 December 2011

Specification URIs
This version:
http://docs.oasis-open.org/wss-m/wss/v1.1.1/cos01/wss-rel-token-profile-v1.1.1-cos01.doc
(Authoritative)
http://docs.oasis-open.org/wss-m/wss/v1.1.1/cos01/wss-rel-token-profile-v1.1.1-cos01.html
http://docs.oasis-open.org/wss-m/wss/v1.1.1/cos01/wss-rel-token-profile-v1.1.1-cos01.pdf

Previous version:
http://docs.oasis-open.org/wss-m/wss/v1.1.1/csd01/wss-rel-token-profile-v1.1.1-csd01.doc
(Authoritative)
http://docs.oasis-open.org/wss-m/wss/v1.1.1/csd01/wss-rel-token-profile-v1.1.1-csd01.html
http://docs.oasis-open.org/wss-m/wss/v1.1.1/csd01/wss-rel-token-profile-v1.1.1-csd01.pdf

Latest version:
http://docs.oasis-open.org/wss-m/wss/v1.1.1/wss-rel-token-profile-v1.1.1.doc (Authoritative)
http://docs.oasis-open.org/wss-m/wss/v1.1.1/wss-rel-token-profile-v1.1.1.html
http://docs.oasis-open.org/wss-m/wss/v1.1.1/wss-rel-token-profile-v1.1.1.pdf

Technical Committee:
OASIS Web Services Security Maintenance (WSS-M) TC

Chair:
David Turner (david.turner@microsoft.com), Microsoft

Editors:
Ronald Monzillo (ronald.monzillo@sun.com), Sun Microsystems
Chris Kaler (ckaler@microsoft.com), Microsoft
Anthony Nadalin (drolsecure@us.ibm.com), IBM
Phillip Hallam-Baker (pbaker@verisign.com), Verisign
Carlo Milono (cmilono@tibco.com), Tibco
Thomas DeMartini (Thomas.DeMartini@ContentGuard.com), ContentGuard, Inc.

Additional artifacts:
This prose specification is one component of a multi-part Work Product which includes:
- Web Services Security Kerberos Token Profile Version 1.1.1
- Web Services Security Rights Expression Language (REL) Token Profile Version 1.1.1 (this document)
- Web Services Security SAML Token Profile Version 1.1.1
- Web Services Security: SOAP Message Security Version 1.1.1
- Web Services Security SOAP Message with Attachments (SwA) Profile Version 1.1.1
- Web Services Security Username Token Profile Version 1.1.1
- Web Services Security X.509 Certificate Token Profile Version 1.1.1
- XML schemas: http://docs.oasis-open.org/wss-m/wss/v1.1.1/cos01/xsd/
Related work:
This specification supersedes:
- Web Services Security Rights Expression Language (REL) Token Profile 1.1. 01 February 2006. OASIS Standard.

Abstract:
This document describes how to use ISO/IEC 21000-5 Rights Expressions with the Web Services Security (WSS) specification.
This document integrates specific error corrections or editorial changes to the preceding specification, within the scope of the Web Services Security and this TC.
This document introduces a third digit in the numbering convention where the third digit represents a consolidation of error corrections, bug fixes or editorial formatting changes (e.g., 1.1.1); it does not add any new features beyond those of the base specifications (e.g., 1.1).

Status:
This document was last revised or approved by the OASIS Web Services Security Maintenance (WSS-M) 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/wss-m/.
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/wss-m/ipr.php).

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

[WSS-REL-Token-Profile-V1.1.1]
http://docs.oasis-open.org/wss-m/wss/v1.1.1/cos01/wss-rel-token-profile-v1.1.1-cos01.html.
Notices

Copyright © OASIS Open 2012. All Rights Reserved.

All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual Property Rights Policy (the "OASIS IPR Policy"). The full Policy may be found at the OASIS website.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to OASIS, except as needed for the purpose of developing any document or deliverable produced by an OASIS Technical Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

OASIS requests that any OASIS Party or any other party that believes it has patent claims that would necessarily be infringed by implementations of this OASIS Committee Specification or OASIS Standard, to notify OASIS TC Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification.

OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this specification by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification. OASIS may include such claims on its website, but disclaims any obligation to do so.

OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS' procedures with respect to rights in any document or deliverable produced by an OASIS Technical Committee can be found on the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this OASIS Committee Specification or OASIS Standard, can be obtained from the OASIS TC Administrator. OASIS makes no representation that any information or list of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.

The name "OASIS" is a trademark of OASIS, the owner and developer of this specification, and should be used only to refer to the organization and its official outputs. OASIS welcomes reference to, and implementation and use of, specifications, while reserving the right to enforce its marks against misleading uses. Please see http://www.oasis-open.org/who/trademark.php for above guidance.
# Table of Contents

1. Introduction (Informative) .............................................................. 5

2. Notations and Terminology (Normative) ........................................ 6
   2.1 Notational Conventions .......................................................... 6
   2.2 Namespaces ............................................................................. 6
   2.3 Terminology ................................................................. 6

3. Usage (Normative) ........................................................................ 7
   3.1 Token Types ........................................................................... 7
   3.2 Processing Model .................................................................... 7
   3.3 Attaching Security Tokens ....................................................... 7
   3.4 Identifying and Referencing Security Tokens ............................ 7
   3.5 Authentication ....................................................................... 10
     3.5.1 <r:KeyHolder> Principal ............................................... 10
   3.6 Confidentiality ....................................................................... 12
     3.6.1 <r:KeyHolder> Principal ............................................... 12
   3.7 Error Codes ........................................................................... 14

4. Types of Licenses (Informative) .................................................... 15
   4.1 Attribute Licenses ................................................................. 15
   4.2 Sender Authorization ............................................................ 15
   4.3 Issuer Authorization .............................................................. 16

5. Threat Model and Countermeasures (Informative) ......................... 18
   5.1 Eavesdropping ..................................................................... 18
   5.2 Replay .................................................................................. 18
   5.3 Message Insertion ............................................................... 18
   5.4 Message Deletion ............................................................... 18
   5.5 Message Modification ........................................................ 19
   5.6 Man-in-the-Middle .............................................................. 19

6. References .................................................................................. 20

7. Conformance ............................................................................... 21
   A. Acknowledgements .............................................................. 22
   B. Revision History .................................................................... 26
1 Introduction (Informative)

The Web Services Security: SOAP Message Security [WS-Security] specification proposes a standard set of SOAP extensions that can be used when building secure Web services to implement message level integrity and confidentiality. This specification describes the use of ISO/IEC 21000-5 Rights Expressions with respect to the WS-Security specification.
2 Notations and Terminology (Normative)

2.1 Notational Conventions

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [KEYWORDS].

Namespace URIs (of the general form "some-URI") represent some application-dependent or context-dependent URI as defined in [URI].

This specification is designed to work with the general SOAP message structure and message processing model, and should be applicable to any version of SOAP. The current SOAP 1.2 namespace URI is used herein to provide detailed examples, but there is no intention to limit the applicability of this specification to a single version of SOAP.

2.2 Namespaces

The following namespaces are used in this document:

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Namespace</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td><a href="http://www.w3.org/2003/05/soap-envelope">http://www.w3.org/2003/05/soap-envelope</a></td>
</tr>
<tr>
<td>ds</td>
<td><a href="http://www.w3.org/2000/09/xmldsig#">http://www.w3.org/2000/09/xmldsig#</a></td>
</tr>
<tr>
<td>xenc</td>
<td><a href="http://www.w3.org/2001/04/xmlenc#">http://www.w3.org/2001/04/xmlenc#</a></td>
</tr>
<tr>
<td>wsse</td>
<td><a href="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd</a></td>
</tr>
<tr>
<td>wsse11</td>
<td><a href="http://docs.oasis-open.org/wss/oasis-wss-wssecurity-secext-1.1.xsd">http://docs.oasis-open.org/wss/oasis-wss-wssecurity-secext-1.1.xsd</a></td>
</tr>
<tr>
<td>wsu</td>
<td><a href="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd</a></td>
</tr>
<tr>
<td>r</td>
<td>urn:mpeg:mpeg21:2003:01-REL-R-NS</td>
</tr>
<tr>
<td>sx</td>
<td>urn:mpeg:mpeg21:2003:01-REL-SX-NS</td>
</tr>
</tbody>
</table>

Table 1 Namespace Prefixes

2.3 Terminology


Defined below are the basic definitions for additional terminology used in this specification.

License – ISO/IEC 21000-5 Rights Expression
3 Usage (Normative)

This section describes the syntax and processing rules for the use of licenses with the Web Services Security: Soap Message Security specification [WS-Security].

3.1 Token Types

When a URI value is used to indicate a license according to this profile, its value MUST be http://docs.oasis-open.org/wss/oasis-wss-rel-token-profile-1.0.pdf#license.

Note: This URI is for both the ValueType and TokenType attributes. It is also for use by any elements or attributes that require a token type URI and are defined in another specification taking advantage of REL Tokens.

3.2 Processing Model

The processing model for WS-Security with licenses is no different from that of WS-Security with other token formats as described in Web Services Security: SOAP Message Security [WS-Security].

At the token level, a processor of licenses MUST conform to the required validation and processing rules defined in ISO/IEC 21000-5 [REL].

3.3 Attaching Security Tokens

Licenses are attached to SOAP messages using WS-Security by placing the license element inside the <wsse:Security> header. The following example illustrates a SOAP message with a license.

```
<S:Envelope xmlns:S="...">
  <S:Header>
    <wsse:Security xmlns:wsse="...">
      <r:license xmlns:r="...">
      ...
    </r:license>
  </wsse:Security>
  <S:Body>
    ...
  </S:Body>
</S:Envelope>
```

3.4 Identifying and Referencing Security Tokens

The Web Services Security: SOAP Message Security [WS-Security] specification defines the wsu:Id attribute as the common mechanism for identifying security tokens (the specification describes the reasons for this). Licenses have an additional identification mechanism available: their licenseld attribute, the value of which is a URI. The following example shows a license that uses both mechanisms:

```
<r:license xmlns:r="..." xmlns:wsu="...">
  licenseId="urn:foo:SecurityToken:ef375268"
  wsu:Id="SecurityToken-ef375268"
</r:license>
```

Licenses can be referenced either according to their location or their licenseld. Location references are dependent on location and can be either local or remote. Licenseld references are not dependent on location.
Local location references are RECOMMENDED when they can be used. Remote location references are OPTIONAL for cases where it is not feasible to transmit licenses with the SOAP message. LicenseId references are OPTIONAL for cases where location is unknown or cannot be indicated.

WS-Security specifies that tokens are referenced using the <wsse:SecurityTokenReference> element. Implementations compliant with this profile SHOULD set the /wsse:SecurityTokenReference/wsse:Reference/@ValueType attribute to http://docs.oasis-open.org/wss/oasis-wss-rel-token-profile-1.0.pdf#license when using wsse:SecurityTokenReference to refer to a license by licenseId. This is OPTIONAL when referring to a license by location.

The following table demonstrates the use of the <wsse:SecurityTokenReference> element to refer to licenses.

<table>
<thead>
<tr>
<th>By Location</th>
<th>Local</th>
<th>Remote</th>
</tr>
</thead>
</table>
| Location    | <wsse:SecurityTokenReference> |<wsse:Reference URI="#SecurityToken-ef375268"/>
|            |<wsse:SecurityTokenReference> |<wsse:Reference URI="#SecurityToken-ef375268"/>

Table 2. <wsse:SecurityTokenReference>

The following example demonstrates how a <wsse:SecurityTokenReference> can be used to indicate that the message parts specified inside the <ds:SignedInfo> element were signed using a key from the license referenced by licenseId in the <ds:KeyInfo> element.

```
<S:Envelope xmlns:S="..." xmlns:wsse="...">
  <S:Header>
    <wsse:Security xmlns:wsse="...">
      <r:license xmlns:r="..." licenseId="urn:foo:SecurityToken:ef375268" xmlns:wsu="..." wsu:Id="SecurityToken-ef375268">
        ...
      </r:license>
      ...
      <ds:Signature>
        <ds:SignedInfo>
          ...
          <wsse:SecurityTokenReference>
            <wsse:Reference URI="#SecurityToken-ef375268"/>
          </wsse:SecurityTokenReference>
          ...
        </ds:SignedInfo>
        <ds:SignatureValue>...
        </ds:SignatureValue>
        <ds:KeyInfo>
          <wsse:SecurityTokenReference>
            <wsse:Reference URI="#SecurityToken-ef375268"/>
          </wsse:SecurityTokenReference>
          ...
        </ds:KeyInfo>
      </ds:Signature>
    </wsse:Security>
  </S:Header>
  <S:Body>
    ...
  </S:Body>
```

wss-rel-token-profile-v1.1.1-cos01
Standards Track Work Product Copyright © OASIS Open 2012. All Rights Reserved.
15 December 2011
Page 8 of 26
The following example shows a signature over a local license using a location reference to that license.
The example demonstrates how the integrity of an (unsigned) license can be preserved by signing it in
the <wsse:Security> header.

```
<S:Envelope xmlns:S="..." xmlns:wsu="...">
  <S:Header>
    <wsse:Security xmlns:wsse="...">
      <r:license xmlns:r="..." wsu:Id="SecurityToken-ef375268">
        ...
      </r:license>
      ...
      <wsse:SecurityTokenReference wsu:Id="Str1">
        <wsse:Reference
          URI="#SecurityToken-ef375268"/>
      </wsse:SecurityTokenReference>
    ...
    <ds:Signature>
      <ds:SignedInfo>
        ...
        <ds:Reference URI="#Str1">
          <ds:Transform
            Algorithm="http://schemas.xmlsoap.org/2003/06/STR-Transform">
            <ds:CanonicalizationMethod
              Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315"/>
            ...
          <ds:DigestValue>...</ds:DigestValue>
        </ds:Reference>
        ...
      </ds:SignedInfo>
      ...
      <ds:KeyInfo>...</ds:KeyInfo>
    </ds:Signature>
  </wsse:Security>
</S:Header>
<S:Body>
  ...
</S:Body>
</S:Envelope>
```

Note: since licenses allow the use of the wsu:Id attribute, it is usually not necessary to use the STR-
Transform because the license can be referred to directly in the ds:SignedInfo as shown in the following
example:

```
<S:Envelope xmlns:S="..." xmlns:ds="...">
  <S:Header>
    <wsse:Security xmlns:wsse="...">
      <r:license xmlns:r="..." xmlns:wsu="..." wsu:Id="SecurityToken-ef375268">
        ...
      </r:license>
    ...
    <ds:Signature>
      <ds:SignedInfo>
        ...
        <ds:Reference URI="#SecurityToken-ef375268">
          ...
        </ds:Reference>
      </ds:SignedInfo>
    </ds:Signature>
  </wsse:Security>
</S:Header>
<S:Body>
  ...
</S:Body>
</S:Envelope>
```
3.5 Authentication

The Web Services Security: SOAP Message Security [WS-Security] specification does not dictate how claim confirmation must be performed. As well, the REL allows for multiple types of confirmation. This profile of WS-Security REQUIRES that message senders and receivers support claim confirmation for <r:keyHolder> principals. It is RECOMMENDED that an XML Signature be used to establish the relationship between the message sender and the claims. This is especially RECOMMENDED whenever the SOAP message exchange is conducted over an unprotected transport.

The following table enumerates the mandatory principals to be supported by claim confirmation and summarizes their associated processing models. It should be noted that this table is not all-encompassing, and it is envisioned that future specifications may expand this table over time.

<table>
<thead>
<tr>
<th>Principal</th>
<th>RECOMMENDED Processing Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;r:keyHolder&gt;</td>
<td>The message sender adds (to the security header) an XML Signature that can be verified with the key information specified in the &lt;r:keyHolder&gt; of the referenced license.</td>
</tr>
</tbody>
</table>

Table 3. Processing Rules for Claim Confirmation

Note that the high-level processing model described in the following sections does not differentiate between message author and message sender as would be necessary to guard against replay attacks. The high-level processing model also does not take into account requirements for authentication of receiver by sender or for message or token confidentiality. These concerns must be addressed by means other than those described in the high-level processing model. If confidentiality of the token in the message is important, then use the approach defined by [WS-Security] to encrypt the token.

3.5.1 <r:keyHolder> Principal

The following sections describe the <r:keyHolder> method of establishing the correspondence between a SOAP message sender and the claims within a license.

Sender

The message sender MUST include within the <wsse:Security> header element a <r:license> containing at least one <r:grant> to an <r:keyHolder> identifying the key to be used to confirm the claims. If the message sender includes an <r:license> containing more than one <r:grant> to an <r:keyHolder>, then all of those <r:keyHolder> elements MUST be equal.

In order for the receiver to perform claim confirmation, the sender MUST demonstrate knowledge of the confirmation key. The sender MAY accomplish this by using the confirmation key to sign content from...
within the message and by including the resulting <ds:Signature> element in the <wsse:Security> header element. <ds:Signature> elements produced for this purpose MUST conform to the canonicalization and token inclusion rules defined in the core WS-Security specification and this profile specification. Licenses that contain at least one <r:grant> to an <r:keyHolder> SHOULD contain an <r:issuer> with a <ds:Signature> element that identifies the license issuer to the relying party and protects the integrity of the confirmation key established by the license issuer.

**Receiver**

If the receiver determines that the sender has demonstrated knowledge of a confirmation key as specified in an <r:keyHolder>, then the claims (found in the licenses) pertaining to that <r:keyHolder> MAY be attributed to the sender. If one of these claims is an identity and if the conditions of that claim are satisfied, then any elements of the message whose integrity is protected by the confirmation key MAY be considered to have been authored by that identity.

**Example**

The following example illustrates how a license security token having an <r:keyHolder> principal can be used with a <ds:Signature> to establish that John Doe is requesting a stock report on FOO.

```xml
<S:Envelope xmlns:S="...">
  <S:Header>
    <wsse:Security xmlns:wsse="...">
      <r:license xmlns:r="..." licenseId="urn:foo:SecurityToken:ef375268">
        <r:grant>
          <r:keyHolder>
            <r:info>
              <ds:KeyValue>...</ds:KeyValue>
            </r:info>
          </r:keyHolder>
        </r:grant>
        <r:possessProperty/>
        <sx:commonName xmlns:sx="...">John Doe</sx:commonName>
      </r:issuer>
      <ds:Signature>...</ds:Signature>
    </r:license>
    <ds:Signature>
      <ds:SignedInfo>
        
        <ds:Reference URI="#MsgBody">
          
          </ds:Reference>
          </ds:SignedInfo>
          </ds:SignatureValue>
          </ds:KeyInfo>
          </wsse:SecurityTokenReference>
          </wsse:Reference>
          URI="urn:foo:SecurityToken:ef375268"
          ValueType="http://docs.oasis-open.org/wss/oasis-wss-rel-token-profile-1.0.pdf#license"
        </wsse:SecurityTokenReference>
      </ds:KeyInfo>
    </ds:Signature>
  </wsse:Security>
</S:Header>
</S:Envelope>
```
3.6 Confidentiality

This section details how licenses may be used to protect the confidentiality of a SOAP message within WS-Security. The Web Services Security: SOAP Message Security [WS-Security] specification does not dictate how confidentiality must be performed. As well, the REL allows for multiple types of confidentiality. This profile of WS-Security REQUIRES that message senders and receivers support confidentiality for \(<r: KEYHOLDER>\) principals. It is RECOMMENDED that XML Encryption be used to ensure confidentiality. This is especially RECOMMENDED whenever the SOAP message exchange is conducted over an unprotected transport.

The following table enumerates the mandatory principals to be supported for confidentiality and summarizes their associated processing models. It should be noted that this table is not all-encompassing, and it is envisioned that future specifications may expand this table over time.

<table>
<thead>
<tr>
<th>Principal</th>
<th>RECOMMENDED Processing Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>(&lt;r: KEYHOLDER&gt;)</td>
<td>The message sender adds (to the security header) either 1) an (&lt;xenc: ReferenceList&gt;) that points to one or more (&lt;xenc: EncryptedData&gt;) elements that can be decrypted with a key which can be determined from information specified in the (&lt;r: KEYHOLDER&gt;) of the referenced license or 2) an (&lt;xenc: EncryptedKey&gt;) that can be decrypted with a key determined from information specified in the (&lt;r: KEYHOLDER&gt;) of the referenced license.</td>
</tr>
</tbody>
</table>

Table 4. Processing Rules for Confidentiality

Note that this section deals only with Confidentiality. Details of authentication of the sender by the receiver must be addressed by means other than those described in this section (see the previous section).

3.6.1 \(<r: KEYHOLDER>\) Principal

The following sections describe the \(<r: KEYHOLDER>\) method of establishing confidentiality using a license.

Sender

The message sender MUST include within the \(<wsse: Security>\) header element a \(<r: license>\) containing at least one \(<r: grant>\) to an \(<r: KEYHOLDER>\) identifying the key used to encrypt some data or key. If the message sender includes an \(<r: license>\) containing more than one \(<r: grant>\) to an \(<r: KEYHOLDER>\), then all of those \(<r: KEYHOLDER>\) elements MUST be equal.

In order for the receiver to know when to decrypt the data or key, the sender MUST indicate the encryption in the message. The sender MAY accomplish this by placing an \(<xenc: EncryptedData>\) or
<xenc:EncryptedKey> in the appropriate place in the message and by including the resulting
<xenc:ReferenceList> or <xenc:EncryptedKey> element in the <wsse:Security> header element.
<xenc:ReferenceList> or <xenc:EncryptedKey> elements produced for this purpose MUST conform to the
rules defined in the core WS-Security specification and this profile specification.

Receiver

If the receiver determines that he has knowledge of a decryption key as specified in an <r:keyHolder>,
then he MAY decrypt the associated data or key. In the case of decrypting a key, he may then
recursively decrypt any data or key that that key can decrypt.

Example

The following example illustrates how a license containing a <r:keyHolder> principal can be used with
XML encryption schema elements to protect the confidentiality of a message using a separate encryption
key given in the <xenc:EncryptedKey> in the security header.
In this example, the r:license element provides information about the recipient's RSA public key (i.e.,
KeyValue in keyHolder) used to encrypt the symmetric key carried in the EncryptedKey element. The
recipient uses this information to determine the correct private key to use in decrypting the symmetric key.
The symmetric key is then used to decrypt the EncryptedData child of the Body element.

```xml
<S:Envelope xmlns:S="..." xmlns:ds="...">
  <S:Header>
    <wsse:Security xmlns:wsse="...">
      <r:license xmlns:r="..." licenseId="urn:foo:SecurityToken:ef375268">
        <r:grant>
          <r:keyHolder>
            <r:info>
              <ds:KeyValue>...</ds:KeyValue>
            </r:info>
          </r:keyHolder>
          <r:possessProperty/>
          <sx:commonName xmlns:sx="...">SOME COMPANY</sx:commonName>
        </r:grant>
        <r:issuer>
          <ds:Signature>...</ds:Signature>
        </r:issuer>
      </r:license>
      <xenc:EncryptedKey xmlns:xenc="http://www.w3.org/2001/04/xmlenc#">
        <xenc:EncryptionMethod Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-1_5"/>
        <KeyInfo xmlns="http://www.w3.org/2000/09/xmldsig#">
          <wsse:SecurityTokenReference>
            <wsse:Reference URI="urn:foo:SecurityToken:ef375268"/>
          </wsse:SecurityTokenReference>
        </KeyInfo>
        <xenc:CipherData>
          <xenc:CipherValue>dNYS...fQ=</xenc:CipherValue>
        </xenc:CipherData>
        <xenc:ReferenceList>
          <xenc:DataReference URI="#enc"/>
        </xenc:ReferenceList>
      </xenc:EncryptedKey>
    </wsse:Security>
  </S:Header>
    <xenc:EncryptedData Id="enc" Type="http://www.w3.org/2001/04/xmlenc#Content" xmlns:xenc="http://www.w3.org/2001/04/xmlenc#">
      <xenc:EncryptionMethod/>
    </xenc:EncryptedData>
  </S:Body>
</S:Envelope>
```
3.7 Error Codes

It is RECOMMENDED that the error codes defined in the Web Services Security: SOAP Message Security [WS-Security] specification are used. However, implementations MAY use custom errors, defined in private namespaces if they desire. Care should be taken not to introduce security vulnerabilities in the errors returned.
4 Types of Licenses (Informative)

4.1 Attribute Licenses

In addition to key information, licenses can carry information about attributes of those keys. Examples of such information on a client are e-mail address or common name. A service’s key, on the other hand, might be associated with a DNS name and common name.

The following is an example client attribute license.

```xml
<r:license xmlns:r="..." xmlns:ds="..."
licenseId="urn:foo:SecurityToken:ef375268">
  <r:inventory>
    <r:keyHolder licensePartId="client">
      <r:info>
        <ds:KeyValue>FDFEWEFF…</ds:KeyValue>
      </r:info>
      <r:keyHolder/>
    </r:inventory>
    <r:grant>
      <r:keyHolder licensePartIdRef="client"/>
      <r:possessProperty/>
      <sx:commonName>John Doe</sx:commonName>
    </r:grant>
    <r:grant>
      <r:keyHolder licensePartIdRef="client"/>
      <r:possessProperty/>
      <sx:emailName>jd@foo.com</sx:emailName>
    </r:grant>
    <r:issuer>
      <ds:Signature>...</ds:Signature>
    </r:issuer>
  </r:license>
```

The following is an example service attribute license.

```xml
<r:license xmlns:r="..." xmlns:ds="..."
licenseId="urn:foo:SecurityToken:ef375268">
  <r:inventory>
    <r:keyHolder licensePartId="service">
      <r:info>
        <ds:KeyValue>FDFEWEFF…</ds:KeyValue>
      </r:info>
      <r:keyHolder/>
    </r:inventory>
    <r:grant>
      <r:keyHolder licensePartIdRef="service"/>
      <r:possessProperty/>
      <sx:commonName>MyService Company</sx:commonName>
    </r:grant>
    <r:grant>
      <r:keyHolder licensePartIdRef="service"/>
      <r:possessProperty/>
      <sx:dnsName>www.myservice.com</sx:dnsName>
    </r:grant>
    <r:issuer>
      <ds:Signature>...</ds:Signature>
    </r:issuer>
  </r:license>
```

Additional examples of and processing rules for the use of attribute licenses can be found in the above sections on Authentication and Confidentiality.

4.2 Sender Authorization

Licenses may be used by a sender as proof of authorization to perform a certain action on a particular resource. This WS-Security specification does not describe how authorization must be performed. In the
context of web services, a sender can send to a receiver an authorization license in the security header as proof of authorization to call the sender. Typically, this authorization license is signed by a trusted authority and conforms to the syntax pattern specified below.

```
<license xmlns:r="..." licenseId="urn:foo:SecurityToken:ef375268">
  <grant>
    <keyHolder>
      <info>
        <KeyValue>FDFEWFF...</KeyValue>
      </info>
    </keyHolder>
    <rightUri definition='...'/>
    <someResource/>
    <someCondition/>
  </grant>
  <issuer>
    <Signature>...</Signature>
  </issuer>
</license>
```

The above license contains an authorization grant authorizing the keyholder (sender's public key), the right to exercise the right identified in the <rightUri> element. The resource in the license typically corresponds to the semantics of the URI given in the definition attribute of the <rightUri> element. The entire license along with the <Signature> element in the <issuer> certifies the fact that the principal (<keyholder>) is granted the authorization to exercise the right in the <rightUri> element over the specified resource. The integrity of the license is usually protected with a digital signature contained within the <Signature>.

### 4.3 Issuer Authorization

To enunciate that a particular issuer is allowed to issue particular types of licenses, one can use the kind of license described here. Issuer authorization licenses can accompany other licenses in the security header such as those used for authentication, sender authorization, or other issuer authorizations. These issuer authorization licenses might help complete the authorization proof that is required for authorizing or authenticating a particular sender.

The following license is an example issuer authorization license for authorizing an issuer to issue a simple attribute license.

```
<license xmlns:r="..." licenseId="urn:foo:SecurityToken:ef375268">
  <grant>
    <forAll varName='K'/>
    <forAll varName='P'/>
    <keyHolder>
      <info>
        <KeyValue>FDFEWFF...</KeyValue>
      </info>
    </keyHolder>
    <issue/>
    <grant>
      <keyHolder varRef='K'/>
      <possessProperty/>
      <propertyAbstract varRef='P'/>
    </grant>
    <issuer>
      <Signature>...</Signature>
    </issuer>
  </grant>
</license>
```

The following license is an example issuer authorization license for authorizing an issuer to issue sender authorization licenses.

```
<license xmlns:r="..." licenseId="urn:foo:SecurityToken:ef375268">
  <grant>
    <forAll varName='K'/>
    <forAll varName='R'/>
    <keyHolder>
      <info>
        <KeyValue>FDFEWFF...</KeyValue>
      </info>
    </keyHolder>
    <issue/>
    <grant>
      <keyHolder varRef='K'/>
      <possessProperty/>
      <propertyAbstract varRef='R'/>
    </grant>
    <issuer>
      <Signature>...</Signature>
    </issuer>
  </grant>
</license>
```
The following license is an example issuer authorization license for authorizing an issuer to issue (to other issuers) issuer authorization licenses allowing those other issuers to issue simple attribute licenses, such as those that can be used for authentication or confidentiality.
5 Threat Model and Countermeasures (Informative)

This section addresses the potential threats that a SOAP message may encounter and the countermeasures that may be taken to thwart such threats. A SOAP message containing licenses may face threats in various contexts. This includes the cases where the message is in transit, being routed through a number of intermediaries, or during the period when the message is in storage.

The use of licenses with WS-Security introduces no new threats beyond those identified for the REL or WS-Security with other types of security tokens. Message alteration and eavesdropping can be addressed by using the integrity and confidentiality mechanisms described in WS-Security. Replay attacks can be addressed by using of message timestamps and caching, as well as other application-specific tracking mechanisms. For licenses, ownership is verified by the use of keys; man-in-the-middle attacks are generally mitigated. It is strongly RECOMMENDED that all relevant and immutable message data be signed. It should be noted that transport-level security MAY be used to protect the message and the security token. In order to trust licenses, they SHOULD be signed natively and/or using the mechanisms outlined in WS-Security. This allows readers of the licenses to be certain that the licenses have not been forged or altered in any way. It is strongly RECOMMENDED that the <r:license> elements be signed (either within the token, as part of the message, or both).

The following few sections elaborate on the afore-mentioned threats and suggest countermeasures.

5.1 Eavesdropping

Eavesdropping is a threat to the confidentiality of the message, and is common to all types of network protocols. The routing of SOAP messages through intermediaries increases the potential incidences of eavesdropping. Additional opportunities for eavesdropping exist when SOAP messages are persisted.

To provide maximum protection from eavesdropping, licenses, license references, and sensitive message content SHOULD be encrypted such that only the intended audiences can view their content. This removes threats of eavesdropping in transit, but does not remove risks associated with storage or poor handling by the receiver.

Transport-layer security MAY be used to protect the message from eavesdropping while in transport, but message content must be encrypted above the transport if it is to be protected from eavesdropping by intermediaries.

5.2 Replay

The reliance on authority protected (e.g. signed) licenses to <r:keyHolder> principals precludes all but the key holder from binding the licenses to a SOAP message. Although this mechanism effectively restricts message authorship to the holder of the confirmation key, it does not preclude the capture and resubmission of the message by other parties.

Replay attacks can be addressed by using message timestamps and caching, as well as other application-specific tracking mechanisms.

5.3 Message Insertion

This profile of WS-Security is not vulnerable to message insertion attacks. Higher-level protocols built on top of SOAP and WS-Security should avoid introducing message insertion threats and provide proper countermeasures for any they do introduce.

5.4 Message Deletion

This profile of WS-Security is not vulnerable to message deletion attacks other than denial of service. Higher-level protocols built on top of SOAP and WS-Security should avoid introducing message deletion threats and provide proper countermeasures for any they do introduce.
5.5 Message Modification

Message Modification poses a threat to the integrity of a message. The threat of message modification can be thwarted by signing the relevant and immutable content by the key holder. The receivers SHOULD only trust the integrity of those segments of the message that are signed by the key holder.

To ensure that message receivers can have confidence that received licenses have not been forged or altered since their issuance, licenses appearing in <wsse:Security> header elements SHOULD be integrity protected (e.g. signed) by their issuing authority. It is strongly RECOMMENDED that a message sender sign any <r:license> elements that it is confirming and that are not signed by their issuing authority.

Transport-layer security MAY be used to protect the message and contained licenses and/or license references from modification while in transport, but signatures are required to extend such protection through intermediaries.

5.6 Man-in-the-Middle

This profile of WS-Security is not vulnerable to man-in-the-middle attacks. Higher-level protocols built on top of SOAP and WS-Security should avoid introducing Man-in-the-Middle threats and provide proper countermeasures for any they do introduce.
6 References

7 Conformance

An implementation conforms to this specification if it meets the requirements in Sections 2.1, 2.2 and 3.
A. Acknowledgements

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

Participants:

Current Contributors:

- Tom Rutt (Fujitsu Limited)
- Jacques Durand (Fujitsu Limited)
- Calvin Powers (IBM)
- Kelvin Lawrence (IBM)
- Michael McIntosh (Individual)
- Thomas Hardjono (M.I.T.)
- David Turner (Microsoft Corporation)
- Anthony Nadalin (Microsoft Corporation)
- Monica Martin (Microsoft Corporation)
- Marc Goodner (Microsoft Corporation)
- Peter Davis (Neustar)
- Hal Lockhart (Oracle Corporation)
- Rich Levinson (Oracle Corporation)
- Anil Saldhana (Red Hat)
- Martin Raepple (SAP AG)
- Federico Rossini (Telecom Italia S.p.a.)
- Carlo Milono (TIBCO Software Inc.)
- Don Adams (TIBCO Software Inc.)
- Jerry Smith (US Department of Defense (DoD))

Previous Contributors:

- Michael Hu (Actional)
- Maneesh Sahu (Actional)
- Duane Nickull (Adobe Systems)
- Gene Thurston (AmberPoint)
- Frank Siebenlist (Argonne National Laboratory)
- Peter Dapkus (BEA)
- Hal Lockhart (BEA Systems)
- Denis Pilipchuk (BEA Systems)
- Corinna Witt (BEA Systems)
- Steve Anderson (BMC Software)
- Rich Levinson (Computer Associates)
- Thomas DeMartini (ContentGuard)
- Guillermo Lao (ContentGuard)
- TJ Pannu (ContentGuard)
- Xin Wang (ContentGuard)
- Merlin Hughes (Cybertrust)
<table>
<thead>
<tr>
<th>Name</th>
<th>Company</th>
<th>Company Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dale</td>
<td>Moberg</td>
<td>Cyclone Commerce</td>
</tr>
<tr>
<td>Shawn</td>
<td>Sharp</td>
<td>Cyclone Commerce</td>
</tr>
<tr>
<td>Rich</td>
<td>Salz</td>
<td>Datapower</td>
</tr>
<tr>
<td>Ganesh</td>
<td>Vaideeswaran</td>
<td>Documentum</td>
</tr>
<tr>
<td>Sam</td>
<td>Wei</td>
<td>EMC</td>
</tr>
<tr>
<td>John</td>
<td>Hughes</td>
<td>Entegrity</td>
</tr>
<tr>
<td>Tim</td>
<td>Moses</td>
<td>Entrust</td>
</tr>
<tr>
<td>Carolina</td>
<td>Canales-Valenzuela</td>
<td>Ericsson</td>
</tr>
<tr>
<td>Davanum</td>
<td>Srinivas</td>
<td>formerly of Computer Associates</td>
</tr>
<tr>
<td>Mark</td>
<td>Hayes</td>
<td>formerly of VeriSign</td>
</tr>
<tr>
<td>Dana S.</td>
<td>Kaufman</td>
<td>Forum Systems</td>
</tr>
<tr>
<td>Toshihiro</td>
<td>Nishimura</td>
<td>Fujitsu</td>
</tr>
<tr>
<td>Tom</td>
<td>Rutt</td>
<td>Fujitsu</td>
</tr>
<tr>
<td>Kefeng</td>
<td>Chen</td>
<td>GeoTrust</td>
</tr>
<tr>
<td>Irving</td>
<td>Reid</td>
<td>Hewlett-Packard</td>
</tr>
<tr>
<td>Kojiro</td>
<td>Nakayama</td>
<td>Hitachi</td>
</tr>
<tr>
<td>Yutaka</td>
<td>Kudo</td>
<td>Hitachi</td>
</tr>
<tr>
<td>Jason</td>
<td>Rouault</td>
<td>HP</td>
</tr>
<tr>
<td>Paula</td>
<td>Austel</td>
<td>IBM</td>
</tr>
<tr>
<td>Derek</td>
<td>Fu</td>
<td>IBM</td>
</tr>
<tr>
<td>Maryann</td>
<td>Hondo</td>
<td>IBM</td>
</tr>
<tr>
<td>Kelvin</td>
<td>Lawrence</td>
<td>IBM</td>
</tr>
<tr>
<td>Michael</td>
<td>McIntosh</td>
<td>IBM</td>
</tr>
<tr>
<td>Anthony</td>
<td>Nadalin</td>
<td>IBM</td>
</tr>
<tr>
<td>Nataraj</td>
<td>Nagaratnam</td>
<td>IBM</td>
</tr>
<tr>
<td>Bruce</td>
<td>Rich</td>
<td>IBM</td>
</tr>
<tr>
<td>Ron</td>
<td>Williams</td>
<td>IBM</td>
</tr>
<tr>
<td>Bob</td>
<td>Blakley</td>
<td>IBM</td>
</tr>
<tr>
<td>Joel</td>
<td>Farrell</td>
<td>IBM</td>
</tr>
<tr>
<td>Satoshi</td>
<td>Hada</td>
<td>IBM</td>
</tr>
<tr>
<td>Hiroshi</td>
<td>Maruyama</td>
<td>IBM</td>
</tr>
<tr>
<td>David</td>
<td>Melgar</td>
<td>IBM</td>
</tr>
<tr>
<td>Kent</td>
<td>Tamura</td>
<td>IBM</td>
</tr>
<tr>
<td>Wayne</td>
<td>Vicknair</td>
<td>IBM</td>
</tr>
<tr>
<td>Don</td>
<td>Flinn</td>
<td>Individual</td>
</tr>
<tr>
<td>Phil</td>
<td>Griffin</td>
<td>Individual</td>
</tr>
<tr>
<td>Bob</td>
<td>Morgan</td>
<td>Individual/Internet2</td>
</tr>
<tr>
<td>Kate</td>
<td>Cherry</td>
<td>Lockheed Martin</td>
</tr>
<tr>
<td>Paul</td>
<td>Cotton</td>
<td>Microsoft</td>
</tr>
<tr>
<td>Vijay</td>
<td>Gajjala</td>
<td>Microsoft</td>
</tr>
<tr>
<td>Martin</td>
<td>Gudgin</td>
<td>Microsoft</td>
</tr>
<tr>
<td>Chris</td>
<td>Kaler</td>
<td>Microsoft</td>
</tr>
</tbody>
</table>
Bob Atkinson Microsoft
Keith Ballinger Microsoft
Allen Brown Microsoft
Giovanni Della-Libera Microsoft
Alan Geller Microsoft
Johannes Klein Microsoft
Scott Konersmann Microsoft
Chris Kurt Microsoft
Brian LaMacchia Microsoft
Paul Leach Microsoft
John Manferdelli Microsoft
John Shewchuk Microsoft
Dan Simon Microsoft
Hervey Wilson Microsoft
Jeff Hodges Neustar/Sun
Frederick Hirsch Nokia
Senthil Sengodan Nokia
Abbie Barbir Nortel
Lloyd Burch Novell
Ed Reed Novell
Charles Knouse Oblix
Vamsi Motukuru Oracle
Vipin Samar Oracle
Jerry Schwarz Oracle
Prateek Mishra Principal Identity
Eric Gravengaard Reactivity
Stuart King Reed Elsevier
Ben Hammond RSA Security
Rob Philpott RSA Security
Andrew Nash RSA Security
Peter Rostin RSA Security
Martijn de Boer SAP
Blake Dournaee Sarvega
Sundeep Peechu Sarvega
Pete Wenzel SeeBeyond
Jonathan Tourzan Sony
Yassir Elley Sun
Manveen Kaur Sun Microsystems
Ronald Monzillo Sun Microsystems
Jan Alexander Systinet
Michael Nguyen The IDA of Singapore
Don Adams TIBCO
Symon          Chang          TIBCO Software
John           Weiland        US Navy
Hans           Granqvist      VeriSign
Philip         Hallam-Baker   VeriSign
Hemma          Prafullchandra VeriSign
Morten         Jorgensen      Vordel
## B. Revision History

<table>
<thead>
<tr>
<th>Revision</th>
<th>Date</th>
<th>Editor</th>
<th>Changes Made</th>
</tr>
</thead>
<tbody>
<tr>
<td>WD01</td>
<td>17-January-2011</td>
<td>Carlo Milono</td>
<td>Corrected/added hyperlinks where missing; added Status section</td>
</tr>
<tr>
<td>WD02</td>
<td>8-February-2011</td>
<td>Carlo Milono</td>
<td>Added Related Work to reflect v1.1.1 of the specs; changed References for SOAP Message Security to reflect v1.1.1; Changed WD# to 2; Added Date; Moved Current Members to Previous and added new Current Members; saved document under wd02; entered the Revision History Merged Old Current Contributors with Old Previous, created a New Current Contributors.</td>
</tr>
<tr>
<td>CSD01</td>
<td>2-May-2011</td>
<td>TC Admin</td>
<td>Generated from WD02</td>
</tr>
<tr>
<td>CSD02-draft</td>
<td>16-May-11</td>
<td>David Turner</td>
<td>Added conformance statement and corrected a few formatting issues.</td>
</tr>
</tbody>
</table>