



Web Services Security XrML-based Rights Expression Language Token Profile

Working Draft 02, 13 January 2003

Document identifier:

WSS-REL-02

Location:

TBD

Editors:

Phillip Hallam-Baker, Verisign
Chris Kaler, Microsoft
Ronald Monzillo, Sun
Anthony Nadalin, IBM

Contributors:

TEXT TO BE REVISED TO INCLUDE ADDITIONAL CONTRIBUTORS AS NECESSARY

Thomas DeMartini, ContentGuard	Phillip Hallam-Baker, Verisign
Maryann Hondo, IBM	Chris Kaler, Microsoft
Guillermo Lao, ContentGuard	Hiroshi Maruyama, IBM
Anthony Nadalin, IBM	Nataraj Nagaratnam, IBM
TJ Pannu, ContentGuard	Hemma Prefullchandra, VeriSign
John Shewchuck, Microsoft	Xin Wang, ContentGuard

Abstract:

This document describes how to use eXtensible rights Markup Language (XrML)-based Rights Expression Language (REL) licenses with the [WS-Security](#) specification.

Status:

Committee members should send comments on this specification to the wss@lists.oasis-open.org list. Others should subscribe to and send comments to the wss-comment@lists.oasis-open.org list. To subscribe, visit <http://lists.oasis-open.org/ob/adm.pl>.

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing

37 terms, please refer to the Intellectual Property Rights section of the Security
38 Services TC web page at <http://www.oasis-open.org/committees/wss> The
39 OASIS policy on intellectual Property Rights is described at [http://www.oasis-](http://www.oasis-open.org/who/intellectualproperty.shtml)
40 [open.org/who/intellectualproperty.shtml](http://www.oasis-open.org/who/intellectualproperty.shtml).

41 **Table of Contents**

42	1	Introduction.....	4
43	2	Notations and Terminology	5
44	2.1	Notational Conventions	5
45	2.2	Namespaces	5
46	2.3	Terminology	6
47	3	Usage.....	7
48	3.1	Processing Model.....	7
49	3.2	Attaching Security Tokens	7
50	3.3	Identifying and Referencing Security Tokens	8
51	3.4	Proof-of-Possession of Security Tokens.....	12
52	3.5	Error Codes	15
53	3.6	Threat Model and Countermeasures	15
54	4	Acknowledgements	18
55	5	References	19
56		Appendix A. Revision History	20
57		Appendix B. Notices	21
58			

59 **1 Introduction**

60 The [WS-Security](#) specification proposes a standard set of [SOAP](#) extensions that can
61 be used when building secure Web services to implement message level integrity and
62 confidentiality. This specification describes the use of eXtensible rights Markup
63 Language (XrML)-based Rights Expression Language (REL) licenses with respect to
64 the [WS-Security](#) specification.
65

66 2 Notations and Terminology

67
68
69
70

This section specifies the notations, namespaces, and terminology used in this specification.

71 2.1 Notational Conventions

72 The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",
73 "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this
74 document are to be interpreted as described in RFC2119.

75
76
77
78
79
80
81

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.

82 This specification is designed to work with the general XrML2 license structure and
83 processing model, and should be applicable to any XrML2-based rights expression
84 language. The current XrML 2.1 namespace URI is used herein to provide detailed
85 examples, but there is no intention to limit the applicability of this specification to a
86 single version of an XrML2-based rights expression language.
87

88 2.2 Namespaces

89
90
91
92

The XML namespace URIs that MUST be used by implementations of this specification are as follows (note that different elements in this specification are from different namespaces):

93
94
95
96
97
98
99

```
http://schemas.xmlsoap.org/ws/2002/xx/secext  
http://schemas.xmlsoap.org/ws/2002/xx/utility
```

The following namespaces are used in this document:

Prefix	Namespace
S	http://www.w3.org/2001/12/soap-envelope
ds	http://www.w3.org/2000/09/xmldsig#

Prefix	Namespace
xenc	http://www.w3.org/2001/04/xmlenc#
wsse	http://schemas.xmlsoap.org/ws/2002/xx/secext
wsu	http://schemas.xmlsoap.org/ws/2002/xx/utility
r	http://www.xrml.org/schema/2002/05/xrml2core
sx	http://www.xrml.org/schema/2002/05/xrml2sx

Table 1. Namespace Prefixes

100

101

102 **2.3 Terminology**

103

104 This specification employs the terminology defined in the [WS-Security](#) Core
105 Specification.

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

108 [TBS]

109 3 Usage

110 This section describes the profile (specific elements, mechanisms and procedures) for
111 the XrML-based REL Token Profile of [WS-Security](#).

112

113 **Identification:** urn:oasis:names:tc:WSS:1.0:profiles:WSS-REL-profile

114

115 **Contact information:** TBD

116

117 **Description:** Given below.

118

119 **Updates:** None.

120

121 3.1 Processing Model

122

123 The processing model for [WS-Security](#) with licenses is no different from that of
124 [WS-Security](#) with other token formats as described in [WS-Security](#).

125

126 At the token level, a processor of XrML-based REL security tokens MUST conform to
127 the required validation and processing rules defined in the respective REL
128 specification.

129

130 3.2 Attaching Security Tokens

131

132 REL licenses are attached to SOAP messages using [WS-Security](#) by placing the
133 license elements inside the `<wsse:Security>` header. The following example
134 illustrates a SOAP message with a license token.

135

```
136 <S:Envelope xmlns:S="...">  
137   <S:Header>  
138     <wsse:Security xmlns:wsse="...">  
139       <r:license xmlns:r="...">  
140         ...  
141       </r:license>  
142     ...  
143   </wsse:Security>  
144 </S:Header>  
145 <S:Body>  
146   ...  
147 </S:Body>  
148 </S:Envelope>
```

149

150 3.3 Identifying and Referencing Security Tokens

151

152 The [WS-Security](#) specification defines the *wsu:Id* attribute as the common
153 mechanism for identifying security tokens (the specification describes the reasons for
154 this). Licenses have an additional identification mechanism available: their *licenseId*
155 attribute, the value of which is a URI. The following example shows a license that
156 uses both mechanisms:

157

```
158 <r:license xmlns:r="..." xmlns:wsu="..."  
159   licenseId="urn:foo:SecurityToken:ef375268"  
160   wsu:Id="SecurityToken-ef375268">  
161   ...  
162 </r:license>
```

163

164 Licenses can be referenced either according to their *licenseId* or their location.
165 *LicenseId* references are not dependent on location. Location references are
166 dependent on location and can be either local or remote.

167

168 References may occur in three different contexts:

169

170 ? The reference may be contained inside the `<ds:KeyInfo>` element within an
171 XML signature. The reference in this case points to the license that contains
172 the key that was used to sign the digest of the `<ds:SignedInfo>`. The
173 receiver may use this reference to verify the integrity of the
174 `<ds:SignedInfo>`.

175

176 ? The reference may also occur within an element other than the
177 `<ds:Signature>` element. This may be useful to indicate where a service can
178 find other licenses for additional security-related processing.

179

180 ? The license may be referenced from within the `<ds:SignedInfo>` element of
181 an XML signature. To ensure the integrity of the license, a signing authority
182 may sign the license and place the resulting signature within a
183 `<ds:Signature>` element. In this case, the `<ds:SignedInfo>` element of the
184 `<ds:Signature>` contains a `<ds:Reference>` element that points to the
185 license.

186

187 The following few sections demonstrate how to reference licenses from these
188 contexts.

189 **3.3.1 License Referenced from the <ds:KeyInfo>**
 190 **Element of an XML Signature**

191 A license can be referenced from within the <ds:KeyInfo> element of a
 192 <ds:Signature> element. WS-Security specifies that this is accomplished using the
 193 <wsse:SecurityTokenReference> element.

194
 195 Implementations compliant with this profile SHOULD set the
 196 /wsse:SecurityTokenReference/wsse:Reference/@ValueType attribute to r:license
 197 when using wsse:SecurityTokenReference to refer to a license by licenseld. This is
 198 not necessary when referring to a license by location.

199
 200 The following table demonstrates the use of the <wsse:SecurityTokenReference>
 201 element to refer to licenses.
 202

By licenseld		<pre><wsse:SecurityTokenReference> <wsse:Reference URI="urn:foo:SecurityToken:ef375268" ValueType="r:license" /> </wsse:SecurityTokenReference></pre>
By Location	Local	<pre><wsse:SecurityTokenReference> <wsse:Reference URI="#SecurityToken-ef375268" /> </wsse:SecurityTokenReference></pre>
	Remote	<pre><wsse:SecurityTokenReference> <wsse:Reference URI="http://www.foo.com/ef375268.xml" /> </wsse:SecurityTokenReference></pre>

203 **Table 2. <wsse:SecurityTokenReference>**

204
 205 The following example demonstrates how a <wsse:SecurityTokenReference> can be
 206 used to indicate that the message parts specified inside the <ds:SignedInfo>
 207 element were signed using a key from the license referenced by licenseld in the
 208 <ds:KeyInfo> element.

209
 210 <S:Envelope xmlns:S="...">
 211 <S:Header>

```

212 <wsse:Security xmlns:wsse="...">
213   <r:license xmlns:r="..." licenseId="urn:foo:SecurityToken:ef375268">
214     ...
215   </r:license>
216   ...
217   <ds:Signature>
218     <ds:SignedInfo>
219       ...
220     </ds:SignedInfo>
221     <ds:SignatureValue>...</ds:SignatureValue>
222     <ds:KeyInfo>
223       <wsse:SecurityTokenReference>
224         <wsse:Reference
225           URI="urn:foo:SecurityToken:ef375268"
226           ValueType="r:license"
227         />
228       </wsse:SecurityTokenReference>
229     </ds:KeyInfo>
230   </ds:Signature>
231 </wsse:Security>
232 </S:Header>
233 <S:Body>
234   ...
235 </S:Body>
236 </S:Envelope>

```

237 **3.3.2 License Referenced from Elements Other Than** 238 **<ds:Signature>**

239 A license can be referenced from elements other than <ds:Signature>. WS-Security
240 specifies that this is accomplished using the <wsse:SecurityTokenReference>
241 element. (For details on the use of the <wsse:SecurityTokenReference> element to
242 refer to licenses, please see Table 2 in 3.3.1).

243
244 The following example demonstrates how a <wsse:SecurityTokenReference> can be
245 used to refer to a license from directly within the <wsse:Security> header element
246 (just one such element that is an element other than a <ds:Signature>). In this
247 case, we choose to show a location reference to a remote license.

```

249 <S:Envelope xmlns:S="...">
250   <S:Header>
251     <wsse:Security xmlns:wsse="...">
252       ...
253     <wsse:SecurityTokenReference>
254       <wsse:Reference
255         URI="http://www.foo.com/ef375268.xml"
256       />
257     </wsse:SecurityTokenReference>

```

```

258     ...
259     </wsse:Security>
260 </S:Header>
261 <S:Body>
262     ...
263 </S:Body>
264 </S:Envelope>
265

```

3.3.3 License Referenced from the <ds:SignedInfo> Element of an XML Signature

266 A license can be referenced from within the <ds:SignedInfo> element of a
 267 <ds:Signature> element. DIGSIG specifies that this is accomplished using the
 268 <ds:Reference> element. The following table demonstrates the use of the
 269 <ds:Reference> element to refer to licenses.
 270
 271
 272

By licenseld		<pre> <ds:Reference URI="urn:foo:SecurityToken:ef375268"> <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" /> <ds:DigestValue>...</ds:DigestValue> </ds:Reference> </pre>
By Location	Local	<pre> <ds:Reference URI="#SecurityToken-ef375268"> <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" /> <ds:DigestValue>...</ds:DigestValue> </ds:Reference> </pre>
	Remote	<pre> <ds:Reference URI="http://www.foo.com/ef375268.xml"> <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" /> <ds:DigestValue>...</ds:DigestValue> </ds:Reference> </pre>

Table 3. <ds:Reference>

273
 274

275 The following example shows a signature over a local license using a location
276 reference to that license. The example demonstrates how the integrity of an
277 (unsigned) license can be preserved by signing it in the <wsse:Security> header.
278

```
279 <S:Envelope xmlns:S="...">
280   <S:Header>
281     <wsse:Security xmlns:wsse="...">
282       <r:license xmlns:r="..." xmlns:wsu="..." wsu:Id="SecurityToken-ef375268">
283         ...
284       </r:license>
285       ...
286     <ds:Signature>
287       <ds:SignedInfo>
288         ...
289         <ds:Reference URI="#SecurityToken-ef375268">
290           <ds:DigestMethod
291             Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"
292           />
293           <ds:DigestValue>...</ds:DigestValue>
294         </ds:Reference>
295       </ds:SignedInfo>
296     <ds:SignatureValue>...</ds:SignatureValue>
297     <ds:KeyInfo>...</ds:KeyInfo>
298   </ds:Signature>
299 </wsse:Security>
300 </S:Header>
301 <S:Body>
302   ...
303 </S:Body>
304 </S:Envelope>
```

305 **3.4 Proof-of-Possession of Security Tokens**

306 The [WS-Security](#) specification does not dictate how claim confirmation must be
307 performed. As well, XrML-based RELs allow for multiple types of confirmation. The
308 REL profile of WS-Security requires that message senders and receivers support
309 claim confirmation for <r:keyHolder> principals. It is strongly RECOMMENDED that
310 an XML Signature be used to establish the relationship between the message sender
311 and the claims. This is especially RECOMMENDED whenever the SOAP message
312 exchange is conducted over an unprotected transport.
313

314 The following table enumerates the mandatory principals to be supported by claim
315 confirmation and summarizes their associated processing models. It should be noted
316 that this table is not all-encompassing, and it is envisioned that future specifications
317 may expand this table over time.
318

Principal	RECOMMENDED Processing Rules
<r:keyHolder>	The message sender adds (to the security header) an XML Signature that can be verified with the key information specified in the <r:keyHolder> of the referenced REL license.

Table 4. Processing Rules for Claim Confirmation

319

320

321 Note that the high-level processing model described in the following sections does
 322 not differentiate between message author and message sender as would be
 323 necessary to guard against replay attacks. The high-level processing model also does
 324 not take into account requirements for authentication of receiver by sender or for
 325 message or token confidentiality. These concerns must be addressed by means other
 326 than those described in the high-level processing model.

327 **3.4.1 <r:keyHolder> Principal**

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

331 **3.4.1.1 Sender**

332 The message sender **MUST** include within the <wsse:Security> header element a
 333 <r:license> containing at least one <r:grant> to an <r:keyHolder> identifying the
 334 key to be used to confirm the claims.

335

336 In order for the receiver to perform claim confirmation, the sender **MUST**
 337 demonstrate knowledge of the confirmation key. The sender **MAY** accomplish this by
 338 using the confirmation key to sign content from within the message and by including
 339 the resulting <ds:Signature> element in the <wsse:Security> header element.
 340 <ds:Signature> elements produced for this purpose **MUST** conform to the
 341 canonicalization and token inclusion rules defined in the core WS-Security
 342 specification and this profile specification.

343

344 Licenses that contain at least one <r:grant> to an <r:keyHolder> **SHOULD** contain
 345 an <r:issuer> with a <ds:Signature> element that protects the integrity of the
 346 confirmation key established by the license issuer.

347 **3.4.1.2 Receiver**

348 If the receiver determines that the sender has demonstrated knowledge of a
 349 confirmation key as specified in an <r:keyHolder>, then the claims (found in the

350 licenses) pertaining to that <r:keyHolder> MAY be attributed to the sender. If one
351 of these claims is an identity and if the conditions of that claim are satisfied, then
352 any elements of the message whose integrity is protected by the confirmation key
353 MAY be considered to have been authored by that identity.

354 3.4.1.3 Example

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

```
358 <S:Envelope xmlns:S="...">
359   <S:Header>
360     <wsse:Security xmlns:wsse="...">
361       <r:license xmlns:r="..." licenseId="urn:foo:SecurityToken:ef375268">
362         <r:grant>
363           <r:keyHolder>
364             <r:info>
365               <ds:KeyValue>...</ds:KeyValue>
366             </r:info>
367           </r:keyHolder>
368           <r:possessProperty/>
369           <sx:commonName xmlns:sx="...">John Doe</sx:commonName>
370         </r:grant>
371         <r:issuer>
372           <ds:Signature>...</ds:Signature>
373         </r:issuer>
374       </r:license>
375     </S:Header>
376     <ds:Signature>
377       <ds:SignedInfo>
378         ...
379         <ds:Reference URI="#MsgBody">
380           <ds:DigestMethod
381             Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"
382           />
383           <ds:DigestValue>...</ds:DigestValue>
384         </ds:Reference>
385       </ds:SignedInfo>
386     </ds:Signature>
387     <ds:SignatureValue>...</ds:SignatureValue>
388     <ds:KeyInfo>
389       <wsse:SecurityTokenReference>
390         <wsse:Reference
391           URI="urn:foo:SecurityToken:ef375268"
392           ValueType="r:license"
393         />
394     </ds:KeyInfo>
395   </S:Envelope>
```

```
396     </wsse:SecurityTokenReference>
397     </ds:KeyInfo>
398   </ds:Signature>
399
400   </wsse:Security>
401 </S:Header>
402
403 <S:Body @wsu:Id="MsgBody" xmlns:wsu="...">
404   <ReportRequest>
405     <TickerSymbol>FOO</TickerSymbol>
406   </ReportRequest>
407 </S:Body>
408
409 </S:Envelope>
410
```

411 **3.5 Error Codes**

412 It is RECOMMENDED to use the error codes defined in the [WS-Security](#) specification.
413 However, implementations MAY use custom errors, defined in private namespaces if
414 they desire. Care should be taken not to introduce security vulnerabilities in the
415 errors returned.
416

417 **3.6 Threat Model and Countermeasures**

418 This section addresses the potential threats that a SOAP message may encounter
419 and the countermeasures that may be taken to thwart such threats. A SOAP
420 message containing XrML-based REL licenses may face threats in various contexts.
421 This includes the cases where the message is in transit, being routed through a
422 number of intermediaries, or during the period when the message is in storage.
423

424 The use of XrML-based REL licenses with WS-Security introduces no new threats
425 beyond those identified for the XrML-based REL or WS-Security with other types of
426 security tokens. Message alteration and eavesdropping can be addressed by using
427 the integrity and confidentiality mechanisms described in WS-Security. Replay
428 attacks can be addressed by using of message timestamps and caching, as well as
429 other application-specific tracking mechanisms. For XrML-based REL licenses
430 ownership is verified by use of keys, man-in-the-middle attacks are generally
431 mitigated. It is strongly RECOMMENDED that all relevant and immutable message
432 data be signed. It should be noted that transport-level security MAY be used to
433 protect the message and the security token. In order to trust license tokens, they
434 SHOULD be signed natively and/or using the mechanisms outlined in WS-Security.
435 This allows readers of the tokens to be certain that the tokens have not been forged
436 or altered in any way. It is strongly RECOMMENDED that the <r:license> elements
437 be signed (either within the token, as part of the message, or both).
438

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

441 **3.6.1 Eavesdropping**

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

446
447 To provide maximum protection from eavesdropping, licenses, license references,
448 and sensitive message content SHOULD be encrypted such that only the intended
449 audiences can view their content. This removes threats of eavesdropping in transit,
450 but does not remove risks associated with storage or poor handling by the receiver.

451
452 Transport-layer security MAY be used to protect the message from eavesdropping
453 while in transport, but message content must be encrypted above the transport if it
454 is to be protected from eavesdropping by intermediaries.

455 **3.6.2 Replay**

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

461
462 Replay attacks can be addressed by using message timestamps and caching, as well
463 as other application-specific tracking mechanisms.

464 **3.6.3 Message Insertion**

465 The XrML-based REL token profile of WS-Security is not vulnerable to message
466 insertion attacks. Higher-level protocols built on top of SOAP and WS-Security should
467 avoid introducing message insertion threats and provide proper countermeasures for
468 any they do introduce.

469 **3.6.4 Message Deletion**

470 The XrML-based REL token profile of WS-Security is not vulnerable to message
471 deletion attacks. Higher-level protocols built on top of SOAP and WS-Security should
472 avoid introducing message deletion threats and provide proper countermeasures for
473 any they do introduce.

474 **3.6.5 Message Modification**

475 Message Modification poses a threat to the integrity of a message. The threat of
476 message modification can be thwarted by signing the relevant and immutable

477 content by the key holder. The receivers SHOULD only trust the integrity of those
478 segments of the message that are signed by the key holder.

479

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

486

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

490 **3.6.6 Man-in-the-Middle**

491 The XrML-based REL token profile of WS-Security is not vulnerable to man-in-the-
492 middle attacks. Higher-level protocols built on top of SOAP and WS-Security should
493 avoid introducing Man-in-the-Middle threats and provide proper countermeasures for
494 any they do introduce.

495

496 **4 Acknowledgements**

497 This specification was developed as a result of joint work of many individuals from
498 the WSS TC including:
499 TBD
500

501

5 References

502 **[DIGSIG]** Informational RFC 2828, "[Internet Security Glossary](#)," May 2000.
503 **[KEYWORDS]** S. Bradner, "Key words for use in RFCs to Indicate Requirement
504 Levels," [RFC 2119](#), Harvard University, March 1997.
505 **[MPEG-REL]** Text of ISO/IEC Final Committee Draft 21000-5 Rights Expression
506 Language, December 2002.
507 **[SOAP]** W3C Note, "SOAP: Simple Object Access Protocol 1.1," 08 May 2000.
508 W3C Working Draft, Nilo Mitra (Editor), [SOAP Version 1.2 Part 0: Primer](#), June 2002.
509 W3C Working Draft, [SOAP Version 1.2 Part 1: Messaging Framework](#), Martin Gudgin,
510 Marc Hadley, Noah Mendelsohn, Jean-Jacques Moreau, Henrik Frystyk Nielsen
511 (Editors), June 2002.
512 W3C Working Draft, Martin Gudgin, Marc Hadley, Noah Mendelsohn, Jean-Jacques
513 Moreau, Henrik Frystyk Nielsen (Editors), [SOAP Version 1.2 Part 2: Adjuncts](#), June
514 2002.
515 **[URI]** T. Berners-Lee, R. Fielding, L. Masinter, "Uniform Resource Identifiers (URI):
516 Generic Syntax," [RFC 2396](#), MIT/LCS, U.C. Irvine, Xerox Corporation, August 1998.
517 224
518 **[WS-Security]** TBS – point to the OASIS core draft
519 **[XML-ns]** W3C Recommendation, "[Namespaces in XML](#)," 14 January 1999.
520 **[XML Signature]** W3C Recommendation, "[XML Signature Syntax and Processing](#),"
521 12 February 2002.
522 **[XML Token]** Contribution to the WSS TC, Chris Kaler (Editor), WS-Security Profile
523 for XML-based Tokens, August 2002. 230
524 **[XrML]** ContentGuard, eXtensible rights Markup Language Core 2.1 Specification, 20
525 May 2002.

526

Appendix A. Revision History

Rev	Date	What
01	19-Sep-02	Initial draft produced by extracting XrML related content from [XML token]
02	13-Jan-03	Cleaned up, fleshed out, added examples.

527

Appendix B. Notices

529 OASIS takes no position regarding the validity or scope of any intellectual property
530 or other rights that might be claimed to pertain to the implementation or use of the
531 technology described in this document or the extent to which any license under such
532 rights might or might not be available; neither does it represent that it has made any
533 effort to identify any such rights. Information on OASIS's procedures with respect to
534 rights in OASIS specifications can be found at the OASIS website. Copies of claims of
535 rights made available for publication and any assurances of licenses to be made
536 available, or the result of an attempt made to obtain a general license or permission
537 for the use of such proprietary rights by implementors or users of this specification,
538 can be obtained from the OASIS Executive Director.

539 OASIS invites any interested party to bring to its attention any copyrights, patents or
540 patent applications, or other proprietary rights which may cover technology that may
541 be required to implement this specification. Please address the information to the
542 OASIS Executive Director.

543 Copyright © The Organization for the Advancement of Structured Information
544 Standards [OASIS] 2002. All Rights Reserved.

545 This document and translations of it may be copied and furnished to others, and
546 derivative works that comment on or otherwise explain it or assist in its
547 implementation may be prepared, copied, published and distributed, in whole or in
548 part, without restriction of any kind, provided that the above copyright notice and
549 this paragraph are included on all such copies and derivative works. However, this
550 document itself does not be modified in any way, such as by removing the copyright
551 notice or references to OASIS, except as needed for the purpose of developing
552 OASIS specifications, in which case the procedures for copyrights defined in the
553 OASIS Intellectual Property Rights document must be followed, or as required to
554 translate it into languages other than English.

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

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