



Web Services Security X.509 Certificate Token Profile

Monday, 15 March 2004

Document identifier:

{WSS: SOAP Message Security }-{X509 Profile }-{1.0} (Word) (PDF)

Location:

<http://www.docs.oasis-open.org/wss/2003/12/oasis-200401-wss-x509-token-profile-1.0>

<http://www.oasis-open.org/committees/documents.php>

Editors:

Phillip Hallam-Baker, VeriSign

Chris Kaler, Microsoft

Ronald Monzillo, Sun

Anthony Nadalin, IBM

Contributors:

Gene	Thurston	AmberPoint
Frank	Siebenlist	Argonne National Lab
Merlin	Hughes	Baltimore Technologies
Irving	Reid	Baltimore Technologies
Peter	Dapkus	BEA
Hal	Lockhart	BEA
Symon	Chang	CommerceOne
Srinivas	Davanum	Computer Associates
Thomas	DeMartini	ContentGuard
Guillermo	Lao	ContentGuard
TJ	Pannu	ContentGuard
Shawn	Sharp	Cyclone Commerce
Ganesh	Vaideeswaran	Documentum
Sam	Wei	Documentum
John	Hughes	Entegrity
Tim	Moses	Entrust
Toshihiro	Nishimura	Fujitsu
Tom	Rutt	Fujitsu
Jason	Rouault	HP
Yutaka	Kudo	Hitachi
Paula	Austel	IBM
Maryann	Hondo	IBM
Michael	McIntosh	IBM
Kelvin	Lawrence	IBM (co-Chair)
Anthony	Nadalin	IBM
Nataraj	Nagaratnam	IBM
Don	Flinn	Individual

43	Bob	Morgan	Individual
44	Paul	Cotton	Microsoft
45	Vijay	Gajjala	Microsoft
46	Chris	Kaler	Microsoft (co-Chair)
47	Chris	Kurt	Microsoft
48	John	Shewchuk	Microsoft
49	Prateek	Mishra	Netegrity
50	Frederick	Hirsch	Nokia
51	Senthil	Sengodan	Nokia
52	Lloyd	Burch	Novell
53	Ed	Reed	Novell
54	Charles	Knouse	Oblix
55	Steve	Anderson	OpenNetwork (Sec)
56	Vipin	Samar	Oracle
57	Jerry	Schwarz	Oracle
58	Eric	Gravengaard	Reactivity
59	Stuart	King	Reed Elsevier
60	Andrew	Nash	RSA Security
61	Rob	Philpott	RSA Security
62	Peter	Rostin	RSA Security
63	Martijn	de Boer	SAP
64	Blake	Dournaee	Sarvega
65	Pete	Wenzel	SeeBeyond
66	Jonathan	Tourzan	Sony
67	Yassir	Elley	Sun Microsystems
68	Jeff	Hodges	Sun Microsystems
69	Ronald	Monzillo	Sun Microsystems
70	Jan	Alexander	Systinet
71	Michael	Nguyen	The IDA of Singapore
72	Don	Adams	TIBCO
73	John	Weiland	US Navy
74	Phillip	Hallam-Baker	VeriSign
75	Morten	Jorgensen	Vordel

Contributors of input documents (if not already listed above) :

76	Bob	Blakley	IBM
77	Joel	Farrell	IBM
78	Satoshi	Hada	IBM
79	Hiroshi	Maruyama	IBM
80	David	Melgar	IBM
81	Bob	Atkinson	Microsoft
82	Allen	Brown	Microsoft
83	Giovanni	Della-Libera	Microsoft
84	Johannes	Klein	Microsoft
85	Scott	Konersmann	Microsoft
86	Brian	LaMacchia	Microsoft
87	Paul	Leach	Microsoft
88	John	Manferdelli	Microsoft
89	Dan	Simon	Microsoft
90	Hervey	Wilson	Microsoft
91	Hemma	Prafullchandra	VeriSign

Abstract:

94 This document describes how to use X.509 Certificates with the Web Services Security:
95 SOAP Message Security specification [WS-Security] specification.

96 **Status:**
97 This is an interim draft.
98 Committee members should send comments on this specification to the [wss@lists.oasis-](mailto:wss@lists.oasis-open.org)
99 [open.org](mailto:wss@lists.oasis-open.org) list. Others should subscribe to and send comments to the [comment@lists.oasis-open.org](mailto:wss-
100 <a href=) list. To subscribe, visit [open.org/ob/adm.pl](http://lists.oasis-
101 <a href=).
102 For information on whether any patents have been disclosed that may be essential to
103 implementing this specification, and any offers of patent licensing terms, please refer to
104 the Intellectual Property Rights section of the WS-Security TC web page
105 (<http://www.oasis-open.org/committees/wss/ipr.php>).

Table of Contents

107	1	Introduction (Non-Normative)	2
108	2	Notations and Terminology (Normative)	2
109	2.1	Notational Conventions	2
110	2.2	Namespaces	2
111	2.3	Terminology.....	2
112	3	Usage (Normative).....	2
113	3.1	Token types.....	2
114	3.1.1	#X509v3 Token Type.....	2
115	3.1.2	#X509PKIPathv1 Token Type	2
116	3.1.3	#PKCS7 Token Type	2
117	3.2	Token References.....	2
118	3.2.1	Reference to a Subject Key Identifier	2
119	3.2.2	Reference to a Security Token	2
120	3.2.3	Reference to an Issuer and Serial Number	2
121	3.3	Signature	2
122	3.3.1	Key Identifier	2
123	3.3.2	Reference to a Binary Security Token	2
124	3.3.3	Reference to an Issuer and Serial Number	2
125	3.4	Encryption	2
126	3.5	Error Codes	2
127	4	Threat Model and Countermeasures (Non-Normative)	2
128	5	References.....	2
129		Appendix A: Revision History	2
130		Appendix B: Notices	2
131			

132 **1 Introduction (Non-Normative)**

133 This specification describes the use of the X.509 authentication framework with the Web Services
134 Security: SOAP Message Security specification [WS-Security].

135 An X.509 certificate specifies a binding between a public key and a set of attributes that includes
136 (at least) a subject name, issuer name, serial number and validity interval. This binding may be
137 subject to subsequent revocation advertised by mechanisms that include issuance of CRLs,
138 OCSP tokens or mechanisms that are outside the X.509 framework, such as XKMS.

139 An X.509 certificate may be used to validate a public key that may be used to authenticate a
140 SOAP message or to identify the public key with SOAP message that has been encrypted.

2 Notations and Terminology (Normative)

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

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 RFC 2119.

When describing abstract data models, this specification uses the notational convention used by the XML Infoset. Specifically, abstract property names always appear in square brackets (e.g., [some property]).

When describing concrete XML schemas, this specification uses a convention where each member of an element's [children] or [attributes] property is described using an XPath-like notation (e.g., /x:MyHeader/x:SomeProperty/@value1). The use of {any} indicates the presence of an element wildcard (<xs:any/>). The use of @{any} indicates the presence of an attribute wildcard (<xs:anyAttribute/>).

2.2 Namespaces

The XML Namespace [XML-ns] URIs that MUST be used by implementations of this specification are as follows (note that elements used in this specification are defined in one or other of these namespaces):

```
http://www.docs.oasis-open.org/wss/2004/01/oasis-200401-wss-  
wssecurity-secext-1.0.xsd  
http://www.docs.oasis-open.org/wss/2004/01/oasis-200401-wss-  
wssecurity-utility-1.0.xsd
```

The following namespace prefixes are used in this document:

Prefix	Namespace
S11	http://schemas.xmlsoap.org/soap/envelope/
S12	http://www.w3.org/2003/05/soap-envelope
ds	http://www.w3.org/2000/09/xmldsig#
xenc	http://www.w3.org/2001/04/xmlenc#
wsse	http://www.docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd
wsu	http://www.docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd

166

Table 1- Namespace prefixes

167 **2.3 Terminology**

168 This specification adopts the terminology defined in Web Services Security: SOAP Message
169 Security specification [WS-Security].

170 Readers are presumed to be familiar with the definitions of terms in the Internet Security Glossary
171 [Glossary].

172 3 Usage (Normative)

173 This specification describes the syntax and processing rules for the use of the X.509
174 authentication framework with the Web Services Security: SOAP Message Security specification
175 [WS-Security].

176 3.1 Token types

177 This profile defines the syntax of, and processing rules for, three types of binary security token
178 using the URI values specified in Table 2 (note that URI fragments are relative to the URI for this
179 specification).

180

Token	ValueType URI	Description
Single certificate	#X509v3	An X.509 v3 signature-verification certificate
Certificate Path	#X509PKIPathv1	An ordered list of X.509 certificates packaged in a PKIPath
Set of certificates and CRLs	#PKCS7	A list of X.509 certificates and (optionally) CRLs packaged in a PKCS#7 wrapper

181

Table 2 – Token types

182 3.1.1 X509v3 Token Type

183 The type of the end-entity that is authenticated by a certificate used in this manner is a matter of
184 policy that is outside the scope of this specification.

185 3.1.2 X509PKIPathv1 Token Type

186 The #X509PKIPathv1 token type MAY be used to represent a certificate path.

187 3.1.3 PKCS7 Token Type

188 The #PKCS7 token type MAY be used to represent a certificate path. It is RECOMMENDED that
189 applications use the PKIPath object for this purpose instead.

190 The order of the certificates in a PKCS#7 data structure is not significant. If an ordered certificate
191 path is converted to PKCS#7 encoded bytes and then converted back, the order of the
192 certificates may not be preserved. Processors SHALL NOT assume any significance to the order
193 of the certificates in the data structure. See [PKCS7] for more information.

194 3.2 Token References

195 In order to ensure a consistent processing model across all the token types supported by WSS:
196 SOAP Message Security, the <wsse:SecurityTokenReference> element SHALL be used to
197 specify all references to X.509 token types in signature or encryption elements that comply with
198 this profile.

199

200 A <wsse:SecurityTokenReference> element MAY reference an X.509 token type by one of
201 the following means:

202 Reference to a Subject Key Identifier

203 The <wsse:SecurityTokenReference> element contains a
204 <wsse:KeyIdentifier> element that specifies the token data by means of a X.509
205 SubjectKeyIdentifier reference.

206 Reference to a Binary Security Token

207 The <wsse:SecurityTokenReference> element contains a <wsse:Reference>
208 element that references a local <wsse:BinarySecurityToken> element or a remote
209 data source that contains the token data itself.

210 Reference to an Issuer and Serial Number

211 The <wsse:SecurityTokenReference> element contains a <ds:X509Data> element
212 that contains a <ds:X509IssuerSerial> element that uniquely identifies an end
213 entity certificate by its X.509 Issuer and Serial Number.

214 3.2.1 Reference to a Subject Key Identifier

215 The <wsse:KeyIdentifier> element is used to specify a reference to an X.509 certificate by
216 means of a reference to its X.509 SubjectKeyIdentifier attribute. This profile defines the syntax of,
217 and processing rules for referencing a Subject Key Identifier using the URI values specified in
218 Table 3 (note that URI fragments are relative to the URI for this specification).

219

Subject Key Identifier	ValueType URI	Description
Certificate Key Identifier	#X509SubjectKeyIdentifier	Value of the certificate's X.509 SubjectKeyIdentifier

220

Table 3 – Subject Key Identifier

221 The <wsse:SecurityTokenReference> element from which the reference is made contains
222 the <wsse:KeyIdentifier> element. The <wsse:KeyIdentifier> element MUST have a
223 ValueType attribute with the value #X509SubjectKeyIdentifier and its contents MUST be the
224 value of the certificate's X.509 SubjectKeyIdentifier extension, encoded as per the
225 <wsse:KeyIdentifier> element's EncodingType attribute. For the purposes of this
226 specification, the value of the SubjectKeyIdentifier extension is the contents of the KeyIdentifier
227 octet string, excluding the encoding of the octet string prefix.

228 3.2.2 Reference to a Security Token

229 The <wsse:Reference> element is used to reference an X.509 security token value by means of
230 a URI reference.

231 The URI reference MAY be internal in which case the URI reference SHOULD be a bare name
232 XPointer reference to a <wsse:BinarySecurityToken> element contained in a preceding
233 message header that contains the binary X.509 security token data.

234 3.2.3 Reference to an Issuer and Serial Number

235 The <ds:X509IssuerSerial> element is used to specify a reference to an X.509 security
236 token by means of the certificate issuer name and serial number.

237 The <ds:X509IssuerSerial> element is a direct child of the <ds:X509Data> element that is
238 in turn a direct child of the <wsse:SecurityTokenReference> element in which the
239 reference is made.

240 3.3 Signature

241 Signed data MAY specify the certificate associated with the signature using any of the X.509
242 security token types and references defined in this specification.

243 An X.509 certificate specifies a binding between a public key and a set of attributes that includes
244 (at least) a subject name, issuer name, serial number and validity interval. Other attributes may
245 specify constraints on the use of the certificate or affect the recourse that may be open to a
246 relying party that depends on the certificate. A given public key may be specified in more than
247 one X.509 certificate; consequently a given public key may be bound to two or more distinct sets
248 of attributes.

249 It is therefore necessary to ensure that a signature created under an X.509 certificate token
250 uniquely and irrefutably specifies the certificate under which the signature was created.

251 Implementations SHOULD protect against a certificate substitution attack by including either the
252 certificate itself or an immutable and unambiguous reference to the certificate within the scope of
253 the signature according to the method used to reference the certificate as described in the
254 following sections.

255 3.3.1 Key Identifier

256 The <wsse:KeyIdentifier> element does not guarantee an immutable and unambiguous
257 reference to the certificate referenced. Consequently implementations that use this form of
258 reference within a signature SHOULD employ the STR Dereferencing Transform within a
259 reference to the signature key information in order to ensure that the referenced certificate is
260 signed, and not just the ambiguous reference. The form of the reference is a bare name
261 reference as defined by the XPointer specification [XPointer].

262 The following example shows a certificate referenced by means of a KeyIdentifier. The scope of
263 the signature is the <ds:SignedInfo> element which includes both the message body (#body)
264 and the signing certificate by means of a reference to the <ds:KeyInfo> element which
265 references it (#keyinfo). Since the <ds:KeyInfo> element only contains a mutable reference to
266 the certificate rather than the certificate itself, a transformation is specified which replaces the
267 reference to the certificate with the certificate. The <ds:KeyInfo> element specifies the signing
268 key by means of a <wsse:SecurityTokenReference> element which contains a
269 <wsse:KeyIdentifier> element which specifies the X.509 subject key identifier of the signing
270 certificate.

```
271 <S11:Envelope xmlns:S11="...">  
272   <S11:Header>  
273     <wsse:Security  
274       xmlns:wsse="..."  
275       xmlns:wsu="...">  
276       <ds:Signature  
277         xmlns:ds="http://www.w3.org/2000/09/xmldsig#">  
278         <ds:SignedInfo>...  
279           <ds:Reference URI="#body">...</ds:Reference>  
280           <ds:Reference URI="#keyinfo">  
281             <ds:Transforms>  
282               <ds:Transform Algorithm="...#STR-Transform">  
283                 <wsse:TransformationParameters>  
284                   <ds:CanonicalizationMethod Algorithm="..." />  
285                 </wsse:TransformationParameters>  
286               </ds:Transform>  
287             </ds:Transforms>...
```

```

288         </ds:Reference>
289     </ds:SignedInfo>
290     <ds:SignatureValue>HFLP...</ds:SignatureValue>
291     <ds:KeyInfo Id="keyinfo">
292         <wsse:SecurityTokenReference>
293             <wsse:KeyIdentifier EncodingType="...#Base64Binary"
294                 ValueType="...#X509SubjectKeyIdentifier">
295                 MIGfMa0GCSq...
296             </wsse:KeyIdentifier>
297         </wsse:SecurityTokenReference>
298     </ds:KeyInfo>
299 </ds:Signature>
300 </wsse:Security>
301 </S11:Header>
302 <S11:Body wsu:Id="body"
303     xmlns:wsu=".../">
304     ...
305 </S11:Body>
306 </S11:Envelope>

```

307 3.3.2 Reference to a Binary Security Token

308 The signed data SHOULD contain a core bare name reference (as defined by the XPointer
309 specification [XPointer]) to the <wsse:BinarySecurityToken> element that contains the
310 security token referenced, or a core reference to the external data source containing the security
311 token.

312 The following example shows a certificate embedded in a <wsse:BinarySecurityToken>
313 element and referenced by URI within a signature. The certificate is included in the
314 <wsse:Security> header as a <wsse:BinarySecurityToken> element with identifier
315 binarytoken. The scope of the signature defined by a <ds:Reference> element within the
316 <ds:SignedInfo> element includes the signing certificate which is referenced by means of the
317 URI bare name pointer #binarytoken. The <ds:KeyInfo> element specifies the signing key
318 by means of a <wsse:SecurityTokenReference> element which contains a
319 <wsse:Reference> element which references the certificate by means of the URI bare name
320 pointer #binarytoken.

```

321 <S11:Envelope xmlns:S11="...">
322     <S11:Header>
323         <wsse:Security
324             xmlns:wsse="..."
325             xmlns:wsu="...">
326             <wsse:BinarySecurityToken
327                 wsu:Id="binarytoken"
328                 ValueType="wsse:X509v3"
329                 EncodingType="wsse:Base64Binary">
330                 MIEZzCCA9CgAwIBAgIQEmtJZc0...
331             </wsse:BinarySecurityToken>
332             <ds:Signature
333                 xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
334                 <ds:SignedInfo>...
335                 <ds:Reference URI="#body">...</ds:Reference>
336                 <ds:Reference URI="#binarytoken">...</ds:Reference>
337             </ds:SignedInfo>
338             <ds:SignatureValue>HFLP...</ds:SignatureValue>
339             <ds:KeyInfo>
340                 <wsse:SecurityTokenReference>
341                     <wsse:Reference URI="#binarytoken" />
342                 </wsse:SecurityTokenReference>
343             </ds:KeyInfo>

```

```

344     </ds:Signature>
345   </wsse:Security>
346 </S11:Header>
347 <S11:Body wsu:Id="body"
348   xmlns:wsu="...">
349   ...
350 </S11:Body>
351 </S11:Envelope>

```

352 3.3.3 Reference to an Issuer and Serial Number

353 The signed data SHOULD contain a core bare name reference (as defined by the XPointer
 354 specification [XPointer]) to the <ds:KeyInfo> element that contains the security token
 355 reference.

356 The following example shows a certificate referenced by means of its issuer name and serial
 357 number. In this example the certificate is not included in the message. The scope of the signature
 358 defined by the <ds:SignedInfo> element includes both the message body (#body) and the key
 359 information element (#keyInfo). The <ds:KeyInfo> element contains a
 360 <wsse:SecurityTokenReference> element which specifies the issuer and serial number of
 361 the specified certificate by means of the <ds:X509IssuerSerial> element.

```

362 <S11:Envelope xmlns:S11="...">
363   <S11:Header>
364     <wsse:Security
365       xmlns:wsse="..."
366       xmlns:wsu="...">
367       <ds:Signature
368         xmlns:ds="...">
369         <ds:SignedInfo>...
370         <ds:Reference URI="#body"></ds:Reference>
371         <ds:Reference URI="#keyinfo"></ds:Reference>
372       </ds:SignedInfo>
373       <ds:SignatureValue>HFLP...</ds:SignatureValue>
374       <ds:KeyInfo Id="keyinfo">
375         <wsse:SecurityTokenReference>
376           <ds:X509Data>
377             <ds:X509IssuerSerial>
378               <ds:X509IssuerName>
379                 DC=ACMECorp, DC=com
380               </ds:X509IssuerName>
381               <ds:X509SerialNumber>12345678</X509SerialNumber>
382             </ds:X509IssuerSerial>
383           </ds:X509Data>
384         </wsse:SecurityTokenReference>
385       </ds:KeyInfo>
386     </ds:Signature>
387   </wsse:Security>
388 </S11:Header>
389 <S11:Body wsu:Id="body"
390   xmlns:wsu="...">
391   ...
392 </S11:Body>
393 </S11:Envelope>

```

394 3.4 Encryption

395 Encrypted keys or data MAY identify a key required for decryption by identifying the
 396 corresponding key used for encryption by means of any of the X.509 security token types or
 397 references specified herein.

398 Since the sole purpose is to identify the decryption key it is not necessary to specify either a trust
399 path or the specific contents of the certificate itself.

400 It is RECOMMENDED that implementations specify an encryption key by reference to the Issuer
401 and Serial Number of an X509v3 certificate security token.

402 The following example shows a decryption key referenced by means of the issuer name and
403 serial number of an associated certificate. In this example the certificate is not included in the
404 message. The <ds:KeyInfo> element contains a <wsse:SecurityTokenReference>
405 element which specifies the issuer and serial number of the specified certificate by means of the
406 <ds:X509IssuerSerial> element.

```
407 <S11:Envelope
408     xmlns:S11="..."
409     xmlns:ds="..."
410     xmlns:wsse="..."
411     xmlns:xenc="...">
412   <S11:Header>
413     <wsse:Security>
414       <xenc:EncryptedKey>
415         <xenc:EncryptionMethod Algorithm="..." />
416         <ds:KeyInfo>
417           <wsse:SecurityTokenReference>
418             <ds:X509IssuerSerial>
419               <ds:X509IssuerName>
420                 DC=ACMECorp, DC=com
421               </ds:X509IssuerName>
422               <ds:X509SerialNumber>12345678</X509SerialNumber>
423             </ds:X509IssuerSerial>
424           </wsse:SecurityTokenReference>
425         </ds:KeyInfo>
426       <xenc:CipherData>
427         <xenc:CipherValue>...</xenc:CipherValue>
428       </xenc:CipherData>
429       <xenc:ReferenceList>
430         <xenc:DataReference URI="#encrypted" />
431       </xenc:ReferenceList>
432     </xenc:EncryptedKey>
433   </wsse:Security>
434 </S11:Header>
435 <S11:Body>
436   <xenc:EncryptedData Id="encrypted" Type="...">
437     <xenc:CipherData>
438       <xenc:CipherValue>...</xenc:CipherValue>
439     </xenc:CipherData>
440   </xenc:EncryptedData>
441 </S11:Body>
442 </S11:Envelope>
```

443 3.5 Error Codes

444 When using X.509 certificates, the error codes defined in the WSS: SOAP Message Security
445 specification [WS-Security] MUST be used.

446 If an implementation requires the use of a custom error it is recommended that a sub-code be
447 defined as an extension of one of the codes defined in the WSS: SOAP Message Security
448 specification [WS-Security].

449 **4 Threat Model and Countermeasures (Non-**
450 **Normative)**

451 The use of X.509 certificate token introduces no new threats beyond those identified in WSS:
452 SOAP Message Security specification [WS-Security].

453 Message alteration and eavesdropping can be addressed by using the integrity and confidentiality
454 mechanisms described in WSS: SOAP Message Security [WS-Security]. Replay attacks can be
455 addressed by using message timestamps and caching, as well as other application-specific
456 tracking mechanisms. For X.509 certificates, identity is authenticated by use of keys, man-in-the-
457 middle attacks are generally mitigated.

458 It is strongly RECOMMENDED that all relevant and immutable message data be signed.

459 It should be noted that a transport-level security protocol such as SSL or TLS [RFC2246] MAY be
460 used to protect the message and the security token as an alternative to or in conjunction with
461 WSS: SOAP Message Security specification [WS-Security].

5 References

- 462
- 463 **[Glossary]** Informational RFC 2828, *Internet Security Glossary*, May 2000.
464 <http://www.ietf.org/rfc/rfc2828.txt>
- 465 **[KEYWORDS]** S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*,
466 RFC 2119, Harvard University, March 1997,
467 <http://www.ietf.org/rfc/rfc2119.txt>
- 468 **[RFC2246]** T. Dierks, C. Allen., *The TLS Protocol Version, 1.0*. IETF RFC 2246
469 January 1999. <http://www.ietf.org/rfc/rfc2246.txt>
- 470 **[SOAP11]** W3C Note, "SOAP: Simple Object Access Protocol 1.1," 08 May 2000.
- 471 **[SOAP12]** W3C Recommendation, "http://www.w3.org/TR/2003/REC-soap12-part1-
472 20030624/", 24 June 2003
- 473 **[URI]** T. Berners-Lee, R. Fielding, L. Masinter, "Uniform Resource Identifiers
474 (URI): Generic Syntax," RFC 2396, MIT/LCS, U.C. Irvine, Xerox
475 Corporation, August 1998. <http://www.ietf.org/rfc/rfc2396.txt>
- 476 **[WS-Security]** OASIS,"Web Services Security: SOAP Message Security" 19 January
477 2004, [http://www.docs.oasis-open.org/wss/2004/01/oasis-200401-wss-](http://www.docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0)
478 [soap-message-security-1.0](http://www.docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0)
- 479 **[XML-ns]** T. Bray, D. Hollander, A. Layman. *Namespaces in XML. W3C*
480 *Recommendation*. January 1999. [http://www.w3.org/TR/1999/REC-xml-](http://www.w3.org/TR/1999/REC-xml-names-19990114)
481 [names-19990114](http://www.w3.org/TR/1999/REC-xml-names-19990114)
- 482 **[XML Signature]** D. Eastlake, J. R., D. Solo, M. Bartel, J. Boyer , B. Fox , E. Simon. *XML-*
483 *Signature Syntax and Processing*, W3C Recommendation, 12 February
484 2002. <http://www.w3.org/TR/xmlsig-core/>
- 485 **[PKCS7]** *PKCS #7: Cryptographic Message Syntax Standard* RSA Laboratories,
486 November 1, 1993. [http://www.rsasecurity.com/rsalabs/pkcs/pkcs-](http://www.rsasecurity.com/rsalabs/pkcs/pkcs-7/index.html)
487 [7/index.html](http://www.rsasecurity.com/rsalabs/pkcs/pkcs-7/index.html)
- 488 **[X509]** ITU-T Recommendation X.509 (1997 E): Information Technology - *Open*
489 *Systems Interconnection - The Directory: Authentication Framework*,
490 June 1997.
- 491 **[XPointer]** Paul Grosso, Eve Maler, Jonathan Marsh, Norman Walsh, *XML Pointer*
492 *Language (XPointer)*, W3C Recommendation 25 March 2003
493 <http://www.w3.org/TR/xptr-framework/>
- 494
- 495

Appendix A: Revision History

Rev	Date	What
01	18-Sep-02	Initial draft based on input documents and editorial review
03	30-Jan-03	Changes in title
04	19-May-03	Added by reference and pkipath modes of cert identification. Added section 1 introduction, changes to formatting etc.
05	6 June 2003	
06	20 June 2003	Included examples showing how tokens must be referenced from signatures and cipher values. Defined how key-agreement keys are to be conveyed in a Security header.
07	4 August 2003	Modifications to KeyIdentifier handling and use of SecurityTokenReference. Changes to the acknowledgements section.
08	6 August 2003	Reorganization of major sections to simplify flow
09	14 August 2003	Editorial corrections raised in off list emails.
10	19 August 2003	Editorial corrections raised in profile teleconference.
11	09 January 2004	Editorial corrections raised in forum
12	15 January 2004	Editorial correction, amend X509IssuerSerial usage
13	19 January 2004	Editorial corrections for name space and document name
14	17 February 2004	Editorial corrections per Karl Best

Appendix B: Notices

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

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

511 Copyright © OASIS Open 2002-2004. *All Rights Reserved.*

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

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

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

528