



1

2

Web Services Security X.509 Certificate Token Profile

3

4

Tuesday, 17 February 2004

5

Document identifier:

6

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

7

Location:

8

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

9

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

10

Editors:

11

Phillip Hallam-Baker, VeriSign

12

Chris Kaler, Microsoft

13

Ronald Monzillo, Sun

14

Anthony Nadalin, IBM

15

Contributors:

16

Gene Thurston AmberPoint

17

Frank Siebenlist Argonne National Lab

18

Merlin Hughes Baltimore Technologies

19

Irving Reid Baltimore Technologies

20

Peter Dapkus BEA

21

Hal Lockhart BEA

22

Symon Chang CommerceOne

23

Thomas DeMartini ContentGuard

24

Guillermo Lao ContentGuard

25

TJ Pannu ContentGuard

26

Shawn Sharp Cyclone Commerce

27

Ganesh Vaideeswaran Documentum

28

Sam Wei Documentum

29

John Hughes Entegry

30

Tim Moses Entrust

31

Toshihiro Nishimura Fujitsu

32

Tom Rutt Fujitsu

33

Jason Rouault HP

34

Yutaka Kudo Hitachi

35

Maryann Hondo IBM

36

Kelvin Lawrence IBM (co-Chair)

37

Anthony Nadalin IBM

38

Nataraj Nagaratnam IBM

39

Don Flinn Individual

40

Bob Morgan Individual

41

Paul Cotton Microsoft

42

Vijay Gajjala Microsoft

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

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

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

89 **Abstract:**

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

92 **Status:**

93 This is an interim draft.

94 Committee members should send comments on this specification to the wss@lists.oasis-
95 open.org list. Others should subscribe to and send comments to the wss-

96 comment@lists.oasis-open.org list. To subscribe, visit <http://lists.oasis->
97 [open.org/ob/adm.pl](http://lists.oasis-open.org/ob/adm.pl).
98 For information on whether any patents have been disclosed that may be essential to
99 implementing this specification, and any offers of patent licensing terms, please refer to
100 the Intellectual Property Rights section of the WS-Security TC web page
101 (<http://www.oasis-open.org/committees/wss/ipr.php>).

Table of Contents

103	1	Introduction (Non-Normative)	5
104	2	Notations and Terminology (Normative)	6
105	2.1	Notational Conventions	6
106	2.2	Namespaces	6
107	2.3	Terminology.....	7
108	3	Usage (Normative).....	8
109	3.1	Token types.....	8
110	3.1.1	#X509v3 Token Type.....	8
111	3.1.2	#X509PKIPathv1 Token Type	8
112	3.1.3	#PKCS7 Token Type	8
113	3.2	Token References.....	8
114	3.2.1	Reference to a Subject Key Identifier	9
115	3.2.2	Reference to a Security Token	9
116	3.2.3	Reference to an Issuer and Serial Number	9
117	3.3	Signature	10
118	3.3.1	Key Identifier	10
119	3.3.2	Reference to a Binary Security Token	11
120	3.3.3	Reference to an Issuer and Serial Number	12
121	3.4	Encryption	12
122	3.5	Error Codes	13
123	4	Threat Model and Countermeasures (Non-Normative)	14
124	5	References.....	15
125		Appendix A: Revision History	16
126		Appendix B: Notices	17
127			

128 **1 Introduction (Non-Normative)**

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

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

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

2 Notations and Terminology (Normative)

137

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

2.1 Notational Conventions

139

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

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

146 When describing concrete XML schemas, this specification uses a convention where each
147 member of an element's [children] or [attributes] property is described using an XPath-like
148 notation (e.g., /x:MyHeader/x:SomeProperty/@value1). The use of {any} indicates the presence
149 of an element wildcard (<xs:any/>). The use of @{any} indicates the presence of an attribute
150 wildcard (<xs:anyAttribute/>).
151

2.2 Namespaces

152

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

```
156     http://www.docs.oasis-open.org/wss/2004/01/oasis-200401-wss-  
157     wssecurity-secext-1.0.xsd  
158     http://www.docs.oasis-open.org/wss/2004/01/oasis-200401-wss-  
159     wssecurity-utility-1.0.xsd  
160
```

161 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

162

Table 1- Namespace prefixes

163 **2.3 Terminology**

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

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

168 3 Usage (Normative)

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

172 3.1 Token types

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

176

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

177

Table 2 – Token types

178 3.1.1 X509v3 Token Type

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

181 3.1.2 X509PKIPathv1 Token Type

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

183 3.1.3 PKCS7 Token Type

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

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

190 3.2 Token References

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

195

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

198 Reference to a Subject Key Identifier

199 The <wsse:SecurityTokenReference> element contains a
200 <wsse:KeyIdentifier> element that specifies the token data by means of a X.509
201 SubjectKeyIdentifier reference.

202 Reference to a Binary Security Token

203 The <wsse:SecurityTokenReference> element contains a <wsse:Reference>
204 element that references a local <wsse:BinarySecurityToken> element or a remote
205 data source that contains the token data itself.

206 Reference to an Issuer and Serial Number

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

210 3.2.1 Reference to a Subject Key Identifier

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

215

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

216

Table 3 – Subject Key Identifier

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

224 3.2.2 Reference to a Security Token

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

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

230 3.2.3 Reference to an Issuer and Serial Number

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

233 The <ds:X509IssuerSerial> element is a direct child of the <ds:X509Data> element that is
234 in turn a direct child of the <wsse:SecurityTokenReference> element in which the
235 reference is made.

236 3.3 Signature

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

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

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

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

251 3.3.1 Key Identifier

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

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

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

```

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

```

303 3.3.2 Reference to a Binary Security Token

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

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

```

317 <S11:Envelope xmlns:S11="...">
318   <S11:Header>
319     <wsse:Security
320       xmlns:wsse="..."
321       xmlns:wsu="...">
322       <wsse:BinarySecurityToken
323         wsu:Id="binarytoken"
324         ValueType="wsse:X509v3"
325         EncodingType="wsse:Base64Binary">
326         MIEEZzCCA9CgAwIBAgIQEmtJZc0...
327       </wsse:BinarySecurityToken>
328     <ds:Signature
329       xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
330       <ds:SignedInfo>...
331         <ds:Reference URI="#body">...</ds:Reference>
332         <ds:Reference URI="#binarytoken">...</ds:Reference>
333       </ds:SignedInfo>
334       <ds:SignatureValue>HFLP...</ds:SignatureValue>
335       <ds:KeyInfo>
336         <wsse:SecurityTokenReference>
337           <wsse:Reference URI="#binarytoken" />
338         </wsse:SecurityTokenReference>
339       </ds:KeyInfo>

```

```

340     </ds:Signature>
341   </wsse:Security>
342 </S11:Header>
343 <S11:Body wsu:Id="body"
344   xmlns:wsu="...">
345   ...
346 </S11:Body>
347 </S11:Envelope>

```

348 3.3.3 Reference to an Issuer and Serial Number

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

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

```

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

```

390 3.4 Encryption

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

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

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

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

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

439 3.5 Error Codes

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

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

445 **4 Threat Model and Countermeasures (Non-**
446 **Normative)**

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

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

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

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

5 References

- 458
- 459 **[Glossary]** Informational RFC 2828, *Internet Security Glossary*, May 2000.
460 <http://www.ietf.org/rfc/rfc2828.txt>
- 461 **[KEYWORDS]** S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*,
462 RFC 2119, Harvard University, March 1997,
463 <http://www.ietf.org/rfc/rfc2119.txt>
- 464 **[RFC2246]** T. Dierks, C. Allen., *The TLS Protocol Version, 1.0*. IETF RFC 2246
465 January 1999. <http://www.ietf.org/rfc/rfc2246.txt>
- 466 **[SOAP11]** W3C Note, "SOAP: Simple Object Access Protocol 1.1," 08 May 2000.
- 467 **[SOAP12]** W3C Recommendation, "http://www.w3.org/TR/2003/REC-soap12-part1-
468 20030624/", 24 June 2003
- 469 **[URI]** T. Berners-Lee, R. Fielding, L. Masinter, "Uniform Resource Identifiers
470 (URI): Generic Syntax," RFC 2396, MIT/LCS, U.C. Irvine, Xerox
471 Corporation, August 1998. <http://www.ietf.org/rfc/rfc2396.txt>
- 472 **[WS-Security]** OASIS, "Web Services Security: SOAP Message Security" 19 January
473 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)
474 [soap-message-security-1.0](http://www.docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0)
- 475 **[XML-ns]** T. Bray, D. Hollander, A. Layman. *Namespaces in XML. W3C*
476 *Recommendation*. January 1999. [http://www.w3.org/TR/1999/REC-xml-](http://www.w3.org/TR/1999/REC-xml-names-19990114)
477 [names-19990114](http://www.w3.org/TR/1999/REC-xml-names-19990114)
- 478 **[XML Signature]** D. Eastlake, J. R., D. Solo, M. Bartel, J. Boyer , B. Fox , E. Simon. *XML-*
479 *Signature Syntax and Processing*, W3C Recommendation, 12 February
480 2002. <http://www.w3.org/TR/xmlsig-core/>
- 481 **[PKCS7]** *PKCS #7: Cryptographic Message Syntax Standard* RSA Laboratories,
482 November 1, 1993. [http://www.rsasecurity.com/rsalabs/pkcs/pkcs-](http://www.rsasecurity.com/rsalabs/pkcs/pkcs-7/index.html)
483 [7/index.html](http://www.rsasecurity.com/rsalabs/pkcs/pkcs-7/index.html)
- 484 **[X509]** ITU-T Recommendation X.509 (1997 E): Information Technology - *Open*
485 *Systems Interconnection - The Directory: Authentication Framework*,
486 June 1997.
- 487 **[XPointer]** Paul Grosso, Eve Maler, Jonathan Marsh, Norman Walsh, *XML Pointer*
488 *Language (XPointer)*, W3C Recommendation 25 March 2003
489 <http://www.w3.org/TR/xptr-framework/>
- 490
- 491

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

494

Appendix B: Notices

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

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

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

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

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

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

524