



Web Services Security X.509 Certificate Token Profile

Monday, 19 January 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
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
Maryann	Hondo	IBM
Kelvin	Lawrence	IBM (co-Chair)
Anthony	Nadalin	IBM
Nataraj	Nagaratnam	IBM
Don	Flinn	Individual
Bob	Morgan	Individual
Paul	Cotton	Microsoft
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	Manferdell	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/jpr.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	9
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.....	11
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 Security: SOAP Message Security specification [WS-Security].

130

131

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

132

133

134

135

An X.509 certificate may be used to validate a public key that may be used to authenticate a WS-Security-enhanced message or to identify the public key with which a WS-Security-enhanced message has been encrypted.

136

137

138

2 Notations and Terminology (Normative)

139

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

140

2.1 Notational Conventions

141

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.

142

143

144

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]).

145

146

147

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/>).

148

149

150

151

152

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

153

2.2 Namespaces

154

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):

155

156

157

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

158

159

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

160

161

162

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

163

Table 1- Namespace prefixes

164 **2.3 Terminology**

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

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

169 3 Usage (Normative)

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

173 3.1 Token types

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

Deleted: Table 2

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

178 *Table 2 – Token types*

179 3.1.1 #X509v3 Token Type

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

182 3.1.2 #X509PKIPathv1 Token Type

183 The `wsse:X509PKIPathv1` token type MAY be used to represent a certificate path.

184 3.1.3 #PKCS7 Token Type

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

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

191 3.2 Token References

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

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.

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

220 3.2.2 Reference to a Security Token

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

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

226 3.2.3 Reference to an Issuer and Serial Number

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

229 The `<ds:X509IssuerSerial>` element is a direct child of the `<ds:X509Data>` element that is
230 in turn a direct child of the `<wsse:SecurityTokenReference>` element in which the
231 reference is made.

232 3.3 Signature

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

235 An X.509 certificate specifies a binding between a public key and a set of attributes that includes
236 (at least) a subject name, issuer name, serial number and validity interval. Other attributes may
237 specify constraints on the use of the certificate or affect the recourse that may be open to a
238 relying party that depends on the certificate. A given public key may be specified in more than

239 one X.509 certificate; consequently a given public key may be bound to two or more distinct sets
240 of attributes.

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

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

247 3.3.1 Key Identifier

248 The <wsse:KeyIdentifier> element does not guarantee an immutable and unambiguous
249 reference to the certificate referenced. Consequently implementations that use this form of
250 reference within a signature SHOULD employ the <wsse:SecurityTokenReference>
251 element dereferencing transform within a reference to the signature key information in order to
252 ensure that the referenced certificate is signed, and not just the ambiguous reference. The form
253 of the reference is a bare name reference as defined by the XPointer specification [XPointer].

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

```
263 <S11:Envelope xmlns:S="http://www.w3.org/2002/12/soap-envelope">  
264   <S11:Header>  
265     <wsse:Security  
266       xmlns:wsse="..."  
267       xmlns:wsu="...">  
268       <ds:Signature  
269         xmlns:ds="http://www.w3.org/2000/09/xmldsig#">  
270         <ds:SignedInfo>...  
271           <ds:Reference URI="#body">...</ds:Reference>  
272           <ds:Reference URI="#keyinfo">  
273             <ds:Transforms>  
274               <ds:Transform Algorithm=".../STR-Transform">  
275                 <wsse:TransformationParameters  
276                   <ds:CanonicalizationMethod Algorithm="..."/>  
277                   </wsse:TransformationParameters>  
278                 </ds:Transform>  
279               </ds:Transforms>...  
280             </ds:Reference>  
281           </ds:SignedInfo>  
282           <ds:SignatureValue>HFLP...</ds:SignatureValue>  
283           <ds:KeyInfo Id="keyinfo">  
284             <wsse:SecurityTokenReference>  
285               <wsse:KeyIdentifier EncodingType="...#Base64Binary"  
286                 ValueType="...#X509SubjectKeyIdentifier">  
287                 MIGfMa0GCSq...  
288               </wsse:KeyIdentifier>  
289             </wsse:SecurityTokenReference>  
290           </ds:KeyInfo>  
291         </ds:Signature>  
292       </wsse:Security>  
293     </S11:Header>
```

```
294 <S11:Body wsu:Id="body"
295     xmlns:wsu="...">
296     ...
297 </S11:Body>
298 </S11:Envelope>
```

299 3.3.2 Reference to a Binary Security Token

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

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

```
312 <S11:Envelope xmlns:S11="...">
313   <S11:Header>
314     <wsse:Security
315       xmlns:wsse="..."
316       xmlns:wsu="...">
317       <wsse:BinarySecurityToken
318         wsu:Id="binarytoken"
319         ValueType="wsse:X509v3"
320         EncodingType="wsse:Base64Binary">
321         MIEZzCCA9CgAwIBAgIQEmtJzc0...
322       </wsse:BinarySecurityToken>
323       <ds:Signature
324         xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
325         <ds:SignedInfo>...
326         <ds:Reference URI="#body">...</ds:Reference>
327         <ds:Reference URI="#binarytoken">...</ds:Reference>
328       </ds:SignedInfo>
329       <ds:SignatureValue>HFLP...</ds:SignatureValue>
330       <ds:KeyInfo>
331         <wsse:SecurityTokenReference>
332           <wsse:Reference URI="#binarytoken" />
333         </wsse:SecurityTokenReference>
334       </ds:KeyInfo>
335     </ds:Signature>
336   </wsse:Security>
337 </S11:Header>
338 <S11:Body wsu:Id="body"
339   xmlns:wsu="...">
340   ...
341 </S11:Body>
342 </S11:Envelope>
```

343 3.3.3 Reference to an Issuer and Serial Number

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

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

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

Formatted: English U.S.

Formatted: English U.S.

385 3.4 Encryption

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

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

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

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

```
398 <S11:Envelope
399   xmlns:S11="..."
400   xmlns:ds="..."
```

```

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

```

434 3.5 Error Codes

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

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

440 **4 Threat Model and Countermeasures (Non-**
441 **Normative)**

442 The use of X.509 certificates with WS-Security introduces no new threats beyond those identified
443 in WSS: SOAP Message Security specification [WS-Security].

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

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

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

453

5 References

454

[Glossary] Informational RFC 2828, *Internet Security Glossary*, May 2000.
<http://www.ietf.org/rfc/rfc2828.txt>

455

456

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

457

458

459

[RFC2246] T. Dierks, C. Allen., *The TLS Protocol Version, 1.0*. IETF RFC 2246
January 1999. <http://www.ietf.org/rfc/rfc2246.txt>

460

461

[SOAP11] W3C Note, "SOAP: Simple Object Access Protocol 1.1," 08 May 2000.

462

[SOAP12] W3C Recommendation, "<http://www.w3.org/TR/2003/REC-soap12-part1-20030624/>", 24 June 2003

463

464

[URI] T. Berners-Lee, R. Fielding, L. Masinter, "Uniform Resource Identifiers (URI): Generic Syntax," RFC 2396, MIT/LCS, U.C. Irvine, Xerox Corporation, August 1998. <http://www.ietf.org/rfc/rfc2396.txt>

465

466

467

[WS-Security] OASIS, "Web Services Security: SOAP Message Security" 19 January 2004, <http://www.docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0>

468

469

470

[XML-ns] T. Bray, D. Hollander, A. Layman. *Namespaces in XML. W3C Recommendation*. January 1999. <http://www.w3.org/TR/1999/REC-xml-names-19990114>

471

472

473

[XML Signature] D. Eastlake, J. R., D. Solo, M. Bartel, J. Boyer , B. Fox , E. Simon. *XML-Signature Syntax and Processing*, W3C Recommendation, 12 February 2002. <http://www.w3.org/TR/xmlsig-core/>

474

475

476

[PKCS7] *PKCS #7: Cryptographic Message Syntax Standard* RSA Laboratories, November 1, 1993. <http://www.rsasecurity.com/rsalabs/pkcs/pkcs-7/index.html>

477

478

479

[X509] ITU-T Recommendation X.509 (1997 E): Information Technology - *Open Systems Interconnection - The Directory: Authentication Framework*, June 1997.

480

481

482

[XPointer] Paul Grosso, Eve Maler, Jonathan Marsh, Norman Walsh, *XML Pointer Language (XPointer)*, W3C Recommendation 25 March 2003
<http://www.w3.org/TR/xptr-framework/>

483

484

485

486

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

Appendix B: Notices

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

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

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

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

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

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

519