



Web Services Security X.509 Certificate Token Profile

OASIS Standard 200401, March 2004

Document identifier:

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

Document Location:

<http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0>

Errata Location:

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

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)

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

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

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

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

97 **Status:**
98 This is an interim draft.

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

102 For information on whether any patents have been disclosed that may be essential to implementing this
103 specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section
104 of the WS-Security TC web page (<http://www.oasis-open.org/committees/wss/ipr.php>).

Table of Contents

106	1	Introduction (Non-Normative)	5
107	2	Notations and Terminology (Normative).....	6
108	2.1	Notational Conventions.....	6
109	2.2	Namespaces.....	6
110	2.3	Terminology.....	6
111	3	Usage (Normative)	7
112	3.1	Token types	7
113	3.1.1	#X509v3 Token Type	7
114	3.1.2	#X509PKIPathv1 Token Type	7
115	3.1.3	#PKCS7 Token Type.....	7
116	3.2	Token References	7
117	3.2.1	Reference to a Subject Key Identifier	8
118	3.2.2	Reference to a Security Token.....	8
119	3.2.3	Reference to an Issuer and Serial Number	8
120	3.3	Signature	8
121	3.3.1	Key Identifier.....	9
122	3.3.2	Reference to a Binary Security Token.....	10
123	3.3.3	Reference to an Issuer and Serial Number	10
124	3.4	Encryption.....	11
125	3.5	Error Codes	12
126	4	Threat Model and Countermeasures (Non-Normative)	13
127	5	References	14
128		Appendix A: Revision History.....	15
129		Appendix B: Notices.....	16
130			

131 **1 Introduction (Non-Normative)**

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

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

138 An X.509 certificate may be used to validate a public key that may be used to authenticate a SOAP message or to
139 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://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd
wsu	http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd

Table 1- Namespace prefixes

2.3 Terminology

This specification adopts the terminology defined in Web Services Security: SOAP Message Security specification [WS-Security]. Readers are presumed to be familiar with the definitions of terms in the Internet Security Glossary [Glossary].

3 Usage (Normative)

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

3.1 Token types

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

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

Table 2 – Token types

3.1.1 X509v3 Token Type

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

3.1.2 X509PKIPathv1 Token Type

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

3.1.3 PKCS7 Token Type

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

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

3.2 Token References

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

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

Reference to a Subject Key Identifier

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

196 Reference to a Binary Security Token
 197 The <wsse:SecurityTokenReference> element contains a <wsse:Reference> element that
 198 references a local <wsse:BinarySecurityToken> element or a remote data source that contains the token
 199 data itself.
 200 Reference to an Issuer and Serial Number
 201 The <wsse:SecurityTokenReference> element contains a <ds:X509Data> element that contains a
 202 <ds:X509IssuerSerial> element that uniquely identifies an end entity certificate by its X.509 Issuer and
 203 Serial Number.

204 3.2.1 Reference to a Subject Key Identifier

205 The <wsse:KeyIdentifier> element is used to specify a reference to an X.509 certificate by means of a
 206 reference to its X.509 SubjectKeyIdentifier attribute. This profile defines the syntax of, and processing rules for
 207 referencing a Subject Key Identifier using the URI values specified in Table 3 (note that URI fragments are relative to
 208 the URI for this specification).
 209

Subject Key Identifier	ValueType URI	Description
Certificate Key Identifier	#x509SubjectKeyIdentifier	Value of the certificate's X.509 SubjectKeyIdentifier

210 *Table 3 – Subject Key Identifier*

211 The <wsse:SecurityTokenReference> element from which the reference is made contains the
 212 <wsse:KeyIdentifier> element. The <wsse:KeyIdentifier> element MUST have a
 213 ValueType attribute with the value #x509SubjectKeyIdentifier and its contents MUST be the value of the
 214 certificate's X.509 SubjectKeyIdentifier extension, encoded as per the <wsse:KeyIdentifier> element's
 215 EncodingType attribute. For the purposes of this specification, the value of the SubjectKeyIdentifier extension is
 216 the contents of the KeyIdentifier octet string, excluding the encoding of the octet string prefix.

217 3.2.2 Reference to a Security Token

218 The <wsse:Reference> element is used to reference an X.509 security token value by means of a URI reference.
 219 The URI reference MAY be internal in which case the URI reference SHOULD be a bare name XPointer reference to a
 220 <wsse:BinarySecurityToken> element contained in a preceding message header that contains the binary
 221 X.509 security token data.

222 3.2.3 Reference to an Issuer and Serial Number

223 The <ds:X509IssuerSerial> element is used to specify a reference to an X.509 security token by means of
 224 the certificate issuer name and serial number.
 225 The <ds:X509IssuerSerial> element is a direct child of the <ds:X509Data> element that is in turn a direct
 226 child of the <wsse:SecurityTokenReference> element in which the reference is made.

227 3.3 Signature

228 Signed data MAY specify the certificate associated with the signature using any of the X.509 security token types and
 229 references defined in this specification.
 230 An X.509 certificate specifies a binding between a public key and a set of attributes that includes (at least) a subject
 231 name, issuer name, serial number and validity interval. Other attributes may specify constraints on the use of the
 232 certificate or affect the recourse that may be open to a relying party that depends on the certificate. A given public key
 233 may be specified in more than one X.509 certificate; consequently a given public key may be bound to two or more
 234 distinct sets of attributes.
 235 It is therefore necessary to ensure that a signature created under an X.509 certificate token uniquely and irrefutably
 236 specifies the certificate under which the signature was created.

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

240 3.3.1 Key Identifier

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

246 The following example shows a certificate referenced by means of a KeyIdentifier. The scope of the signature is the
247 <ds:SignedInfo> element which includes both the message body (#body) and the signing certificate by means
248 of a reference to the <ds:KeyInfo> element which references it (#keyinfo). Since the <ds:KeyInfo>
249 element only contains a mutable reference to the certificate rather than the certificate itself, a transformation is
250 specified which replaces the reference to the certificate with the certificate. The <ds:KeyInfo> element specifies
251 the signing key by means of a <wsse:SecurityTokenReference> element which contains a
252 <wsse:KeyIdentifier> element which specifies the X.509 subject key identifier of the signing certificate.

```
253 <S11:Envelope xmlns:S11="...">
254   <S11:Header>
255     <wsse:Security
256       xmlns:wsse="..."
257       xmlns:wsu="...">
258       <ds:Signature
259         xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
260         <ds:SignedInfo>...
261         <ds:Reference URI="#body">...</ds:Reference>
262         <ds:Reference URI="#keyinfo">
263           <ds:Transforms>
264             <ds:Transform Algorithm="...#STR-Transform">
265               <wsse:TransformationParameters>
266                 <ds:CanonicalizationMethod Algorithm="..."/>
267               </wsse:TransformationParameters>
268             </ds:Transform>
269           </ds:Transforms>...
270         </ds:Reference>
271       </ds:SignedInfo>
272       <ds:SignatureValue>HFLP...</ds:SignatureValue>
273       <ds:KeyInfo Id="keyinfo">
274         <wsse:SecurityTokenReference>
275           <wsse:KeyIdentifier EncodingType="...#Base64Binary"
276             ValueType="...#X509SubjectKeyIdentifier">
277             MIGfMa0GCSq...
278           </wsse:KeyIdentifier>
279         </wsse:SecurityTokenReference>
280       </ds:KeyInfo>
281     </ds:Signature>
282   </wsse:Security>
283 </S11:Header>
284 <S11:Body wsu:Id="body"
285   xmlns:wsu=".../">
286   ...
287 </S11:Body>
```

289 3.3.2 Reference to a Binary Security Token

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

293 The following example shows a certificate embedded in a <wsse:BinarySecurityToken> element and
 294 referenced by URI within a signature. The certificate is included in the <wsse:Security> header as a
 295 <wsse:BinarySecurityToken> element with identifier binarytoken. The scope of the signature
 296 defined by a <ds:Reference> element within the <ds:SignedInfo> element includes the signing
 297 certificate which is referenced by means of the URI bare name pointer #binarytoken. The <ds:KeyInfo>
 298 element specifies the signing key by means of a <wsse:SecurityTokenReference> element which
 299 contains a <wsse:Reference> element which references the certificate by means of the URI bare name pointer
 300 #binarytoken.

```

301 <S11:Envelope xmlns:S11="...">
302   <S11:Header>
303     <wsse:Security
304       xmlns:wsse="..."
305       xmlns:wsu="...">
306       <wsse:BinarySecurityToken
307         wsu:Id="binarytoken"
308         ValueType="wsse:X509v3"
309         EncodingType="wsse:Base64Binary">
310         MIIIEZzCCA9CgAwIBAgIQEmtJZc0...
311       </wsse:BinarySecurityToken>
312       <ds:Signature
313         xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
314         <ds:SignedInfo>...
315         <ds:Reference URI="#body">...</ds:Reference>
316         <ds:Reference URI="#binarytoken">...</ds:Reference>
317       </ds:SignedInfo>
318       <ds:SignatureValue>HFLP...</ds:SignatureValue>
319       <ds:KeyInfo>
320         <wsse:SecurityTokenReference>
321           <wsse:Reference URI="#binarytoken" />
322         </wsse:SecurityTokenReference>
323       </ds:KeyInfo>
324     </ds:Signature>
325   </wsse:Security>
326 </S11:Header>
327   <S11:Body wsu:Id="body"
328     xmlns:wsu="...">
329     ...
330   </S11:Body>
331 </S11:Envelope>

```

332 3.3.3 Reference to an Issuer and Serial Number

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

335 The following example shows a certificate referenced by means of its issuer name and serial number. In this example
 336 the certificate is not included in the message. The scope of the signature defined by the <ds:SignedInfo>

337 element includes both the message body (#body) and the key information element (#keyInfo). The
338 <ds:KeyInfo> element contains a <wsse:SecurityTokenReference> element which specifies the
339 issuer and serial number of the specified certificate by means of the <ds:X509IssuerSerial> element.

```
340 <S11:Envelope xmlns:S11="...">
341   <S11:Header>
342     <wsse:Security
343       xmlns:wsse="..."
344       xmlns:wsu="...">
345       <ds:Signature
346         xmlns:ds="...">
347         <ds:SignedInfo>...
348           <ds:Reference URI="#body"></ds:Reference>
349           <ds:Reference URI="#keyinfo"></ds:Reference>
350         </ds:SignedInfo>
351         <ds:SignatureValue>HFLP...</ds:SignatureValue>
352         <ds:KeyInfo Id="keyinfo">
353           <wsse:SecurityTokenReference>
354             <ds:X509Data>
355               <ds:X509IssuerSerial>
356                 <ds:X509IssuerName>
357                   DC=ACMECorp, DC=com
358                 </ds:X509IssuerName>
359                 <ds:X509SerialNumber>12345678</X509SerialNumber>
360               </ds:X509IssuerSerial>
361             </ds:X509Data>
362           </wsse:SecurityTokenReference>
363         </ds:KeyInfo>
364       </ds:Signature>
365     </wsse:Security>
366   </S11:Header>
367   <S11:Body wsu:Id="body"
368     xmlns:wsu="...">
369     ...
370   </S11:Body>
371 </S11:Envelope>
```

372 3.4 Encryption

373 Encrypted keys or data MAY identify a key required for decryption by identifying the corresponding key used for
374 encryption by means of any of the X.509 security token types or references specified herein.
375 Since the sole purpose is to identify the decryption key it is not necessary to specify either a trust path or the specific
376 contents of the certificate itself.
377 It is RECOMMENDED that implementations specify an encryption key by reference to the Issuer and Serial Number of
378 an X509v3 certificate security token.
379 The following example shows a decryption key referenced by means of the issuer name and serial number of an
380 associated certificate. In this example the certificate is not included in the message. The <ds:KeyInfo> element
381 contains a <wsse:SecurityTokenReference> element which specifies the issuer and serial number of
382 the specified certificate by means of the <ds:X509IssuerSerial> element.

```
383 <S11:Envelope
384   xmlns:S11="..."
385   xmlns:ds="..."
386   xmlns:wsse="..."
387   xmlns:xenc="...">
```

```

388 <S11:Header>
389   <wsse:Security>
390     <xenc:EncryptedKey>
391       <xenc:EncryptionMethod Algorithm="..." />
392       <ds:KeyInfo>
393         <wsse:SecurityTokenReference>
394           <ds:X509IssuerSerial>
395             <ds:X509IssuerName>
396               DC=ACMECorp, DC=com
397             </ds:X509IssuerName>
398             <ds:X509SerialNumber>12345678</X509SerialNumber>
399           </ds:X509IssuerSerial>
400         </wsse:SecurityTokenReference>
401       </ds:KeyInfo>
402       <xenc:CipherData>
403         <xenc:CipherValue>...</xenc:CipherValue>
404       </xenc:CipherData>
405       <xenc:ReferenceList>
406         <xenc:DataReference URI="#encrypted" />
407       </xenc:ReferenceList>
408     </xenc:EncryptedKey>
409   </wsse:Security>
410 </S11:Header>
411 <S11:Body>
412   <xenc:EncryptedData Id="encrypted" Type="...">
413     <xenc:CipherData>
414       <xenc:CipherValue>...</xenc:CipherValue>
415     </xenc:CipherData>
416   </xenc:EncryptedData>
417 </S11:Body>
418 </S11:Envelope>

```

419 3.5 Error Codes

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

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

424

4 Threat Model and Countermeasures (Non-Normative)

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

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

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

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

5 References

- 435
- 436 [Glossary] Informational RFC 2828, *Internet Security Glossary*, May 2000.
437 <http://www.ietf.org/rfc/rfc2828.txt>
- 438 [KEYWORDS] S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*, RFC 2119,
439 Harvard University, March 1997, <http://www.ietf.org/rfc/rfc2119.txt>
- 440 [RFC2246] T. Dierks, C. Allen., *The TLS Protocol Version, 1.0*. IETF RFC 2246 January 1999.
441 <http://www.ietf.org/rfc/rfc2246.txt>
- 442 [SOAP11] W3C Note, "SOAP: Simple Object Access Protocol 1.1," 08 May 2000.
- 443 [SOAP12] W3C Recommendation, "<http://www.w3.org/TR/2003/REC-soap12-part1-20030624/>", 24
444 June 2003
- 445 [URI] T. Berners-Lee, R. Fielding, L. Masinter, "Uniform Resource Identifiers (URI): Generic
446 Syntax," RFC 2396, MIT/LCS, U.C. Irvine, Xerox Corporation, August 1998.
447 <http://www.ietf.org/rfc/rfc2396.txt>
- 448 [WS-Security] OASIS, "Web Services Security: SOAP Message Security" 19 January 2004,
449 [http://www.docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-
450 1.0](http://www.docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0)
- 451 [XML-ns] T. Bray, D. Hollander, A. Layman. *Namespaces in XML. W3C Recommendation*. January
452 1999. <http://www.w3.org/TR/1999/REC-xml-names-19990114>
- 453 [XML Signature] D. Eastlake, J. R., D. Solo, M. Bartel, J. Boyer , B. Fox , E. Simon. *XML-Signature
454 Syntax and Processing*, W3C Recommendation, 12 February 2002.
455 <http://www.w3.org/TR/xmlsig-core/>
- 456 [PKCS7] *PKCS #7: Cryptographic Message Syntax Standard* RSA Laboratories, November 1,
457 1993. <http://www.rsasecurity.com/rsalabs/pkcs/pkcs-7/index.html>
- 458 [X509] ITU-T Recommendation X.509 (1997 E): Information Technology - *Open Systems
459 Interconnection - The Directory: Authentication Framework*, June 1997.
- 460 [XPointer] Paul Grosso, Eve Maler, Jonathan Marsh, Norman Walsh, *XML Pointer Language
461 (XPointer)*, W3C Recommendation 25 March 2003 <http://www.w3.org/TR/xptr-framework/>
- 462
- 463

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 pkpath 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

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

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

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

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

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

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

490