



1

2 **Web Services Security**
3 **X.509 Certificate Token Profile 1.1**

4 **OASIS Committee Specification, 14 November**
5 **2005**

6 **OASIS Identifier:**

7 `wss-v1.1-spec-cs-X509-token-profile`

8 **Document Location:**

9 `http://docs.oasis-open.org/wss/oasis-wss-x509-token-profile-1.1`

10 **Technical Committee:**

11 Web Service Security (WSS)

12 **Chairs:**

13 Kelvin Lawrence, IBM

14 Chris Kaler, Microsoft

15 **Editors:**

16 Anthony Nadalin, IBM

17 Chris Kaler, Microsoft

18 Ronald Monzillo, Sun

19 Phillip Hallam-Baker, Verisign

20

21 **Abstract:**

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

24 **Status:**

25 Committee members should send comments on this specification to the `wss@lists.oasis-`
26 `open.org` list. Others should subscribe to and send comments to the `wss-`
27 `comment@lists.oasis-open.org` list. To subscribe, visit `open.org/ob/adm.pl`.

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

33 **Notices**

34 OASIS takes no position regarding the validity or scope of any intellectual property or other rights
35 that might be claimed to pertain to the implementation or use of the technology described in this
36 document or the extent to which any license under such rights might or might not be available;
37 neither does it represent that it has made any effort to identify any such rights. Information on
38 OASIS's procedures with respect to rights in OASIS specifications can be found at the OASIS
39 website. Copies of claims of rights made available for publication and any assurances of licenses
40 to be made available, or the result of an attempt made to obtain a general license or permission
41 for the use of such proprietary rights by implementors or users of this specification, can be
42 obtained from the OASIS Executive Director. OASIS invites any interested party to bring to its
43 attention any copyrights, patents or patent applications, or other proprietary rights which may
44 cover technology that may be required to implement this specification. Please address the
45 information to the OASIS Executive Director.

46

47 Copyright (C) OASIS Open 2002-2005. All Rights Reserved.

48

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

58

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

61

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

67

68 OASIS has been notified of intellectual property rights claimed in regard to some or all of the
69 contents of this specification. For more information consult the online list of claimed rights.

70

71 This section is non-normative.

72 **Table of Contents**

73 1 Introduction (Non-Normative) 4
74 2 Notations and Terminology (Normative) 5
75 2.1 Notational Conventions 5
76 2.2 Namespaces 5
77 2.3 Terminology 6
78 3 Usage (Normative) 8
79 3.1 Token types 8
80 3.1.1 X509v3 Token Type 8
81 3.1.2 X509PKIPathv1 Token Type 8
82 3.1.3 PKCS7 Token Type 8
83 3.2 Token References 9
84 3.2.1 Reference to an X.509 Subject Key Identifier 9
85 3.2.2 Reference to a Security Token 10
86 3.2.3 Reference to an Issuer and Serial Number 10
87 3.3 Signature 10
88 3.3.1 Key Identifier 11
89 3.3.2 Reference to a Binary Security Token 12
90 3.3.3 Reference to an Issuer and Serial Number 13
91 3.4 Encryption 13
92 3.5 Error Codes 15
93 4 Threat Model and Countermeasures (Non-Normative) 16
94 5 References 17
95 Appendix A: Acknowledgments 19
96 Appendix B: Revision History 22
97

98 **1 Introduction (Non-Normative)**

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

101

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

106

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

109

110 Note that Sections 2.1, 2.2, all of 3, and indicated parts of 5 are normative. All other sections are
111 non-normative.

112 2 Notations and Terminology (Normative)

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

114 2.1 Notational Conventions

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

118

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

122

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

128

129 2.2 Namespaces

130 Namespace URIs (of the general form "some-URI") represents some application-dependent or
131 context-dependent URI as defined in RFC 3986 [URI]. This specification is designed to work with
132 the general SOAP [SOAP11, SOAP12] message structure and message processing model, and
133 should be applicable to any version of SOAP. The current SOAP 1.1 namespace URI is used
134 herein to provide detailed examples, but there is no intention to limit the applicability of this
135 specification to a single version of SOAP.

136

137 The namespaces used in this document are shown in the following table (note that for brevity, the
138 examples use the prefixes listed below but do not include the URIs – those listed below are
139 assumed).

140

141 `http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-`
142 `1.0.xsd`

143 `http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-`
144 `1.0.xsd`

145 `http://docs.oasis-open.org/wss/oasis-wss-wssecurity-secext-1.1.xsd`

146 The following namespace prefixes are used in this document:

Prefix	Namespace
S11	<code>http://schemas.xmlsoap.org/soap/envelope/</code>

S12	http://www.w3.org/2003/05/soap-envelope
ds	http://www.w3.org/2000/09/xmlsig#
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
wsse11	http://docs.oasis-open.org/wss/oasis-wss-wssecurity-secext-1.1.xsd
wsu	http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd

147

Table 1- Namespace prefixes

148 URI fragments defined in this specification are relative to the following base URI unless otherwise
149 stated:

150

151 [http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-](http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0)
152 [profile-1.0](http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0)

153

154 The following table lists the full URI for each URI fragment referred to in this specification.

URI Fragment	Full URI
#Base64Binary	http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#Base64Binary
#STR-Transform	http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#STR-Transform
#PKCS7	http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#PKCS7
#X509v3	http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509v3
#X509PKIPathv1	http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509PKIPathv1
#X509SubjectKeyIdentifier	http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0#X509SubjectKeyIdentifier

155

156 2.3 Terminology

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

159

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

162 3 Usage (Normative)

163 This specification describes the syntax and processing rules for the use of the X.509
164 authentication framework with the Web Services Security: SOAP Message Security specification
165 [WS-Security]. For the purposes of determining the order of preference of reference types, the
166 use of IssuerSerial within X509Data should be considered to be a form of Key Identifier

167 3.1 Token types

168 This profile defines the syntax of, and processing rules for, three types of binary security token
169 using the URI values specified in Table 2.

170

171 If the `ValueType` attribute is missing, the receiver may interpret it either based on a prior
172 agreement or by parsing the content.

173

Token	ValueType URI	Description
Single certificate	#X509v3	An X.509 v3 certificate capable of signature-verification at a minimum
Single certificate	#x509v1	An X.509 v1 certificate capable of signature-verification at a minimum.
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

174

Table 2 – Token types

175 3.1.1 X509v3 Token Type

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

178 3.1.2 X509PKIPathv1 Token Type

179 The `x509PKIPathv1` token type MAY be used to represent a certificate path.

180 3.1.3 PKCS7 Token Type

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

183

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

188 3.2 Token References

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

193

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

196

- 197 • Reference to a Subject Key Identifier
198 The <wsse:SecurityTokenReference> element contains a
199 <wsse:KeyIdentifier> element that specifies the token data by means of a
200 X.509 SubjectKeyIdentifier reference. A subject key identifier may only be used to
201 reference an X.509v3 certificate.”
202
- 203 • Reference to a Binary Security Token
204 The <wsse:SecurityTokenReference> element contains a wsse:Reference>
205 element that references a local <wsse:BinarySecurityToken> element or a
206 remote data source that contains the token data itself.
207
- 208 • Reference to an Issuer and Serial Number
209 The <wsse:SecurityTokenReference> element contains a <ds:X509Data>
210 element that contains a <ds:X509IssuerSerial> element that uniquely identifies
211 an end entity certificate by its X.509 Issuer and Serial Number.

212 3.2.1 Reference to an X.509 Subject Key Identifier

213 The <wsse:KeyIdentifier> element is used to specify a reference to an X.509v3 certificate
214 by means of a reference to its X.509 SubjectKeyIdentifier attribute. This profile defines the syntax
215 of, and processing rules for referencing a Subject Key Identifier using the URI values specified in
216 Table 3 (note that URI fragments are relative to [http://docs.oasis-](http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0)
217 [open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0](http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0)).

218

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

219

Table 3 – Subject Key Identifier

220 The <wsse:SecurityTokenReference> element from which the reference is made contains
221 the <wsse:KeyIdentifier> element. The <wsse:KeyIdentifier> element MUST have a
222 valueType attribute with the value #X509SubjectKeyIdentifier and its contents MUST be
223 the value of the certificate's X.509v3 SubjectKeyIdentifier extension, encoded as per the
224 <wsse:KeyIdentifier> element's EncodingType attribute. For the purposes of this
225 specification, the value of the SubjectKeyIdentifier extension is the contents of the KeyIdentifier
226 octet string, excluding the encoding of the octet string prefix.

227 3.2.2 Reference to a Security Token

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

230

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

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

241 3.3 Signature

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

244

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

251

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

254

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

259 3.3.1 Key Identifier

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

266

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

276

```
277 <S11:Envelope xmlns:S11="...">
278   <S11:Header>
279     <wsse:Security
280       xmlns:wsse="..."
281       xmlns:wsu="...">
282       <ds:Signature
283         xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
284         <ds:SignedInfo>...
285         <ds:Reference URI="#body">...</ds:Reference>
286         <ds:Reference URI="#keyinfo">
287           <ds:Transforms>
288             <ds:Transform Algorithm="...#STR-Transform">
289               <wsse:TransformationParameters>
290                 <ds:CanonicalizationMethod Algorithm="..." />
291               </wsse:TransformationParameters>
292             </ds:Transform>
293           </ds:Transforms>...
294         </ds:Reference>
295       </ds:SignedInfo>
296       <ds:SignatureValue>HFLP...</ds:SignatureValue>
297       <ds:KeyInfo Id="keyinfo">
298         <wsse:SecurityTokenReference>
299           <wsse:KeyIdentifier EncodingType="...#Base64Binary"
300             ValueType="...#X509SubjectKeyIdentifier">
301             MIGfMa0GCSq...
302           </wsse:KeyIdentifier>
303         </wsse:SecurityTokenReference>
304       </ds:KeyInfo>
305     </ds:Signature>
306   </wsse:Security>
307 </S11:Header>
308 <S11:Body wsu:Id="body"
309   xmlns:wsu=".../">
310   ...
```

```
311     </S11:Body>
312 </S11:Envelope>
```

313 3.3.2 Reference to a Binary Security Token

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

318

319 The following example shows a certificate embedded in a <wsse:BinarySecurityToken>
320 element and referenced by URI within a signature. The certificate is included in the
321 <wsse:Security> header as a <wsse:BinarySecurityToken> element with identifier
322 binarytoken. The scope of the signature defined by a <ds:Reference> element within the
323 <ds:SignedInfo> element includes the signing certificate which is referenced by means of the
324 URI bare name pointer #binarytoken. The <ds:KeyInfo> element specifies the signing key
325 by means of a <wsse:SecurityTokenReference> element which contains a
326 <wsse:Reference> element which references the certificate by means of the URI bare name
327 pointer #binarytoken.

328

```
329 <S11:Envelope xmlns:S11="...">
330   <S11:Header>
331     <wsse:Security
332       xmlns:wsse="..."
333       xmlns:wsu="...">
334       <wsse:BinarySecurityToken
335         wsu:Id="binarytoken"
336         ValueType="...#X509v3"
337         EncodingType="...#Base64Binary">
338         MIEZzCCA9CgAwIBAgIQEmtJZc0...
339       </wsse:BinarySecurityToken>
340       <ds:Signature
341         xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
342         <ds:SignedInfo>...
343           <ds:Reference URI="#body">...</ds:Reference>
344           <ds:Reference URI="#binarytoken">...</ds:Reference>
345         </ds:SignedInfo>
346         <ds:SignatureValue>HFLP...</ds:SignatureValue>
347         <ds:KeyInfo>
348           <wsse:SecurityTokenReference>
349             <wsse:Reference URI="#binarytoken" />
350           </wsse:SecurityTokenReference>
351         </ds:KeyInfo>
352       </ds:Signature>
353     </wsse:Security>
354   </S11:Header>
355   <S11:Body wsu:Id="body"
356     xmlns:wsu="...">
357     ...
358   </S11:Body>
359 </S11:Envelope>
```

360 3.3.3 Reference to an Issuer and Serial Number

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

364

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

371

```
372 <S11:Envelope xmlns:S11="...">
373   <S11:Header>
374     <wsse:Security
375       xmlns:wsse="..."
376       xmlns:wsmu="...">
377       <ds:Signature
378         xmlns:ds="...">
379         <ds:SignedInfo>...
380         <ds:Reference URI="#body"></ds:Reference>
381         <ds:Reference URI="#keyinfo"></ds:Reference>
382       </ds:SignedInfo>
383       <ds:SignatureValue>HFLP...</ds:SignatureValue>
384       <ds:KeyInfo Id="keyinfo">
385         <wsse:SecurityTokenReference>
386           <ds:X509Data>
387             <ds:X509IssuerSerial>
388               <ds:X509IssuerName>
389                 DC=ACMECorp, DC=com
390               </ds:X509IssuerName>
391               <ds:X509SerialNumber>12345678</ds:X509SerialNumber>
392             </ds:X509IssuerSerial>
393           </ds:X509Data>
394         </wsse:SecurityTokenReference>
395       </ds:KeyInfo>
396     </ds:Signature>
397   </wsse:Security>
398 </S11:Header>
399 <S11:Body wsu:Id="body"
400   xmlns:wsmu="...">
401   ...
402 </S11:Body>
403 </S11:Envelope>
```

404 3.4 Encryption

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

408

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

411

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

417

```
418 <S11:Envelope  
419     xmlns:S11="..."  
420     xmlns:ds="..."  
421     xmlns:wsse="..."  
422     xmlns:xenc="...">  
423   <S11:Header>  
424     <wsse:Security>  
425       <xenc:EncryptedKey>  
426         <xenc:EncryptionMethod Algorithm="..." />  
427         <ds:KeyInfo>  
428           <wsse:SecurityTokenReference>  
429             <ds:X509Data>  
430               <ds:X509IssuerSerial>  
431                 <ds:X509IssuerName>  
432                   DC=ACMECorp, DC=com  
433                 </ds:X509IssuerName>  
434                 <ds:X509SerialNumber>12345678</ds:X509SerialNumber>  
435               </ds:X509IssuerSerial>  
436             </ds:X509Data>  
437           </wsse:SecurityTokenReference>  
438         </ds:KeyInfo>  
439         <xenc:CipherData>  
440           <xenc:CipherValue>...</xenc:CipherValue>  
441         </xenc:CipherData>  
442         <xenc:ReferenceList>  
443           <xenc:DataReference URI="#encrypted" />  
444         </xenc:ReferenceList>  
445       </xenc:EncryptedKey>  
446     </wsse:Security>  
447   </S11:Header>  
448   <S11:Body>  
449     <xenc:EncryptedData Id="encrypted" Type="...">  
450       <xenc:CipherData>  
451         <xenc:CipherValue>...</xenc:CipherValue>  
452       </xenc:CipherData>  
453     </xenc:EncryptedData>  
454   </S11:Body>  
455 </S11:Envelope>
```

456

457 The following example shows a decryption key referenced by means of the Thumbprint of an
458 associated certificate. In this example the certificate is not included in the message. The
459 <ds:KeyInfo> element contains a <wsse:SecurityTokenReference> element which
460 specifies the Thumbprint of the specified certificate by means of the <http://docs.oasis->

461 open.org/wss/oasis-wss-soap-message-security-1.1#ThumbprintSHA1 attribute of
462 the <wsse:KeyIdentifier> element.

```
463 <S11:Envelope
464   xmlns:S11="..."
465   xmlns:ds="..."
466   xmlns:wsse="..."
467   xmlns:xenc="...">
468   <S11:Header>
469     <wsse:Security>
470       <xenc:EncryptedKey>
471         <xenc:EncryptionMethod Algorithm="..."/>
472         <ds:KeyInfo>
473           <wsse:SecurityTokenReference>
474             <wsse:KeyIdentifier
475               ValueType="http://docs.oasis-open.org/wss/oasis-wss-
476 soap-message-security-1.1#ThumbPrintSHA1" >LKiQ/CmFrJDJqCLFcjlhIsmZ/+0=
477             </wsse:KeyIdentifier>
478           </wsse:SecurityTokenReference>
479         </ds:KeyInfo>
480       <xenc:CipherData>
481         <xenc:CipherValue>...</xenc:CipherValue>
482       </xenc:CipherData>
483       <xenc:ReferenceList>
484         <xenc:DataReference URI="#encrypted"/>
485       </xenc:ReferenceList>
486     </xenc:EncryptedKey>
487   </wsse:Security>
488 </S11:Header>
489 <S11:Body>
490   <xenc:EncryptedData Id="encrypted" Type="...">
491     <xenc:CipherData>
492       <xenc:CipherValue>...</xenc:CipherValue>
493     </xenc:CipherData>
494   </xenc:EncryptedData>
495 </S11:Body>
496 </S11:Envelope>
```

497

498 3.5 Error Codes

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

501

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

505

506 **4 Threat Model and Countermeasures (Non-**
507 **Normative)**

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

510

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

516

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

518

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

522

5 References

523 The following are normative references

- 524 **[Glossary]** Informational RFC 2828, *Internet Security Glossary*, May 2000.
525 <http://www.ietf.org/rfc/rfc2828.txt>
- 526 **[KEYWORDS]** S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*,
527 RFC 2119, Harvard University, March 1997,
528 <http://www.ietf.org/rfc/rfc2119.txt>
- 529 **[RFC2246]** T. Dierks, C. Allen., *The TLS Protocol Version, 1.0*. IETF RFC 2246
530 January 1999. <http://www.ietf.org/rfc/rfc2246.txt>
- 531 **[SOAP11]** W3C Note, "SOAP: Simple Object Access Protocol 1.1," 08 May 2000.
- 532 **[SOAP12]** W3C Recommendation, "SOAP Version 1.2 Part 1: Messaging
533 Framework", 23 June 2003.
- 534 **[URI]** T. Berners-Lee, R. Fielding, L. Masinter, "Uniform Resource Identifiers
535 (URI): Generic Syntax," RFC 3986, MIT/LCS, Day Software, Adobe
536 Systems, January 2005.
- 537 **[WS-Security]** A. Nadalin et al., *Web Services Security: SOAP Message Security 1.1*
538 (WS-Security 2004), OASIS Standard, [http://docs.oasis-](http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.1.pdf)
539 [open.org/wss/2004/01/oasis-200401-wss-soap-message-security-](http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.1.pdf)
540 1.1.pdf.
- 541 **[PKCS7]** *PKCS #7: Cryptographic Message Syntax Standard* RSA Laboratories,
542 November 1, 1993. [http://www.rsasecurity.com/rsalabs/pkcs/pkcs-](http://www.rsasecurity.com/rsalabs/pkcs/pkcs-7/index.html)
543 7/index.html
- 544 **[PKIPATH]** [http://www.itu.int/rec/recommendation.asp?type=items&lang=e&parent=T-](http://www.itu.int/rec/recommendation.asp?type=items&lang=e&parent=T-REC-X.509-200110-S!Cor1)
545 REC-X.509-200110-S!Cor1
- 546 **[X509]** ITU-T Recommendation X.509 (1997 E): *Information Technology - Open*
547 *Systems Interconnection - The Directory: Authentication Framework*,
548 June 1997.

549

550 The following are non-normative references

- 551 **[XML-ns]** T. Bray, D. Hollander, A. Layman. *Namespaces in XML. W3C*
552 *Recommendation*. January 1999. [http://www.w3.org/TR/1999/REC-xml-](http://www.w3.org/TR/1999/REC-xml-names-19990114)
553 names-19990114
- 554 **[XML Encrypt]** W3C Recommendation, "XML Encryption Syntax and Processing," 10
555 December 2002
- 556 **[XML Signature]** D. Eastlake, J. R., D. Solo, M. Bartel, J. Boyer , B. Fox , E. Simon. *XML-*
557 *Signature Syntax and Processing*, W3C Recommendation, 12 February
558 2002.

559

560

Appendix A: Acknowledgments

Current Contributors:

Michael	Hu	Actional
Maneesh	Sahu	Actional
Duane	Nickull	Adobe Systems
Gene	Thurston	AmberPoint
Frank	Siebenlist	Argonne National Laboratory
Hal	Lockhart	BEA Systems
Denis	Pilipchuk	BEA Systems
Corinna	Witt	BEA Systems
Steve	Anderson	BMC Software
Rich	Levinson	Computer Associates
Thomas	DeMartini	ContentGuard
Merlin	Hughes	Cybertrust
Dale	Moberg	Cyclone Commerce
Rich	Salz	Datapower
Sam	Wei	EMC
Dana S.	Kaufman	Forum Systems
Toshihiro	Nishimura	Fujitsu
Kefeng	Chen	GeoTrust
Irving	Reid	Hewlett-Packard
Kojiro	Nakayama	Hitachi
Paula	Austel	IBM
Derek	Fu	IBM
Maryann	Hondo	IBM
Kelvin	Lawrence	IBM
Michael	McIntosh	IBM
Anthony	Nadalin	IBM
Nataraj	Nagaratnam	IBM
Bruce	Rich	IBM
Ron	Williams	IBM
Don	Flinn	Individual
Kate	Cherry	Lockheed Martin
Paul	Cotton	Microsoft
Vijay	Gajjala	Microsoft
Martin	Gudgin	Microsoft
Chris	Kaler	Microsoft
Frederick	Hirsch	Nokia
Abbie	Barbir	Nortel
Prateek	Mishra	Oracle
Vamsi	Motukuru	Oracle
Ramana	Turlapi	Oracle
Ben	Hammond	RSA Security
Rob	Philpott	RSA Security
Blake	Dournaee	Sarvega
Sundeep	Pechu	Sarvega

Coumara	Radja	Sarvega
Pete	Wenzel	SeeBeyond
Manveen	Kaur	Sun Microsystems
Ronald	Monzillo	Sun Microsystems
Jan	Alexander	Systinet
Symon	Chang	TIBCO Software
John	Weiland	US Navy
Hans	Granqvist	VeriSign
Phillip	Hallam-Baker	VeriSign
Hemma	Prafullchandra	VeriSign

564

Previous Contributors:

Peter	Dapkus	BEA
Guillermo	Lao	ContentGuard
TJ	Pannu	ContentGuard
Xin	Wang	ContentGuard
Shawn	Sharp	Cyclone Commerce
Ganesh	Vaideeswaran	Documentum
Tim	Moses	Entrust
Carolina	Canales-Valenzuela	Ericsson
Tom	Rutt	Fujitsu
Yutaka	Kudo	Hitachi
Jason	Rouault	HP
Bob	Blakley	IBM
Joel	Farrell	IBM
Satoshi	Hada	IBM
Hiroshi	Maruyama	IBM
David	Melgar	IBM
Kent	Tamura	IBM
Wayne	Vicknair	IBM
Phil	Griffin	Individual
Mark	Hayes	Individual
John	Hughes	Individual
Peter	Rostin	Individual
Davanum	Srinivas	Individual
Bob	Morgan	Individual/Internet2
Bob	Atkinson	Microsoft
Keith	Ballinger	Microsoft
Allen	Brown	Microsoft
Giovanni	Della-Libera	Microsoft
Alan	Geller	Microsoft
Johannes	Klein	Microsoft
Scott	Konersmann	Microsoft
Chris	Kurt	Microsoft
Brian	LaMacchia	Microsoft
Paul	Leach	Microsoft
John	Manferdelli	Microsoft
John	Shewchuk	Microsoft
Dan	Simon	Microsoft

Hervey	Wilson	Microsoft
Jeff	Hodges	Neustar
Senthil	Sengodan	Nokia
Lloyd	Burch	Novell
Ed	Reed	Novell
Charles	Knouse	Oblix
Vipin	Samar	Oracle
Jerry	Schwarz	Oracle
Eric	Gravengaard	Reactivity
Andrew	Nash	Reactivity
Stuart	King	Reed Elsevier
Martijn	de Boer	SAP
Jonathan	Tourzan	Sony
Yassir	Elley	Sun
Michael	Nguyen	The IDA of Singapore
Don	Adams	TIBCO
Morten	Jorgensen	Vordel

565

566

Appendix B: Revision History

567

Rev	Date	By Whom	What
-----	------	---------	------