



Web Services Security

X.509 Certificate Token Profile 1.1

OASIS Public Review Draft – 28 June 2005

OASIS Identifier:

{product-productVersion-artifactType-stage-descriptiveName-revision.form (Word) (PDF)
(HTML)}

Document Location:

wss-v1.1-spec-pr-x509TokenProfile-01

Location:

Persistent:

[persistent location]

This Version:

<http://docs.oasis-open.org/wss/2005/xx/wss-v1.1-spec-pr-x509TokenProfile-0>

Previous Version:

none

Technical Committee:

Web Service Security (WSS)

Chairs:

Kelvin Lawrence, IBM

Chris Kaler, Microsoft

Editors:

Anthony Nadalin, IBM

Chris Kaler, Microsoft

Ronald Monzillo, Sun

Phillip Hallam-Baker, Verisign

Abstract:

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

Status:

This is an interim draft.

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

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

41 **Notices**

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

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

54 Copyright © The Organization for the Advancement of Structured Information Standards [OASIS]
55 2002-2005. All Rights Reserved.

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

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

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

72 **Table of Contents**

73 1 Introduction (Non-Normative) 5
74 2 Notations and Terminology (Normative) 6
75 2.1 Notational Conventions 6
76 2.2 Namespaces 6
77 2.3 Terminology 7
78 3 Usage (Normative) 9
79 3.1 Token types 9
80 3.1.1 X509 Token Type 9
81 3.1.2 X509PKIPathv1 Token Type 9
82 3.1.3 PKCS7 Token Type 9
83 3.2 Token References 10
84 3.2.1 Reference to an X.509 Subject Key Identifier 10
85 3.2.2 Reference to a Security Token 11
86 3.2.3 Reference to an Issuer and Serial Number 11
87 3.2.4 Thumbprint References 11
88 3.3 Signature 11
89 3.3.1 Key Identifier 12
90 3.3.2 Reference to a Binary Security Token 13
91 3.3.3 Reference to an Issuer and Serial Number 14
92 3.4 Encryption 15
93 3.5 Error Codes 16
94 4 Threat Model and Countermeasures (Non-Normative) 17
95 5 References 18
96 Appendix A: Acknowledgments **Error! Bookmark not defined.**
97 Appendix B: Revision History 21
98

99 **1 Introduction (Non-Normative)**

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

102

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

107

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

110

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

113 2 Notations and Terminology (Normative)

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

115 2.1 Notational Conventions

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

119

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

123

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

129

130 2.2 Namespaces

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

137

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

141

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

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

146 `http://docs.oasis-open.org/wss/2005/xx/oasis-2005xx-wss-`
147 `wssecurity-secext-1.1.xsd`

148

149 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/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/2005/xx/oasis-2005xx-wss-wssecurity-secext-1.1.xsd
wsu	http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd

150

Table 1- Namespace prefixes

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

153

154 [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)
 155 [profile-1.0](http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0)

156

157 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

158

159 2.3 Terminology

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

162

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

165 3 Usage (Normative)

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

170 3.1 Token types

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

173

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

176

Token	ValueType URI	Description
Single certificate	#X509v3	An X.509 v3 signature-verification certificate
Single certificate	#x509v1	An X.509 v1 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

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

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

199

200 Reference to a Subject Key Identifier

201 The <wsse:SecurityTokenReference> element contains a
 202 <wsse:KeyIdentifier> element that specifies the token data by means of a X.509
 203 SubjectKeyIdentifier reference. A subject key identifier may only be used to reference an
 204 X.509v3 certificate.”

205

206 Reference to a Binary Security Token

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

210

211 Reference to an Issuer and Serial Number

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

215 **3.2.1 Reference to an X.509 Subject Key Identifier**

216 The <wsse:KeyIdentifier> element is used to specify a reference to an X.509v3 certificate
 217 by means of a reference to its X.509 SubjectKeyIdentifier attribute. This profile defines the syntax
 218 of, and processing rules for referencing a Subject Key Identifier using the URI values specified in
 219 Table 3 (note that URI fragments are relative to the URI for this specification).

220

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

221

Table 3 – Subject Key Identifier

222 The <wsse:SecurityTokenReference> element from which the reference is made contains
 223 the <wsse:KeyIdentifier> element. The <wsse:KeyIdentifier> element MUST have a
 224 ValueType attribute with the value #X509SubjectKeyIdentifier and its contents MUST be the
 225 value of the certificate's X.509v3 SubjectKeyIdentifier extension, encoded as per the

226 <wsse:KeyIdentifier> element's EncodingType attribute. For the purposes of this
227 specification, the value of the SubjectKeyIdentifier extension is the contents of the KeyIdentifier
228 octet string, excluding the encoding of the octet string prefix.

229 3.2.2 Reference to a Security Token

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

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

235 3.2.3 Reference to an Issuer and Serial Number

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

238

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

242 3.2.4 Thumbprint References

243 The <wsse:KeyIdentifier> element is used to specify a reference to an X.509 certificate by
244 means of a reference to its X.509 Thumbprint attribute. This profile defines the syntax of, and the
245 processing rules for referencing a Thumbprint using the URI values specified below (note that the
246 URI fragment is relative to [http://docs.oasis-open.org/wss/2005/xx/oasis-2005xx-
247 wss-soap-message-security-1.1](http://docs.oasis-open.org/wss/2005/xx/oasis-2005xx-wss-soap-message-security-1.1)):

248

Subject Key Identifier	ValueType URI	Description
Thumbprint	#X509ThumbprintSHA1	The thumbprint of the X.509 certificate

249

250 The <wsse:SecurityTokenReference> element from which the reference is made contains a
251 <wsse:KeyIdentifier> element. The <wsse:KeyIdentifier> element MUST have a
252 ValueType attribute with the value or [http://docs.oasis-
253 open.org/wss/2005/xx/oasis-2005xx-wss-soap-message-security-
254 1.1#ThumbprintSHA1](http://docs.oasis-open.org/wss/2005/xx/oasis-2005xx-wss-soap-message-security-1.1#ThumbprintSHA1) and its contents MUST be the thumbprint for the desired certificate . If
255 the certificate does not contain a X.509 Thumbprint extension, then one is computed as the SHA1
256 of the raw octets which would be encoded within the <wsse:BinarySecurityToken> element
257 were it to be included. The thumbprint is encoded as per the <wsse:KeyIdentifier>
258 element's EncodingType attribute. The default encoding is Base64. Implementations compliant
259 with this specification MAY support such a certificate reference mechanism.

260 3.3 Signature

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

263

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

270

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

273

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

278 3.3.1 Key Identifier

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

285

286 The following example shows a certificate referenced by means of a KeyIdentifier. The scope of
287 the signature is the <ds:SignedInfo> element which includes both the message body (#body)
288 and the signing certificate by means of a reference to the <ds:KeyInfo> element which
289 references it (#keyinfo). Since the <ds:KeyInfo> element only contains a mutable reference to
290 the certificate rather than the certificate itself, a transformation is specified which replaces the
291 reference to the certificate with the certificate. The <ds:KeyInfo> element specifies the signing
292 key by means of a <wsse:SecurityTokenReference> element which contains a
293 <wsse:KeyIdentifier> element which specifies the X.509 subject key identifier of the signing
294 certificate.

295

```
296 <S11:Envelope xmlns:S11="...">  
297   <S11:Header>  
298     <wsse:Security  
299       xmlns:wsse="..."  
300       xmlns:wssu="...">  
301       <ds:Signature  
302         xmlns:ds="http://www.w3.org/2000/09/xmldsig#">  
303         <ds:SignedInfo>...  
304         <ds:Reference URI="#body">...</ds:Reference>  
305         <ds:Reference URI="#keyinfo">  
306           <ds:Transforms>  
307             <ds:Transform Algorithm="...#STR-Transform">  
308               <wsse:TransformationParameters>  
309                 <ds:CanonicalizationMethod Algorithm="..." />  
310               </wsse:TransformationParameters>
```

```

311         </ds:Transform>
312     </ds:Transforms>...
313 </ds:Reference>
314 </ds:SignedInfo>
315 <ds:SignatureValue>HFLP...</ds:SignatureValue>
316 <ds:KeyInfo Id="keyinfo">
317     <wsse:SecurityTokenReference>
318         <wsse:KeyIdentifier EncodingType="...#Base64Binary"
319             ValueType="...#X509SubjectKeyIdentifier">
320             MIGfMa0GCSq...
321         </wsse:KeyIdentifier>
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 3.3.2 Reference to a Binary Security Token

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

337

338 The following example shows a certificate embedded in a <wsse:BinarySecurityToken>
339 element and referenced by URI within a signature. The certificate is included in the
340 <wsse:Security> header as a <wsse:BinarySecurityToken> element with identifier
341 binarytoken. The scope of the signature defined by a <ds:Reference> element within the
342 <ds:SignedInfo> element includes the signing certificate which is referenced by means of the
343 URI bare name pointer #binarytoken. The <ds:KeyInfo> element specifies the signing key
344 by means of a <wsse:SecurityTokenReference> element which contains a
345 <wsse:Reference> element which references the certificate by means of the URI bare name
346 pointer #binarytoken.

```

347 <S11:Envelope xmlns:S11="...">
348     <S11:Header>
349         <wsse:Security
350             xmlns:wsse="..."
351             xmlns:wsu="...">
352             <wsse:BinarySecurityToken
353                 wsu:Id="binarytoken"
354                 ValueType="...#X509v3"
355                 EncodingType="...#Base64Binary">
356                 MIIIEZzCCA9CgAwIBAgIQEmtJZc0...
357             </wsse:BinarySecurityToken>
358             <ds:Signature
359                 xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
360                 <ds:SignedInfo>...
361                 <ds:Reference URI="#body">...</ds:Reference>
362                 <ds:Reference URI="#binarytoken">...</ds:Reference>

```

```

363     </ds:SignedInfo>
364     <ds:SignatureValue>HFLP...</ds:SignatureValue>
365     <ds:KeyInfo>
366         <wsse:SecurityTokenReference>
367             <wsse:Reference URI="#binarytoken" />
368         </wsse:SecurityTokenReference>
369     </ds:KeyInfo>
370 </ds:Signature>
371 </wsse:Security>
372 </S11:Header>
373 <S11:Body wsu:Id="body"
374     xmlns:wsu="...">
375     ...
376 </S11:Body>
377 </S11:Envelope>

```

378 3.3.3 Reference to an Issuer and Serial Number

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

382

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

389

```

390 <S11:Envelope xmlns:S11="...">
391   <S11:Header>
392     <wsse:Security
393       xmlns:wsse="..."
394       xmlns:wsu="...">
395     <ds:Signature
396       xmlns:ds="...">
397       <ds:SignedInfo>...
398         <ds:Reference URI="#body"></ds:Reference>
399         <ds:Reference URI="#keyinfo"></ds:Reference>
400       </ds:SignedInfo>
401       <ds:SignatureValue>HFLP...</ds:SignatureValue>
402       <ds:KeyInfo Id="keyinfo">
403         <wsse:SecurityTokenReference>
404           <ds:X509Data>
405             <ds:X509IssuerSerial>
406               <ds:X509IssuerName>
407                 DC=ACMECorp, DC=com
408               </ds:X509IssuerName>
409               <ds:X509SerialNumber>12345678</X509SerialNumber>
410             </ds:X509IssuerSerial>
411           </ds:X509Data>
412         </wsse:SecurityTokenReference>
413       </ds:KeyInfo>
414     </ds:Signature>

```

```

415     </wsse:Security>
416 </S11:Header>
417 <S11:Body wsu:Id="body"
418     xmlns:wsu="...">
419     ...
420 </S11:Body>
421 </S11:Envelope>

```

3.4 Encryption

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

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

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

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

```

439 <S11:Envelope
440     xmlns:S11="..."
441     xmlns:ds="..."
442     xmlns:wsse="..."
443     xmlns:xenc="...">
444 <S11:Header>
445     <wsse:Security>
446         <xenc:EncryptedKey>
447             <xenc:EncryptionMethod Algorithm="..." />
448             <ds:KeyInfo>
449                 <wsse:SecurityTokenReference>
450                     <ds:X509Data>
451                         <ds:X509IssuerSerial>
452                             <ds:X509IssuerName>
453                                 DC=ACMECorp, DC=com
454                             </ds:X509IssuerName>
455                             <ds:X509SerialNumber>12345678</X509SerialNumber>
456                         </ds:X509IssuerSerial>
457                     </ds:X509Data>
458                 </wsse:SecurityTokenReference>
459             </ds:KeyInfo>
460             <xenc:CipherData>
461                 <xenc:CipherValue>...</xenc:CipherValue>
462             </xenc:CipherData>
463             <xenc:ReferenceList>
464                 <xenc:DataReference URI="#encrypted"/>

```

```
465         </xenc:ReferenceList>
466         </xenc:EncryptedKey>
467     </wsse:Security>
468 </S11:Header>
469 <S11:Body>
470     <xenc:EncryptedData Id="encrypted" Type="...">
471         <xenc:CipherData>
472             <xenc:CipherValue>...</xenc:CipherValue>
473         </xenc:CipherData>
474     </xenc:EncryptedData>
475 </S11:Body>
476 </S11:Envelope>
```

477 **3.5 Error Codes**

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

480

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

484

485 **4 Threat Model and Countermeasures (Non-**
486 **Normative)**

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

489

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

495

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

497

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

5 References

501

502 The following are normative references

- 503 **[Glossary]** Informational RFC 2828, *Internet Security Glossary*, May 2000.
504 <http://www.ietf.org/rfc/rfc2828.txt>
- 505 **[KEYWORDS]** S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*,
506 RFC 2119, Harvard University, March 1997,
507 <http://www.ietf.org/rfc/rfc2119.txt>
- 508 **[RFC2246]** T. Dierks, C. Allen., *The TLS Protocol Version, 1.0*. IETF RFC 2246
509 January 1999. <http://www.ietf.org/rfc/rfc2246.txt>
- 510 **[SOAP11]** W3C Note, "SOAP: Simple Object Access Protocol 1.1," 08 May 2000.
- 511 **[SOAP12]** W3C Recommendation, "SOAP Version 1.2 Part 1: Messaging
512 Framework", 23 June 2003.
- 513 **[URI]** T. Berners-Lee, R. Fielding, L. Masinter, "Uniform Resource Identifiers
514 (URI): Generic Syntax," RFC 3986, MIT/LCS, Day Software, Adobe
515 Systems, January 2005.

516

517 The following are non-normative references

- 518 **[WS-Security]** OASIS, "Web Services Security: SOAP Message Security" 19 January
519 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)
520 [soap-message-security-1.0](http://www.docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0)
- 521 **[XML-ns]** T. Bray, D. Hollander, A. Layman. *Namespaces in XML. W3C*
522 *Recommendation*. January 1999. [http://www.w3.org/TR/1999/REC-xml-](http://www.w3.org/TR/1999/REC-xml-names-19990114)
523 [names-19990114](http://www.w3.org/TR/1999/REC-xml-names-19990114)
- 524 **[XML Encrypt]** W3C Recommendation, "XML Encryption Syntax and Processing," 10
525 December 2002
- 526 **[XML Signature]** D. Eastlake, J. R., D. Solo, M. Bartel, J. Boyer , B. Fox , E. Simon. *XML-*
527 *Signature Syntax and Processing*, W3C Recommendation, 12 February
528 2002. <http://www.w3.org/TR/xmlsig-core/>
- 529 **[PKCS7]** *PKCS #7: Cryptographic Message Syntax Standard* RSA Laboratories,
530 November 1, 1993. [http://www.rsasecurity.com/rsalabs/pkcs/pkcs-](http://www.rsasecurity.com/rsalabs/pkcs/pkcs-7/index.html)
531 [7/index.html](http://www.rsasecurity.com/rsalabs/pkcs/pkcs-7/index.html)
- 532 **[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-!Cor1)
533 [REC-X.509-200110-!Cor1](http://www.itu.int/rec/recommendation.asp?type=items&lang=e&parent=T-REC-X.509-200110-!Cor1)
- 534 **[X509]** ITU-T Recommendation X.509 (1997 E): Information Technology - *Open*
535 *Systems Interconnection - The Directory: Authentication Framework*,
536 June 1997.

537

538

Appendix A: Acknowledgments

Contributors:

Gene	Thurston	AmberPoint
Frank	Siebenlist	Argonne National Lab
Merlin	Hughes	Baltimore Technologies
Irving	Reid	Baltimore Technologies
Peter	Dapkus	BEA
Hal	Lockhart	BEA
Steve	Anderson	BMC (Sec)
Srinivas	Davanum	Computer Associates
Thomas	DeMartini	ContentGuard
Guillermo	Lao	ContentGuard
TJ	Pannu	ContentGuard
Shawn	Sharp	Cyclone Commerce
Ganesh	Vaideeswaran	Documentum
Sam	Wei	Documentum
John	Hughes	Entegrity
Tim	Moses	Entrust
Toshihiro	Nishimura	Fujitsu
Tom	Rutt	Fujitsu
Jason	Rouault	HP
Yutaka	Kudo	Hitachi
Paula	Austel	IBM
Maryann	Hondo	IBM
Michael	McIntosh	IBM
Kelvin	Lawrence	IBM (co-Chair)
Anthony	Nadalin	IBM
Nataraj	Nagaratnam	IBM
Don	Flinn	Individual
Bob	Morgan	Individual
Paul	Cotton	Microsoft
Vijay	Gajjala	Microsoft
Chris	Kaler	Microsoft (co-Chair)
Chris	Kurt	Microsoft
John	Shewchuk	Microsoft
Prateek	Mishra	Netegrity
Frederick	Hirsch	Nokia
Senthil	Sengodan	Nokia
Lloyd	Burch	Novell
Ed	Reed	Novell
Charles	Knouse	Oblix
Vipin	Samar	Oracle
Jerry	Schwarz	Oracle
Eric	Gravengaard	Reactivity
Stuart	King	Reed Elsevier
Andrew	Nash	RSA Security
Rob	Philpott	RSA Security

Peter	Rostin	RSA Security
Martijn	de Boer	SAP
Blake	Dournaee	Sarvega
Pete	Wenzel	SeeBeyond
Jonathan	Tourzan	Sony
Yassir	Elley	Sun Microsystems
Jeff	Hodges	Sun Microsystems
Ronald	Monzillo	Sun Microsystems
Jan	Alexander	Systinet
Michael	Nguyen	The IDA of Singapore
Don	Adams	TIBCO
Symon	Chang	TIBCO
John	Weiland	US Navy
Phillip	Hallam-Baker	VeriSign
Morten	Jorgensen	Vordel

541

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

Bob	Blakley	IBM
Joel	Farrell	IBM
Satoshi	Hada	IBM
Hiroshi	Maruyama	IBM
David	Melgar	IBM
Bob	Atkinson	Microsoft
Allen	Brown	Microsoft
Giovanni	Della-Libera	Microsoft
Johannes	Klein	Microsoft
Scott	Konersmann	Microsoft
Brian	LaMacchia	Microsoft
Paul	Leach	Microsoft
John	Manferdelli	Microsoft
Dan	Simon	Microsoft
Hervey	Wilson	Microsoft
Hemma	Prafullchandra	VeriSign

543

544

Appendix B: Revision History

Rev	Date	By Whom	What
WGD 1.1	2004-09-13	Anthony Nadalin	Initial version cloned from the Vwesion 1.1 and Errata
WGD 1.1	2005-03-22	Anthony Nadalin	Issue 373
WGD 1.1	2005-05-11	Anthony Nadalin	Issue 388
WGD 1.1	2005-05-17	Anthony Nadalin	Formatting Issues
WGD 1.1	2005-06-14	Anthony Nadalin	Fix Example

545