



# Web Services Security X.509 Certificate Token Profile 1.1

OASIS Standard Specification, 1 February 2006

**OASIS Identifier:**

wss-v1.1-spec-os-X509TokenProfile

**Document Location:**

<http://docs.oasis-open.org/wss/v1.1/>

**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 OASIS Standard document produced by the Web Services Security Technical Committee. It was approved by the OASIS membership on 1 February 2006. Check the current location noted above for possible errata to this document.

Technical Committee members should send comments on this specification to the technical Committee's email list. Others should send comments to the Technical Committee by using the "Send A Comment" button on the Technical Committee's web page at [www.oasisopen.org/committees/wss](http://www.oasisopen.org/committees/wss).

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

---

37 **Notices**

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

50

51 Copyright (C) OASIS Open 2002-2006. All Rights Reserved.

52

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

62

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

65

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

71

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

74

75 This section is non-normative.

---

76 **Table of Contents**

77 1 Introduction (Non-Normative) ..... 4  
78 2 Notations and Terminology (Normative) ..... 5  
79 2.1 Notational Conventions ..... 5  
80 2.2 Namespaces ..... 5  
81 2.3 Terminology ..... 6  
82 3 Usage (Normative) ..... 8  
83 3.1 Token types ..... 8  
84 3.1.1 X509v3 Token Type ..... 8  
85 3.1.2 X509PKIPathv1 Token Type ..... 8  
86 3.1.3 PKCS7 Token Type ..... 8  
87 3.2 Token References ..... 9  
88 3.2.1 Reference to an X.509 Subject Key Identifier ..... 9  
89 3.2.2 Reference to a Security Token ..... 10  
90 3.2.3 Reference to an Issuer and Serial Number ..... 10  
91 3.3 Signature ..... 10  
92 3.3.1 Key Identifier ..... 11  
93 3.3.2 Reference to a Binary Security Token ..... 12  
94 3.3.3 Reference to an Issuer and Serial Number ..... 13  
95 3.4 Encryption ..... 13  
96 3.5 Error Codes ..... 15  
97 4 Threat Model and Countermeasures (Non-Normative) ..... 16  
98 5 References ..... 17  
99 Appendix A: Acknowledgments ..... 19  
100 Appendix B: Revision History ..... 22  
101

---

## 102 **1 Introduction (Non-Normative)**

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

105

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

110

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

113

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

---

## 116 2 Notations and Terminology (Normative)

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

### 118 2.1 Notational Conventions

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

122

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

126

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

132

### 133 2.2 Namespaces

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

140

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

144

145 `http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-`  
146 `1.0.xsd`

147 `http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-`  
148 `1.0.xsd`

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

150 The following namespace prefixes are used in this document:

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

S12	<a href="http://www.w3.org/2003/05/soap-envelope">http://www.w3.org/2003/05/soap-envelope</a>
ds	<a href="http://www.w3.org/2000/09/xmlsig#">http://www.w3.org/2000/09/xmlsig#</a>
xenc	<a href="http://www.w3.org/2001/04/xmlenc#">http://www.w3.org/2001/04/xmlenc#</a>
wsse	<a href="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd</a>
wsse11	<a href="http://docs.oasis-open.org/wss/oasis-wss-wssecurity-secext-1.1.xsd">http://docs.oasis-open.org/wss/oasis-wss-wssecurity-secext-1.1.xsd</a>
wsu	<a href="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd</a>

151

*Table 1- Namespace prefixes*

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

154

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

157

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

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

159

## 160 2.3 Terminology

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

163

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

---

## 166 3 Usage (Normative)

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

### 171 3.1 Token types

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

174

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

177

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

178

*Table 2 – Token types*

#### 179 3.1.1 X509v3 Token Type

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

#### 182 3.1.2 X509PKIPathv1 Token Type

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

#### 184 3.1.3 PKCS7 Token Type

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

187

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

### 192 3.2 Token References

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

197

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

200

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

#### 216 3.2.1 Reference to an X.509 Subject Key Identifier

217 The <wsse:KeyIdentifier> element is used to specify a reference to an X.509v3 certificate  
218 by means of a reference to its X.509 SubjectKeyIdentifier attribute. This profile defines the syntax  
219 of, and processing rules for referencing a Subject Key Identifier using the URI values specified in  
220 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)  
221 [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)).

222

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

223

*Table 3 – Subject Key Identifier*

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

### 231 3.2.2 Reference to a Security Token

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

234

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

### 238 3.2.3 Reference to an Issuer and Serial Number

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

241

242 The <ds:X509IssuerSerial> element is a direct child of the <ds:X509Data> element that is  
243 in turn a direct child of the <wsse:SecurityTokenReference> element in which the  
244 reference is made

## 245 3.3 Signature

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

248

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

255

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

258

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

263

### 3.3.1 Key Identifier

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

270

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

280

```
281 <S11:Envelope xmlns:S11="...">
282   <S11:Header>
283     <wsse:Security
284       xmlns:wsse="..."
285       xmlns:wsu="...">
286       <ds:Signature
287         xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
288         <ds:SignedInfo>...
289         <ds:Reference URI="#body">...</ds:Reference>
290         <ds:Reference URI="#keyinfo">
291           <ds:Transforms>
292             <ds:Transform Algorithm="...#STR-Transform">
293               <wsse:TransformationParameters>
294                 <ds:CanonicalizationMethod Algorithm="..." />
295               </wsse:TransformationParameters>
296             </ds:Transform>
297           </ds:Transforms>...
298         </ds:Reference>
299       </ds:SignedInfo>
300       <ds:SignatureValue>HFLP...</ds:SignatureValue>
301       <ds:KeyInfo Id="keyinfo">
302         <wsse:SecurityTokenReference>
303           <wsse:KeyIdentifier EncodingType="...#Base64Binary"
304             ValueType="...#X509SubjectKeyIdentifier">
305             MIGfMa0GCSq...
306           </wsse:KeyIdentifier>
307         </wsse:SecurityTokenReference>
308       </ds:KeyInfo>
309     </ds:Signature>
310   </wsse:Security>
311 </S11:Header>
312 <S11:Body wsu:Id="body"
313   xmlns:wsu=".../">
314   ...
```

```
315     </S11:Body>
316 </S11:Envelope>
```

### 317 3.3.2 Reference to a Binary Security Token

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

322

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

332

```
333 <S11:Envelope xmlns:S11="...">
334   <S11:Header>
335     <wsse:Security
336       xmlns:wsse="..."
337       xmlns:wsu="...">
338       <wsse:BinarySecurityToken
339         wsu:Id="binarytoken"
340         ValueType="...#X509v3"
341         EncodingType="...#Base64Binary">
342         MIEZzCCA9CgAwIBAgIQEmtJZc0...
343       </wsse:BinarySecurityToken>
344       <ds:Signature
345         xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
346         <ds:SignedInfo>...
347           <ds:Reference URI="#body">...</ds:Reference>
348           <ds:Reference URI="#binarytoken">...</ds:Reference>
349         </ds:SignedInfo>
350         <ds:SignatureValue>HFLP...</ds:SignatureValue>
351         <ds:KeyInfo>
352           <wsse:SecurityTokenReference>
353             <wsse:Reference URI="#binarytoken" />
354           </wsse:SecurityTokenReference>
355         </ds:KeyInfo>
356       </ds:Signature>
357     </wsse:Security>
358   </S11:Header>
359   <S11:Body wsu:Id="body"
360     xmlns:wsu="...">
361     ...
362   </S11:Body>
363 </S11:Envelope>
```

### 364 3.3.3 Reference to an Issuer and Serial Number

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

368

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

375

```
376 <S11:Envelope xmlns:S11="...">
377   <S11:Header>
378     <wsse:Security
379       xmlns:wsse="..."
380       xmlns:wsmu="...">
381       <ds:Signature
382         xmlns:ds="...">
383         <ds:SignedInfo>...
384         <ds:Reference URI="#body"></ds:Reference>
385         <ds:Reference URI="#keyinfo"></ds:Reference>
386       </ds:SignedInfo>
387       <ds:SignatureValue>HFLP...</ds:SignatureValue>
388       <ds:KeyInfo Id="keyinfo">
389         <wsse:SecurityTokenReference>
390           <ds:X509Data>
391             <ds:X509IssuerSerial>
392               <ds:X509IssuerName>
393                 DC=ACMECorp, DC=com
394               </ds:X509IssuerName>
395               <ds:X509SerialNumber>12345678</ds:X509SerialNumber>
396             </ds:X509IssuerSerial>
397           </ds:X509Data>
398         </wsse:SecurityTokenReference>
399       </ds:KeyInfo>
400     </ds:Signature>
401   </wsse:Security>
402 </S11:Header>
403 <S11:Body wsu:Id="body"
404   xmlns:wsmu="...">
405   ...
406 </S11:Body>
407 </S11:Envelope>
```

### 408 3.4 Encryption

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

412

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

415

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

421

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

460

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

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

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

501

### 502 3.5 Error Codes

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

505

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

509

---

510 **4 Threat Model and Countermeasures (Non-**  
511 **Normative)**

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

514

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

520

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

522

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

---

## 5 References

526

527 The following are normative references

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

553

554 The following are non-normative references

- 555       **[XML-ns]**        T. Bray, D. Hollander, A. Layman. *Namespaces in XML. W3C*  
556                            *Recommendation*. January 1999. [http://www.w3.org/TR/1999/REC-xml-](http://www.w3.org/TR/1999/REC-xml-names-19990114)  
557                            [names-19990114](http://www.w3.org/TR/1999/REC-xml-names-19990114)
- 558       **[XML Encrypt]**       W3C Recommendation, "XML Encryption Syntax and Processing," 10  
559                            December 2002
- 560       **[XML Signature]**    D. Eastlake, J. R., D. Solo, M. Bartel, J. Boyer , B. Fox , E. Simon. *XML-*  
561                            *Signature Syntax and Processing*, W3C Recommendation, 12 February  
562                            2002.

563

564



## 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

568

**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

569

570

---

## Appendix B: Revision History

571

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