



# Web Services Security X509 Certificate Token Profile

Working Draft 04, 19th May 2003

**Document identifier:**

WSS-X509-04

**Location:**

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

**Editors:**

Phillip Hallam-Baker, VeriSign

Chris Kaler, Microsoft

Ronald Monzillo, Sun

Anthony Nadalin, IBM

**Contributors:**

TBD – Revise this list to include WSS TC contributors

Bob Atkinson, Microsoft

Giovanni Della-Libera, Microsoft

Satoshi Hada, IBM

Phillip Hallam-Baker, VeriSign

Maryann Hondo, IBM

Chris Kaler, Microsoft

Johannes Klein, Microsoft

Brian LaMacchia, Microsoft

Paul Leach, Microsoft

John Manferdelli, Microsoft

Hiroshi Maruyama, IBM

Anthony Nadalin, IBM

Nataraj Nagaratnam, IBM

Hemma Prafullchandra, VeriSign

John Shewchuk, Microsoft

Dan Simon, Microsoft

Kent Tamura, IBM

Hervey Wilson, Microsoft

**Abstract:**

This document describes how to use X509 Certificates with the [WS-Security](#) specification.

**Status:**

This is an interim draft. Please send comments to the editors.

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

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 Security Services TC web page (<http://www.oasis-open.org/who/intellectualproperty.shtml>).

---

30 **Table of Contents**

31 1 Introduction ..... 3  
32 2 Notations and Terminology..... 4  
33 2.1 Notational Conventions ..... 4  
34 2.2 Namespaces ..... 4  
35 2.3 Terminology ..... 4  
36 3 Usage ..... 5  
37 3.1 Processing Model ..... 5  
38 3.2 Attaching Security Tokens ..... 5  
39 3.3 Identifying Certificates ..... 6  
40 3.3.1 Identifying End Entity Certificates by Value ..... 6  
41 3.3.2 Identifying and Referencing Certificate Chains ..... 6  
42 3.3.3 Identifying End Entity Certificates by Reference ..... 7  
43 3.4 Authentication..... 7  
44 3.5 Encryption ..... 8  
45 3.6 Error Codes ..... 8  
46 3.7 Threat Model and Countermeasures ..... 8  
47 4 Acknowledgements ..... 9  
48 5 References .....10  
49 Appendix A: Revision History.....11  
50 Appendix B: Notices .....12  
51

---

52 **1 Introduction**

53 This specification describes the use of X509 certificates with respect to the [WS-Security](#)  
54 specification.

55 An X.509 Certificate specifies a binding between a public key and a set of attributes that include a  
56 subject name, issuer name, serial number and validity interval. This binding may be subject to  
57 subsequent revocation advertised by mechanisms that include issue of CRLs, OCSP tokens or  
58 mechanisms that are outside the X.509 framework such as XKMS.

59 An X.509 Certificate may be used to establish the authenticity of a public key used to authenticate  
60 a WS-Security enhanced message or to identify the public key under which a WS-Security  
61 enhanced message is encrypted.

62 Note that Section 1 is non-normative.

---

## 63 2 Notations and Terminology

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

### 65 2.1 Notational Conventions

66 The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD",  
67 "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be  
68 interpreted as described in RFC2119.

69 Namespace URIs (of the general form "some-URI") represent some application-dependent or  
70 context-dependent URI as defined in [RFC2396](#).

71 This specification is designed to work with the general [SOAP](#) message structure and message  
72 processing model, and should be applicable to any version of [SOAP](#). The current SOAP 1.2  
73 namespace URI is used herein to provide detailed examples, but there is no intention to limit the  
74 applicability of this specification to a single version of [SOAP](#).

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

### 76 2.2 Namespaces

77 The [XML namespace](#) URIs that MUST be used by implementations of this specification are as  
78 follows (note that different elements in this specification are from different namespaces):

79 `http://schemas.xmlsoap.org/ws/2002/xx/secext`  
80 `http://schemas.xmlsoap.org/ws/2002/xx/utility`

81 The following namespaces are used in this document:

Prefix	Namespace
S	<a href="http://www.w3.org/2001/12/soap-envelope">http://www.w3.org/2001/12/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://schemas.xmlsoap.org/ws/2002/xx/secext">http://schemas.xmlsoap.org/ws/2002/xx/secext</a>
wsu	<a href="http://schemas.xmlsoap.org/ws/2002/xx/utility">http://schemas.xmlsoap.org/ws/2002/xx/utility</a>

### 82 2.3 Terminology

83 This specification employs the terminology defined in the WS-Security Core Specification.

---

## 84 3 Usage

85 This section describes the profile (specific mechanisms and procedures) for the X509  
86 binding of [WS-Security](#).

87 **Identification:** urn:oasis:names:tc:WSS:1.0:profiles:WSS-X509-token

88 **Contact information:** TBD

89 **Description:** Given below.

90 **Updates:** None.

### 91 3.1 Processing Model

92 The processing model for [WS-Security](#) with X509 certificates is no different from that  
93 of [WS-Security](#) with other token formats as described in [WS-Security](#) .

### 94 3.2 Attaching Security Tokens

95 X.509 Certificates that are attached as security tokens within a [WS-Security](#)  
96 enhanced message SHOULD be attached by means of the  
97 `<wsse:BinarySecurityToken>` element.

98 The [WS-Security](#) specification indicates that X.509 certificates MAY be described  
99 inside of a `<ds:KeyInfo>` element, however, it is RECOMMENDED that they be  
100 specified using a `<wsse:BinarySecurityToken>`. If, however, an implementation  
101 needs to use `<ds:KeyInfo>`, it SHOULD place the `<ds:KeyInfo>` element as a child  
102 of the `<wsse:Security>` header rather than embedded within the signature. This  
103 allows receivers to have a single processing model.

104 The following values are defined for the ValueType attribute of the  
105 `<wsse:BinarySecurityToken>` element.

QName	Description
wsse:X509v3	X.509 v3 end entity certificate
wsse:PKCS7	An X.509 certificate chain packaged in a PKCS#7 wrapper

106 The following example illustrates a SOAP message with an X509 Certificate.

```
107 <S:Envelope xmlns:S="...">
108   <S:Header>
109     <wsse:Security xmlns:wsse="...">
110
111       <wsse:BinarySecurityToken
112         xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/04/secext "
113         Id="myToken"
114         ValueType="wsse:X509v3"
115         EncodingType="wsse:Base64Binary">
116         MII EZzCCA9CgAwIBAgIQEmtJZc0...
117       </wsse:BinarySecurityToken>
118
119       ...
120     </wsse:Security>
121   </S:Header>
```

122  
123  
124  
125

```
<S:Body>
  ...
</S:Body>
</S:Envelope>
```

## 126 3.3 Identifying Certificates

### 127 3.3.1 Identifying End Entity Certificates by Value

128 An attached X.509 certificate that identifies an end entity is attached by means of  
129 the `wsse:BinarySecurityToken` element and referenced by means of a  
130 `wsse:SecurityTokenReference` element that contains a `wsse:KeyIdentifier` element.  
131 The `wsu:Id` attribute of the `wsse:KeyIdentifier` element references the value of the  
132 `wsu:Id` attribute specified in the `wsse:BinarySecurityToken`.

133 The following example shows a SOAP message that contains an X509v3 Certificate as  
134 a binary token:

135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154

```
Example TBS
<S:Envelope xmlns:S="...">
  <S:Header>
    <wsse:Security xmlns:wsse="...">
      <wsse:BinarySecurityToken
        xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/04/secext"
        Id="myToken"
        ValueType="wsse:X509v3"
        EncodingType="wsse:Base64Binary">
        MIEZzCCA9CgAwIBAgIQEmtJZc0...
      </wsse:BinarySecurityToken>
    </wsse:Security>
  </S:Header>
  <S:Body>
    ...
  </S:Body>
</S:Envelope>
```

### 155 3.3.2 Identifying and Referencing Certificate Chains

156 An attached X.509 certificate that identifies an end entity is attached by means of  
157 the `wsse:BinarySecurityToken` element and referenced by means of a  
158 `wsse:SecurityTokenReference` element that contains a `wsse:KeyIdentifier` element.  
159 The `wsu:Id` attribute of the `wsse:KeyIdentifier` element references the value of the  
160 `wsu:Id` attribute specified in the `wsse:BinarySecurityToken`.

161 The `wsse:BinarySecurityToken` element contains a PKCS#7 SignedData object, with  
162 the only significant field being certificates. In particular, the signature and the  
163 contents are ignored. If no certificates are present, a zero-length CertPath is  
164 assumed. Warning: PKCS#7 does not maintain the order of certificates in a  
165 certification path. This means that if a CertPath is converted to PKCS#7 encoded  
166 bytes and then converted back, the order of the certificates may change, potentially  
167 rendering the CertPath invalid. Users should be aware of this behavior. See [PKCS7]  
168 for more information.

169 The following example shows a SOAP message that contains an X509v3 Certificate  
170 chain encoded inside a PKCS#7 package:

```
171 Example TBS
172 <S:Envelope xmlns:S="...">
173   <S:Header>
174     <wsse:Security xmlns:wsse="...">
175
176       <wsse:BinarySecurityToken
177         xmlns:wsse="http://schemas.xmlsoap.org/ws/2002/04/secext "
178         Id="myToken"
179         ValueType="wsse:PKCS7"
180         EncodingType="wsse:Base64Binary">
181         MIIIEZzCCA9CgAwIBAgIQEmtJZc0...
182       </wsse:BinarySecurityToken>
183
184       ...
185     </wsse:Security>
186   </S:Header>
187   <S:Body>
188     ...
189   </S:Body>
190 </S:Envelope>
```

### 191 3.3.3 Identifying End Entity Certificates by Reference

192 An X.509 certificate that identifies an end entity that is not attached to the message  
193 payload is referenced by means of the Issuer Subject Name and Serial Number using  
194 the XML Signature <X509IssuerSerial> element.

195 The following example shows a SOAP message that identifies an X.509v3 end entity  
196 certificate by reference to the Issuer and serial number:

```
197 Example TBS
198 <S:Envelope xmlns:S="...">
199   <S:Header>
200     <wsse:Security xmlns:wsse="...">
201
202       <ds:KeyInfo>
203         <X509Data>
204           <X509IssuerSerial>
205             <X509IssuerName>CN=TAMURA Kent, OU=TRL, O=IBM,
206             L=Yamato-shi, ST=Kanagawa, C=JP</X509IssuerName>
207             <X509SerialNumber>12345678</X509SerialNumber>
208           </X509IssuerSerial>
209           <X509SKI>31d97bd7</X509SKI>
210         </X509Data>
211       </ds:KeyInfo>
212
213       ...
214     </wsse:Security>
215   </S:Header>
216   <S:Body>
217     ...
218   </S:Body>
219 </S:Envelope>
```

## 220 3.4 Authentication

221 When an X.509 certificate is used to specify a signature key, the [signature](#) algorithm  
222 MUST be a digital signature algorithm.

223 The value of the signature key is the value of the public key specified in the  
224 certificate.

### 225 **3.5 Encryption**

226 When an X.509 certificate is used to specify an encryption key, the encryption  
227 algorithm MUST be a public key encryption algorithm.

228 The certificate that specifies the encryption key SHOULD be identified by reference  
229 since the receiver only requires the use of the certificate to identify the  
230 corresponding encryption key and does not require the certificate or the certificate  
231 chain to authenticate the public key it holds.

232 The value of the encryption key is the value of the public key specified in the  
233 certificate.

### 234 **3.6 Error Codes**

235 When using X509 Certificates the error codes defined in the [WS-Security](#)  
236 specification MUST be used.

237 If an implementation requires the use of a custom error it is recommended that a  
238 sub-code be defined as an extension of one of the codes defined in the [WS-Security](#)  
239 specification.

### 240 **3.7 Threat Model and Countermeasures**

241 The use of X509 certificates with [WS-Security](#) introduces no new threats beyond  
242 those identified for WS-Security with other types of security tokens.

243 Message alteration and eavesdropping can be addressed by using the integrity and  
244 confidentiality mechanisms described in WS-Security. Replay attacks can be  
245 addressed by using message timestamps and caching, as well as other application-  
246 specific tracking mechanisms. For X.509 certificates ownership is verified by use of  
247 keys, man-in-the-middle attacks are generally mitigated.

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

250 It should be noted that transport-level security MAY be used to protect the message  
251 and the security token.

---

252 **4 Acknowledgements**

253 This specification was developed as a result of joint work of many individuals from  
254 the WSS TC including: TBD

255 The input specifications for this document were developed as a result of joint work  
256 with many individuals and teams, including: Keith Ballinger, Microsoft, Bob Blakley,  
257 IBM, Allen Brown, Microsoft, Joel Farrell, IBM, Mark Hayes, VeriSign, Kelvin  
258 Lawrence, IBM, Scott Konersmann, Microsoft, David Melgar, IBM, Dan Simon,  
259 Microsoft, Wayne Vicknair, IBM.

---

## 5 References

- 260
- 261     **[DIGSIG]**            Informational RFC 2828, "[Internet Security Glossary](#)," May 2000.
- 262     **[KEYWORDS]**        S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels,"  
263     [RFC 2119](#), Harvard University, March 1997
- 264     **[SOAP]**             W3C Note, "[SOAP: Simple Object Access Protocol 1.1](#)," 08 May 2000.
- 265     **[URI]**              T. Berners-Lee, R. Fielding, L. Masinter, "Uniform Resource Identifiers  
266     (URI): Generic Syntax," [RFC 2396](#), MIT/LCS, U.C. Irvine, Xerox  
267     Corporation, August 1998.
- 268     **[WS-Security]**     TBS – point to the OASIS draft
- 269     **[XML-ns]**            W3C Recommendation, "[Namespaces in XML](#)," 14 January 1999.
- 270     **[XML Signature]**   W3C Recommendation, "[XML Signature Syntax and Processing](#)," 12  
271     February 2002.
- 272     **[PKCS7]**            **TBS** <http://www.rsasecurity.com/rsalabs/pkcs/pkcs-7/index.html>
- 273     **[X509]**             **TBS**
- 274     **[PKIPATH]**         **TBS**  
275     [ftp://ftp.bull.com/pub/OSIdirectory/DefectResolution/TechnicalCorrigenda](ftp://ftp.bull.com/pub/OSIdirectory/DefectResolution/TechnicalCorrigenda/ApprovedTechnicalCorrigendaToX.509/8%7CX.509-TC1(4th).pdf)  
276     [/ApprovedTechnicalCorrigendaToX.509/8%7CX.509-TC1\(4th\).pdf](ftp://ftp.bull.com/pub/OSIdirectory/DefectResolution/TechnicalCorrigenda/ApprovedTechnicalCorrigendaToX.509/8%7CX.509-TC1(4th).pdf).  
277

---

## Appendix A: Revision History

Rev	Date	What
01	18-Sep-02	Initial draft based on input documents and editorial review
03	30-Jan-03	Changes in title
04	19-May-03	Added by reference and pkipath modes of cert identification. Added section 1 introduction, changes to formatting etc.

---

## Appendix B: Notices

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

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

293 Copyright © OASIS Open 2002. *All Rights Reserved.*

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

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

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

310