



Web Services Security Kerberos Token Profile 1.1

Committee Draft – Tuesday 14 June 2005

OASIS identifier:

{WSS: SOAP Message Security }-{Kerberos Token Profile }-{1.1} (Word) (PDF)

Location:

<http://docs.oasis-open.org/wss/2005/xx/oasis-2005xx-wss-kerberos-token-profile-1.1>

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 Kerberos [Kerb] tickets (specifically the AP-REQ packet) with the WSS: SOAP Message Security [WSS] 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 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>.

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>).

Notices

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

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

45 Copyright © The Organization for the Advancement of Structured Information Standards [OASIS]
46 2002-2005. All Rights Reserved.

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

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

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

63	Table of Contents	
64	1 Introduction	4
65	2 Notations and Terminology	5
66	2.1 Notational Conventions	5
67	2.2 Namespaces	5
68	2.3 Terminology	6
69	3 Usage	7
70	3.1 Processing Model	7
71	3.2 Attaching Security Tokens	7
72	3.3 Identifying and Referencing Kerberos Tokens	8
73	3.4 Authentication	9
74	3.5 Encryption	10
75	3.6 Principal Name	10
76	3.7 Error Codes	10
77	4 Threat Model and Countermeasures	11
78	5 References	12
79	Appendix A. Acknowledgments	13
80	Appendix B. Revision History	16
81		

82 1 Introduction

83 This specification describes the use of Kerberos [Kerb] tokens with respect to the WSS: SOAP
84 Message Security specification [WSS].

85 Specifically, this document defines how to encode Kerberos tickets and attach them to SOAP
86 messages. As well, it specifies how to add signatures and encryption to the SOAP message, in
87 accordance with WSS: SOAP Message Security, which uses and references the Kerberos
88 tokens.

89 For interoperability concerns, and for some security concerns, the specification is limited to using
90 the AP-REQ packet (service ticket and authenticator) defined by Kerberos as the Kerberos token.
91 This allows a service to authenticate the ticket and interoperate with existing Kerberos
92 implementations.

93 It should be noted that how the AP-REQ is obtained is out of scope of this specification as are
94 scenarios involving other ticket types and user-to-user interactions.

95 Note that Sections 2.1, 2.2, all of 3, and indicated parts of 6 are normative. All other sections are
96 non-normative.

97 2 Notations and Terminology

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

99 2.1 Notational Conventions

100 The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD",
101 "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be
102 interpreted as described in RFC2119 [2119].

103

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

106

107 This specification is designed to work with the general SOAP [S11, S12] message structure and
108 message processing model, and should be applicable to any version of SOAP. The current SOAP
109 1.2 namespace URI is used herein to provide detailed examples, but there is no intention to limit
110 the applicability of this specification to a single version of SOAP.

111 2.2 Namespaces

112 The XML namespace [XML-ns] URIs that MUST be used by implementations of this specification
113 are as follows (note that different elements in this specification are from different namespaces):

114

```
115 http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-  
116 secext-1.0.xsd  
117 http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-  
118 utility-1.0.xsd  
119 http://docs.oasis-open.org/wss/2005/xx/oasis-2005xx-wss-wssecurity-  
120 secext-1.1.xsd
```

121

122 Note that this specification does not introduce new schema elements.

123 The following namespaces are used in this document:

Prefix	Namespace
S11	http://schemas.xmlsoap.org/soap/envelope/
S12	http://www.w3.org/2003/05/soap-envelope
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-seceext-1.1.xsd
wsu	http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd
ds	http://www.w3.org/2000/09/xmldsig#
xenc	http://www.w3.org/2001/04/xmlenc#

124

125 The URLs provided for the *wsse* and *wsu* namespaces can be used to obtain the schema files.
 126 URI fragments defined in this specification are relative to the following base URI unless otherwise
 127 specified:

128 <http://docs.oasis-open.org/wss/2005/xx/oasis-2005xx-wss-kerberos-token-profile-1.1>

129 2.3 Terminology

130 Readers are presumed to be familiar with the terms in the Internet Security Glossary [ISG].

131

132 This specification employs the terminology defined in the WSS: SOAP Message Security Core
 133 Specification [WSS].

134

135 The following (non-normative) table defines additional acronyms and abbreviations for this
 136 document.

Term	Definition
SHA	Secure Hash Algorithm
SOAP	Simple Object Access Protocol
URI	Uniform Resource Identifier
XML	Extensible Markup Language

137

138 3 Usage

139 This section describes the profile (specific mechanisms and procedures) for the
140 Kerberos binding of WSS: SOAP Message Security.

141 **Identification:** <http://docs.oasis-open.org/wss/2005/xx/oasis-2005xx-wss-kerberos-token-profile-1.1>
142

143 3.1 Processing Model

144 The processing model for WSS: SOAP Message Security with Kerberos tokens is no
145 different from that of WSS: SOAP Message Security with other token formats as
146 described in WSS: SOAP Message Security.

147 3.2 Attaching Security Tokens

148 Kerberos tokens are attached to SOAP messages using WSS: SOAP Message Security by using
149 the `<wsse:BinarySecurityToken>` described in WSS: SOAP Message Security. When using
150 this element, the `@ValueType` attribute **MUST** be specified. This specification defines two values
151 for this token as defined in the table below:

URI	Description
http://docs.oasis-open.org/wss/2005/xx/oasis-2005xx-wss-kerberos-token-profile-1.1#Kerberosv5_AP_REQ	Kerberos v5 AP-REQ as defined in the Kerberos specification. This ValueType is used when the ticket is an AP Request.
http://docs.oasis-open.org/wss/2005/xx/oasis-2005xx-wss-kerberos-token-profile-1.1#GSS_Kerberosv5_AP_REQ	A GSS wrapped Kerberos v5 AP-REQ as defined in the GSSAPI specification. This ValueType is used when the ticket is an AP Request (ST + Authenticator).

152 It should be noted that the URIs in the table above also serves as the official URIs
153 identifying the Kerberos token defined in this specification.

154

155 Both token types defined in this section use the type 0x8003 defined in RFC1964 for the
156 checksum field of the authenticator inside the AP_REQ.

157

158 The octet sequence of the either the GSS wrapped Kerberos ticket or the Kerberos
159 ticket (e.g. AP-REQ) is encoded using the indicated algorithm (e.g. base 64) and the
160 result is placed inside of the `<wsse:BinarySecurityToken>` element.

161 The following example illustrates a SOAP message with a Kerberos token.

162 `<S11:Envelope xmlns:S11="...">`

163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178

```
<S11:Header>
  <wsse:Security xmlns:wsse="...">
    <wsse:BinarySecurityToken
      xmlns:wsse="... "
      wsu:Id="myToken"
      ValueType="...#Kerberosv5_AP_REQ"
      EncodingType="...#Base64Binary">
      MIEZzCCA9CgAwIBAgIQEmtJZc0...
    </wsse:BinarySecurityToken>
    ...
  </wsse:Security>
</S11:Header>
<S11:Body>
  ...
</S11:Body>
</S11:Envelope>
```

179

3.3 Identifying and Referencing Kerberos Tokens

180

181 A Kerberos Token is referenced by means of the `<wsse:SecurityTokenReference>`
182 element. This mechanism, defined in WSS: SOAP Message Security, provides different
183 referencing mechanisms. The following list identifies the supported and unsupported
184 mechanisms:

185 The `wsu:Id` MAY be specified on the `<wsse:BinarySecurityToken>` element allowing the
186 token to be directly referenced.

187 A `<wsse:KeyIdentifier>` element MAY be used which specifies the identifier for the
188 Kerberos ticket. This value is computed as the SHA1 of the pre-encoded octets that were used to
189 form the contents of the `<wsse:BinarySecurityToken>` element. The
190 `<wsse:KeyIdentifier>` element contains the encoded form of the KeyIdentifier which is
191 defined as the base64 encoding of the SHA1 result.

192 Key Name references MUST NOT be used.

193 When a Kerberos Token is referenced using `<wsse:SecurityTokenReference>` the
194 `@ValueType` attribute is not required. If specified, the URI listed above as Kerberos token type
195 MUST be specified.

196 The `<wsse:SecurityTokenReference>` element from which the reference is made contains
197 the `<wsse:KeyIdentifier>` element. The `<wsse:KeyIdentifier>` element MUST have a
198 `ValueType` attribute with the value `#Kerberosv5APREQSHA1` and its contents MUST be the
199 SHA1 of GSS wrapped or unwrapped AP-REQ, encoded as per the `<wsse:KeyIdentifier>`
200 element's `EncodingType` attribute.

201

Reference Identifier	ValueType URI	Description
Kerberos v5 AP-REQ	#Kerberosv5APREQSHA1	SHA1 of the v5 AP-REQ octets, either GSS wrapped Kerberos AP-REQ or just the Kerberos AP-REQ.

202

203 The following example illustrates using ID references to a Kerberos token:

204

```
205 <S11:Envelope xmlns:S11="...">
206   <S11:Header>
207     <wsse:Security xmlns:wsse="...">
208       <wsse:BinarySecurityToken
209         xmlns:wsse="..."
210         wsu:Id="myToken"
211         ValueType="...#Kerberosv5_AP_REQ"
212         EncodingType="...#Base64Binary">
213         MIEZzCCA9CgAwIBAgIQEmtJZc0...
214       </wsse:BinarySecurityToken>
215       ...
216       <wsse:SecurityTokenReference>
217         <wsse:Reference URI="#myToken" />
218       </wsse:SecurityTokenReference>
219       ...
220     </wsse:Security>
221   </S11:Header>
222   <S11:Body>
223     ...
224   </S11:Body>
225 </S11:Envelope>
226
```

227

228 The AP-REQ packet is included in the initial message to the service, but need not be attached to
229 subsequent messages exchanged between the involved parties. Consequently, the KeyIdentifier
230 reference mechanism SHOULD be used on subsequent exchanges as illustrated in the example
231 below:

232

```
233 <S11:Envelope xmlns:S11="...">
234   <S11:Header>
235     <wsse:Security xmlns:wsse="...">
236       ...
237       <wsse:SecurityTokenReference
238 <wsse:KeyIdentifier    ValueType="...#Kerberosv5APREQSHA1">
239         EZzCCA9CgAwIB...
240         <wsse:KeyIdentifier>
241         </wsse:KeyIdentifier>
242       </wsse:SecurityTokenReference>
243       ...
244     </wsse:Security>
245   </S11:Header>
246   <S11:Body>
247     ...
248   </S11:Body>
249 </S11:Envelope>
```

250 3.4 Authentication

251 When a Kerberos ticket is referenced as a signature key, the signature algorithm [DSIG] MUST
252 be a hashed message authentication code.

253

254 When a Kerberos ticket is referenced as an encryption key, the encryption algorithm MUST be a
255 symmetric encryption algorithm.

256

257 The value of the signature or encryption key is constructed from the value of the Kerberos sub-
258 key when it is present in the authenticator or a session key from the ticket if the sub-key is
259 absent, either by using the Kerberos sub-key or session key directly or using a key derived from
260 that key using a mechanism agreed to by the communicating parties.

261 **3.5 Encryption**

262 When a Kerberos ticket is referenced as an encryption key, the encryption algorithm MUST be a
263 symmetric encryption algorithm.

264

265 The value of the signature or encryption key is constructed from the value of the Kerberos sub-
266 key when it is present in the authenticator or a session key from the ticket if the sub-key is
267 absent, either by using the Kerberos sub-key or session key directly or using a key derived from
268 that key using a mechanism agreed to by the communicating parties..

269 **3.6 Principal Name**

270 Kerberos principal name definition and mapping of non-Kerberos names to Kerberos V principal
271 names are out of scope of this document.

272 **3.7 Error Codes**

273 When using Kerberos tokens, it is RECOMMENDED to use the error codes defined in the WSS:
274 SOAP Message Security specification. However, implementations MAY use custom errors,
275 defined in private namespaces if they desire. Care should be taken not to introduce security
276 vulnerabilities in the errors returned.

277

4 Threat Model and Countermeasures

278 The use of Kerberos assertion tokens with WSS: SOAP Message Security introduces no new
279 message-level threats beyond those identified for Kerberos itself or by WSS: SOAP Message
280 Security with other types of security tokens.

281

282 One potential threat is that of key re-use. The mechanisms described in WSS: SOAP Message
283 Security can be used to prevent replay of the message; however, it is possible that for some
284 service scopes, there are host security concerns of key hijacking within a Kerberos infrastructure.
285 The use of the AP-REQ and its associated authenticator and sequencer mitigate this threat.

286

287 Message alteration and eavesdropping can be addressed by using the integrity and confidentiality
288 mechanisms described in WSS: SOAP Message Security. Replay attacks can be addressed by
289 using message timestamps and caching, as well as other application-specific tracking
290 mechanisms. For Kerberos tokens ownership is verified by use of keys, so man-in-the-middle
291 attacks are generally mitigated.

292

293 It is strongly recommended that GSS wrapped AP-REQ used or that unwrapped AP-REQ be
294 combined with timestamp be used to prevent replay attack.

295

296 It is strongly recommended that all relevant and immutable message data be signed to prevent
297 replay attacks.

298

299 It should be noted that transport-level security MAY be used to protect the message and the
300 security token if either a wrapped AP-REQ or that unwrapped AP-REQ be combined with
301 timestamp and signature are not being used.

5 References

302

303 The following are normative references

304 **[2119]** S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels,"
305 [RFC 2119](#), Harvard University, March 1997

306 **[Kerb]** J. Kohl and C. Neuman, "The Kerberos Network Authentication Service
307 (V5)," [RFC 1510](#), September 1993, <http://www.ietf.org/rfc/rfc1510.txt> .

308 **[KEYWORDS]** S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels,"
309 [RFC 2119](#), Harvard University, March 1997

310 **[S11]** W3C Note, "[SOAP: Simple Object Access Protocol 1.1](#)," 08 May 2000.

311 **[S12]** W3C Recommendation, "SOAP Version 1.2 Part 1: Messaging
312 Framework", 23 June 2003.

313 **[URI]** T. Berners-Lee, R. Fielding, L. Masinter, "Uniform Resource Identifiers
314 (URI): Generic Syntax," RFC 3986, MIT/LCS, Day Software, Adobe
315 Systems, January 2005.

316 The following are non-normative references

317 **[ISG]** Informational RFC 2828, "[Internet Security Glossary](#)," May 2000.

318 **[WSS]** A. Nadalin et al., Web Services Security: SOAP Message Security 1.0
319 (WS-Security 2004), OASIS Standard 200401, March 2004,
320 [http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-](http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf)
321 [message-security-1.0.pdf](http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf).

322 **[XML-ns]** W3C Recommendation, "[Namespaces in XML](#)," 14 January 1999.

323 **[DSIG]** D. Eastlake, J. R., D. Solo, M. Bartel, J. Boyer , B. Fox , E. Simon. *XML-*
324 *Signature Syntax and Processing*, W3C Recommendation, 12 February
325 2002. <http://www.w3.org/TR/xmlsig-core/>.

326

Appendix A. Acknowledgments

327

This specification was developed as a result of joint work of many individuals from the WSS TC.

328

The input specifications for this document were developed as a result of joint work with many

329

individuals and teams, including: Keith Ballinger, Microsoft, Bob Blakley, IBM, Allen Brown,

330

Microsoft, Joel Farrell, IBM, Mark Hayes, VeriSign, Kelvin Lawrence, IBM, Scott Konersmann,

331

Microsoft, David Melgar, IBM, Dan Simon, Microsoft, Wayne Vicknair, IBM.

332

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)
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
Yutaka	Kudo	Hitachi
Jason	Rouault	HP
Bob	Blakley	IBM
Joel	Farrell	IBM
Satoshi	Hada	IBM
Maryann	Hondo	IBM
Hiroshi	Maruyama	IBM

David	Melgar	IBM
Anthony	Nadalin	IBM
Nataraj	Nagaratnam	IBM
Wayne	Vicknair	IBM
Kelvin	Lawrence	IBM (co-Chair)
Don	Flinn	Individual
Bob	Morgan	Individual
Bob	Atkinson	Microsoft
Keith	Ballinger	Microsoft
Allen	Brown	Microsoft
Paul	Cotton	Microsoft
Giovanni	Della-Libera	Microsoft
Vijay	Gajjala	Microsoft
Johannes	Klein	Microsoft
Scott	Konermann	Microsoft
Chris	Kurt	Microsoft
Brian	LaMacchia	Microsoft
Paul	Leach	Microsoft
John	Manferdell	Microsoft
John	Shewchuk	Microsoft
Dan	Simon	Microsoft
Hervey	Wilson	Microsoft
Chris	Kaler	Microsoft (co-Chair)
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
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
Mark	Hays	Verisign
Hemma	Prafullchandra	VeriSign

333

Appendix B. 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	20-Jan-04	Revise based on comments, switch to new URLs and formats and recent decisions in TC
05	27-Jul-04	Revise based on comments and recent decisions in TC
06	16-May-05	Revise based on comments and recent decisions in TC. Issues 381, 382, 383, 384, 385, 386, 387
07	17-May-05	Formatting Issues
08	14-June-05	Issues 396