



Web Services Security SAML Token Binding

Working Draft 02, 23 September 2002

Document identifier:

WSS-SAML-01

Location:

TBD

Editors:

Phillip Hallam-Baker, VeriSign

Chris Kaler, Microsoft

Ronald Monzillo, Sun

Anthony Nadalin, IBM

Contributors:

TBD – Revise this list to include WSS TC contributors

Phillip Hallam-Baker, VeriSign

Jeff Hodges, Sun Microsystems

Maryann Hondo, IBM

Chris Kaler, Microsoft

Eve Maler, Sun Microsystems

Hiroshi Maruyama, IBM

Chris McLaren, Netegrity

Prateek Mishra, Netegrity

Anthony Nadalin, IBM

Nataraj Nagarathnam, IBM

Hemma Prafullchandra, VeriSign

Irving Reid, Baltimore

Krishna Sankar, Cisco

John Shewchuk, Microsoft

Abstract:

This document describes how to use Security Assertion Markup Language (SAML) assertions 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 <mailto: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

28 terms, please refer to the Intellectual Property Rights section of the Security
29 Services TC web page ([http://www.oasis-](http://www.oasis-open.org/who/intellectualproperty.shtml)
30 [open.org/who/intellectualproperty.shtml](http://www.oasis-open.org/who/intellectualproperty.shtml)).
31

31 **Table of Contents**

32 1 Introduction 5
33 1.1 Goals and Requirements 5
34 1.1.1 Requirements 5
35 1.1.2 Non-Goals 5
36 2 Notations and Terminology 6
37 2.1 Notational Conventions 6
38 2.2 Namespaces 6
39 2.3 Terminology 7
40 3 Usage 8
41 3.1 Processing Model 8
42 3.2 Attaching Security Tokens 8
43 3.3 Identifying and Referencing Security Tokens 9
44 3.4 Proof-of-Possession of Security Tokens 9
45 3.5 Error Codes 10
46 3.6 Threat Model and Countermeasures 14
47 4 Acknowledgements 17
48 5 References 18
49 Appendix A: Revision History 20
50 Appendix B: Notices 21
51

52 1 Introduction

53 The [WS-Security](#) specification proposes a standard set of [SOAP](#) extensions that can
54 be used when building secure Web services to implement message level integrity and
55 confidentiality. This specification describes the use of Security Assertion Markup
56 Language (SAML) assertions with respect to the [WS-Security](#) specification.

57 1.1 Goals and Requirements

58 The goal of this specification is to define the use of SAML assertions in the context of
59 [WS-Security](#) including for the purpose of securing [SOAP](#) message exchanges.

60 The requirements to be satisfied by this specification are listed below.

61 1.1.1 Requirements

62 TBS

63

64 1.1.2 Non-Goals

65 The following topics are outside the scope of this document:

66 TBS

67

68 2 Notations and Terminology

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

71 2.1 Notational Conventions

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

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

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

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

84 2.2 Namespaces

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

88 <http://schemas.xmlsoap.org/ws/2002/xx/secext>
89 <http://schemas.xmlsoap.org/ws/2002/xx/utility>

90 The following namespaces are used in this document:

91

Prefix	Namespace
S	http://www.w3.org/2001/12/soap-envelope
ds	http://www.w3.org/2000/09/xmldsig#
xenc	http://www.w3.org/2001/04/xmlenc#
wsse	http://schemas.xmlsoap.org/ws/2002/xx/secext
wsu	http://schemas.xmlsoap.org/ws/2002/xx/utility
saml	urn: oasis:names:tc:SAML:1.0:assertion

samlp	urn: oasis:names:tc:SAML:1.0:protocol
-------	---------------------------------------

92 **2.3 Terminology**

93 This specification employs the terminology defined in the [WS-Security](#) Core
94 Specification.

95 Defined below are the basic definitions for additional terminology used in this
96 specification.

97 [TBS]

98 3 Usage

99 This section describes the specific mechanisms and procedures for the SAML binding
100 of [WS-Security](#).

101 **Identification:** urn:oasis:names:tc:WSS:1.0:bindings:WSS-SAML-binding

102 **Contact information:** TBD

103 **Description:** Given below.

104 **Updates:** None.

105 3.1 Processing Model

106 The SAML binding of [WS-Security](#) extends the token-independent processing model
107 defined by the core [WS-Security](#) specification.

108 When a receiver processes a <wsse:Security> header containing SAML assertions, it
109 MUST make an explicit determination of the relationship between the subject of each
110 assertion and the sender of the message. Two methods for establishing this
111 correspondence, *holder-of-key* and *sender-vouches* are described below. Senders
112 and receivers implementing the SAML binding of [WS-Security](#) MUST implement the
113 processing necessary to support both of these subject confirmation methods.

114 3.2 Attaching Security Tokens

115 SAML assertions are attached to SOAP messages using [WS-Security](#) by placing
116 assertion elements inside a <wsse:Security> header. The following example
117 illustrates a SOAP message containing a SAML assertion in a <wsse:Security>
118 header.

```
119 <S:Envelope xmlns:S="...">  
120   <S:Header>  
121     <wsse:Security xmlns:wsse="...">  
122       <saml:Assertion  
123         MajorVersion="1"  
124         MinorVersion="0"  
125         AssertionID="SecurityToken-ef375268"  
126         Issuer="elliottw1"  
127         IssueInstant="2002-07-23T11:32:05.6228146-07:00"  
128         xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion">  
129         ...  
130       </saml:Assertion>  
131       ...  
132     </wsse:Security>  
133   </S:Header>  
134   <S:Body>  
135     ...  
136   </S:Body>  
137 </S:Envelope>  
138
```


139 3.3 Identifying and Referencing Security Tokens

140 The [WS-Security](#) specification defines the `wsu:id` attribute as the common
141 mechanism for referencing security tokens by "Id". Because the
142 `<saml:AssertionIDReference>` element does not provide for attribute
143 extensibility, this binding encapsulates `<saml:AssertionIDReference>` elements in
144 the `<wsse:SecurityTokenReference>` element such that the `wsu:id` attribute of the
145 encapsulating element can be used to identify assertions according to the common
146 [WS-Security](#) mechanism. When this element is encountered within a reference, the
147 recipient, if it supports the SAML binding of [WS-Security](#), MUST interpret the
148 contained element as a `<saml:AssertionIDReference>`.

149 The following example illustrates a message with an [XML Signature](#) that references a
150 SAML assertion token.

```
151 <S:Envelope xmlns:S="...">
152   <S:Header>
153     <wsse:Security xmlns:wsse="...">
154       <saml:Assertion
155         MajorVersion="1"
156         MinorVersion="0"
157         AssertionID="SecurityToken-ef375268"
158         Issuer="elliottw1"
159         IssueInstant="2002-07-23T11:32:05.6228146-07:00"
160         xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion">
161         ...
162       </saml:Assertion>
163       <ds:Signature xmlns:ds="...">
164         ...
165         <ds:KeyInfo>
166           <wsse:SecurityTokenReference>
167             <saml:AssertionIDReference>
168               SecurityToken-ef375268
169             </saml:AssertionIDReference>
170           </wsse:SecurityTokenReference>
171         </ds:KeyInfo>
172       </ds:Signature>
173       ...
174     </wsse:Security>
175   </S:Header>
176   <S:Body>
177     ...
178   </S:Body>
179 </S:Envelope>
180
```

181 3.4 Proof-of-Possession of Security Tokens

182 As previously stated, the SAML binding of [WS-Security](#) requires that message
183 senders and receivers support the holder-of-key and sender-vouches methods of
184 subject confirmation. Additional subject confirmation mechanisms may also be
185 supported. It is strongly RECOMMENDED that an XML signature be used to establish
186 the relationship between the message sender and the attached assertions. This is
187 especially RECOMMENDED whenever the SOAP message exchange is conducted over
188 an unprotected transport.

189 Any processor of SAML assertions MUST conform to the required validation and
190 processing rules defined in the SAML specification.

191 The following table enumerates the mandatory subject confirmation methods and
192 summarizes their associated processing models:

Mechanism	RECOMMENDED Processing Rules
<code>urn:oasis:names:tc:SAML:1.0:cm:holder-of-key</code>	The requestor (the subject) includes an XML Signature that can be verified with the key information in the referenced security token.
<code>urn:ietf:rfc:3075</code>	The requestor (the subject) includes an XML Signature that can be verified with the key information in the referenced security token.
<code>Urn:oasis:names:tc:SAML:1.0:cm:sender-vouches</code>	The requestor (the sender, different from the subject) vouches for the verification of the subject. The receiver MUST have an existing trust relationship with the requestor to accept this. It is RECOMMENDED that the requestor sign the token and the message or use a secure transport.

193 Note that the high level processing model described in the following sections does
194 not differentiate between message author and message sender as would be
195 necessary to guard against replay attacks. The high-level processing model also does
196 not take into account requirements for authentication of receiver by sender, or for
197 message or assertion confidentiality. These concerns must be addressed by means
198 other than those described in the high-level processing model.

199 **3.4.1 Holder-of-key Subject Confirmation Method**

200 The following sections describe the holder-of-key method of establishing the
201 correspondence between a SOAP message sender and the subject of SAML assertions
202 added to the SOAP message according to the SAML binding of [WS-Security](#).

203 **3.4.1.1 Sender**

204 A message sender uses the holder-of-key confirmation method to demonstrate that
205 it is the subject of the assertions in the message. The assertions included in a
206 message that the sender will confirm by the holder-of-key method MUST include the
207 following `<saml:SubjectConfirmation>` element:

```
208 <saml:SubjectConfirmation>  
209 <saml:ConfirmationMethod>
```

```
210 urn:oasis:names:tc:SAML:1.0:cm:holder-of-key
211 </saml:ConfirmationMethod>
212 <ds:KeyInfo>...</ds:KeyInfo>
213 </saml:SubjectConfirmation>
```

214 The `<saml:SubjectConfirmation>` element MUST include a `<ds:KeyInfo>` element
215 that identifies the public or secret key to be used to confirm the identity of the
216 subject.

217 To satisfy the associated confirmation method processing of the message receiver,
218 the sender MUST demonstrate knowledge of the key of the subject. The sender MAY
219 accomplish this by using the key of the subject to sign content within the message
220 and by including the resulting `<ds:Signature>` element in the `<wsse:Security>`
221 header.

222 `<ds:Signature>` elements produced for this purpose MUST conform to the
223 canonicalization and token inclusion rules defined in the core [WS-Security](#)
224 specification.

225 3.4.1.2 Receiver

226 A message receiver SHOULD NOT accept assertions containing a holder-of-key
227 `<saml:ConfirmationMethod>` unless the message sender has demonstrated
228 knowledge of the key identified by the `<ds:keyInfo>` element of the
229 `<saml:SubjectConfirmation>` element: If the receiver determines that the sender
230 has demonstrated knowledge of a subject confirmation key, then the SAML
231 assertions containing the confirmation key MAY be attributed to the sender and any
232 elements of the message whose integrity is protected by the subject confirmation
233 key MAY be considered to have been authored by the subject.

234 3.4.1.3 Example

235 The following example illustrates the use of the holder-of-key subject confirmation
236 method to establish the correspondence between the SOAP message author and the
237 subject of the SAML assertions in the `<wsse:Security>` header:

```
238 <?xml:version="1.0" encoding="UTF-8"?>
239 <SOAP-ENV:Envelope
240   xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
241   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
242   xmlns:xsd="http://www.w3.org/2001/XMLSchema">
243
244   <SOAP-ENV:Header>
245     <wsse:Security>
246       <saml:Assertion
247         xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"
248         MajorVersion="1" MinorVersion="0"
249         AssertionID="2sxJu9g/vvLG9sAN9bKp/8q0NKU="
250         Issuer="www.example.com"
251         IssueInstant="2002-06-19T16:58:33.173Z">
252         <saml:Conditions
253           NotBefore="2002-06-19T16:53:33.173Z"
254           NotOnOrAfter="2002-06-19T17:08:33.173Z"/>
255
256         <saml:AuthenticationStatement
257           AuthenticationMethod="urn:oasis:names:tc:SAML:1.0:am:password"
258           AuthenticationInstant="2002-06-19T16:57:30.000Z">
259           <saml:Subject>
```

```

260         <saml:NameIdentifier
261             NameQualifier="www.example.com"
262             Format="">
263 uid=joe,ou=people,ou=saml-demo,o=example.com
264         </saml:NameIdentifier>
265         <saml:SubjectConfirmation>
266             <saml:ConfirmationMethod>
267 urn:oasis:names:tc:SAML:1.0:cm:holder-of-key
268             </saml:ConfirmationMethod>
269             <ds:KeyInfo>
270                 <ds:KeyValue>...</ds:KeyValue>
271             </ds:KeyInfo>
272         </saml:SubjectConfirmation>
273     </saml:Subject>
274 </saml:AuthenticationStatement>
275
276     <saml:AttributeStatement>
277         <saml:Subject>
278             <saml:NameIdentifier
279                 NameQualifier="www.example.com"
280                 Format="">
281 uid=joe,ou=people,ou=saml-demo,o=baltimore.com
282             </saml:NameIdentifier>
283             <saml:SubjectConfirmation>
284                 <saml:ConfirmationMethod>
285 urn:oasis:names:tc:SAML:1.0:cm:holder-of-key
286                 </saml:ConfirmationMethod>
287                 <ds:KeyInfo>
288                     <ds:KeyValue>...</ds:KeyValue>
289                 </ds:KeyInfo>
290             </saml:SubjectConfirmation>
291         </saml:Subject>
292
293         <saml:Attribute
294             AttributeName="MemberLevel"
295             AttributeNamespace="http://www.oasis-
296 open.org/Catalyst2002/attributes">
297             <saml:AttributeValue>gold</saml:AttributeValue>
298         </saml:Attribute>
299         <saml:Attribute
300             AttributeName="E-mail"
301             AttributeNamespace="http://www.oasis-
302 open.org/Catalyst2002/attributes">
303             <saml:AttributeValue>joe@yahoo.com</saml:AttributeValue>
304         </saml:Attribute>
305     </saml:AttributeStatement>
306     <ds:Signature>...</ds:Signature>
307 </saml:Assertion>
308 <ds:Signature>
309     <ds:SignedInfo>...</ds:SignedInfo>
310     <ds:SignatureValue>
311 HJJWbvqW9E84vJVQkjjLLA6nNvBX7mY00TZhWbDFNDElgsCSXZ5Ekw==
312     </ds:SignatureValue>
313 </ds:Signature>
314 </wsse:Security>
315 </SOAP-ENV:Header>
316
317 <SOAP-ENV:Body>
318     <ReportRequest>
319         <TickerSymbol>SUNW</TickerSymbol>
320     </ReportRequest>
321 </SOAP-ENV:Body>
322 </SOAP-ENV:Envelope>

```

323 **3.4.2 Sender-vouches Subject Confirmation Method**

324 The following sections describe the sender-vouches method of establishing the
325 correspondence between a SOAP message sender and the SAML assertions added to
326 the SOAP message according to the SAML binding of [WS-Security](#).

327 **3.4.2.1 Sender**

328 A message sender uses the sender-vouches confirmation method to assert that it is
329 acting on behalf of the subjects of the assertions in the message. The assertions
330 included in a message that the sender will confirm by the sender-vouches method
331 MUST include the following `<saml:SubjectConfirmation>` element:

```
332 <saml:SubjectConfirmation>  
333   <saml:ConfirmationMethod>  
334   urn:oasis:names:tc:SAML:1.0:cm:sender-vouches  
335   </saml:ConfirmationMethod>  
336 </saml:SubjectConfirmation>
```

337 To satisfy the associated confirmation method processing of the receiver, the sender
338 MUST use its key to integrity protect the assertions and those elements of the SOAP
339 message that the sender is vouching for. The sender MAY accomplish this by
340 including in the corresponding `<wsse:Security>` header a `<ds:Signature>` element
341 that the sender prepares by using its key to sign the assertions and relevant
342 message content. As defined by the [XML Signature Specification](#), the sender MAY
343 identify its key by including a `<ds:KeyInfo>` element within the `<ds:Signature>`
344 element.

345 A `<ds:Signature>` element produced for this purpose MUST conform to the
346 canonicalization and token inclusion rules defined in the core [WS-Security](#)
347 specification.

348 **3.4.2.2 Receiver**

349 A message receiver SHOULD NOT accept assertions containing a sender-vouches
350 `<saml:ConfirmationMethod>` unless the assertions and SOAP message content being
351 vouched for by the sender are integrity protected by a sender who is trusted by the
352 receiver to act on behalf of the subject of the assertions.

353 **3.4.2.3 Example**

354 The following example illustrates a senders use of the sender-vouches subject
355 confirmation method with an associated `<ds:Signature>` element to establish its
356 identity and to assert that it has sent message elements on behalf of the subjects of
357 the contained assertions:

```
358 <SOAP-ENV:Envelope  
359   xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">  
360   <SOAP-ENV:Header  
361     xmlns:saml="..."  
362     <wsse:Security>  
363       <wsse:SecurityTokenReference>  
364         <saml:AssertionIDReference>XVB12#$21abc</AssertionIDReference>  
365         <wsse:Reference URI="http://www.example.com/SAMLservice"/>  
366       </wsse:SecurityTokenReference>  
367     <saml:Assertion>...</saml:Assertion>
```

```

368     <ds:Signature>...
369         <ds:KeyInfo>...</ds:KeyInfo>
370     </ds:Signature>
371 </wsse:Security>
372 </SOAP-ENV:Header>
373 <SOAP-ENV:Body>
374     ...
375 </SOAP-ENV:Body>
376 </SOAP-ENV:Envelope>

```

377 3.5 Error Codes

378 It is RECOMMENDED that systems implementing the SAML binding of [WS-Security](#)
379 respond with the error codes defined in the core [WS-Security](#) specification.
380 Implementations that chose to respond with custom errors, defined in private
381 namespaces, SHOULD take care not to introduce any security vulnerabilities as a
382 result of the information returned in their error responses.

383 A receiver that is unable to process the SAML assertions contained in a
384 `<wsse:Security>` header SHOULD use one of the fault codes listed in the core [WS-](#)
385 [Security](#) specification to report the error. The RECOMMENDED correspondence
386 between the common assertion processing failures and the error codes defined in the
387 core [WS-security](#) specification are defined in the following table:

Assertion Processing Error	RECOMMENDED Error
A referenced SAML assertion could not be retrieved.	<code>wsse:SecurityTokenUnavailable</code>
An assertion contains a <code><saml:Condition></code> element that the receiver does not understand.	<code>wsse:UnsupportedSecurityToken</code>
A signature within an assertion or including an assertion is invalid.	<code>wsse:FailedCheck</code>
The issuer of an assertion is not acceptable to the receiver.	<code>wsse:InvalidSecurityToken</code>
The receiver does not understand the extension schema used in a assertion.	<code>wsse:UnsupportedSecurityToken</code>

388 3.6 Threat Model and Countermeasures

389 This document defines the mechanisms and procedures for securely attaching SAML
390 assertions to SOAP messages. SOAP messages are used in multiple contexts,
391 specifically including cases where the message is transported without an active
392 session, the message is persisted, or the message is routed through a number of
393 intermediaries. Such a general context of use suggests that users of this binding
394 must be concerned with a variety of threats. The following sections describe the
395 vulnerability of the SAML token binding of [WS-Security](#) to a variety of threats. In

396 general, the use of SAML assertions with [WS-Security](#) introduces no new threats
397 beyond those identified for SAML or by the core [WS-Security](#) specification.

398 The following sections provide an overview of the characteristics of the threat model,
399 and the countermeasures that SHOULD be adopted for each perceived threat.

400 **3.6.1 Eavesdropping**

401 Eavesdropping is a threat to the SAML token binding of WS-Security in the same
402 manner as it is a threat to any network protocol. The routing of SOAP messages
403 through intermediaries increases the potential incidences of eavesdropping.
404 Additional opportunities for eavesdropping exist when SOAP messages are persisted.

405 To provide maximum protection from eavesdropping, assertions and sensitive
406 message content SHOULD be encrypted such that only the intended audiences can
407 view the material. This removes threats of eavesdropping in transit, but MAY not
408 remove risks associated with storage by the receiver or poor handling of the clear
409 text by the receiver.

410 Transport-layer security MAY be used to protect the message and contained SAML
411 assertions from eavesdropping while in transport, but message content MUST be
412 encrypted above the transport if it is to be protected from eavesdropping by
413 intermediaries.

414 **3.6.2 Replay**

415 The reliance on assertions with a holders-of-key subject confirmation mechanism
416 precludes all but a holder of the key from binding the assertions to a SOAP message.
417 Although this mechanism affectively restricts message authorship to the holder of
418 the subject key, it does not preclude the capture and resubmission of the message
419 by other parties.

420 Assertions that contain a sender-vouches confirmation mechanism introduce another
421 dimension to replay vulnerability because the assertions impose no restriction on the
422 senders who may use or reuse the assertions. Any entity coming into contact with
423 such assertions could use them in a message in which they use their identity to
424 vouch for the subject of the assertions.

425 Replay attacks can be addressed by using message timestamps and caching, as well
426 as by using other application-specific tracking mechanisms.

427 **3.6.3 Message Insertion**

428 The SAML token binding of WS-Security is not vulnerable to message insertion
429 attacks.

430 **3.6.4 Message Deletion**

431 The SAML token binding of WS-Security is not vulnerable to message insertion
432 attacks.

433 **3.6.5 Message Modification**

434 The SAML token binding of WS-Security is protected from message modification if
435 the relevant message content is signed by the holder of the key or the vouching
436 sender. It is strongly RECOMMENDED that all relevant and immutable message
437 content be signed by the sender. Receivers SHOULD only consider those portions of
438 the document that are covered by the sender's signature as being subject to the
439 assertions in the message.

440 SAML assertions appearing in <wsse:Security> header elements SHOULD be signed
441 by their issuing Authority such that message receivers can have confidence that the
442 assertions have not been forged or altered since their issuance. It is strongly
443 RECOMMENDED that the message sender also sign the <saml:Assertion> elements
444 (either within the token, as part of the message, or both).

445 Transport-layer security MAY be used to protect the message and contained SAML
446 assertions from modification while in transport, but signatures are required to extend
447 such protection through intermediaries.

448 **3.6.6 Man-in-the-Middle**

449 Assertions with a holder-of-key subject confirmation method are not vulnerable to a
450 MITM attack. Assertions with a sender-vouches subject confirmation method are
451 vulnerable to MITM attacks to the degree that the receiver does not have a trusted
452 binding of key to the vouching sender's identity.

453 **4 Acknowledgements**

454 This specification was developed as a result of joint work of many individuals from
455 the WSS TC including:

456 TBD

5 References

- 457
- 458 **[DIGSI G]** Informational RFC 2828, "[Internet Security Glossary](#)," May
459 2000.
- 460 **[KEYWORDS]** S. Bradner, "Key words for use in RFCs to Indicate Requirement
461 Levels," [RFC 2119](#), Harvard University, March 1997
- 462 **[SAMLBind]** Oasis Committee Specification 01, P. Mishra (Editor) [Bindings
463 and Profiles for the OASIS Security Assertion Markup Language
464 \(SAML\)](#), May 2002.
- 465 **[SAMLCore]** Oasis Committee Specification 01, P. Hallem-Baker, and E.
466 Maler, (Editors), [Assertions and Protocol for the OASIS Security
467 Assertion Markup Language \(SAML\)](#), May 2002.
- 468 **[SAMLReqs]** OASIS Committee Consensus Draft, D. Platt, Evan Prodromou
469 (Editors), [SAML Requirements and Use Cases](#), OASIS,
470 December 2001.
- 471 **[SAMLSecure]** OASIS Committee Specification 01, C. McLaren (Editor),
472 [Security and Privacy Considerations for the OASIS Security
473 Assertion Markup Language \(SAML\)](#) , May 2002.
- 474 **[SOAP]** W3C Note, "[SOAP: Simple Object Access Protocol 1.1](#)," 08 May
475 2000.
- 476 W3C Working Draft, Nilo Mitra (Editor), [SOAP Version 1.2 Part
477 0: Primer](#), June 2002.
- 478 W3C Working Draft, Martin Gudgin, Marc Hadley, Noah
479 Mendelsohn, Jean-Jacques Moreau, Henrik Frystyk Nielsen
480 (Editors), [SOAP Version 1.2 Part 1: Messaging Framework](#), June
481 2002.
- 482 W3C Working Draft, Martin Gudgin, Marc Hadley, Noah
483 Mendelsohn, Jean-Jacques Moreau, Henrik Frystyk Nielsen
484 (Editors), [SOAP Version 1.2 Part 2: Adjuncts](#), June 2002.
- 485 **[URI]** T. Berners-Lee, R. Fielding, L. Masinter, "Uniform Resource
486 Identifiers (URI): Generic Syntax," [RFC 2396](#), MIT/LCS, U.C.
487 Irvine, Xerox Corporation, August 1998.
- 488 **[WS-SAML]** Contribution to the WSS TC, P. Mishra (Editor), [WS-Security
489 Profile of the Security Assertion Markup Language \(SAML\)
490 Working Draft 04](#), Sept 2002.
- 491 **[WS-Security]** TBS – point to the OASIS core draft
- 492 **[XML-ns]** W3C Recommendation, "[Namespaces in XML](#)," 14 January
493 1999.
- 494 **[XML Signature]** W3C Recommendation, "[XML Signature Syntax and
495 Processing](#)," 12 February 2002.

496 **[XML Token]** Contribution to the WSS TC, Chris Kaler (Editor),
497 WS-Security Profile for XML-based Tokens, August 2002.
498

499

Appendix A: Revision History

Rev	Date	What
01	19-Sep-02	Initial draft produced by extracting SAML related content from [XML token]
02	23-Sep-02	Merged in content from SS TC submission

500

501

Appendix B: Notices

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

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

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

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

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

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

534