



1

2

3

Web Services Security SAML Token Binding

4

Working Draft 05, 16 December 2002

5

Document identifier:

6

WSS-SAML-05

7

Location:

8

TBD

9

Editors:

10

Phillip Hallam-Baker, VeriSign

11

Chris Kaler, Microsoft

12

Ronald Monzillo, Sun

13

Anthony Nadalin, IBM

14

Contributors:

15

TBD – Revise this list to include WSS TC contributors

Phillip Hallam-Baker, VeriSign

Prateek Mishra, Netegrity

Jeff Hodges, Sun Microsystems

Anthony Nadalin, IBM

Maryann Hondo, IBM

Nataraj Nagaratnam, IBM

Chris Kaler, Microsoft

Hemma Prafullchandra, VeriSign

Eve Maler, Sun Microsystems

Irving Reid, Baltimore

Hiroshi Maruyama, IBM

Krishna Sankar, Cisco

Chris McLaren, Netegrity

John Shewchuk, Microsoft

16

Abstract:

17

This document describes how to use Security Assertion Markup Language

18

(SAML) assertions with the [WS-Security](#) specification.

19

Status:

20

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

21

22

Committee members should send comments on this specification to

23

wss@lists.oasis-open.org list. Others should subscribe to and send comments

24

to the wss-comment@lists.oasis-open.org list. To subscribe, visit

25

<http://lists.oasis-open.org/ob/adm.pl>.

26

For information on the disclosure of Intellectual Property Rights or licensing

27

terms related to the work of the Web Services Security TC please refer to the

28 Intellectual Property Rights section of the TC web page at [http://www.oasis-](http://www.oasis-open.org/committees/wss/)
29 [open.org/committees/wss/](http://www.oasis-open.org/committees/wss/). The OASIS policy on Intellectual Property Rights
30 is described at <http://www.oasis-open.org/who/intellectualproperty.shtml>.
31

Table of Contents

32	1.....Introduction	
33	45
34	1.1 Goals and Requirements.....	45
35	1.1.1 Requirements.....	45
36	1.1.2 Non-Goals.....	45
37	2.....Notations and Terminology	
38	56
39	2.1 Notational Conventions.....	56
40	2.2 Namespaces.....	56
41	2.3 Terminology.....	67
42	3.....Usage	
43	78
44	3.1 Processing Model.....	78
45	3.2 Attaching Security Tokens.....	78
46	3.3 Identifying and Referencing Security Tokens.....	89
47	3.4 Proof-of-Possession of Security Tokens.....	1110
48	3.5 Error Codes.....	1311
49	3.6 Threat Model and Countermeasures.....	1918
50	4.....Acknowledgements	
51	2220
52	5.....References	
53	2321
54	Appendix A: Revision History.....	2523
55	Appendix B: Notices.....	2624
56		

57 1 Introduction

58 The [WS-Security](#) specification proposes a standard set of [SOAP](#) extensions
59 that can be used when building secure Web services to implement message
60 level integrity and confidentiality. This specification describes the use of
61 Security Assertion Markup Language (SAML) assertions from the
62 `<wsse:Security>` header block defined by the [WS-Security](#) specification.

63 1.1 Goals and Requirements

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

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

68 1.1.1 Requirements

69 TBS

70 1.1.2 Non-Goals

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

72 TBS

73

74 2 Notations and Terminology

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

77 2.1 Notational Conventions

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

81 Namespace URIs (of the general form "some-URI") represent some
82 application-dependent or context-dependent URI as defined in RFC2396.

83 This specification is designed to work with the general SOAP message
84 structure and message processing model, and should be applicable to any
85 version of SOAP. The current SOAP 1.2 namespace URI is used herein to
86 provide detailed examples, but there is no intention to limit the applicability of
87 this specification to a single version of SOAP.

88 Readers are presumed to be familiar with the terms in the Internet Security
89 Glossary.

90 2.2 Namespaces

91 The XML namespace URIs that MUST be used by implementations of this
92 specification are as follows (note that different elements in this specification
93 are from different namespaces):

94 `http://schemas.xmlsoap.org/ws/2002/xx/secext`
95 `http://schemas.xmlsoap.org/ws/2002/xx/utility`

96 The following namespaces are used in this document:

97

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

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

98 **2.3 Terminology**

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

101 Defined below are the basic definitions for additional terminology used in this
102 specification.

103 [TBS]

3 Usage

104

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

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

108 **Contact information:** TBD

109 **Description:** Given below.

110 **Updates:** None.

3.1 Processing Model

111

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

114 When a receiver processes a `<wsse:Security>` header containing or referencing
115 SAML assertions, it MUST select, based on its policy, the signatures and assertions
116 that it will process. It is assumed that a receiver's signature selection policy may rely
117 on semantic labeling of `<wsse:SecurityTokenReference>` elements occurring in the
118 `<ds:KeyInfo>` elements within the signatures. It is also assumed that the assertions
119 selected for validation and processing will include those referenced from the
120 `<ds:KeyInfo>` and `<ds:SignedInfo>` elements of the selected signatures.

121 As part of its validation and processing of the selected assertions, the receiver MUST
122 make an explicit determination of the relationship between the subject of each
123 assertion and the sender of the message. Two methods for establishing this
124 correspondence, `holder-of-key` and `sender-vouches` are described below. Senders
125 and receivers implementing the SAML binding of [WS-Security](#) MUST implement the
126 processing necessary to support both of these subject confirmation methods.

3.2 Attaching Security Tokens

127

128 SAML assertions are attached to SOAP messages using [WS-Security](#) by placing
129 assertion elements or references to assertions inside a `<wsse:Security>` header.
130 The following example illustrates a SOAP message containing a SAML assertion in a
131 `<wsse:Security>` header.

132

133

134

135

136

137

138

139

140

141

142

143

144

```
<S:Envelope xmlns:S="...">
  <S:Header>
    <wsse:Security xmlns:wsse="...">
      <saml:Assertion
        MajorVersion="1"
        MinorVersion="0"
        AssertionID="SecurityToken-ef375268"
        Issuer="elliottw1"
        IssueInstant="2002-07-23T11:32:05.6228146-07:00"
        xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion">
        ...
      </saml:Assertion>
      ...
    </wsse:Security>
  </S:Header>
</S:Envelope>
```

145
146
147
148
149
150

```
</wsse:Security>
</S:Header>
<S:Body>
...
</S:Body>
</S:Envelope>
```

151 3.3 Identifying and Referencing Security Tokens

152 The [WS-Security](#) specification defines the `<wsse:SecurityTokenReference>` element
153 for referencing security tokens. Three forms of token references are defined [by this](#)
154 [element and the element schema includes provision for defining additional reference](#)
155 [forms should they be necessary. The three forms of token references defined by the](#)
156 [<wsse:SecurityTokenReference> element are defined as follows:-](#)

157 ~~□ An element reference — a security token specific XML element that contains an~~
158 ~~identifier and perhaps locator of a security token within the message or at some~~
159 ~~external location:-~~

160 ~~A URI reference — a generic element that conveys in its attributes, the security token~~
161 ~~URI and token type value (i.e. ValueType) that define the location and perhaps~~
162 ~~identifier of a security token occurring either within the message or at some external~~
163 ~~location. A URI containing only a fragment identifier is interpreted as identifying the~~
164 ~~corresponding security token within the message in which the fragment identifier~~
165 ~~occurs:-~~

166 • A key identifier reference – a generic element (i.e. `<wsse:KeyIdentifier>`) that
167 conveys a security token identifier and indicates in its attributes (as necessary)
168 the type of the token being identified (i.e. the `ValueType`), the identifier encoding
169 type (i.e. the `EncodingType`), and any other parameters necessary to reference
170 the security token.

171 When a key identifier is used to reference a SAML assertion the `ValueType`
172 attribute must contain the value “saml:Assertion” and the `<wsse:KeyIdentifier>`
173 element must contain as its element value the corresponding `AssertionID`.

174 The SAML binding of WSS-Security prescribes the use of the following attributes
175 within a key identifier reference when the referenced assertion must be acquired
176 from the assertion authority.

177 [/wsse:SecurityTokenReference/KeyIdentifier/@saml:Location](#)

178 This optional attribute is used to carry a URI reference describing how to
179 locate the SAML authority. As defined by SAMLCore, the syntax of the URI will
180 depend on the protocol binding defined by the `saml:Binding` attribute of the
181 `<wsse:KeyIdentifier>`. For example, a binding based on HTTP will be a web
182 URL, while a binding based on SMTP might use the “mailto” scheme.

183 [/wsse:SecurityTokenReference/keyIdentifier/@saml:Binding](#)

184 A URI reference identifying the SAML protocol binding to use in
185 communicating with the SAML authority. SAML protocol bindings are assigned
186 a URI reference in `SAMLBind`.

187 { Note to TC: this mechanism should be extended to support artifact
188 references”

189 • ~~a generic element that conveys in its attributes, the security token identifier (i.e.~~
190 ~~wsu:id) and token type value (i.e. ValueType) that identifies a security token~~
191 ~~with matching wsu:id and ValueType occurring within a <wsse:Security>~~
192 ~~header of the message. Identifier references may only be used to reference~~
193 ~~security tokens that carry matching attributes, which approximately restricts their~~
194 ~~use to Binary Security Tokens attributed as a result of their encapsulation in~~
195 ~~XML. A key name reference – a <ds:KeyName> element contains a string value key~~
196 ~~identifier, and the referenced token or tokens are those that contain a matching~~
197 ~~identity value.~~

198 The syntax of SAML assertion identifiers does not facilitate their differentiation
199 from other identifier forms. For this reason, key name reference forms SHOULD
200 not be used to reference SAML assertions.

201 • A Direct or URI reference – a generic element (i.e. <wsse:Reference>) that
202 identifies a security token by URI. If only a fragment is specified, then the
203 reference is to the security token within the document whose wsu:Id attribute
204 value matches the fragment. Otherwise, the reference is to the (potentially
205 external) security token identified by the URI.

206 The SAML assertion schema does not include or provide for inclusion of the
207 wsu:Id attribute. For this reason, a URI reference cannot be used to (directly)
208 reference a SAML assertion.

209 ~~A URI reference containing a URL may be combined with a token-specific element~~
210 ~~reference to yield a location-qualified reference.~~

211 In ~~t~~he SAML binding of WS-security, ~~a referenced~~ SAML assertions may be
212 referenced in three contexts:

213 • A SAML assertion may be referenced from a <ds:KeyInfo> element of a
214 <ds:Signature> element in a <wsse:Security> header. In this case, the assertion
215 contains the key used in the signature calculation.

216 • A SAML assertion may be referenced from a <wsse:Security> header or from an
217 element (other than a signature) in the header.

218 • A SAML assertion may be referenced from a <ds:Reference> element within the
219 <ds:SignedInfo> element of a <ds:Signature> element in a <wsse:Security>
220 header. In this case, the referenced assertion is being signed by the containing
221 signature.

222 In each of these contexts, the referenced assertion may be:

223 • local – in which case, it is included in the <wsse:Security> header containing the
224 reference.

225 • remote – in which case it is not included in the <wsse:Security> header
226 containing the reference, but may occur in another part of the SOAP message or
227 may be available at the location identified by the reference which may be an
228 assertion authority.

229 In the SAML binding of WS-Security, the preferred method to reference SAML
230 assertions is by key identifier reference.

231 A SAML assertion that exists in a <wsse:Security> header may be referenced from
232 the <wsse:Security> header, a header element, or from the <ds:KeyInfo> element
233 of a <ds:Signature> element in the header by using a key identifier reference.

234 Methods to reference SAML assertion from a <ds:Reference> element remain to be
235 formalized.

236 ~~-is identified by a <saml:AssertionIDReference> occurring either as~~
237 ~~an element reference or as a String value fragment identifier in a URI~~
238 ~~reference.~~

239 **3.3.1 SAML Assertion Referenced from Header or** 240 **Element Reference Elements**

241 A SAML assertion may be referenced from a <wsse:Security> header or from an
242 element (other than a signature) in the header. The following examples demonstrate
243 the use of a key identifier reference in a <wsse:Security> header to reference a local
244 SAML assertion. A <wsse:SecurityTokenReference> containing a
245 <saml:AssertionIDReference> element containing a SAML assertion identifier may
246 be used to reference a SAML assertion occurring within the <wsse:Security> header
247 of the SOAP message in which the reference occurs. The following example
248 illustrates the use of a <wsse:securityTokenReference> containing a
249 <saml:AssertionIDReference> within the <keyInfo> of an XML Signature element
250 to reference the SAML assertion (in the <wsse:Security> header) that contains the
251 key used to compute the signature.

```
252 <S:Envelope xmlns:S="...">
253   <S:Header>
254     <wsse:Security xmlns:wsse="...">
255       <saml:Assertion
256         MajorVersion="1"
257         MinorVersion="0"
258         AssertionID="SecurityToken-ef375268"
259         Issuer="elliottw1"
260         IssueInstant="2002-07-23T11:32:05.6228146-07:00"
261         xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion">
262         ...
263       </saml:Assertion>
264       <wsse:SecurityTokenReference
265         <wsse:KeyIdentifier wsu:id="..."
266           ValueType="saml:Assertion"
267           SecurityToken-ef375268
268         </wsse:KeyIdentifier>
269       </wsse:SecurityTokenReference>
270     <del><ds:Signature xmlns:ds="...">
271       ...
272     </del><ds:KeyInfo>
273       <del><wsse:SecurityTokenReference>
274         <del><saml:AssertionIDReference>
275           SecurityToken-ef375268
276         </del></saml:AssertionIDReference>
277       </del></wsse:SecurityTokenReference>
278     </del></ds:KeyInfo>
279     </del></ds:Signature>
280     ...
281   </del></wsse:Security>
282 </S:Header>
283 <S:Body>
```

284
285
286

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

287 A SAML assertion that exists outside of a <wsse:Security> header may be
288 referenced from the <wsse:Security> header element by including (in the reference)
289 saml:Location and saml:Binding attributes that define the address and protocol to
290 use to acquire the identified assertion at a SAML assertion authority or responder.

291
292
293
294
295
296
297
298

```
<wsse:SecurityTokenReference
  <wsse:KeyIdentifier wsu:id="..."
    ValueType="saml:Assertion"
    saml:Location=http://www.fabrikam123.com/elliottw1
    saml:Binding="urn:oasis:names:tc:SAML:1.0:bindings:SOAP-binding"
    SecurityToken-ef375268
  </wsse:KeyIdentifier>
</wsse:SecurityTokenReference>
```

299 **3.3.2 ~~URI References to SAML assertion~~ referenced from** 300 **KeyInfo**

301 The following examples demonstrate the use of a key identifier reference from within
302 a <ds:KeyInfo> element of a <ds:Signature> element in a <wsse:Security> header.

303 ~~As depicted in the following example depicts the use of,~~ a key identifier reference
304 containing a SAML AssertionID (as its value) to reference a local assertion identified
305 by AssertionID. { It is presumed that the default encoding type is xsi:string} .

306
307
308
309
310
311
312

```
<ds:KeyInfo>
  <wsse:SecurityTokenReference>
    <wsse:KeyIdentifier wsu:id="..."
      ValueType="saml:Assertion"
      SecurityToken-ef375268
    </wsse:KeyIdentifier>
  </wsse:SecurityTokenReference>
```

313 ~~</ds:KeyInfo>URI reference containing only a fragment identifier~~
314 ~~consisting of a <saml:AssertionIDReference> may be used to reference~~
315 ~~a SAML assertion occurring within the <wsseSecurity> header of the~~
316 ~~SOAP message in which the reference occurs. A URI reference containing~~
317 ~~an XML path expression can be used to reference a SAML assertion~~
318 ~~occurring anywhere within the containing SOAP message.~~

319
320
321
322
323
324

```
<wsse:SecurityTokenReference>
  <wsse:Reference URI="#SecurityToken-ef375268"
    ValueType="saml:IDReferenceType">
  </wsse:Reference>
</wsse:SecurityTokenReference>
```

325 The following example extends the previous example with the inclusion of
326 saml:Location and saml:Binding attributes that define the address and protocol to
327 use to acquire the identified assertion at a SAML assertion authority or
328 responder. ~~The following example demonstrates the use of a URI reference in~~
329 ~~conjunction with a <saml:AssertionIDReference> to define the location of the SAML~~
330 ~~responder at which the identified assertion may be obtained.~~

331
332
333
334

```
<ds:KeyInfo>
  <wsse:SecurityTokenReference>
    <wsse:KeyIdentifier wsu:id="..."
      ValueType="saml:Assertion"
```

```

335     saml:Binding="urn:oasis:names:tc:SAML:1.0:bindings:SOAP-binding"
336     saml:Location="http://www.fabrikam123.com/elliottw1"
337     SecurityToken-ef375268
338     </wsse:KeyIdentifier>
339     </wsse:SecurityTokenReference>
340 </ds:KeyInfo>
341 </wsse:SecurityTokenReference>
342 <saml:AssertionIDReference>SecurityToken-ef375268
343 </saml:AssertionIDReference>
344 <wsse:Reference-URI="http://www.fabrikam123.com/elliottw1"
345 </wsse:Reference>
346 </wsse:SecurityTokenReference>

```

347 **3.3.3 SAML assertion referenced from SignedInfo-identifier**
348 **References to SAML Assertions**

349 Methods to reference SAML assertion from <ds:Reference> elements remain to be
350 formalized. One issue that remains to be resolved is how to differentiate whether it is
351 the reference or the referenced assertion that is to be digested. SAML assertions may
352 not be referenced by identifier references because the <saml:Assertion> element
353 schema does not include the wsu:id and ValueType attributes.

354 **3.4 Proof-of-Possession of Security Tokens**

355 The SAML binding of **WS-Security** requires that message senders and receivers
356 support the holder-of-key and sender-vouches methods of subject confirmation. It is
357 strongly RECOMMENDED that an XML signature be used to establish the relationship
358 between the message sender and the attached assertions. This is especially
359 RECOMMENDED whenever the SOAP message exchange is conducted over an
360 unprotected transport.

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

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

Mechanism	RECOMMENDED Processing Rules
urn:oasis:names:tc:SAML:1.0:cm:holder-of-key	The requestor includes an XML Signature that can be verified with the key information in the <saml:ConfirmationMethod> of the SAML assertion referenced by the Signature.
Urn:oasis:names:tc:SAML:1.0:cm:sender-vouches	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.
--	--

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

371 3.4.1 Holder-of-key Subject Confirmation Method

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

376 3.4.1.1 Sender

377 A message sender uses the holder-of-key confirmation method to
378 demonstrate that it is authorized to act as the subject of the assertions in the
379 message. The assertions included in a message that the sender will confirm
380 by the holder-of-key method MUST include the following
381 `<saml:SubjectConfirmation>` element:

```
382 <saml:SubjectConfirmation>  
383   <saml:ConfirmationMethod>  
384     urn:oasis:names:tc:SAML:1.0:cm:holder-of-key  
385   </saml:ConfirmationMethod>  
386   <ds:KeyInfo>...</ds:KeyInfo>  
387 </saml:SubjectConfirmation>
```

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

391 To satisfy the associated confirmation method processing of the message
392 receiver, the sender MUST demonstrate knowledge of the confirmation key.
393 The sender MAY accomplish this by using the confirmation key to sign content
394 within the message and by including the resulting `<ds:Signature>` element in
395 the `<wsse:Security>` header.

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

399 SAML assertions that contain a holder-of-key `<saml:SubjectConfirmation>`
400 element SHOULD contain a `<ds:Signature>` element that protects the
401 integrity of the confirmation `<ds:KeyInfo>` established by the assertion
402 authority.

403 The canonicalization method used to produce the <ds:Signature>
404 elements used to protect the integrity of SAML assertions MUST support the
405 validation of these <ds:Signature> elements in contexts (such as
406 <wsse:Security> header elements) other than those in which the signatures
407 were calculated.

408 3.4.1.2 Receiver

409 Of the SAML assertions it selects for processing, a message receiver
410 **MUST SHOULD** NOT accept assertions containing a holder-of-key
411 <saml:ConfirmationMethod>, unless the receiver has validated the integrity
412 of the assertions ~~the assertions are signed and validated as described above~~
413 and the message sender has demonstrated knowledge of the key identified by
414 the <ds:keyInfo> element of the <saml:SubjectConfirmation> element. If
415 the receiver determines that the sender has demonstrated knowledge of a
416 subject confirmation key, then the SAML assertions containing the
417 confirmation key MAY be attributed to the sender and any elements of the
418 message whose integrity is protected by the subject confirmation key MAY be
419 considered to have been authored by the subject.

420 3.4.1.3 Example

421 The following example illustrates the use of the holder-of-key subject
422 confirmation method to establish the correspondence between the SOAP
423 message author and the subject of the SAML assertions in the
424 <wsse:Security> header:

```
425 <?xml:version="1.0" encoding="UTF-8"?>
426 <S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"
427   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
428   xmlns:xsd="http://www.w3.org/2001/XMLSchema">
429
430 <S:Header>
431 <wsse:Security>
432
433 <saml:Assertion
434   xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"
435   MajorVersion="1" MinorVersion="0"
436   AssertionID="2sxJu9g/vvLG9sAN9bKp/8q0NKU="
437   Issuer="www.example.com"
438   IssueInstant="2002-06-19T16:58:33.173Z">
439 <saml:Conditions
440   NotBefore="2002-06-19T16:53:33.173Z"
441   NotOnOrAfter="2002-06-19T17:08:33.173Z"/>
442
443 <saml:AuthenticationStatement
444   AuthenticationMethod="urn:oasis:names:tc:SAML:1.0:am:password"
445   AuthenticationInstant="2002-06-19T16:57:30.000Z">
446 <saml:Subject>
447 <saml:NameIdentifier
448   NameQualifier="www.example.com"
449   Format="">
450   uid=joe,ou=people,ou=saml-demo,o=example.com
451 </saml:NameIdentifier>
452 <saml:SubjectConfirmation>
453 <saml:ConfirmationMethod>
454   urn:oasis:names:tc:SAML:1.0:cm:holder-of-key
```

```

455         </saml:ConfirmationMethod>
456         <ds:KeyInfo>
457             <ds:KeyValue>...</ds:KeyValue>
458         </ds:KeyInfo>
459     </saml:SubjectConfirmation>
460 </saml:Subject>
461 </saml:AuthenticationStatement>
462
463 <saml:AttributeStatement>
464     <saml:Subject>
465         <saml:NameIdentifier
466             NameQualifier="www.example.com"
467             Format="">
468             uid=joe,ou=people,ou=saml-demo,o=baltimore.com
469         </saml:NameIdentifier>
470         <saml:SubjectConfirmation>
471             <saml:ConfirmationMethod>
472                 urn:oasis:names:tc:SAML:1.0:cm:holder-of-key
473             </saml:ConfirmationMethod>
474             <ds:KeyInfo>
475                 <ds:KeyValue>...</ds:KeyValue>
476             </ds:KeyInfo>
477         </saml:SubjectConfirmation>
478     </saml:Subject>
479
480     <saml:Attribute
481         AttributeName="MemberLevel"
482         AttributeNamespace="http://www.oasis-
483 open.org/Catalyst2002/attributes">
484         <saml:AttributeValue>gold</saml:AttributeValue>
485     </saml:Attribute>
486     <saml:Attribute
487         AttributeName="E-mail"
488         AttributeNamespace="http://www.oasis-
489 open.org/Catalyst2002/attributes">
490         <saml:AttributeValue>joe@yahoo.com</saml:AttributeValue>
491     </saml:Attribute>
492 </saml:AttributeStatement>
493 <ds:Signature>...</ds:Signature>
494 </saml:Assertion>
495
496 <ds:Signature>
497     <ds:SignedInfo>
498         <ds:CanonicalizationMethod Algorithm=
499             "http://www.w3.org/2001/10/xml-exc-c14n#" />
500         <ds:SignatureMethod Algorithm=
501             "http://www.w3.org/2000/09/xmldsig#hmac-sha1" />
502         </ds:Reference>
503         <ds:Reference URI="#MsgBody">
504             <ds:DigestMethod Algorithm=
505                 "http://www.w3.org/2000/09/xmldsig#sha1" />
506             <ds:DigestValue>GyGsF0Pi4xPU...</ds:DigestValue>
507         </ds:Reference>
508     ...</ds:SignedInfo>
509     <ds:SignatureValue>HJJWbvqW9E84vJVQk...</ds:SignatureValue>
510     <ds:KeyInfo>
511         <wsse:SecurityTokenReference>
512             <saml:AssertionIDReference>"2sxJu9g/vvLG9sAN9bKp/8q0NKU="
513             </saml:AssertionIDReference>
514         </wsse:SecurityTokenReference>
515     </ds:KeyInfo>
516 </ds:Signature>
517

```

```

518 </wsse:Security>
519 </S:Header>
520
521 <S:Body>
522   <ReportRequest>
523     <TickerSymbol>SUNW</TickerSymbol>
524   </ReportRequest>
525 </S:Body>
526 </S:Envelope>

```

527 3.4.2 Sender-vouches Subject Confirmation Method

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

532 3.4.2.1 Sender

533 A message sender uses the sender-vouches confirmation method to assert
534 that it is acting on behalf of the subjects of the assertions in the message.
535 The assertions included in a message that the sender will confirm by the
536 sender-vouches method MUST include the following
537 <saml:SubjectConfirmation> element:

```

538 <saml:SubjectConfirmation>
539   <saml:ConfirmationMethod>
540     urn:oasis:names:tc:SAML:1.0:cm:sender-vouches
541   </saml:ConfirmationMethod>
542 </saml:SubjectConfirmation>

```

543 To satisfy the associated confirmation method processing of the receiver, the
544 sender MUST ~~use its key to~~ integrity protect the assertions and those
545 elements of the SOAP message that ~~it the sender~~ is vouching for. The sender
546 MAY accomplish this by including in the corresponding <wsse:Security>
547 header a <ds:Signature> element that the sender prepares by using its key
548 to sign the assertions and relevant message content. As defined by the [XML](#)
549 [Signature](#) Specification, the sender MAY identify its key by including a
550 <ds:KeyInfo> element within the <ds:Signature> element.

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

554 3.4.2.2 Receiver

555 Of the SAML assertions it selects for processing, a message receiver
556 ~~MUST SHOULD~~ NOT accept assertions containing a sender-vouches
557 <saml:ConfirmationMethod> unless the assertions and SOAP message
558 content being vouched for by the sender are integrity protected by a sender
559 who is trusted by the receiver to act on behalf of the subject of the
560 assertions.

561 3.4.2.3 Example

562 The following example illustrates a sender's use of the sender-vouches
563 subject confirmation method with an associated `<ds:Signature>` element to
564 establish its identity and to assert that it has sent message elements on
565 behalf of the subjects of the contained assertions:

```
566 <?xml:version="1.0" encoding="UTF-8"?>
567 <S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"
568   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
569   xmlns:xsd="http://www.w3.org/2001/XMLSchema">
570
571   <S:Header>
572     <wsse:Security>
573
574       <saml:Assertion
575         xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"
576         MajorVersion="1" MinorVersion="0"
577         AssertionID="2sxJu9g/vvLG9sAN9bKp/8q0NKU="
578         Issuer="www.example.com"
579         IssueInstant="2002-06-19T16:58:33.173Z">
580         <saml:Conditions
581           NotBefore="2002-06-19T16:53:33.173Z"
582           NotOnOrAfter="2002-06-19T17:08:33.173Z"/>
583
584         <saml:AuthenticationStatement
585           AuthenticationMethod="urn:oasis:names:tc:SAML:1.0:am:password"
586           AuthenticationInstant="2002-06-19T16:57:30.000Z">
587           <saml:Subject>
588             <saml:NameIdentifier
589               NameQualifier="www.example.com"
590               Format="">
591               uid=joe,ou=people,ou=saml-demo,o=example.com
592             </saml:NameIdentifier>
593             <saml:SubjectConfirmation>
594               <saml:ConfirmationMethod>
595                 urn:oasis:names:tc:SAML:1.0:cm:sender-vouches
596               </saml:ConfirmationMethod>
597             </saml:SubjectConfirmation>
598           </saml:Subject>
599         </saml:AuthenticationStatement>
600
601         <saml:AttributeStatement>
602           <saml:Subject>
603             <saml:NameIdentifier
604               NameQualifier="www.example.com"
605               Format="">
606               uid=joe,ou=people,ou=saml-demo,o=baltimore.com
607             </saml:NameIdentifier>
608             <saml:SubjectConfirmation>
609               <saml:ConfirmationMethod>
610                 urn:oasis:names:tc:SAML:1.0:cm:sender-vouches
611               </saml:ConfirmationMethod>
612             </saml:SubjectConfirmation>
613           </saml:Subject>
614
615           <saml:Attribute
616             AttributeName="MemberLevel"
617             AttributeNamespace="http://www.oasis-
618 open.org/Catalyst2002/attributes">
619             <saml:AttributeValue>gold</saml:AttributeValue>
620           </saml:Attribute>

```

```

621     <saml:Attribute
622       AttributeName="E-mail"
623       AttributeNamespace="http://www.oasis-
624 open.org/Catalyst2002/attributes">
625       <saml:AttributeValue>joe@yahoo.com</saml:AttributeValue>
626     </saml:Attribute>
627   </saml:AttributeStatement>
628 </saml:Assertion>
629
630 <ds:Signature>
631   <ds:SignedInfo>
632     <ds:CanonicalizationMethod Algorithm=
633       "http://www.w3.org/2001/10/xml-exc-c14n#" />
634     <ds:SignatureMethod Algorithm=
635       "http://www.w3.org/2000/09/xmldsig#hmac-sha1" />
636     <ds:Reference URI="#2sxJu9g/vvLG9sAN9bKp/8q0NKU="
637       Type="saml:IDReferenceType">
638       <ds:DigestMethod Algorithm=
639         "http://www.w3.org/2000/09/xmldsig#sha1" />
640       <ds:DigestValue>GyGsF0Pi4xPU...</ds:DigestValue>
641     </ds:Reference>
642     <ds:Reference URI="#MsgBody">
643       <ds:DigestMethod Algorithm=
644         "http://www.w3.org/2000/09/xmldsig#sha1" />
645       <ds:DigestValue>GyGsF0Pi4xPU...</ds:DigestValue>
646     </ds:Reference>
647   </ds:SignedInfo>
648   <ds:SignatureValue>JWbvqW94vJVQkA...</ds:SignatureValue>
649   <ds:KeyInfo>
650     <X509Data>
651       <X509SubjectName>portal@yahoo.com</X509SubjectName>
652     </X509Data>
653   </ds:KeyInfo>
654 </ds:Signature>
655
656 </wsse:Security>
657 </S:Header>
658
659 <S:Body wsu:Id="MsgBody">
660   <ReportRequest>
661     <TickerSymbol>SUNW</TickerSymbol>
662   </ReportRequest>
663 </S:Body>
664
665 </S:Envelope>

```

3.5 Error Codes

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

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

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

678 3.6 Threat Model and Countermeasures

679 This document defines the mechanisms and procedures for securely attaching
680 SAML assertions to SOAP messages. SOAP messages are used in multiple
681 contexts, specifically including cases where the message is transported
682 without an active session, the message is persisted, or the message is routed
683 through a number of intermediaries. Such a general context of use suggests
684 that users of this binding must be concerned with a variety of threats. The
685 following sections describe the vulnerability of the SAML token binding of WS-
686 Security. In general, the use of SAML assertions with [WS-Security](#) introduces
687 no new threats beyond those identified for SAML or by the core [WS-Security](#)
688 specification.

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

691 3.6.1 Eavesdropping

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

697 To provide maximum protection from eavesdropping, assertions, [assertion](#)
698 [references](#), and sensitive message content SHOULD be encrypted such that only the
699 intended audiences can view their content. This removes threats of eavesdropping in
700 transit, but MAY not remove risks associated with storage or poor handling -by the
701 receiver.

702 Transport-layer security MAY be used to protect the message and contained SAML
703 assertions [and/or references](#) from eavesdropping while in transport, but message

704 content MUST be encrypted above the transport if it is to be protected from
705 eavesdropping by intermediaries.

706 3.6.2 Replay

707 The reliance on authority protected (e.g. signed) assertions with a holder-of-
708 key subject confirmation mechanism precludes all but a holder of the key
709 from binding the assertions to a SOAP message. Although this mechanism
710 affectively restricts message authorship to the holder of the confirmation key,
711 it does not preclude the capture and resubmission of the message by other
712 parties.

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

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

720 3.6.3 Message Insertion

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

723 3.6.4 Message Deletion

724 The SAML token binding of WS-Security is not vulnerable to message deletion
725 attacks.

726 3.6.5 Message Modification

727 The SAML token binding of WS-Security is protected from message modification if
728 the relevant message content is integrity protected signed by the holder of the key
729 or by the vouching sender. Therefore, it is strongly RECOMMENDED that all relevant
730 and immutable message content be signed by the holder of the key or by the
731 vouching sender (as the case warrants). Receivers SHOULD only consider those
732 portions of the document that are integrity protected by the appropriate entity
733 ~~covered by the sender's signature~~ as being subject to the assertions in the message.

734 ~~SAML assertions appearing in <wss:Security> header elements SHOULD be signed~~
735 ~~by their issuing authority. To ensure so~~ that message receivers can have confidence
736 that ~~received the~~ assertions have not been forged or altered since their issuance.
737 SAML assertions and assertion references appearing in <wss:Security> header
738 elements MUST be integrity protected (e.g. signed) by their issuing authority or the
739 vouching sender (as the case warrants). It is strongly RECOMMENDED that a
740 message sender ~~sign~~ sign any <saml:Assertion> elements that it is confirming and
741 that are not signed by their issuing authority.

742 Transport-layer security MAY be used to protect the message and contained SAML
743 assertions [and/or assertion references](#) from modification while in transport, but
744 signatures are required to extend such protection through intermediaries.

745 **3.6.6 Man-in-the-Middle**

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

750 **4 Acknowledgements**

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

753 TBD

5 References

- 754
- 755 **[DIGSIG]** Informational RFC 2828, "[Internet Security Glossary](#)," May
756 2000.
- 757 **[KEYWORDS]** S. Bradner, "Key words for use in RFCs to Indicate Requirement
758 Levels," [RFC 2119](#), Harvard University, March 1997
- 759 **[SAMLBind]** Oasis Committee Specification 01, P. Mishra (Editor) [Bindings
760 and Profiles for the OASIS Security Assertion Markup Language
761 \(SAML\)](#), May 2002.
- 762 **[SAMLCore]** Oasis Committee Specification 01, P. Hallem-Baker, and E.
763 Maler, (Editors), [Assertions and Protocol for the OASIS Security
764 Assertion Markup Language \(SAML\)](#), May 2002.
- 765 **[SAMLReqs]** OASIS Committee Consensus Draft, D. Platt, Evan Prodromou
766 (Editors), [SAML Requirements and Use Cases](#), OASIS,
767 December 2001.
- 768 **[SAMLSecure]** OASIS Committee Specification 01, C. McLaren (Editor),
769 [Security and Privacy Considerations for the OASIS Security
770 Assertion Markup Language \(SAML\)](#) , May 2002.
- 771 **[SOAP]** W3C Note, "[SOAP: Simple Object Access Protocol 1.1](#)," 08 May
772 2000.
- 773 W3C Working Draft, Nilo Mitra (Editor), [SOAP Version 1.2 Part
774 0: Primer](#), June 2002.
- 775 W3C Working Draft, Martin Gudgin, Marc Hadley, Noah
776 Mendelsohn, Jean-Jacques Moreau, Henrik Frystyk Nielsen
777 (Editors), [SOAP Version 1.2 Part 1: Messaging Framework](#), June
778 2002.
- 779 W3C Working Draft, Martin Gudgin, Marc Hadley, Noah
780 Mendelsohn, Jean-Jacques Moreau, Henrik Frystyk Nielsen
781 (Editors), [SOAP Version 1.2 Part 2: Adjuncts](#), June 2002.
- 782 **[URI]** T. Berners-Lee, R. Fielding, L. Masinter, "Uniform Resource
783 Identifiers (URI): Generic Syntax," [RFC 2396](#), MIT/LCS, U.C.
784 Irvine, Xerox Corporation, August 1998.
- 785 **[WS-SAML]** Contribution to the WSS TC, P. Mishra (Editor), [WS-Security
786 Profile of the Security Assertion Markup Language \(SAML\)
787 Working Draft 04](#), Sept 2002.
- 788 **[WS-Security]** TBS – point to the OASIS core draft
- 789 **[XML-ns]** W3C Recommendation, "[Namespaces in XML](#)," 14 January
790 1999.
- 791 **[XML Signature]** W3C Recommendation, "[XML Signature Syntax and
792 Processing](#)," 12 February 2002.

793 **[XML Token]** Contribution to the WSS TC, Chris Kaler (Editor),
794 WS-Security Profile for XML-based Tokens, August 2002.
795

796

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
03	18-Nov-02	Resolved issues raised by TC
04	09-Dec-02	Refined confirmation mechanisms, and added signing example

797

798

Appendix B: Notices

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

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

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

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

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

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