



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 Nagarathnam, 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	4
34	1.1 Goals and Requirements.....	4
35	1.1.1 Requirements.....	4
36	1.1.2 Non-Goals.....	4
37	2.....	Notations and Terminology
38	5
39	2.1 Notational Conventions.....	5
40	2.2 Namespaces.....	5
41	2.3 Terminology.....	6
42	3.....	Usage
43	7
44	3.1 Processing Model.....	7
45	3.2 Attaching Security Tokens.....	7
46	3.3 Identifying and Referencing Security Tokens.....	8
47	3.4 Proof-of-Possession of Security Tokens.....	10
48	3.5 Error Codes.....	11
49	3.6 Threat Model and Countermeasures.....	18
50	4.....	Acknowledgements
51	20
52	5.....	References
53	21
54	Appendix A: Revision History.....	23
55	Appendix B: Notices.....	24
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

2 Notations and Terminology

74

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

2.1 Notational Conventions

77

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.

2.2 Namespaces

90

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]

104 3 Usage

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.

111 3.1 Processing Model

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.

127 3.2 Attaching Security Tokens

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 <S:Envelope xmlns:S="...">  
133   <S:Header>  
134     <wsse:Security xmlns:wsse="...">  
135       <saml:Assertion  
136         MajorVersion="1"  
137         MinorVersion="0"  
138         AssertionID="SecurityToken-ef375268"  
139         Issuer="elliottw1"  
140         IssueInstant="2002-07-23T11:32:05.6228146-07:00"  
141         xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion">  
142         ...  
143       </saml:Assertion>  
144       ...
```

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

162 When a key identifier is used to reference a SAML assertion the `ValueType`
163 attribute must contain the value "saml:Assertion" and the `<wsse:KeyIdentifier>`
164 element must contain as its element value the corresponding `AssertionID`.

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

168 `/wsse:SecurityTokenReference/KeyIdentifier/@saml:Location`

169 This optional attribute is used to carry a URI reference describing how to
170 locate the SAML authority. As defined by [SAMLCore](#), the syntax of the URI will
171 depend on the protocol binding defined by the `saml:Binding` attribute of the
172 `<wsse:KeyIdentifier>`. For example, a binding based on HTTP will be a web
173 URL, while a binding based on SMTP might use the "mailto" scheme.

174 `/wsse:SecurityTokenReference/keyIdentifier/@saml:Binding`

175 A URI reference identifying the SAML protocol binding to use in
176 communicating with the SAML authority. SAML protocol bindings are assigned
177 a URI reference in [SAMLBind](#).

178 { Note to TC: this mechanism should be extended to support artifact
179 references"

- 180 • A key name reference – a `<ds:KeyName>` element contains a string value key
181 identifier, and the referenced token or tokens are those that contain a *matching*
182 identity value.

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

- 186 • A Direct or URI reference – a generic element (i.e. `<wsse:Reference>`) that
187 identifies a security token by URI. If only a fragment is specified, then the

188 reference is to the security token within the document whose *wsu:Id* attribute
189 value matches the fragment. Otherwise, the reference is to the (potentially
190 external) security token identified by the URI.

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

194 In the SAML binding of [WS-security](#), SAML assertions may be referenced in three
195 contexts:

- 196 • A SAML assertion may be referenced from a `<ds:KeyInfo>` element of a
197 `<ds:Signature>` element in a `<wsse:Security>` header. In this case, the assertion
198 contains the key used in the signature calculation.
- 199 • A SAML assertion may be referenced from a `<wsse:Security>` header or from an
200 element (other than a signature) in the header.
- 201 • A SAML assertion may be referenced from a `<ds:Reference>` element within the
202 `<ds:SignedInfo>` element of a `<ds:Signature>` element in a `<wsse:Security>`
203 header. In this case, the referenced assertion is being signed by the containing
204 signature.

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

- 206 • local – in which case, it is included in the `<wsse:Security>` header containing the
207 reference.
- 208 • remote – in which case it is not included in the `<wsse:Security>` header
209 containing the reference, but may occur in another part of the SOAP message or
210 may be available at the location identified by the reference which may be an
211 assertion authority.

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

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

217 Methods to reference SAML assertion from a `<ds:Reference>` element remain to be
218 formalized.

219 **3.3.1 SAML Assertion Referenced from Header or Element**

220 A SAML assertion may be referenced from a `<wsse:Security>` header or from an
221 element (other than a signature) in the header. The following examples demonstrate
222 the use of a key identifier reference in a `<wsse:Security>` header to reference a local
223 SAML assertion.

```
224 <S:Envelope xmlns:S="...">  
225   <S:Header>  
226     <wsse:Security xmlns:wsse="...">  
227       <saml:Assertion  
228         MajorVersion="1"  
229         MinorVersion="0"
```

```

230         AssertionID="SecurityToken-ef375268"
231         Issuer="elliottw1"
232         IssueInstant="2002-07-23T11:32:05.6228146-07:00"
233         xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion">
234         ...
235     </saml:Assertion>
236     <wsse:SecurityTokenReference
237         <wsse:KeyIdentifier wsu:id="..."
238             ValueType="saml:Assertion"
239             SecurityToken-ef375268
240         </wsse:KeyIdentifier>
241     </wsse:SecurityTokenReference>
242 </S:Header>
243 <S:Body>
244     ...
245 </S:Body>
246 </S:Envelope>

```

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

```

251 <wsse:SecurityTokenReference
252     <wsse:KeyIdentifier wsu:id="..."
253         ValueType="saml:Assertion"
254         saml:Location="http://www.fabrikam123.com/elliottw1"
255         saml:Binding="urn:oasis:names:tc:SAML:1.0:bindings:SOAP-binding"
256         SecurityToken-ef375268
257     </wsse:KeyIdentifier>
258 </wsse:SecurityTokenReference>

```

259 3.3.2 SAML assertion referenced from KeyInfo

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

262 The following example depicts the use of a key identifier reference containing a SAML
263 AssertionID (as its value) to reference a local assertion identified by AssertionID. { It
264 is presumed that the default encoding type is xsi:string} .

```

265 <ds:KeyInfo>
266     <wsse:SecurityTokenReference>
267         <wsse:KeyIdentifier wsu:id="..."
268             ValueType="saml:Assertion"
269             SecurityToken-ef375268
270         </wsse:KeyIdentifier>
271     </wsse:SecurityTokenReference>
272 </ds:KeyInfo>

```

273 The following example extends the previous example with the inclusion of
274 saml:Location and saml:Binding attributes that define the address and protocol to
275 use to acquire the identified assertion at a SAML assertion authority or responder.

```

276 <ds:KeyInfo>
277     <wsse:SecurityTokenReference>
278         <wsse:KeyIdentifier wsu:id="..."
279             ValueType="saml:Assertion"
280             saml:Binding="urn:oasis:names:tc:SAML:1.0:bindings:SOAP-binding"
281             saml:Location="http://www.fabrikam123.com/elliottw1"

```

282
283
284
285

```
SecurityToken-ef375268
</wsse:KeyIdentifier>
</wsse:SecurityTokenReference>
</ds:KeyInfo>
```

286 3.3.3 SAML assertion referenced from SignedInfo

287 Methods to reference SAML assertion from < ds:Reference> elements remain to be
288 formalized. One issue that remains to be resolved is how to differentiate whether it is
289 the reference or the referenced assertion that is to be digested.

290 3.4 Proof-of-Possession of Security Tokens

291 The SAML binding of [WS-Security](#) requires that message senders and receivers
292 support the holder-of-key and sender-vouches methods of subject confirmation. It is
293 strongly RECOMMENDED that an XML signature be used to establish the relationship
294 between the message sender and the attached assertions. This is especially
295 RECOMMENDED whenever the SOAP message exchange is conducted over an
296 unprotected transport.

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

299 The following table enumerates the mandatory subject confirmation methods and
300 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.

301 Note that the high level processing model described in the following sections does
302 not differentiate between message author and message sender as would be
303 necessary to guard against replay attacks. The high-level processing model also does
304 not take into account requirements for authentication of receiver by sender, or for

305 message or assertion confidentiality. These concerns must be addressed by means
306 other than those described in the high-level processing model.

307 **3.4.1 Holder-of-key Subject Confirmation Method**

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

312 **3.4.1.1 Sender**

313 A message sender uses the holder-of-key confirmation method to
314 demonstrate that it is authorized to act as the subject of the assertions in the
315 message. The assertions included in a message that the sender will confirm
316 by the holder-of-key method **MUST** include the following
317 `<saml:SubjectConfirmation>` element:

```
318 <saml:SubjectConfirmation>  
319   <saml:ConfirmationMethod>  
320     urn:oasis:names:tc:SAML:1.0:cm:holder-of-key  
321   </saml:ConfirmationMethod>  
322   <ds:KeyInfo>...</ds:KeyInfo>  
323 </saml:SubjectConfirmation>
```

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

327 To satisfy the associated confirmation method processing of the message
328 receiver, the sender **MUST** demonstrate knowledge of the confirmation key.
329 The sender **MAY** accomplish this by using the confirmation key to sign content
330 within the message and by including the resulting `<ds:Signature>` element in
331 the `<wsse:Security>` header.

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

335 SAML assertions that contain a holder-of-key `<saml:SubjectConfirmation>`
336 element **SHOULD** contain a `<ds:Signature>` element that protects the
337 integrity of the confirmation `<ds:KeyInfo>` established by the assertion
338 authority.

339 The canonicalization method used to produce the `<ds:Signature>`
340 elements used to protect the integrity of SAML assertions **MUST** support the
341 validation of these `<ds:Signature>` elements in contexts (such as
342 `<wsse:Security>` header elements) other than those in which the signatures
343 were calculated.

344 **3.4.1.2 Receiver**

345 Of the SAML assertions it selects for processing, a message receiver **MUST**
346 **NOT** accept assertions containing a holder-of-key

347 <saml:ConfirmationMethod>, unless the receiver has validated the integrity
348 of the assertions and the message sender has demonstrated knowledge of the
349 key identified by the <ds:keyInfo> element of the
350 <saml:SubjectConfirmation> element. If the receiver determines that the
351 sender has demonstrated knowledge of a subject confirmation key, then the
352 SAML assertions containing the confirmation key MAY be attributed to the
353 sender and any elements of the message whose integrity is protected by the
354 subject confirmation key MAY be considered to have been authored by the
355 subject.

356 3.4.1.3 Example

357 The following example illustrates the use of the holder-of-key subject
358 confirmation method to establish the correspondence between the SOAP
359 message author and the subject of the SAML assertions in the
360 <wsse:Security> header:

```
361 <?xml:version="1.0" encoding="UTF-8"?>
362 <S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"
363   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
364   xmlns:xsd="http://www.w3.org/2001/XMLSchema">
365
366   <S:Header>
367     <wsse:Security>
368
369       <saml:Assertion
370         xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"
371         MajorVersion="1" MinorVersion="0"
372         AssertionID="2sxJu9g/vvLG9sAN9bKp/8q0NKU="
373         Issuer="www.example.com"
374         IssueInstant="2002-06-19T16:58:33.173Z">
375         <saml:Conditions
376           NotBefore="2002-06-19T16:53:33.173Z"
377           NotOnOrAfter="2002-06-19T17:08:33.173Z"/>
378
379         <saml:AuthenticationStatement
380           AuthenticationMethod="urn:oasis:names:tc:SAML:1.0:am:password"
381           AuthenticationInstant="2002-06-19T16:57:30.000Z">
382           <saml:Subject>
383             <saml:NameIdentifier
384               NameQualifier="www.example.com"
385               Format="">
386               uid=joe,ou=people,ou=saml-demo,o=example.com
387             </saml:NameIdentifier>
388             <saml:SubjectConfirmation>
389               <saml:ConfirmationMethod>
390                 urn:oasis:names:tc:SAML:1.0:cm:holder-of-key
391               </saml:ConfirmationMethod>
392               <ds:KeyInfo>
393                 <ds:KeyValue>...</ds:KeyValue>
394               </ds:KeyInfo>
395             </saml:SubjectConfirmation>
396           </saml:Subject>
397         </saml:AuthenticationStatement>
398
399         <saml:AttributeStatement>
400           <saml:Subject>
401             <saml:NameIdentifier
402               NameQualifier="www.example.com"
403               Format="">
```

```

404         uid=joe,ou=people,ou=saml-demo,o=baltimore.com
405     </saml:NameIdentifier>
406     <saml:SubjectConfirmation>
407         <saml:ConfirmationMethod>
408             urn:oasis:names:tc:SAML:1.0:cm:holder-of-key
409         </saml:ConfirmationMethod>
410         <ds:KeyInfo>
411             <ds:KeyValue>...</ds:KeyValue>
412         </ds:KeyInfo>
413     </saml:SubjectConfirmation>
414 </saml:Subject>
415
416     <saml:Attribute
417         AttributeName="MemberLevel"
418         AttributeNamespace="http://www.oasis-
419 open.org/Catalyst2002/attributes">
420         <saml:AttributeValue>gold</saml:AttributeValue>
421     </saml:Attribute>
422     <saml:Attribute
423         AttributeName="E-mail"
424         AttributeNamespace="http://www.oasis-
425 open.org/Catalyst2002/attributes">
426         <saml:AttributeValue>joe@yahoo.com</saml:AttributeValue>
427     </saml:Attribute>
428 </saml:AttributeStatement>
429 <ds:Signature>...</ds:Signature>
430 </saml:Assertion>
431
432 <ds:Signature>
433     <ds:SignedInfo>
434         <ds:CanonicalizationMethod Algorithm=
435             "http://www.w3.org/2001/10/xml-exc-c14n#" />
436         <ds:SignatureMethod Algorithm=
437             "http://www.w3.org/2000/09/xmldsig#hmac-sha1" />
438         </ds:Reference>
439         <ds:Reference URI="#MsgBody">
440             <ds:DigestMethod Algorithm=
441                 "http://www.w3.org/2000/09/xmldsig#sha1" />
442             <ds:DigestValue>GyGsF0Pi4xPU...</ds:DigestValue>
443         </ds:Reference>
444     </ds:SignedInfo>
445     <ds:SignatureValue>HJJWbvqW9E84vJVQk...</ds:SignatureValue>
446     <ds:KeyInfo>
447         <wsse:SecurityTokenReference>
448             <saml:AssertionIDReference>2sxJu9g/vvLG9sAN9bKp/8q0NKU=
449             </saml:AssertionIDReference>
450         </wsse:SecurityTokenReference>
451     </ds:KeyInfo>
452 </ds:Signature>
453
454 </wsse:Security>
455 </S:Header>
456
457 <S:Body>
458     <ReportRequest>
459         <TickerSymbol>SUNW</TickerSymbol>
460     </ReportRequest>
461 </S:Body>
462 </S:Envelope>

```

463 3.4.2 Sender-vouches Subject Confirmation Method

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

468 3.4.2.1 Sender

469 A message sender uses the sender-vouches confirmation method to assert
470 that it is acting on behalf of the subjects of the assertions in the message.
471 The assertions included in a message that the sender will confirm by the
472 sender-vouches method **MUST** include the following
473 `<saml:SubjectConfirmation>` element:

```
474 <saml:SubjectConfirmation>  
475   <saml:ConfirmationMethod>  
476     urn:oasis:names:tc:SAML:1.0:cm:sender-vouches  
477   </saml:ConfirmationMethod>  
478 </saml:SubjectConfirmation>
```

479 To satisfy the associated confirmation method processing of the receiver, the
480 sender **MUST** integrity protect the assertions and those elements of the SOAP
481 message that it is vouching for. The sender **MAY** accomplish this by including
482 in the corresponding `<wsse:Security>` header a `<ds:Signature>` element
483 that the sender prepares by using its key to sign the assertions and relevant
484 message content. As defined by the [XML Signature Specification](#), the sender
485 **MAY** identify its key by including a `<ds:KeyInfo>` element within the
486 `<ds:Signature>` element.

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

490 3.4.2.2 Receiver

491 Of the SAML assertions it selects for processing, a message receiver **MUST**
492 **NOT** accept assertions containing a sender-vouches
493 `<saml:ConfirmationMethod>` unless the assertions and SOAP message
494 content being vouched for by the sender are integrity protected by a sender
495 who is trusted by the receiver to act on behalf of the subject of the
496 assertions.

497 3.4.2.3 Example

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

```
502 <?xml:version="1.0" encoding="UTF-8"?>  
503 <S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"  
504   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
505   xmlns:xsd="http://www.w3.org/2001/XMLSchema">
```

```

506
507 <S:Header>
508 <wsse:Security>
509
510   <saml:Assertion
511     xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"
512     MajorVersion="1" MinorVersion="0"
513     AssertionID="2sxJu9g/vvLG9sAN9bKp/8q0NKU="
514     Issuer="www.example.com"
515     IssueInstant="2002-06-19T16:58:33.173Z">
516     <saml:Conditions
517       NotBefore="2002-06-19T16:53:33.173Z"
518       NotOnOrAfter="2002-06-19T17:08:33.173Z"/>
519
520     <saml:AuthenticationStatement
521       AuthenticationMethod="urn:oasis:names:tc:SAML:1.0:am:password"
522       AuthenticationInstant="2002-06-19T16:57:30.000Z">
523       <saml:Subject>
524         <saml:NameIdentifier
525           NameQualifier="www.example.com"
526           Format="">
527           uid=joe,ou=people,ou=saml-demo,o=example.com
528         </saml:NameIdentifier>
529         <saml:SubjectConfirmation>
530           <saml:ConfirmationMethod>
531             urn:oasis:names:tc:SAML:1.0:cm:sender-vouches
532           </saml:ConfirmationMethod>
533         </saml:SubjectConfirmation>
534       </saml:Subject>
535     </saml:AuthenticationStatement>
536
537     <saml:AttributeStatement>
538       <saml:Subject>
539         <saml:NameIdentifier
540           NameQualifier="www.example.com"
541           Format="">
542           uid=joe,ou=people,ou=saml-demo,o=baltimore.com
543         </saml:NameIdentifier>
544         <saml:SubjectConfirmation>
545           <saml:ConfirmationMethod>
546             urn:oasis:names:tc:SAML:1.0:cm:sender-vouches
547           </saml:ConfirmationMethod>
548         </saml:SubjectConfirmation>
549       </saml:Subject>
550
551         <saml:Attribute
552           AttributeName="MemberLevel"
553           AttributeNamespace="http://www.oasis-
554 open.org/Catalyst2002/attributes">
555           <saml:AttributeValue>gold</saml:AttributeValue>
556         </saml:Attribute>
557         <saml:Attribute
558           AttributeName="E-mail"
559           AttributeNamespace="http://www.oasis-
560 open.org/Catalyst2002/attributes">
561           <saml:AttributeValue>joe@yahoo.com</saml:AttributeValue>
562         </saml:Attribute>
563       </saml:AttributeStatement>
564     </saml:Assertion>
565
566   <ds:Signature>
567     <ds:SignedInfo>
568       <ds:CanonicalizationMethod Algorithm=

```



```

569     "http://www.w3.org/2001/10/xml-exc-c14n#"/>
570     <ds:SignatureMethod Algorithm=
571     "http://www.w3.org/2000/09/xmldsig#hmac-sha1"/>
572     <ds:Reference URI=#2sxJu9g/vvLG9sAN9bKp/8q0NKU=
573     Type="saml:IDReferenceType">
574     <ds:DigestMethod Algorithm=
575     "http://www.w3.org/2000/09/xmldsig#sha1"/>
576     <ds:DigestValue>GyGsF0Pi4xPU...</ds:DigestValue>
577     </ds:Reference>
578     <ds:Reference URI="#MsgBody">
579     <ds:DigestMethod Algorithm=
580     "http://www.w3.org/2000/09/xmldsig#sha1"/>
581     <ds:DigestValue>GyGsF0Pi4xPU...</ds:DigestValue>
582     </ds:Reference>
583     </ds:SignedInfo>
584     <ds:SignatureValue>JWbvqW94vJVQkA...</ds:SignatureValue>
585     <ds:KeyInfo>
586     <X509Data>
587     <X509SubjectName>portal@yahoo.com</X509SubjectName>
588     </X509Data>
589     </ds:KeyInfo>
590     </ds:Signature>
591
592 </wsse:Security>
593 </S:Header>
594
595 <S:Body wsu:Id="MsgBody">
596 <ReportRequest>
597 <TickerSymbol>SUNW</TickerSymbol>
598 </ReportRequest>
599 </S:Body>
600
601 </S:Envelope>

```

602 3.5 Error Codes

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

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

Assertion Processing Error	RECOMMENDED Error
A referenced SAML assertion could not be retrieved.	Wsse:SecurityTokenUnavailable
An assertion contains a <code><saml:Condition></code> element that the receiver does not	Wsse:UnsupportedSecurityToken

understand.	
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

614 3.6 Threat Model and Countermeasures

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

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

627 3.6.1 Eavesdropping

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

633 To provide maximum protection from eavesdropping, assertions, assertion
634 references, and sensitive message content SHOULD be encrypted such that only the
635 intended audiences can view their content. This removes threats of eavesdropping in
636 transit, but MAY not remove risks associated with storage or poor handling by the
637 receiver.

638 Transport-layer security MAY be used to protect the message and contained SAML
639 assertions and/or references from eavesdropping while in transport, but message
640 content MUST be encrypted above the transport if it is to be protected from
641 eavesdropping by intermediaries.

642 3.6.2 Replay

643 The reliance on authority protected (e.g. signed) assertions with a holder-of-
644 key subject confirmation mechanism precludes all but a holder of the key

645 from binding the assertions to a SOAP message. Although this mechanism
646 affectively restricts message authorship to the holder of the confirmation key,
647 it does not preclude the capture and resubmission of the message by other
648 parties.

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

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

656 **3.6.3 Message Insertion**

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

659 **3.6.4 Message Deletion**

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

662 **3.6.5 Message Modification**

663 The SAML token binding of WS-Security is protected from message modification if
664 the relevant message content is integrity protected by the holder of the key or by
665 the vouching sender. Therefore, it is strongly RECOMMENDED that all relevant and
666 immutable message content be signed by the holder of the key or by the vouching
667 sender (as the case warrants). Receivers SHOULD only consider those portions of the
668 document that are integrity protected by the appropriate entity as being subject to
669 the assertions in the message.

670 To ensure that message receivers can have confidence that received assertions have
671 not been forged or altered since their issuance, SAML assertions and assertion
672 references appearing in `<wsse:Security>` header elements MUST be integrity
673 protected (e.g. signed) by their issuing authority or the vouching sender (as the case
674 warrants). It is strongly RECOMMENDED that a message sender sign any
675 `<saml:Assertion>` elements that it is confirming and that are not signed by their
676 issuing authority.

677 Transport-layer security MAY be used to protect the message and contained SAML
678 assertions and/or assertion references from modification while in transport, but
679 signatures are required to extend such protection through intermediaries.

680 **3.6.6 Man-in-the-Middle**

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

685

4 Acknowledgements

686

687

688

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

TBD

5 References

689

- 690 **[DIGSI G]** Informational RFC 2828, "[Internet Security Glossary](#)," May
691 2000.
- 692 **[KEYWORDS]** S. Bradner, "Key words for use in RFCs to Indicate Requirement
693 Levels," [RFC 2119](#), Harvard University, March 1997
- 694 **[SAMLBind]** Oasis Committee Specification 01, P. Mishra (Editor) [Bindings
695 and Profiles for the OASIS Security Assertion Markup Language
696 \(SAML\)](#), May 2002.
- 697 **[SAMLCore]** Oasis Committee Specification 01, P. Hallem-Baker, and E.
698 Maler, (Editors), [Assertions and Protocol for the OASIS Security
699 Assertion Markup Language \(SAML\)](#), May 2002.
- 700 **[SAMLReqs]** OASIS Committee Consensus Draft, D. Platt, Evan Prodromou
701 (Editors), [SAML Requirements and Use Cases](#), OASIS,
702 December 2001.
- 703 **[SAMLSecure]** OASIS Committee Specification 01, C. McLaren (Editor),
704 [Security and Privacy Considerations for the OASIS Security
705 Assertion Markup Language \(SAML\)](#) , May 2002.
- 706 **[SOAP]** W3C Note, "[SOAP: Simple Object Access Protocol 1.1](#)," 08 May
707 2000.
- 708 W3C Working Draft, Nilo Mitra (Editor), [SOAP Version 1.2 Part
709 0: Primer](#), June 2002.
- 710 W3C Working Draft, Martin Gudgin, Marc Hadley, Noah
711 Mendelsohn, Jean-Jacques Moreau, Henrik Frystyk Nielsen
712 (Editors), [SOAP Version 1.2 Part 1: Messaging Framework](#), June
713 2002.
- 714 W3C Working Draft, Martin Gudgin, Marc Hadley, Noah
715 Mendelsohn, Jean-Jacques Moreau, Henrik Frystyk Nielsen
716 (Editors), [SOAP Version 1.2 Part 2: Adjuncts](#), June 2002.
- 717 **[URI]** T. Berners-Lee, R. Fielding, L. Masinter, "Uniform Resource
718 Identifiers (URI): Generic Syntax," [RFC 2396](#), MIT/LCS, U.C.
719 Irvine, Xerox Corporation, August 1998.
- 720 **[WS-SAML]** Contribution to the WSS TC, P. Mishra (Editor), [WS-Security
721 Profile of the Security Assertion Markup Language \(SAML\)
722 Working Draft 04](#), Sept 2002.
- 723 **[WS-Security]** TBS – point to the OASIS core draft
- 724 **[XML-ns]** W3C Recommendation, "[Namespaces in XML](#)," 14 January
725 1999.
- 726 **[XML Signature]** W3C Recommendation, "[XML Signature Syntax and
727 Processing](#)," 12 February 2002.

728 **[XML Token]** Contribution to the WSS TC, Chris Kaler (Editor),
729 WS-Security Profile for XML-based Tokens, August 2002.
730

731

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

732

733 Appendix B: Notices

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

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

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

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

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

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