



1

2

# Web Services Security SAML Token Binding

3

4

## Working Draft 04, 9 December 2002

6

### Document identifier:

7

WSS-SAML-04

8

### Location:

9

TBD

10

### Editors:

11

Phillip Hallam-Baker, VeriSign

12

Chris Kaler, Microsoft

13

Ronald Monzillo, Sun

14

Anthony Nadalin, IBM

15

### Contributors:

16

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

17

### Abstract:

18

This document describes how to use Security Assertion Markup Language

19

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

20

### Status:

21

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

22

23

Committee members should send comments on this specification to

24

[wss@lists.oasis-open.org](mailto:wss@lists.oasis-open.org) list. Others should subscribe to and send comments

25

to the [wss-comment@lists.oasis-open.org](mailto:wss-comment@lists.oasis-open.org) list. To subscribe, visit

26

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

27 For information on the disclosure of Intellectual Property Rights or licensing  
28 terms related to the work of the Web Services Security TC please refer to the  
29 Intellectual Property Rights section of the TC web page at [http://www.oasis-  
30 open.org/committees/wss/](http://www.oasis-open.org/committees/wss/). The OASIS policy on Intellectual Property Rights  
31 is described at <http://www.oasis-open.org/who/intellectualproperty.shtml>.

32

## 32 **Table of Contents**

33	1	Introduction .....	5
34	1.1	Goals and Requirements .....	5
35	1.1.1	Requirements .....	5
36	1.1.2	Non-Goals .....	5
37	2	Notations and Terminology .....	6
38	2.1	Notational Conventions .....	6
39	2.2	Namespaces .....	6
40	2.3	Terminology .....	7
41	3	Usage.....	8
42	3.1	Processing Model .....	8
43	3.2	Attaching Security Tokens .....	8
44	3.3	Identifying and Referencing Security Tokens .....	9
45	3.4	Proof-of-Possession of Security Tokens.....	10
46	3.5	Error Codes .....	11
47	3.6	Threat Model and Countermeasures .....	18
48	4	Acknowledgements .....	20
49	5	References.....	21
50		Appendix A: Revision History .....	23
51		Appendix B: Notices.....	24
52			

---

## 53 **1 Introduction**

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

### 59 **1.1 Goals and Requirements**

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

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

#### 63 **1.1.1 Requirements**

64 TBS

#### 65 **1.1.2 Non-Goals**

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

67 TBS

68

---

## 69 2 Notations and Terminology

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

### 72 2.1 Notational Conventions

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

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

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

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

### 85 2.2 Namespaces

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

89 `http://schemas.xmlsoap.org/ws/2002/xx/secext`  
90 `http://schemas.xmlsoap.org/ws/2002/xx/utility`

91 The following namespaces are used in this document:

92

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
-------	---------------------------------------

93 **2.3 Terminology**

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

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

98 [TBS]

99

## 3 Usage

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

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

103 **Contact information:** TBD

104 **Description:** Given below.

105 **Updates:** None.

### 3.1 Processing Model

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

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

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

### 3.2 Attaching Security Tokens

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

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

140  
141  
142  
143  
144  
145  
146

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

### 147 3.3 Identifying and Referencing Security Tokens

148 The [WS-Security](#) specification defines the `<wsse:SecurityTokenReference>` element  
149 for referencing security tokens. Three forms of token references are defined:

- 150 • An element reference – a security token specific XML element that contains an  
151 identifier and perhaps locator of a security token within the message or at some  
152 external location.
- 153 • A URI reference – a generic element that conveys in its attributes, the security  
154 token URI and token type value (i.e. `ValueType`) that define the location and  
155 perhaps identifier of a security token occurring either within the message or at  
156 some external location. A URI containing only a fragment identifier is interpreted  
157 as identifying the corresponding security token within the message in which the  
158 fragment identifier occurs.
- 159 • A key identifier reference – a generic element that conveys in its attributes, the  
160 security token identifier (i.e. `wsu:id`) and token type value (i.e. `ValueType`) that  
161 identifies a security token with matching `wsu:id` and `ValueType` occurring within  
162 a `<wsse:Security>` header of the message. Identifier references may only be  
163 used to reference security tokens that carry matching attributes, which  
164 approximately restricts their use to Binary Security Tokens attributed as a result  
165 of their encapsulation in XML.

166 A URI reference containing a URL may be combined with a token specific element  
167 reference to yield a location qualified reference.

168 In The SAML binding of [WS-security](#), a referenced SAML assertion is identified by a  
169 `<saml:AssertionIDReference>` occurring either as an element reference or as a  
170 String value fragment identifier in a URI reference.

#### 171 3.3.1 SAML Assertion Reference Elements

172 A `<wsse:SecurityTokenReference>` containing a `<saml:AssertionIDReference>`  
173 element containing a SAML assertion identifier may be used to reference a SAML  
174 assertion occurring within the `<wsse:Security>` header of the SOAP message in  
175 which the reference occurs. The following example illustrates the use of a  
176 `<wsse:securityTokenReference>` containing a `<saml:AssertionIDReference>`  
177 within the `<keyInfo>` of an [XML Signature](#) element to reference the SAML assertion  
178 (in the `<wsse:Security>` header) that contains the key used to compute the  
179 signature.

180  
181  
182  
183  
184

```
<S:Envelope xmlns:S="...">  
  <S:Header>  
    <wsse:Security xmlns:wsse="...">  
      <saml:Assertion  
        MajorVersion="1"
```



```

185         MinorVersion="0"
186         AssertionID="SecurityToken-ef375268"
187         Issuer="elliottw1"
188         IssueInstant="2002-07-23T11:32:05.6228146-07:00"
189         xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion">
190         ...
191     </saml:Assertion>
192     <ds:Signature xmlns:ds="...">
193     ...
194     <ds:KeyInfo>
195         <wsse:SecurityTokenReference>
196             <saml:AssertionIDReference>
197                 SecurityToken-ef375268
198             </saml:AssertionIDReference>
199         </wsse:SecurityTokenReference>
200     </ds:KeyInfo>
201 </ds:Signature>
202     ...
203 </wsse:Security>
204 </S:Header>
205 <S:Body>
206     ...
207 </S:Body>
208 </S:Envelope>

```

### 209 3.3.2 URI References to SAML assertions

210 As depicted in the following example, a URI reference containing only a fragment  
211 identifier consisting of a `<saml:AssertionIDReference>` may be used to reference a  
212 SAML assertion occurring within the `<wsseSecurity>` header of the SOAP message  
213 in which the reference occurs. A URI reference containing an XML path expression  
214 can be used to reference a SAML assertion occurring anywhere within the containing  
215 SOAP message.

```

216 <wsse:SecurityTokenReference>
217   <wsse:Reference URI="#SecurityToken-ef375268"
218                 ValueType="saml:IDReferenceType">
219   </wsse:Reference>
220 </wsse:SecurityTokenReference>

```

221 The following example demonstrates the use of a URI reference in conjunction with a  
222 `<saml:AssertionIDReference>` to define the location of the SAML responder at  
223 which the identified assertion may be obtained.

```

224 <wsse:SecurityTokenReference>
225   <saml:AssertionIDReference>SecurityToken-ef375268
226 </saml:AssertionIDReference>
227   <wsse:Reference URI="http://www.fabrikam123.com/elliottw1"
228 </wsse:Reference>
229 </wsse:SecurityTokenReference>

```

### 230 3.3.3 Identifier References to SAML Assertions

231 SAML assertions may not be referenced by identifier references because the  
232 `<saml:Assertion>` element schema does not include the `wsu:id` and `ValueType`  
233 attributes.

234 **3.4 Proof-of-Possession of Security Tokens**

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

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

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

<b>Mechanism</b>	<b>RECOMMENDED Processing Rules</b>
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.

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

252 **3.4.1 Holder-of-key Subject Confirmation Method**

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

### 256 3.4.1.1 Sender

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

```
261 <saml:SubjectConfirmation>  
262   <saml:ConfirmationMethod>  
263     urn:oasis:names:tc:SAML:1.0:cm:holder-of-key  
264   </saml:ConfirmationMethod>  
265   <ds:KeyInfo>...</ds:KeyInfo>  
266 </saml:SubjectConfirmation>
```

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

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

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

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

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

### 285 3.4.1.2 Receiver

286 Of the SAML assertions it selects for processing, a message receiver SHOULD NOT  
287 accept assertions containing a holder-of-key `<saml:ConfirmationMethod>`, unless  
288 the assertions are signed and validated as described above and the message sender  
289 has demonstrated knowledge of the key identified by the `<ds:keyInfo>` element of  
290 the `<saml:SubjectConfirmation>` element. If the receiver determines that the  
291 sender has demonstrated knowledge of a subject confirmation key, then the SAML  
292 assertions containing the confirmation key MAY be attributed to the sender and any  
293 elements of the message whose integrity is protected by the subject confirmation  
294 key MAY be considered to have been authored by the subject.

### 295 3.4.1.3 Example

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

```

299 <?xml:version="1.0" encoding="UTF-8"?>
300 <S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"
301   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
302   xmlns:xsd="http://www.w3.org/2001/XMLSchema">
303
304 <S:Header>
305 <wsse:Security>
306   <saml:Assertion
307     xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"
308     MajorVersion="1" MinorVersion="0"
309     AssertionID="2sxJu9g/vvLG9sAN9bKp/8q0NKU="
310     Issuer="www.example.com"
311     IssueInstant="2002-06-19T16:58:33.173Z">
312     <saml:Conditions
313       NotBefore="2002-06-19T16:53:33.173Z"
314       NotOnOrAfter="2002-06-19T17:08:33.173Z"/>
315
316     <saml:AuthenticationStatement
317       AuthenticationMethod="urn:oasis:names:tc:SAML:1.0:am:password"
318       AuthenticationInstant="2002-06-19T16:57:30.000Z">
319       <saml:Subject>
320         <saml:NameIdentifier
321           NameQualifier="www.example.com"
322           Format="">
323           uid=joe,ou=people,ou=saml-demo,o=example.com
324         </saml:NameIdentifier>
325         <saml:SubjectConfirmation>
326           <saml:ConfirmationMethod>
327             urn:oasis:names:tc:SAML:1.0:cm:holder-of-key
328           </saml:ConfirmationMethod>
329           <ds:KeyInfo>
330             <ds:KeyValue>...</ds:KeyValue>
331           </ds:KeyInfo>
332         </saml:SubjectConfirmation>
333       </saml:Subject>
334     </saml:AuthenticationStatement>
335
336     <saml:AttributeStatement>
337       <saml:Subject>
338         <saml:NameIdentifier
339           NameQualifier="www.example.com"
340           Format="">
341           uid=joe,ou=people,ou=saml-demo,o=baltimore.com
342         </saml:NameIdentifier>
343         <saml:SubjectConfirmation>
344           <saml:ConfirmationMethod>
345             urn:oasis:names:tc:SAML:1.0:cm:holder-of-key
346           </saml:ConfirmationMethod>
347           <ds:KeyInfo>
348             <ds:KeyValue>...</ds:KeyValue>
349           </ds:KeyInfo>
350         </saml:SubjectConfirmation>
351       </saml:Subject>
352
353       <saml:Attribute
354         AttributeName="MemberLevel"
355         AttributeNamespace="http://www.oasis-
356 open.org/Catalyst2002/attributes">
357         <saml:AttributeValue>gold</saml:AttributeValue>
358       </saml:Attribute>
359     </saml:AttributeStatement>
360   </saml:Assertion>

```

```

361         AttributeNamespace="http://www.oasis-
362 open.org/Catalyst2002/attributes">
363         <saml:AttributeValue>joe@yahoo.com</saml:AttributeValue>
364     </saml:Attribute>
365 </saml:AttributeStatement>
366 <ds:Signature>...</ds:Signature>
367 </saml:Assertion>
368 <ds:Signature>
369 <ds:SignedInfo>...</ds:SignedInfo>
370 <ds:SignatureValue>HJJWbvqW9E84vJVQk...</ds:SignatureValue>
371 <ds:KeyInfo>
372 <wsse:SecurityTokenReference>
373 <saml:AssertionIDReference>"2sxJu9g/vvLG9sAN9bKp/8q0NKU="
374 </saml:AssertionIDReference>
375 </wsse:SecurityTokenReference>
376 </ds:KeyInfo>
377 </ds:Signature>
378 </wsse:Security>
379 </S:Header>
380
381 <S:Body>
382 <ReportRequest>
383 <TickerSymbol>SUNW</TickerSymbol>
384 </ReportRequest>
385 </S:Body>
386 </S:Envelope>

```

### 387 3.4.2 Sender-vouches Subject Confirmation Method

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

#### 391 3.4.2.1 Sender

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

```

396 <saml:SubjectConfirmation>
397 <saml:ConfirmationMethod>
398     urn:oasis:names:tc:SAML:1.0:cm:sender-vouches
399 </saml:ConfirmationMethod>
400 </saml:SubjectConfirmation>

```

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

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

### 412 3.4.2.2 Receiver

413 Of the SAML assertions it selects for processing, a message receiver SHOULD NOT  
414 accept assertions containing a sender-vouches <saml:ConfirmationMethod> unless  
415 the assertions and SOAP message content being vouched for by the sender are  
416 integrity protected by a sender who is trusted by the receiver to act on behalf of the  
417 subject of the assertions.

### 418 3.4.2.3 Example

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

```
423 <?xml:version="1.0" encoding="UTF-8"?>
424 <S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"
425   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
426   xmlns:xsd="http://www.w3.org/2001/XMLSchema">
427
428   <S:Header>
429     <wsse:Security>
430       <saml:Assertion
431         xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"
432         MajorVersion="1" MinorVersion="0"
433         AssertionID="2sxJu9g/vvLG9sAN9bKp/8q0NKU="
434         Issuer="www.example.com"
435         IssueInstant="2002-06-19T16:58:33.173Z">
436         <saml:Conditions
437           NotBefore="2002-06-19T16:53:33.173Z"
438           NotOnOrAfter="2002-06-19T17:08:33.173Z"/>
439
440         <saml:AuthenticationStatement
441           AuthenticationMethod="urn:oasis:names:tc:SAML:1.0:am:password"
442           AuthenticationInstant="2002-06-19T16:57:30.000Z">
443           <saml:Subject>
444             <saml:NameIdentifier
445               NameQualifier="www.example.com"
446               Format="">
447               uid=joe,ou=people,ou=saml-demo,o=example.com
448             </saml:NameIdentifier>
449             <saml:SubjectConfirmation>
450               <saml:ConfirmationMethod>
451                 urn:oasis:names:tc:SAML:1.0:cm:sender-vouches
452               </saml:ConfirmationMethod>
453             </saml:SubjectConfirmation>
454           </saml:Subject>
455           </saml:AuthenticationStatement>
456
457           <saml:AttributeStatement>
458             <saml:Subject>
459               <saml:NameIdentifier
460                 NameQualifier="www.example.com"
461                 Format="">
462                 uid=joe,ou=people,ou=saml-demo,o=baltimore.com
463               </saml:NameIdentifier>
464               <saml:SubjectConfirmation>
465                 <saml:ConfirmationMethod>
466                   urn:oasis:names:tc:SAML:1.0:cm:sender-vouches
467                 </saml:ConfirmationMethod>
```

```

468     </saml:SubjectConfirmation>
469 </saml:Subject>
470
471     <saml:Attribute
472       AttributeName="MemberLevel"
473       AttributeNamespace="http://www.oasis-
474 open.org/Catalyst2002/attributes">
475       <saml:AttributeValue>gold</saml:AttributeValue>
476     </saml:Attribute>
477     <saml:Attribute
478       AttributeName="E-mail"
479       AttributeNamespace="http://www.oasis-
480 open.org/Catalyst2002/attributes">
481       <saml:AttributeValue>joe@yahoo.com</saml:AttributeValue>
482     </saml:Attribute>
483 </saml:AttributeStatement>
484 </saml:Assertion>
485 <ds:Signature>
486   <ds:SignedInfo>
487     <ds:CanonicalizationMethod Algorithm=
488       "http://www.w3.org/2001/10/xml-exc-c14n#" />
489     <ds:SignatureMethod Algorithm=
490       "http://www.w3.org/2000/09/xmldsig#hmac-sha1" />
491     <ds:Reference URI="#2sxJu9g/vvLG9sAN9bKp/8q0NKU="
492       Type="saml:IDReferenceType">
493       <ds:DigestMethod Algorithm=
494         "http://www.w3.org/2000/09/xmldsig#sha1" />
495       <ds:DigestValue>GyGsF0Pi4xPU...</ds:DigestValue>
496     </ds:Reference>
497     <ds:Reference URI="#MsgBody">
498       <ds:DigestMethod Algorithm=
499         "http://www.w3.org/2000/09/xmldsig#sha1" />
500       <ds:DigestValue>GyGsF0Pi4xPU...</ds:DigestValue>
501     </ds:Reference>
502   </ds:SignedInfo>
503   <ds:SignatureValue>JWbvqW94vJVQkA...</ds:SignatureValue>
504   <ds:KeyInfo>
505     <X509Data>
506       <X509SubjectName>portal@yahoo.com</X509SubjectName>
507     </X509Data>
508   </ds:KeyInfo>
509 </ds:Signature>
510 </wsse:Security>
511 </S:Header>
512
513 <S:Body wsu:Id="MsgBody">
514   <ReportRequest>
515     <TickerSymbol>SUNW</TickerSymbol>
516   </ReportRequest>
517 </S:Body>
518
519 </S:Envelope>

```

## 520 3.5 Error Codes

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

526 A receiver that is unable to process the SAML assertions contained in a  
 527 <wsse:Security> header SHOULD use one of the fault codes listed in the core WS-  
 528 Security specification to report the error. The RECOMMENDED correspondence  
 529 between the common assertion processing failures and the error codes defined in the  
 530 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

## 531 **3.6 Threat Model and Countermeasures**

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

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

### 543 **3.6.1 Eavesdropping**

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

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



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

### 556 **3.6.2 Replay**

557 The reliance on authority signed assertions with a holder-of-key subject confirmation  
558 mechanism precludes all but a holder of the key from binding the assertions to a  
559 SOAP message. Although this mechanism affectively restricts message authorship to  
560 the holder of the confirmation key, it does not preclude the capture and resubmission  
561 of the message by other parties.

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

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

### 569 **3.6.3 Message Insertion**

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

### 572 **3.6.4 Message Deletion**

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

### 575 **3.6.5 Message Modification**

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

582 SAML assertions appearing in `<wsse:Security>` header elements SHOULD be signed  
583 by their issuing authority so that message receivers can have confidence that the  
584 assertions have not been forged or altered since their issuance. It is strongly  
585 RECOMMENDED that a message sender sign any `<saml:Assertion>` elements that it  
586 is confirming and that are not signed by their issuing authority.

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

591 **3.6.6 Man-in-the-Middle**

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

---

596 **4 Acknowledgements**

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

599 TBD

---

## 5 References

- 600
- 601     **[ DIGSIG ]**     Informational RFC 2828, "[Internet Security Glossary](#)," May  
602                     2000.
- 603     **[ KEYWORDS ]**    S. Bradner, "Key words for use in RFCs to Indicate Requirement  
604                     Levels," [RFC 2119](#), Harvard University, March 1997
- 605     **[ SAMLBind ]**     Oasis Committee Specification 01, P. Mishra (Editor) [Bindings  
606                     and Profiles for the OASIS Security Assertion Markup Language  
607                     \(SAML\)](#), May 2002.
- 608     **[ SAMLCore ]**     Oasis Committee Specification 01, P. Hallem-Baker, and E.  
609                     Maler, (Editors), [Assertions and Protocol for the OASIS Security  
610                     Assertion Markup Language \(SAML\)](#), May 2002.
- 611     **[ SAMLReqs ]**     OASIS Committee Consensus Draft, D. Platt, Evan Prodromou  
612                     (Editors), [SAML Requirements and Use Cases](#), OASIS,  
613                     December 2001.
- 614     **[ SAMLSecure ]**   OASIS Committee Specification 01, C. McLaren (Editor),  
615                     [Security and Privacy Considerations for the OASIS Security  
616                     Assertion Markup Language \(SAML\)](#) , May 2002.
- 617     **[ SOAP ]**         W3C Note, "[SOAP: Simple Object Access Protocol 1.1](#)," 08 May  
618                     2000.
- 619                     W3C Working Draft, Nilo Mitra (Editor), [SOAP Version 1.2 Part  
620                     0: Primer](#), June 2002.
- 621                     W3C Working Draft, Martin Gudgin, Marc Hadley, Noah  
622                     Mendelsohn, Jean-Jacques Moreau, Henrik Frystyk Nielsen  
623                     (Editors), [SOAP Version 1.2 Part 1: Messaging Framework](#), June  
624                     2002.
- 625                     W3C Working Draft, Martin Gudgin, Marc Hadley, Noah  
626                     Mendelsohn, Jean-Jacques Moreau, Henrik Frystyk Nielsen  
627                     (Editors), [SOAP Version 1.2 Part 2: Adjuncts](#), June 2002.
- 628     **[ URI ]**         T. Berners-Lee, R. Fielding, L. Masinter, "Uniform Resource  
629                     Identifiers (URI): Generic Syntax," [RFC 2396](#), MIT/LCS, U.C.  
630                     Irvine, Xerox Corporation, August 1998.
- 631     **[ WS-SAML ]**     Contribution to the WSS TC, P. Mishra (Editor), [WS-Security  
632                     Profile of the Security Assertion Markup Language \(SAML\)  
633                     Working Draft 04](#), Sept 2002.
- 634     **[ WS-Security ]**   TBS – point to the OASIS core draft
- 635     **[ XML-ns ]**       W3C Recommendation, "[Namespaces in XML](#)," 14 January  
636                     1999.
- 637     **[ XML Signature ]** W3C Recommendation, "[XML Signature Syntax and  
638                     Processing](#)," 12 February 2002.

639        **[XML Token]**      Contribution to the WSS TC, Chris Kaler (Editor),  
640                              WS-Security Profile for XML-based Tokens, August 2002.  
641

642

---

## 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

643

644

## Appendix B: Notices

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

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

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

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

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

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