



Web Services Security: SAML Token Profile 1.1

Working Draft ~~042~~, ~~1317~~ ~~June~~~~May~~ 2005

Document ~~Location~~identifier:

~~oasis-wss-saml-token-profile-1.1(PDF)(Word)~~

~~Location:~~

~~http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1.pdf~~

Errata Location:

~~http://www.oasis-open.org/committees/wss~~

Technical Committee:

Web Services Security (WSS)

Chairs:

<u>Kelvin Lawrence</u>	<u>IBM</u>
<u>Chris Kaler</u>	<u>Microsoft</u>

Editors:

Ronald Monzillo	Sun
Chris Kaler	Microsoft
Anthony Nadalin	IBM
Phillip Hallam-Baker	VeriSign

Abstract:

~~Contributors (voting members of the WSS TC as of Sept 8, 2004)~~

Gene Thurston	AmberPoint
Frank Siebenlist	Argonne National Laboratory
Hal Lockhart	BEA Systems, Inc.
Corinna Witt	BEA Systems, Inc.
Merlin Hughes	Betrusted (Baltimore Technologies)
Davanum Srinivas	Computer Associates
Thomas DeMartini	ContentGuard
Guillermo Lao	ContentGuard
Sam Wei	Documentum

32	Tim Moses	Entrust
33	Dana Kaufman	Forum Systems, Inc.
34	Toshihiro Nishimura	Fujitsu
35	Kefeng Chen	GeoTrust
36	Irving Reid	Hewlett-Packard
37	Kojiro Nakayama	Hitachi
38	Paula Austel	IBM
39	Derek Fu	IBM
40	Maryann Hondo	IBM
41	Kelvin Lawrence	IBM (TC Chair)
42	Michael McIntosh	IBM
43	Anthony Nadalin	IBM
44	Nataraj Nagaratnam	IBM
45	Ron Williams	IBM
46	Don Flinn	Individual
47	Bob Morgan	Internet2
48	Kate Cherry	Lockheed Martin
49	Paul Cotton	Microsoft Corporation
50	Vijay Gajjala	Microsoft Corporation
51	Alan Geller	Microsoft Corporation
52	Chris Kaler	Microsoft Corporation (TC Chair)
53	Rich Levinson	Netegrity, Inc.
54	Prateek Mishra	Netegrity, Inc.
55	Frederick Hirsch	Nokia
56	Senthil Sengodan	Nokia
57	Abbie Barbir	Nortel Networks
58	Lloyd Burch	Novell
59	Charles Knouse	Obliv
60	Steve Anderson	OpenNetwork (Secretary)
61	Vamsi Motukuru	Oracle
62	Ramana Turlapati	Oracle
63	Ben Hammond	RSA Security
64	Andrew Nash	RSA Security
65	Rob Philpott	RSA Security
66	Martijn de Boer	SAP
67	Blake Dournaee	Sarvega
68	Goumara Radja	Sarvega
69	Pete Wenzel	SeeBeyond Technology Corporation
70	Jeff Hodges	Sun Microsystems
71	Ronald Monzillo	Sun Microsystems
72	Jan Alexander	Systinet
73	Symon Chang	Tibco
74	J Weiland	US Dept of the Navy
75	Phillip Hallam Baker	Verisign
76	Maneesh Sahu	Westbridge Technology
77	Contributors of input Documents (if not already listed above):	
78	Hiroshi Maruyama	IBM
79	Chris McLaren	Netegrity
80	Jerry Schwarz	Oracle

81 ~~Eve Maler~~ ~~Sun Microsystems~~
82 ~~Hemma Prafullchandra~~ ~~VeriSign~~

83 **Abstract:**

84 This document describes how to use Security Assertion Markup Language
85 (SAML) V1.1 and V2.0 assertions with the Web Services Security (WSS):
86 SOAP Message Security V1.1 specification.

87 With respect to the description of the use of SAML V1.1, this document
88 subsumes and is totally consistent with the Web Services Security: SAML
89 Token Profile 1.0.

90 **Status:**

91 This is a working draft. Please send comments to the editors.

92 Committee members should send comments on this specification to
93 wss@lists.oasis-open.org list. Others should subscribe to and send comments
94 to the wss-comment@lists.oasis-open.org list. To subscribe, visit
95 <http://lists.oasis-open.org/ob/adm.pl>.

96
97 For information on the disclosure of Intellectual Property Rights or licensing
98 terms related to the work of the Web Services Security TC please refer to the
99 Intellectual Property Rights section of the TC web page at
100 **<http://www.oasis-open.org/committees/wss/>**. The OASIS policy on
101 Intellectual Property Rights is described at
102 **<http://www.oasis-open.org/who/intellectualproperty.shtml>**.

103

103 **Notices**

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

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

118 Copyright © The Organization for the Advancement of Structured Information Standards [OASIS]
119 2002-2005. All Rights Reserved.

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

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

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

137	Table of Contents	
138	1 Introduction.....	6
139	1.1 Goals.....	6
140	1.1.1 Non-Goals	6
141	2 Notations and Terminology	7
142	2.1 Notational Conventions	7
143	2.2 Namespaces	7
144	2.3 Terminology	8
145	3 Usage	9
146	3.1 Processing Model.....	9
147	3.2 SAML Version Differences	9
148	3.2.1 Assertion Identifier	10
149	3.2.2 Relationship of Subjects to Statements.....	10
150	3.2.3 Assertion URI Reference Replaces AuthorityBinding	12
151	3.2.4 Attesting Entity Identifier	12
152	3.3 Attaching Security Tokens	12
153	3.4 Identifying and Referencing Security Tokens	13
154	3.4.1 SAML Assertion Referenced from Header or Element.....	16
155	3.4.2 SAML Assertion Referenced from KeyInfo.....	18 17
156	3.4.3 SAML Assertion Referenced from SignedInfo	19
157	3.4.4 SAML Assertion Referenced from Encrypted Data Reference	20
158	3.5 Subject Confirmation of SAML Assertions	21
159	3.5.1 Holder-of-key Subject Confirmation Method	22
160	3.5.2 Sender-vouches Subject Confirmation Method	26
161	3.6 Error Codes	31 30
162	4 Threat Model and Countermeasures (non-normative).....	33 32
163	4.1 Eavesdropping	33 32
164	4.2 Replay	33 32
165	4.3 Message Insertion	34 33
166	4.4 Message Deletion	34 33
167	4.5 Message Modification	34 33
168	4.6 Man-in-the-Middle	34 33
169	5 References	35 34
170	Appendix A: Revision History	37 36

171 Appendix B: Notices 3937 |
172

173

1 Introduction

174 The WSS: SOAP Message Security specification defines a standard set of SOAP
175 extensions that implement SOAP message authentication and encryption. This
176 specification defines the use of Security Assertion Markup Language (SAML)
177 assertions as security tokens from the `<wsse:Security>` header block defined by the
178 WSS: SOAP Message Security specification.

1.1 Goals

180 The goal of this specification is to define the use of SAML V1.1 and V2.0 assertions in
181 the context of WSS: SOAP Message Security including for the purpose of securing
182 SOAP messages and SOAP message exchanges. To achieve this goal, this profile
183 describes how:

- 184 1. SAML assertions are carried in and referenced from
185 `<wsse:securitywsse:Security>` Headers.
- 186 2. SAML assertions are used with XML signature to bind the subjects and statements
187 of the assertions (i.e., the claims) to a SOAP message.

1.1.1 Non-Goals

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

- 190 1. Defining SAML statement syntax or semantics.
- 191 2. Describing the use of SAML assertions other than for SOAP Message Security.
- 192 3. Describing the use of SAML V1.0 assertions with the Web Services Security
193 (WSS): SOAP Message Security specification.

194 2 Notations and Terminology

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

197 2.1 Notational Conventions

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

201 This document uses the notational conventions defined in the WS-Security SOAP
202 Message Security document.

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

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

210 Readers are presumed to be familiar with the terms in the Internet Security
211 Glossary.

212 2.2 Namespaces

213 The appearance of the following [XML-ns] namespace prefixes in the examples within
214 this specification should be understood to refer to the corresponding namespaces
215 (from the following table) whether or not an XML namespace declaration appears in
216 the example:

Prefix	
S11	http://schemas.xmlsoap.org/soap/envelope/
S12	http://www.w3.org/2003/05/soap-envelope
ds	http://www.w3.org/2000/09/xmldsig#
xenc	http://www.w3.org/2001/04/xmlenc
wsse	http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-01.xsd
wsse11	TBD

wsu	<code>http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd</code>
saml	<code>urn: oasis:names:tc:SAML:1.0:assertion</code>
saml2	<code>urn: oasis:names:tc:SAML:2.0:assertion</code>
samlp	<code>urn: oasis:names:tc:SAML:1.0:protocol</code>

217 **Table-1 Namespace Prefixes**

218 **2.3 Terminology**

219 This specification employs the terminology defined in the WSS: SOAP Message
 220 Security specification. The definitions for additional terminology used in this
 221 specification appear below.

222

223 Attesting Entity – the entity that provides the confirmation evidence that will be used
 224 to establish the correspondence between the subjects and claims of SAML
 225 statements (in SAML assertions) and SOAP message content.

226

227 Confirmation Method Identifier – the value within a SAML `SubjectConfirmation`
 228 element that identifies the subject confirmation process to be used with the
 229 corresponding statements.

230

231 Subject Confirmation – the process of establishing the correspondence between the
 232 subject and claims of SAML statements (in SAML assertions) and SOAP message
 233 content by verifying the confirmation evidence provided by an attesting entity.

234

235 SAML Assertion Authority - An ~~abstract~~ *system entity* that issues *assertions*.

236

237 Subject – A representation of the entity to which the claims in one or more SAML
 238 statements apply.

239 **3 Usage**

240 This section defines the specific mechanisms and procedures for using SAML
241 assertions as security tokens.

242 **3.1 Processing Model**

243 This specification extends the token-independent processing model defined by the
244 WSS: SOAP Message Security specification.

245 When a receiver processes a `<wsse:Security>` header containing or referencing
246 SAML assertions, it selects, based on its policy, the signatures and assertions that it
247 will process. It is assumed that a receiver's signature selection policy MAY rely on
248 semantic labeling¹ of `<wsse:SecurityTokenReference>` elements occurring in the
249 `<ds:KeyInfo>` elements within the signatures. It is also assumed that the assertions
250 selected for validation and processing will include those referenced from the
251 `<ds:KeyInfo>` and `<ds:SignedInfo>` elements of the selected signatures.

252 As part of its validation and processing of the selected assertions, the receiver MUST²
253 establish the relationship between the subject and claims of the SAML statements (of
254 the referenced SAML assertions) and the entity providing the evidence to satisfy the
255 confirmation method defined for the statements (i.e., the attesting entity). Two
256 methods for establishing this correspondence, `holder-of-key` and `sender-vouches`
257 are described below. Systems implementing this specification MUST implement the
258 processing necessary to support both of these subject confirmation methods.

259 **3.2 SAML Version Differences**

260 The following sub-sections describe the differences between SAML V1.1 and V2.0
261 that apply to this specification.

¹ The optional `Usage` attribute of the `<wsse:SecurityTokenReference>` element MAY be used to associate one or more semantic usage labels (as URIs) with a reference and thus use of a Security Token. Please refer to WSS: SOAP Message Security for the details of this attribute.

² When the confirmation method is `urn:oasis:names:tc:SAML:1.0:cm:bearer`, proof of the relationship between the attesting entity and the subject of the statements in the assertion is implicit and no steps need be taken by the receiver to establish this relationship.

262 3.2.1 Assertion Identifier

263 In SAML V1.1 the name of the assertion identifier attribute is "AssertionID". In SAML
264 v2.0 the name of the assertion identifier attribute is "ID". In both versions the type
265 of the identifier attribute is `xs:ID`.

266 3.2.2 Relationship of Subjects to Statements

267 A SAML assertion contains a collection of 0 or more statements. In SAML V1.1, a
268 separate subject with separate subject confirmation methods may be specified for
269 each statement of an assertion. In SAML V2.0, at most one subject and at most one
270 set of subject confirmation methods may be specified for all the statements of the
271 assertion. These distinctions are described in more detail by the following
272 paragraphs.

273 A SAML V1.1 statement that contains a `<saml:Subject>` element (i.e., a subject
274 statement) may contain a `<saml:SubjectConfirmation>` element that defines the
275 rules for confirming the subject and claims of the statement. If present, the
276 `<saml:SubjectConfirmation>` element occurs within the subject element, and
277 defines one or more methods (i.e., `<saml:ConfirmationMethod>` elements) by which
278 the statement may be confirmed and will include a `<ds:KeyInfo>`³ element when any
279 of the specified methods are based on demonstration of a confirmation key. The
280 `<saml:SubjectConfirmation>` element also provides for the inclusion of additional
281 information to be applied in the confirmation method processing via the optional
282 `<saml:SubjectConfirmationData>` element. The following example depicts a SAML
283 V1.1 assertion containing two subject statements with different subjects and
284 different subject confirmation elements.

```
285 <saml:Assertion  
286   ...  
287   <saml:SubjectStatement>  
288     <saml:Subject>  
289       <saml:NameIdentifier  
290         ...  
291       </saml:NameIdentifier>  
292       <saml:SubjectConfirmation>  
293         <saml:ConfirmationMethod>  
294           urn:oasis:names:tc:SAML:1.0:cm:sender-vouches  
295         </saml:ConfirmationMethod>  
296         <saml:ConfirmationMethod>  
297           urn:oasis:names:tc:SAML:1.0:cm:holder-of-key  
298         </saml:ConfirmationMethod>  
299         <ds:KeyInfo>  
300           <ds:KeyValue>...</ds:KeyValue>  
301         </ds:KeyInfo>  
302       </saml:SubjectConfirmation>  
303     </saml:Subject>  
304     ...  
305   </saml:SubjectStatement>  
306   <saml:SubjectStatement>
```

³ When a `<ds:KeyInfo>` element is specified, it identifies the key that applies to all the key confirmed methods of the confirmation element.

```

307     <saml:Subject>
308         <saml:NameIdentifier
309             ...
310         </saml:NameIdentifier>
311         <saml:SubjectConfirmation>
312             <saml:ConfirmationMethod>
313                 urn:oasis:names:tc:SAML:1.0:cm:sender-vouches
314             </saml:ConfirmationMethod>
315         </saml:SubjectConfirmation>
316     </saml:Subject>
317     ...
318 </saml:SubjectStatement>
319 ...
320 </saml:Assertion>

```

321 A SAML V2.0 assertion may contain a single `<saml2:Subject>` that applies to all the
322 statements of the assertion. When a subject is included in A SAML V2.0 assertion, it
323 may contain any number of `<saml2:SubjectConfirmation>` elements, satisfying any
324 of which is sufficient to confirm the subject and all the statements of the assertion.
325 Each `<saml2:SubjectConfirmation>` element identifies a single confirmation
326 method (by attribute value) and may include an optional
327 `<saml2:SubjectConfirmationData>` element that is used to specify optional
328 confirmation method independent condition attributes and to define additional
329 method specific confirmation data. In the case of a key dependent confirmation
330 method, a `<saml2:KeyInfoConfirmationDataType>` that includes 1 or more
331 `<ds:KeyInfo>` elements is included as `<saml2:SubjectConfirmationData>`. In this
332 case, each `<ds:KeyInfo>` element identifies a key that may be demonstrated to
333 confirm the assertion. The following example depicts a SAML V2.0 assertion
334 containing a subject with multiple confirmation elements that apply to all the
335 statements of the assertion.

```

336 <saml2:Assertion
337     ...
338     <saml2:Subject>
339         <saml2:NameID>
340             ...
341         </saml2:NameID>
342         <saml2:SubjectConfirmation
343             Method="urn:oasis:names:tc:SAML:2.0:cm:sender-vouches">
344             <saml2:SubjectConfirmationData
345                 Address="129.148.9.42"
346             </saml2:SubjectConfirmationData>
347         </saml2:SubjectConfirmation>
348         <saml2:SubjectConfirmation
349             Method="urn:oasis:names:tc:SAML:2.0:cm:holder-of-key">
350             <saml2:KeyInfoSubjectConfirmationData>
351                 <ds:KeyInfo>
352                     <ds:KeyValue>...</ds:KeyValue>
353                 </ds:KeyInfo>
354             </saml2:KeyInfoSubjectConfirmationData>
355         </saml2:SubjectConfirmation>
356     </saml2:Subject>
357     ...
358 <saml2:Statement>
359     ...
360 </saml2:Statement>
361

```

362
363
364
365
366
367

```
<saml2:Statement>
...
</saml2:Statement>
...
</saml2:Assertion>
```

368 3.2.3 Assertion URI Reference Replaces AuthorityBinding

369 SAML V1.1 defines the (deprecated) <saml:AuthorityBinding> element so that a
370 relying party can locate and communicate with an assertion authority to acquire a
371 referenced assertion.

372 The <saml:AuthorityBinding> element was removed from SAML V2.0.
373 [SAMLBindV2] requires that an assertion authority support a URL endpoint at which
374 an assertion will be returned in response to an HTTP request with a single query
375 string parameter named ID.

376 For example, if the documented endpoint at an assertion authority is:

```
377 https://saml.example.edu/assertion-authority
```

378 then the following request will cause the assertion with ID "abcde" to be returned:

```
379 https://saml.example.edu/assertion-authority?ID=abcde
```

380 3.2.4 Attesting Entity Identifier

381 The <saml2:SubjectConfirmation> element of SAML V2.0 provides for the optional
382 inclusion of an element (i.e., NameID) to identify the expected attesting entity as
383 distinct from the subject of the assertion.

```
384 <saml2:SubjectConfirmation
385   Method="urn:oasis:names:tc:SAML:2.0:cm:sender-vouches">
386   <NameID>
387     gateway
388   </NameID>
389   <saml2:SubjectConfirmationData>
390     Address="129.148.9.42"
391   </saml2:SubjectConfirmationData>
392 </saml2:SubjectConfirmation>
```

393 3.3 Attaching Security Tokens

394 SAML assertions are attached to SOAP messages using WSS: SOAP Message Security
395 by placing assertion elements or references to assertions inside a <wsse:Security>
396 header. The following example illustrates a SOAP message containing a bearer
397 confirmed SAML V1.1 assertion in a <wsse:Security> header.

```
398 <S12:Envelope>
399   <S12:Header>
400     <wsse:Security>
401
402       <saml:Assertion
403         AssertionID="_a75adf55-01d7-40cc-929f-dbd8372ebdfc"
404         IssueInstant="2003-04-17T00:46:02Z"
```

```

405     Issuer="www.opensaml.org"
406     MajorVersion="1"
407     MinorVersion="1"
408     . . .
409     <saml:AuthenticationStatement>
410         <saml:Subject>
411             <saml:NameIdentifier
412                 NameQualifier="www.example.com"
413                 Format="urn:oasis:names:tc:SAML:1.1:nameid-
414 format:X509SubjectName">
415                 uid=joe,ou=people,ou=saml-demo,o=baltimore.com
416             </saml:NameIdentifier>
417             <saml:SubjectConfirmation>
418                 <saml:ConfirmationMethod>
419                     urn:oasis:names:tc:SAML:1.0:cm:bearer
420                 </saml:ConfirmationMethod>
421             </saml:SubjectConfirmation>
422         </saml:Subject>
423     </saml:AuthenticationStatement>
424
425     </saml:Assertion>
426
427 </wsse:Security>
428 </S12:Header>
429 <S12:Body>
430     . . .
431 </S12:Body>
432 </S12:Envelope>

```

433 3.4 Identifying and Referencing Security Tokens

434 The WSS: SOAP Message Security specification defines the
435 <wsse:SecurityTokenReference> element for referencing security tokens. Three
436 forms of token references are defined by this element and the element schema
437 includes provision for defining additional reference forms should they be necessary.
438 The three forms of token references defined by the
439 <wsse:SecurityTokenReference> element are defined as follows:

- 440 • A key identifier reference – a generic element (i.e., <wsse:KeyIdentifier>) that
441 conveys a security token identifier as an <wsse:EncodedString> and indicates in
442 its attributes (as necessary) the key identifier type (i.e., the `ValueType`), the
443 identifier encoding type (i.e., the `EncodingType`), and perhaps other parameters
444 used to reference the security token.

445 When a key identifier is used to reference a SAML assertion, it MUST contain as
446 its element value the corresponding SAML assertion identifier. The key identifier
447 MUST also contain a `ValueType` attribute and the value of this attribute MUST be
448 the value from Table 2 corresponding to the version of the referenced assertion.

449 The key identifier MUST NOT include an `EncodingType`⁴ attribute and the element
450 content of the key identifier MUST be encoded as `xsi:string`.

451 When a key identifier is used to reference a V1.1 SAML assertion that is not
452 contained in the same message as the key identifier, a
453 `<saml:AuthorityBinding>` element MUST be contained in the
454 `<wsse:SecurityTokenReference>` element containing the key identifier. The
455 contents of the `<saml:AuthorityBinding>` element MUST contain values
456 sufficient for the intended recipients of the `<wsse:SecurityTokenReference>` to
457 acquire the identified assertion from the intended Authority. To this end, the
458 value of the `AuthorityKind` attribute of the `<saml:AuthorityBinding>` element
459 MUST be "samlp:AssertionIdReference".

460 When a key Identifier is used to reference a SAML assertion contained in the
461 same message as the key identifier, a `<saml:AuthorityBinding>` element MUST
462 NOT be included in the `<wsse:SecurityTokenReference>` containing the key
463 identifier.

464 A key identifier ~~MUST NOT~~**MAY only** be used to reference a SAML V2.0 assertion if
465 the assertion is NOT contained in the same message as the key identifier.

466 • A Direct or URI reference – a generic element (i.e., `<wsse:Reference>`) that
467 identifies a security token by URI. If only a fragment identifier is specified, then
468 the reference is to the security token within the document whose local identifier
469 (e.g., `<wsu:Id>` attribute) matches the fragment identifier. Otherwise, the
470 reference is to the (potentially external) security token identified by the URI.

471 A reference to a SAML V2.0 assertion that is NOT contained in the same message
472 MUST be a Direct or URI reference. In this case, the value of the URI attribute
473 must conform to the URI syntax defined in section 3.7.5.1 of [SAMLBindV2]. That
474 is, an HTTP or HTTPS request with a single query string parameter named ID.
475 The reference MUST also contain a `wsse11:TokenType` attribute and the value of
476 this attribute MUST be the `value` from Table 3 identifying the assertion as a
477 SAML V2.0 security token. When a Direct reference is made to a SAML V2.0
478 Assertion, the Direct reference SHOULD NOT contain a `ValueType` attribute.

479 This profile does not describe the use of Direct or URI references to reference
480 V1.1 SAML assertions.

481 • An Embedded reference – a reference that encapsulates a security token.

⁴ "The Errata for Web Services Security: SOAP Message Security Version 1.0" (at <http://www.oasis-open.org/committees/wss>) removed the default designation from the `#Base64Binary` value for the `EncodingType` attribute of the `KeyIdentifier` element. Therefore, omitting a value for `EncodingType` and requiring that Base64 encoding not be performed, as specified by this profile, is consistent with the [WS-Security Specification \(including V1.1\)errata](#).

482 When an Embedded reference is used to encapsulate a SAML assertion, the SAML
 483 assertion MUST be included as a contained element within a <wsse:Embedded>
 484 element within a <wsse:SecurityTokenReference>.

485 This specification describes how SAML assertions may be referenced in four contexts:

- 486 • A SAML assertion may be referenced directly from a <wsse:Security> header
 487 element. In this case, the assertion is being conveyed by reference in the
 488 message.
- 489 • A SAML assertion may be referenced from a <ds:KeyInfo> element of a
 490 <ds:Signature> element in a <wsse:Security> header. In this case, the
 491 assertion contains a SubjectConfirmation element that identifies the key used
 492 in the signature calculation.
- 493 • A SAML assertion reference may be referenced from a <ds:Reference> element
 494 within the <ds:SignedInfo> element of a <ds:Signature> element in a
 495 <wsse:Security> header. In this case, the doubly-referenced assertion is signed
 496 by the containing signature.
- 497 • A SAML assertion reference may occur as encrypted content within an
 498 <xenc:EncryptedData> element referenced from a <xenc:DataReference>
 499 element within an <xenc:ReferenceList> element. In this case, the assertion
 500 reference (which may contain an embedded assertion) is encrypted.

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

- 502 • local – in which case, it is included in the <wsse:Security> header containing
 503 the reference.
- 504 • remote – in which case it is not included in the <wsse:Security> header
 505 containing the reference, but may occur in another part of the SOAP message or
 506 may be available at the location identified by the reference which may be an
 507 assertion authority.

508 A SAML key identifier reference MUST be used for all (local and remote) references
 509 to SAML 1.1 assertions. All (local and remote) references to SAML V2.0 assertions
 510 SHOULD be by Direct reference and all remote references to V2.0 assertions MUST
 511 be by Direct reference URI. A key identifier reference MAY be used to reference a
 512 local V2.0 assertion. To maintain compatibility with Web Services Security: SOAP
 513 Message Security 1.0, the practice of referencing local SAML 1.1 assertions by Direct
 514 <wsse:SecurityTokenReference> reference is not ~~defined by~~ included in this profile.

515 Every key identifier, direct, or embedded reference to a SAML assertion SHOULD
 516 contain a wss11:TokenType attribute and the value of this attribute MUST be the
 517 value from Table 3 that identifies the type and version of the referenced security
 518 token. When the referenced assertion is a SAML V2.0 Assertion the reference MUST
 519 contain a wss11:TokenType attribute (as described above).

Assertion Version	Value
V1.1	http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.0#SAMLAssertionID

V2.0	http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLID
------	--

520 Table-2 Key Identifier ValueType Attribute Values

Assertion Version	Value
V1.1	http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLV1.1
V2.0	http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLV2.0

521 Table-3 TokenType Attribute Values

522 The following subsections define the SAML assertion references that MUST be
523 supported by conformant implementations of this profile. A ~~conformant compatible~~
524 implementation may choose to support the reference forms corresponding to either
525 or both V1.1 or V2.0 SAML assertions.

526 3.4.1 SAML Assertion Referenced from Header or Element

527 All conformant implementations MUST be able to process SAML assertion references
528 occurring in a <wsse:Security> header or in a header element other than a
529 signature to acquire the corresponding assertion. A conformant implementation
530 MUST be able to process any such reference independent of the confirmation method
531 of the referenced assertion.

532 A SAML assertion may be referenced from a <wsse:Security> header or from an
533 element (other than a signature) in the header. The following example demonstrates
534 the use of a key identifier in a <wsse:Security> header to reference a local SAML
535 V1.1 assertion.

```

536 <S12:Envelope>
537   <S12:Header>
538     <wsse:Security>
539       <saml:Assertion
540         AssertionID="_a75adf55-01d7-40cc-929f-dbd8372ebdfc"
541         IssueInstant="2003-04-17T00:46:02Z"
542         Issuer="www.opensaml.org"
543         MajorVersion="1"
544         MinorVersion="1"
545         . . .
546       </saml:Assertion>
547       <wsse:SecurityTokenReference wsu:Id="STR1"
548         wsse11:TokenType="http://docs.oasis-open.org/wss/oasis-wss-
549 saml-token-profile-1.1#SAMLV1.1">
550         <wsse:KeyIdentifier wsu:Id="..."
551           ValueType="http://docs.oasis-open.org/wss/oasis-wss-saml-
552 token-profile-1.0#SAMLAssertionID">
553           _a75adf55-01d7-40cc-929f-dbd8372ebdfc
554         </wsse:KeyIdentifier>
555       </wsse:SecurityTokenReference>
556     </wsse:Security>

```

```
557 </S12:Header>
558 <S12:Body>
559 . . .
560 </S12:Body>
561 </S12:Envelope>
```

562 The following example depicts the use of a key identifier reference to reference a
563 local SAML V2.0 assertion.

```
564 <wsse:SecurityTokenReference
565   wsu:Id="STR1"
566   wss11:TokenType="http://docs.oasis-open.org/wss/oasis-wss-saml-
567   token-profile-1.1#SAMLV2.0">
568   <wsse:KeyIdentifier wsu:Id="..."
569     ValueType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-
570   profile-1.1#SAMLID">
571     _a75adf55-01d7-40cc-929f-dbd8372ebdfc
572   </wsse:KeyIdentifier>
573 </wsse:SecurityTokenReference>
```

574 A SAML V1.1 assertion that exists outside of a `<wsse:Security>` header may be
575 referenced from the `<wsse:Security>` header element by including (in the
576 `<wsse:SecurityTokenReference>`) a `<saml:AuthorityBinding>` element that
577 defines the location, binding, and query that may be used to acquire the identified
578 assertion at a SAML assertion authority or responder.

```
579 <wsse:SecurityTokenReference wsu:Id="STR1"
580   wss11:TokenType="http://docs.oasis-open.org/wss/oasis-wss-saml-
581   token-profile-1.1#SAMLV1.1">
582   <saml:AuthorityBinding>
583     Binding="urn:oasis:names:tc:SAML:1.0:bindings:SOAP-binding"
584     Location="http://www.opensaml.org/SAML-Authority"
585     AuthorityKind="samlp:AssertionIdReference"
586   </saml:AuthorityBinding>
587   <wsse:KeyIdentifier
588     wsu:Id="..."
589     ValueType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-
590   profile-1.0#SAMLAssertionID">
591     _a75adf55-01d7-40cc-929f-dbd8372ebdfc
592   </wsse:KeyIdentifier>
593 </wsse:SecurityTokenReference>
```

594 The following example depicts the use of a Direct or URI reference to reference a
595 SAML V2.0 assertion that exists outside of a `<wsse:Security>` header.

```
596 </wsse:SecurityTokenReference
597   wsu:Id="..."
598   wss11:TokenType="http://docs.oasis-open.org/wss/oasis-wss-saml-
599   token-profile-1.1#SAMLV2.0">
600   <wsse:Reference
601     wsu:Id="..."
602     URI="https://saml.example.edu/assertion-authority?ID=abcde">
603   </wsse:Reference>
604 </wsse:SecurityTokenReference>
```

605 3.4.2 SAML Assertion Referenced from KeyInfo

606 All conformant implementations MUST be able to process SAML assertion references
607 occurring in the <ds:KeyInfo> element of a <ds:Signature> element in a
608 <wsse:Security> header as defined by the holder-of-key confirmation method.

609 The following example depicts the use of a key identifier to reference a local V1.1
610 assertion from <ds:KeyInfo>.

```
611 <ds:KeyInfo>
612   <wsse:SecurityTokenReference wsu:Id="STR1"
613     wsse1:TokenType="http://docs.oasis-open.org/wss/oasis-wss-saml-
614     token-profile-1.1#SAMLV1.1">
615     <wsse:KeyIdentifier wsu:Id="..."
616       ValueType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-
617       profile-1.0#SAMLAssertionID">
618       _a75adf55-01d7-40cc-929f-dbd8372ebdfc
619     </wsse:KeyIdentifier>
620   </wsse:SecurityTokenReference>
621 </ds:KeyInfo>
```

622 A local, V2.0 assertion may be referenced by replacing the values of the Key
623 Identifier ValueType and reference TokenType attributes with the values defined in
624 tables 2 and 3 (respectively) for SAML V2.0 ~~as follows and (repeated below):~~

```
625 <ds:KeyInfo>
626   <wsse:SecurityTokenReference wsu:Id="STR1"
627     wsse1:TokenType="http://docs.oasis-open.org/wss/oasis-wss-saml-
628     token-profile-1.1#SAMLV2.0">
629     <wsse:KeyIdentifier wsu:Id="..."
630       ValueType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-
631       profile-1.0#SAMLID">
632       _a75adf55-01d7-40cc-929f-dbd8372ebdfc
633     </wsse:KeyIdentifier>
634   </wsse:SecurityTokenReference>
635 </ds:KeyInfo>
636 wsse1:TokenType="http://docs.oasis-open.org/wss/oasis-
637 wss-saml-token-profile-1.1#SAMLV2.0"
638 ValueType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-
639 1.1#SAMLID"
```

639 The following example demonstrates the use of a <wsse:SecurityTokenReference>
640 containing a key identifier and a <saml:AuthorityBinding> to communicate
641 information (location, binding, and query) sufficient to acquire the identified V1.1
642 assertion at an identified SAML assertion authority or responder.

```
643 <ds:KeyInfo>
644   <wsse:SecurityTokenReference wsu:Id="STR1"
645     wsse1:TokenType="http://docs.oasis-open.org/wss/oasis-wss-saml-
646     token-profile-1.1#SAMLV1.1">
647     <saml:AuthorityBinding>
648       Binding="urn:oasis:names:tc:SAML:1.0:bindings:SOAP-binding"
649       Location="http://www.opensaml.org/SAML-Authority"
650       AuthorityKind="samlp:AssertionIdReference"
651     </saml:AuthorityBinding>
652     <wsse:KeyIdentifier wsu:Id="..."
653       ValueType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-
654       profile-1.0#SAMLAssertionID">
655       _a75adf55-01d7-40cc-929f-dbd8372ebdfc
```

656
657
658

```
</wsse:KeyIdentifier>  
</wsse:SecurityTokenReference>  
</ds:KeyInfo>
```

659 Remote references to V2.0 assertions are made by Direct reference URI. The
660 following example depicts the use of a Direct reference URI to reference a remote
661 V2.0 assertion from <ds:KeyInfo>.

662
663
664
665
666
667
668
669
670
671
672

```
<ds:KeyInfo>  
  <wsse:SecurityTokenReference  
    wsu:id="STR1"  
    wss11:TokenType="http://docs.oasis-open.org/wss/oasis-wss-saml-  
token-profile-1.1#SAMLV2.0">  
    <wsse:Reference  
      wsu:id="..."  
      URI="https://saml.example.edu/assertion-authority?ID=abcde">  
    </wsse:Reference>  
  </wsse:SecurityTokenReference>  
</ds:KeyInfo>
```

673 <ds:KeyInfo> elements may also occur in <xenc:EncryptedData> and
674 <xenc:EncryptedKey> elements where they serve to identify the encryption key.
675 <ds:KeyInfo> elements may also occur in SAML SubjectConfirmation elements
676 where they identify a key that MUST be demonstrated to confirm the subject of the
677 corresponding statement(s).

678 Conformant implementations of this profile are ~~NOT~~ required to process SAML
679 assertion references occurring within the <ds:keyInfo> elements within
680 <xenc:EncryptedData>, <xenc:EncryptedKey>, or SAML SubjectConfirmation
681 elements.

682 3.4.3 SAML Assertion Referenced from SignedInfo

683 Independent of the confirmation method of the referenced assertion, all conformant
684 implementations MUST be able to process SAML assertions referenced by
685 <wsse:SecurityTokenReference> from <ds:Reference> elements within the
686 <ds:SignedInfo> element of a <ds:Signature> element in a <wsse:Security>
687 header. Embedded references may be digested directly, thus effectively digesting the
688 encapsulated assertion. Other <wsse:SecurityTokenReference> forms must be
689 dereferenced for the referenced assertion to be digested.

690 The core specification, WSS: SOAP Message Security, defines the STR Dereference
691 transform to cause the replacement (in the digest stream) of a
692 <wsse:SecurityTokenReference> with the contents of the referenced token. The
693 STR Dereference transform MUST be specified and applied to digest any SAML
694 assertion that is referenced by a <wsse:SecurityTokenReference> that is not an
695 embedded reference. The STR Dereference transform SHOULD NOT be applied to an
696 embedded reference.

697 The following example demonstrates the use of the STR Dereference transform to
698 dereference a reference to a SAML V1.1 Assertion (i.e., Security Token) such that
699 the digest operation is performed on the security token not its reference.

700

```
<wsse:SecurityTokenReference wsu:Id="STR1"
```

701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736

```
wss11:TokenType="http://docs.oasis-open.org/wss/oasis-wss-saml-
token-profile-1.1#SAML.V1.1">
<saml:AuthorityBinding>
  Binding="urn:oasis:names:tc:SAML:1.0:bindings:SOAP-binding"
  Location="http://www.opensaml.org/SAML-Authority"
  AuthorityKind= "samlp:AssertionIdReference"
</saml:AuthorityBinding>
<wsse:KeyIdentifier wsu:Id="..."
  ValueType="http://docs.oasis-open.org/wss/oasis-2004XX-wss-saml-
token-profile-1.0#SAMLAssertionID">
  _a75adf55-01d7-40cc-929f-dbd8372ebdfc
</wsse:KeyIdentifier>
</wsse:SecurityTokenReference>
. . .
<ds:SignedInfo>
  <ds:CanonicalizationMethod
    Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
  <ds:SignatureMethod
    Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
  <ds:Reference URI="#STR1">
    <Transforms>
      <ds:Transform
        Algorithm="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-soap-message-security-1.0#STR-Transform" />
      <wsse:TransformationParameters>
        <ds:CanonicalizationMethod
          Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
        </wsse:TransformationParameters>
      </ds:Transform>
    </Transforms>
  <ds:DigestMethod
    Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
  <ds:DigestValue>...</ds:DigestValue>
</ds:Reference>
</ds:SignedInfo>
```

737 Note that the URI appearing in the `<ds:Reference>` element identifies the
738 `<wsse:SecurityTokenReference>` element by its `wsu:Id` value. Also note that the
739 STR Dereference transform MUST contain (in `<wsse:TransformationParameters>`) a
740 `<ds:CanonicalizationMethod>` that defines the algorithm to be used to serialize the
741 input node set (of the referenced assertion).

742 As depicted in the other examples of this section, this profile establishes
743 `<wsse:SecurityTokenReference>` forms for referencing V1.1, local V2.0, and
744 remote V2.0 assertions.

745 **3.4.4 SAML Assertion Referenced from Encrypted Data** 746 **Reference**

747 Independent of the confirmation method of the referenced assertion, all conformant
748 implementations MUST be able to process SAML assertion references occurring as
749 encrypted content within the `<xenc:EncryptedData>` elements referenced by Id
750 from the `<xenc:DataReference>` elements of `<xenc:ReferenceList>` elements. An
751 `<xenc:ReferenceList>` element may occur either as a top-level element in a

752 [Security](#)`<wsse:Security>` header, or embedded within an `<xenc:EncryptedKey>`
753 element. In either case, the `<xenc:ReferenceList>` identifies the encrypted content.

754 Such references are similar in format to the references that MAY appear in the
755 `<ds:Reference>` element within `<ds:SignedInfo>`, except the STR Dereference
756 transform does not apply. As shown in the following example, an encrypted
757 `<wsse:SecurityTokenReference>` (which may contain an embedded assertion) is
758 referenced from an `<xenc:DataReference>` by including the identifier of the
759 `<xenc:EncryptedData>` element that contains the encrypted
760 `<wsse:SecurityTokenReference>` in the `<xenc:DataReference>`.

```
761 <xenc:EncryptedData Id="EncryptedSTR1">  
762   <ds:keyInfo:ds:KeyInfo>  
763     . . .  
764   </ds:KeyInfo>  
765   <xenc:CipherData>  
766     <xenc:CipherValue>...</xenc:CipherValue>  
767   </xenc:CipherData>  
768 </xenc:EncryptedData>  
769 <xenc:ReferenceList>  
770   <xenc:DataReference URI="#EncryptedSTR1"/>  
771 </xenc:ReferenceList>
```

772 **3.4.5 SAML Version Support and Backward Compatability**

773 An implementation of this profile MUST satisfy all of its requirements with respect to
774 either or both SAML V1.1 or SAML V2.0 Assertions. An implementation that satisfies
775 the requirements of this profile with respect to SAML V1.1 assertions MUST be able
776 to fully interoperate with any fully compatible implementation of version 1.0 of this
777 profile.

778 An implementation that does not satisfy the requirements of this profile with respect
779 to SAML V1.1 or SAML V2.0 assertions MUST reject a message containing a
780 `<wsse:Security>` header that references or conveys an assertion of the unsupported
781 version. When a message containing an unsupported assertion version is detected,
782 the receiver MAY choose to respond with an appropriate fault as defined in Section
783 3.6, "Error Codes".

784 **3.5 Subject Confirmation of SAML Assertions**

785 The SAML profile of WSS: SOAP Message Security requires that systems support the
786 holder-of-key and sender-vouches methods of subject confirmation. It is strongly
787 RECOMMENDED that an XML signature be used to establish the relationship between
788 the message and the statements of the attached assertions. This is especially
789 RECOMMENDED whenever the SOAP message exchange is conducted over an
790 unprotected transport.

791 Any processor of SAML assertions MUST conform to the required validation and
792 processing rules defined in the corresponding SAML specification including the
793 validation of assertion signatures, the processing of `<saml:Condition>` elements
794 within assertions, and the processing of `<saml2:SubjectConfirmationData>`

795 attributes. [SAMLCoreV1] defines the validation and processing rules for V1.1
 796 assertions, while [SAMLCoreV2] is authoritative for V2.0 assertions.

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

Mechanism	RECOMMENDED Processing Rules
Urn:oasis:names:tc:SAML:1.0:cm:holder-of-key Or urn:oasis:names:tc:SAML:2.0:cm:holder-of-key	The attesting entity demonstrates knowledge of a confirmation key identified in a holder-of-key <code>SubjectConfirmation</code> element within the assertion.
Urn:oasis:names:tc:SAML:1.0:cm:sender-vouches Or urn:oasis:names:tc:SAML:2.0:cm:sender-vouches	The attesting entity, (presumed to be) different from the subject, vouches for the verification of the subject. The receiver MUST have an existing trust relationship with the attesting entity. The attesting entity MUST protect the assertion in combination with the message content against modification by another party. See also section 4.

799 Note that the high level processing model described in the following sections does
 800 not differentiate between the attesting entity and the message sender as would be
 801 necessary to guard against replay attacks. The high-level processing model also does
 802 not take into account requirements for authentication of receiver by sender, or for
 803 message or assertion confidentiality. These concerns must be addressed by means
 804 other than those described in the high-level processing model (i.e., section 3.1).

805 3.5.1 Holder-of-key Subject Confirmation Method

806 The following sections describe the holder-of-key method of establishing the
 807 correspondence between a SOAP message and the subject and claims of SAML
 808 assertions added to the SOAP message according to this specification.

809 3.5.1.1 Attesting Entity

810 An attesting entity demonstrates that it is authorized to act as the subject of a
 811 holder-of-key confirmed SAML statement by demonstrating knowledge of any key
 812 identified in a holder-of-key `SubjectConfirmation` element associated with the
 813 statement by the assertion containing the statement. Statements attested for by the

814 holder-of-key method MUST be associated, within their containing assertion, with
815 one or more holder-of-key `SubjectConfirmation` elements.

816 The `SubjectConfirmation` elements MUST include a `<ds:KeyInfo>` element that
817 identifies a public or secret key⁵ that can be used to confirm the identity of the
818 subject.

819 To satisfy the associated confirmation method processing to be performed by the
820 message receiver, the attesting entity MUST demonstrate knowledge of the
821 confirmation key. The attesting entity MAY accomplish this by using the confirmation
822 key to sign content within the message and by including the resulting
823 `<ds:Signature>` element in the `<wsse:Security>` header. `<ds:Signature>`
824 elements produced for this purpose MUST conform to the `canonicalization` and
825 token pre-pending rules defined in the WSS: SOAP Message Security specification.

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

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

833 **3.5.1.2 Receiver**

834 Of the SAML assertions it selects for processing, a message receiver MUST NOT
835 accept statements of these assertions based on a holder-of-key
836 `SubjectConfirmation` element defined for the statements (within the assertion)
837 unless the receiver has validated the integrity of the assertion and the attesting
838 entity has demonstrated knowledge of a key identified within the confirmation
839 element.

840 If the receiver determines that the attesting entity has demonstrated knowledge of a
841 subject confirmation key, then the subjects and claims of the SAML statements
842 confirmed by the key MAY be attributed to the attesting entity and any content of the
843 message whose integrity is protected by the key MAY be considered to have been
844 provided by the attesting entity.

⁵[SAMLCoreV1] defines `KeyInfo` of `SubjectConfirmation` as containing a
"cryptographic key held by the subject". Demonstration of this key is sufficient to
establish who is (or may act as the) subject. Moreover, since it cannot be proven
that a confirmation key is known (or known only) by the subject whose identity it
establishes, requiring that the key be held by the subject is an untestable
requirement that adds nothing to the strength of the confirmation mechanism. In
[SAMLCoreV2], the OASIS Security Services Technical Committee agreed to remove
the phrase "held by the subject" from the definition of `KeyInfo` within
`SubjectConfirmation(Data)`.

845 3.5.1.3 Example V1.1

846 The following example illustrates the use of the holder-of-key subject confirmation
847 method to establish the correspondence between the SOAP message and the subject
848 of statements of the SAML V1.1 assertions in the <wsse:Security> header:

```
849 <?xml:version version="1.0" encoding="UTF-8"?>
850 <S12:Envelope>
851   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
852   xmlns:xsd="http://www.w3.org/2001/XMLSchema">
853   <S12:Header>
854
855     <wsse:Security>
856       <saml:Assertion
857         AssertionID="_a75adf55-01d7-40cc-929f-dbd8372ebdfc"
858         IssueInstant="2003-04-17T00:46:02Z"
859         IssueInstant="2005-05-27T16:53:33.173Z"
860         Issuer="www.opensaml.org"
861         MajorVersion="1"
862         MinorVersion="1"
863         xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion">
864         <saml:Conditions>
865           NotBefore="2002-06-19T16:53:33.173Z"2005-05-
866           27T16:53:33.173Z"
867           NotOnOrAfter="20052-056-2719T1617:5808:33.1730233.173Z"/>
868         <saml:AttributeStatement>
869           <saml:Subject>
870             <saml:NameIdentifier
871               NameQualifier="www.example.com"
872               Format="urn:oasis:names:tc:SAML:1.1:nameid-
873               format:X509SubjectName">
874               uid=joe,ou=people,ou=saml-demo,o=baltimore.com
875             </saml:NameIdentifier>
876             <saml:SubjectConfirmation>
877               <saml:ConfirmationMethod>
878                 urn:oasis:names:tc:SAML:1.0:cm:holder-of-key
879               </saml:ConfirmationMethod>
880               <ds:KeyInfo>
881                 <ds:KeyValue>...</ds:KeyValue>
882               </ds:KeyInfo>
883             </saml:SubjectConfirmation>
884           </saml:Subject>
885           <saml:Attribute
886             AttributeName="MemberLevel"
887             AttributeNamespace="http://www.oasis.openoasis-open.
888             org/Catalyst2002/attributes">
889             <saml:AttributeValue>gold</saml:AttributeValue>
890           </saml:Attribute>
891           <saml:Attribute
892             AttributeName="E-mail"
893             AttributeNamespace="http://www.oasis.openoasis-open.
894             org/Catalyst2002/attributes">
895             <saml:AttributeValue>joe@yahoo.com</saml:AttributeValue>
896           </saml:Attribute>
897         </saml:AttributeStatement>
898         <ds:Signature>...</ds:Signature>
899       </saml:Assertion>
900     </wsse:Security>
901   </S12:Header>
902   <S12:Body>
```

```

902     <ds:SignedInfo>
903     <ds:CanonicalizationMethod
904         Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
905     <ds:SignatureMethod
906         Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
907     <ds:Reference
908         URI="#MsgBody">
909         <ds:DigestMethod
910             Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
911         <ds:DigestValue>GyGsF0Pi4xPU...</ds:DigestValue>
912     </ds:Reference>
913 </ds:SignedInfo>
914 <ds:SignatureValue>HJJWbvqW9E84vJVQk...</ds:SignatureValue>
915 <ds:KeyInfo>
916     <wsse:SecurityTokenReference wsu:Id="STR1"
917         wssell1:TokenType="http://docs.oasis-open.org/wss/oasis-wss-
918 saml-token-profile-1.1#SAMLV1.1">
919         <wsse:KeyIdentifier wsu:Id="..."
920             ValueType="http://docs.oasis-open.org/wss/oasis-wss-saml-
921 token-profile-1.0#SAMLAssertionID">
922             _a75adf55-01d7-40cc-929f-dbd8372ebdfc
923         </wsse:KeyIdentifier>
924     </wsse:SecurityTokenReference>
925 </ds:KeyInfo>
926 </ds:Signature>
927 </wsse:Security>
928 </S12:Header>
929
930 <S12:Body wsu:Id="MsgBody">
931     <ReportRequest>
932         <TickerSymbol>SUNW</TickerSymbol>
933     </ReportRequest>
934 </S12:Body>
935 </S12:Envelope>

```

936 3.5.1.4 Example V2.0

937 The following example illustrates the use of the holder-of-key subject confirmation
938 method to establish the correspondence between the SOAP message and the subject
939 of the SAML V2.0 assertion in the <wsse:Security> header:

```

940 <?xml:version version="1.0" encoding="UTF-8"?>
941 <S12:Envelope>
942     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
943     xmlns:xsd="http://www.w3.org/2001/XMLSchema">
944     <S12:Header>
945
946         <wsse:Security>
947             <saml2:Assertion
948                 ...
949                 ID="_a75adf55-01d7-40cc-929f-dbd8372ebdfc"
950                 ...>
951                 <saml2:subject>
952                     <saml2:NameID>
953                         ...
954                     </saml2:NameID>
955                     <saml2:SubjectConfirmation
956                         Method="urn:oasis:names:tc:SAML:2.0:cm:holder-of-key">

```

```

957         <saml2:KeyInfoSubjectConfirmationData>
958             <ds:KeyInfo>
959                 <ds:KeyValue>...</ds:KeyValue>
960             </ds:KeyInfo>
961         </saml2:KeyInfoSubjectConfirmationData>
962     </saml2:SubjectConfirmation>
963 </saml2:Subject>
964 <saml2:Statement>
965     ...
966 </saml2:Statement>
967 <ds:Signature>...</ds:Signature>
968 </saml2:Assertion>
969
970 <ds:Signature>
971     <ds:SignedInfo>
972         <ds:CanonicalizationMethod
973             Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
974         <ds:SignatureMethod
975             Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
976         <ds:Reference
977             URI="#MsgBody">
978             <ds:DigestMethod
979                 Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
980             <ds:DigestValue>GyGsF0Pi4xPU...</ds:DigestValue>
981         </ds:Reference>
982     </ds:SignedInfo>
983     <ds:SignatureValue>HJJWbvqW9E84vJVQk...</ds:SignatureValue>
984     <ds:KeyInfo>
985         <wsse:SecurityTokenReference wsu:Id="STR1"
986             wss11:TokenType="http://docs.oasis-open.org/wss/oasis-wss-
987 saml-token-profile-1.1#SAMLV2.0">
988             <wsse:KeyIdentifier wsu:Id="..."
989                 ValueType="http://docs.oasis-open.org/wss/oasis-wss-saml-
990 token-profile-1.1#SAMLID">
991                 _a75adf55-01d7-40cc-929f-dbd8372ebdfc
992             </wsse:KeyIdentifier>
993         </wsse:SecurityTokenReference>
994     </ds:KeyInfo>
995 </ds:Signature>
996 </wsse:Security>
997 </S12:Header>
998
999 <S12:Body wsu:Id="MsgBody">
1000     <ReportRequest>
1001         <TickerSymbol>SUNW</TickerSymbol>
1002     </ReportRequest>
1003 </S12:Body>
1004 </S12:Envelope>

```

1005 3.5.2 Sender-vouches Subject Confirmation Method

1006 The following sections describe the sender-vouches method of establishing the
1007 correspondence between a SOAP message and the SAML assertions added to the
1008 SOAP message according to the SAML profile of WSS: SOAP Message Security.

1009 **3.5.2.1 Attesting Entity**

1010 An attesting entity uses the sender-vouches confirmation method to assert that it is
1011 acting on behalf of the subject of SAML statements attributed with a sender-vouches
1012 `SubjectConfirmation` element. Statements attested for by the sender-vouches
1013 method MUST be associated, within their containing assertion, with one or more
1014 sender-vouches `SubjectConfirmation` elements.

1015 To satisfy the associated confirmation method processing of the receiver, the
1016 attesting entity MUST protect the vouched for SOAP message content such that the
1017 receiver can determine when it has been altered by another party. The attesting
1018 entity MUST also cause the vouched for statements (as necessary) and their binding
1019 to the message contents to be protected such that unauthorized modification can be
1020 detected. The attesting entity MAY satisfy these requirements by including in the
1021 corresponding `<wsse:Security>` header a `<ds:Signature>` element that it prepares
1022 by using its key to sign the relevant message content and assertions. As defined by
1023 the XML Signature specification, the attesting entity MAY identify its key by including
1024 a `<ds:KeyInfo>` element within the `<ds:Signature>` element.

1025 A `<ds:Signature>` element produced for this purpose MUST conform to the
1026 canonicalization and token pre-pending rules defined in the WSS: SOAP Message
1027 Security specification.

1028 **3.5.2.2 Receiver**

1029 Of the SAML assertions it selects for processing, a message receiver MUST NOT
1030 accept statements of these assertions based on a sender-vouches
1031 `SubjectConfirmation` element defined for the statements (within the assertion)
1032 unless the assertions and SOAP message content being vouched for are protected
1033 (as described above) by an attesting entity who is trusted by the receiver to act as
1034 the subjects and with the claims of the statements.

1035 **3.5.2.3 Example V1.1**

1036 The following example illustrates an attesting entity's use of the sender-vouches
1037 subject confirmation method with an associated `<ds:Signature>` element to
1038 establish its identity and to assert that it has sent the message body on behalf of the
1039 subject(s) of the V1.1 assertion referenced by "STR1".

1040 The assertion referenced by "STR1" is not included in the message. "STR1" is
1041 referenced by `<ds:referenceds:Reference>` from `<ds:SignedInfo>`. The
1042 `<ds:referenceds:Reference>` includes the STR-transform to cause the assertion, not
1043 the `<SecurityTokenReferenceSecurityTokenReference>` to be included in the digest
1044 calculation. "STR1" includes a `<saml:AuthorityBinding>` element that utilizes the
1045 remote assertion referencing technique depicted in the example of section 3.3.3.

1046 The SAML V1.1 assertion embedded in the header and referenced by "STR2" from
1047 `<ds:KeyInfo>` corresponds to the attesting entity. The private key corresponding to
1048 the public confirmation key occurring in the assertion is used to sign together the
1049 message body and assertion referenced by "STR1".

1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108

```
<?xml:version version="1.0" encoding="UTF-8"?>
<S12:Envelope>
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <S12:Header>
    <wsse:Security>

      <saml:Assertion
        AssertionID="_a75adf55-01d7-40cc-929f-dbd8372ebdfc"
        IssueInstant="2003-04-17T00:46:02Z"2005-05-27T16:53:33.173Z"

        Issuer="www.opensaml.org"
        MajorVersion="1"
        MinorVersion="1"
        xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion">
        <saml:Conditions>
          NotBefore="20052-056-2179T16:53:33.173Z"
          NotOnOrAfter="20052-056-2719T167:508:33.173Z"/>
        <saml:AttributeStatement>
          <saml:Subject>
            <saml:NameIdentifier
              NameQualifier="www.example.com"
              Format="..."Format="urn:oasis:names:tc:SAML:1.1:nameid-
              format:X509SubjectName">
              uid=proxy,ou=system,ou=saml-demo,o=baltimore.com
            </saml:NameIdentifier>
            <saml:SubjectConfirmation>
              <saml:ConfirmationMethod>
                urn:oasis:names:tc:SAML:1.0:cm:holder-of-key
              </saml:ConfirmationMethod>
              <ds:KeyInfo>
                <ds:KeyValue>...</ds:KeyValue>
              </ds:KeyInfo>
            </saml:SubjectConfirmation>
          </saml:Subject>
          <saml:Attribute>
            . . .
          </saml:Attribute>
          . . .
        </saml:AttributeStatement>
      </saml:Assertion>

      <wsse:SecurityTokenReference wsu:Id="STR1">
        wssell:TokenType="http://docs.oasis-open.org/wss/oasis-wss-
        saml-token-profile-1.1#SAMLV1.1">
        <saml:AuthorityBinding>
          Binding="urn:oasis:names:tc:SAML:1.0:bindings:SOAP-binding"
          Location="http://www.opensaml.org/SAML-Authority"
          AuthorityKind="samlp:AssertionIdReference"
        </saml:AuthorityBinding>
        <wsse:KeyIdentifier wsu:Id="..."
          ValueType="http://docs.oasis-open.org/wss/oasis-wss-saml-
          token-profile-1.0#SAMLAssertionID">
          _a75adf55-01d7-40cc-929f-dbd8372ebdbe
        </wsse:KeyIdentifier>
      </wsse:SecurityTokenReference>

      <ds:Signature>
      <ds:SignedInfo>
```

```

1109     <ds:CanonicalizationMethod
1110         Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
1111     <ds:SignatureMethod
1112         Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
1113     <ds:Reference URI="#STR1">
1114         <Transforms>
1115             <ds:Transform
1116                 Algorithm="http://docs.oasis-
1117 open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#STR-
1118 Transform" />
1119                 <wsse:TransformationParameters>
1120                     <ds:CanonicalizationMethod
1121                         Algorithm="http://www.w3.org/2001/10/xml-exc-
1122 c14n#" />
1123                     </ds:Transform>
1124                 </ds:Transform>
1125             </Transforms>
1126             <ds:DigestMethod
1127                 Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
1128             <ds:DigestValue>...</ds:DigestValue>
1129         </ds:Reference>
1130     <ds:Reference URI="#MsgBody">
1131         <ds:DigestMethod
1132             Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
1133         <ds:DigestValue>...</ds:DigestValue>
1134     </ds:Reference>
1135 </ds:SignedInfo>
1136 <ds:SignatureValue>HJJWbvqW9E84vJVQk...</ds:SignatureValue>
1137 <ds:KeyInfo>
1138     <wsse:SecurityTokenReference wsu:Id="STR2"
1139         wsse11:TokenType="http://docs.oasis-open.org/wss/oasis-wss-
1140 saml-token-profile-1.1#SAMLV1.1">
1141         <wsse:KeyIdentifier wsu:Id="..."
1142             ValueType="http://docs.oasis-open.org/wss/oasis-wss-saml-
1143 token-profile-1.0#SAMLAssertionID">
1144             _a75adf55-01d7-40cc-929f-dbd8372ebdfc
1145         </wsse:KeyIdentifier>
1146     </wsse:SecurityTokenReference>
1147 </ds:KeyInfo>
1148 </ds:Signature>
1149 </wsse:Security>
1150 </S12:Header>
1151
1152 <S12:Body wsu:Id="MsgBody">
1153     <ReportRequest>
1154         <TickerSymbol>SUNW</TickerSymbol>
1155     </ReportRequest>
1156 </S12:Body>
1157 </S12:Envelope>

```

1158 3.5.2.4 Example V2.0

1159 The following example illustrates the mapping of the preceding example to SAML
1160 V2.0 assertions.

```

1161 <?xml:version version="1.0" encoding="UTF-8"?>
1162 <S12:Envelope>
1163     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
1164     xmlns:xsd="http://www.w3.org/2001/XMLSchema">

```

oasis-wss-saml-token-profile-1.1

1317 JuneMay 2005

Copyright © OASIS Open 2003-2005. All Rights Reserved.

Page 30 of 4138

```

1165 <S12:Header>
1166
1167   <wsse:Security>
1168     <saml2:Assertion
1169       ...
1170       ID="_a75adf55-01d7-40cc-929f-dbd8372ebdfc"
1171       ...>
1172     <saml2:subject>
1173       <saml2:NameID>
1174         ...
1175       </saml2:NameID>
1176       <saml2:SubjectConfirmation
1177         Method="urn:oasis:names:tc:SAML:2.0:cm:holder-of-key">
1178         <saml2:KeyInfoSubjectConfirmationData>
1179           <ds:KeyInfo>
1180             <ds:KeyValue>...</ds:KeyValue>
1181             </ds:KeyInfo>
1182           </saml2:KeyInfoSubjectConfirmationData>
1183         </saml2:SubjectConfirmation>
1184       </saml2:Subject>
1185     <saml2:Statement>
1186       ...
1187     </saml2:Statement>
1188     <ds:Signature>...</ds:Signature>
1189   </saml2:Assertion>
1190
1191   <wsse:SecurityTokenReference wsu:Id="STR1"
1192     wssell:TokenType="http://docs.oasis-open.org/wss/oasis-wss-
1193 saml-token-profile-1.1#SAMLV2.0">
1194     <wsse:Reference wsu:Id="..."
1195       URI="https://www.opensaml.org?_a75adf55-01d7-40cc-929f-
1196 dbd8372ebdbe">
1197     </wsse:Reference>
1198   </wsse:SecurityTokenReference>
1199
1200   <ds:Signature>
1201     <ds:SignedInfo>
1202       <ds:CanonicalizationMethod
1203         Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
1204       <ds:SignatureMethod
1205         Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
1206       <ds:Reference URI="#STR1">
1207         <Transforms>
1208           <ds:Transform
1209
1210             Algorithm="http://docs.oasis-open.org/wss/2004/01/oasis-
1211 200401-wss-soap-message-security-1.0#STR-Transform" />
1212           <wsse:TransformationParameters>
1213             <ds:CanonicalizationMethod
1214               Algorithm="http://www.w3.org/2001/10/xml-exc-
1215 c14n#" />
1216             </wsse:TransformationParameters>
1217           </ds:Transform>
1218         </Transforms>
1219         <ds:DigestMethod
1220           Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
1221         <ds:DigestValue>...</ds:DigestValue>
1222       </ds:Reference>
1223     </ds:Reference URI="#MsgBody">

```

```

1224     <ds:DigestMethod
1225         Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
1226     <ds:DigestValue>...</ds:DigestValue>
1227 </ds:Reference>
1228 </ds:SignedInfo>
1229 <ds:SignatureValue>HJJWbvqW9E84vJVQk...</ds:SignatureValue>
1230 <ds:KeyInfo>
1231     <wsse:SecurityTokenReference wsu:Id="STR2"
1232         wss11:TokenType="http://docs.oasis-open.org/wss/oasis-wss-
1233 saml-token-profile-1.1#SAMLV2.0">
1234         <wsse:KeyIdentifier wsu:Id="..."
1235             ValueType="http://docs.oasis-open.org/wss/oasis-wss-saml-
1236 token-profile-1.1#SAMLID">
1237             _a75adf55-01d7-40cc-929f-dbd8372ebdfc
1238         </wsse:KeyIdentifier>
1239     </wsse:SecurityTokenReference>
1240 </ds:KeyInfo>
1241 </ds:Signature>
1242 </wsse:Security>
1243 </S12:Header>
1244
1245 <S12:Body wsu:Id="MsgBody">
1246     <ReportRequest>
1247         <TickerSymbol>SUNW</TickerSymbol>
1248     </ReportRequest>
1249 </S12:Body>
1250 </S12:Envelope>

```

1251 **3.5.3 Bearer Confirmation Method**

1252 This profile does NOT require message receivers to establish the relationship
1253 between a received message and the statements of any bearer confirmed (i.e.,
1254 confirmation method urn:oasis:names:tc:SAML:1.0:cm:bearer) assertions
1255 conveyed or referenced from the message. Conformant implementations of this
1256 profile MUST be able to process references and convey bearer assertions within
1257 <wsse:Security> headers. Any additional processing requirements that pertain
1258 specifically to bearer confirmed assertions are outside the scope of this profile.

1259 **3.6 Error Codes**

1260 When a system that implements the SAML token profile of WSS: SOAP Message
1261 Security does not perform its normal processing because of an error detected during
1262 the processing of a security header, it MAY choose to report the cause of the error
1263 using the SOAP fault mechanism. The SAML token profile of WSS: SOAP Message
1264 Security does not require that SOAP faults be returned for such errors, and systems
1265 that choose to return faults SHOULD take care not to introduce any security
1266 vulnerabilities as a result of the information returned in error responses.

1267 Systems that choose to return faults SHOULD respond with the error codes and fault
1268 strings defined in the WSS: SOAP Message Security specification. The
1269 RECOMMENDED correspondence between the common assertion processing failures
1270 and the error codes defined in WSS: SOAP Message Security are defined in the
1271 following table:

Assertion Processing Error	RECOMMENDED Error(Faultcode)
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
<u>The receiver does not support the SAML version of a referenced or included assertion.</u>	<u>wsse:UnsupportedSecurityToken</u>

1272 The preceding table defines fault codes in a form suitable for use with SOAP 1.1. The
1273 WSS: SOAP Message Security specification describes how to map SOAP 1.1 fault
1274 constructs to the SOAP 1.2 fault constructs.

1275

4 Threat Model and Countermeasures (non-normative)

1276

1277 This document defines the mechanisms and procedures for securely attaching SAML
1278 assertions to SOAP messages. SOAP messages are used in multiple contexts,
1279 specifically including cases where the message is transported without an active
1280 session, the message is persisted, or the message is routed through a number of
1281 intermediaries. Such a general context of use suggests that users of this profile must
1282 be concerned with a variety of threats.

1283 In general, the use of SAML assertions with WSS: SOAP Message Security introduces
1284 no new threats beyond those identified for SAML or by the WSS: SOAP Message
1285 Security specification. The following sections provide an overview of the
1286 characteristics of the threat model, and the countermeasures that SHOULD be
1287 adopted for each perceived threat.

4.1 Eavesdropping

1288

1289 Eavesdropping is a threat to the SAML token profile of WSS: SOAP Message Security
1290 in the same manner as it is a threat to any network protocol. The routing of SOAP
1291 messages through intermediaries increases the potential incidences of
1292 eavesdropping. Additional opportunities for eavesdropping exist when SOAP
1293 messages are persisted.

1294 To provide maximum protection from eavesdropping, assertions, assertion
1295 references, and sensitive message content SHOULD be encrypted such that only the
1296 intended audiences can view their content. This approach removes threats of
1297 eavesdropping in transit, but MAY not remove risks associated with storage or poor
1298 handling by the receiver.

1299 Transport-layer security MAY be used to protect the message and contained SAML
1300 assertions and/or references from eavesdropping while in transport, but message
1301 content MUST be encrypted above the transport if it is to be protected from
1302 eavesdropping by intermediaries.

4.2 Replay

1303

1304 Reliance on authority-protected (e.g., signed) assertions with a holder-of-key
1305 subject confirmation mechanism precludes all but a holder of the key from binding
1306 the assertions to a SOAP message. Although this mechanism effectively restricts
1307 data origin to a holder of the confirmation key, it does not, by itself, provide the
1308 means to detect the capture and resubmission of the message by other parties.

1309 Assertions that contain a sender-vouches confirmation mechanism introduce another
1310 dimension to replay vulnerability if the assertions impose no restriction on the
1311 entities that may use or reuse the assertions.

1312 Replay attacks can be detected by receivers if message senders include additional
1313 message identifying information (e.g., timestamps, nonces, and or recipient
1314 identifiers) within origin-protected message content and receivers check this
1315 information against previously received values.

1316 **4.3 Message Insertion**

1317 The SAML token profile of WSS: SOAP Message Security is not vulnerable to
1318 message insertion attacks.

1319 **4.4 Message Deletion**

1320 The SAML token profile of WSS: SOAP Message Security is not vulnerable to
1321 message deletion attacks.

1322 **4.5 Message Modification**

1323 Messages constructed according to this specification are protected from message
1324 modification if receivers can detect unauthorized modification of relevant message
1325 content. Therefore, it is strongly RECOMMENDED that all relevant and immutable
1326 message content be signed by an attesting entity. Receivers SHOULD only consider
1327 the correspondence between the subject of the SAML assertions and the SOAP
1328 message content to have been established for those portions of the message that are
1329 protected by the attesting entity against modification by another entity.

1330 To ensure that message receivers can have confidence that received assertions have
1331 not been forged or altered since their issuance, SAML assertions appearing in or
1332 referenced from `<wsse:Security>` header elements MUST be protected against
1333 unauthorized modification (e.g., signed) by their issuing authority or the attesting
1334 entity (as the case warrants). It is strongly RECOMMENDED that an attesting entity
1335 sign any `<saml:Assertion>` elements that it is attesting for and that are not signed
1336 by their issuing authority.

1337 Transport-layer security MAY be used to protect the message and contained SAML
1338 assertions and/or assertion references from modification while in transport, but
1339 signatures are required to extend such protection through intermediaries.

1340 **4.6 Man-in-the-Middle**

1341 Assertions with a holder-of-key subject confirmation method are not vulnerable to a
1342 MITM attack. Assertions with a sender-vouches subject confirmation method are
1343 vulnerable to MITM attacks to the degree that the receiver does not have a trusted
1344 binding of key to the attesting entity's identity.

1345

5 References

- 1346 **[GLOSSARY]** Informational RFC 2828, "*Internet Security Glossary*," May
1347 2000.
- 1348 **[KEYWORDS]** S. Bradner, "Key words for use in RFCs to Indicate Requirement
1349 Levels," *RFC 2119*, Harvard University, March 1997
- 1350 **[SAMLBindV1]** Oasis Standard, E. Maler, P.Mishra, and R. Philpott (Editors),
1351 *Bindings and Profiles for the OASIS Security Assertion Markup
1352 Language (SAML) V1.1*, September 2003.
- 1353 **[SAMLBindV2]** Oasis Standard, S. Cantor, F. Hirsch, J. Kemp, R. Philpott, E.
1354 Maler (Editors), *Bindings for the OASIS Security Assertion
1355 Markup Language (SAML) V2.0*, March 2005.
- 1356 **[SAMLCoreV1]** Oasis Standard, E. Maler, P.Mishra, and R. Philpott (Editors),
1357 *Assertions and Protocols for the OASIS Security Assertion
1358 Markup Language (SAML) V1.1*, September 2003.
- 1359 **[SAMLCoreV2]** Oasis Standard, S. Cantor, J. Kemp, R. Philpott, E. Maler
1360 (Editors), *Assertions and Protocol for the OASIS Security
1361 Assertion Markup Language (SAML) V2.0*, March 2005.
- 1362 **[SOAP]** W3C Note, "SOAP: Simple Object Access Protocol 1.1," 08 May
1363 2000.
- 1364 W3C Working Draft, Nilo Mitra (Editor), *SOAP Version 1.2 Part
1365 0: Primer*, June 2002.
- 1366 W3C Working Draft, Martin Gudgin, Marc Hadley, Noah
1367 Mendelsohn, Jean-Jacques Moreau, Henrik Frystyk Nielsen
1368 (Editors), *SOAP Version 1.2 Part 1: Messaging Framework*, June
1369 2002.
- 1370 W3C Working Draft, Martin Gudgin, Marc Hadley, Noah
1371 Mendelsohn, Jean-Jacques Moreau, Henrik Frystyk Nielsen
1372 (Editors), *SOAP Version 1.2 Part 2: Adjuncts*, June 2002.
- 1373 **[URI]** T. Berners-Lee, R. Fielding, L. Masinter, "Uniform Resource
1374 Identifiers (URI): Generic Syntax," *RFC 2396*, MIT/LCS, U.C.
1375 Irvine, Xerox Corporation, August 1998.
- 1376 **[WS-SAML]** Contribution to the WSS TC, P. Mishra (Editor), WS-Security
1377 Profile of the Security Assertion Markup Language (SAML)
1378 Working Draft 04, Sept 2002.
- 1379 **[WSS: SAML Token Profile]** Oasis Standard, P. Hallem-Baker, A. Nadalin, C.
1380 Kaler, R. Monzillo (Editors), *Web Services Security: SAML
1381 Token Profile 1.0*, December 2004.

1391

Appendix A: Acknowledgements

1392	<u>Maneesh Sahu</u>	<u>Actional Corp</u>
1393	<u>Gene Thurston</u>	<u>AmberPoint</u>
1394	<u>Frank Siebenlist</u>	<u>Argonne National Laboratory</u>
1395	<u>Hal Lockhart</u>	<u>BEA Systems, Inc.</u>
1396	<u>Corinna Witt</u>	<u>BEA Systems, Inc.</u>
1397	<u>Steve Anderson</u>	<u>BMC Software</u>
1398	<u>Davanum Srinivas</u>	<u>Computer Associates</u>
1399	<u>Rich Levinson</u>	<u>Computer Associates</u>
1400	<u>Thomas DeMartini</u>	<u>ContentGuard</u>
1401	<u>Guillermo Lao</u>	<u>ContentGuard</u>
1402	<u>Merlin Hughes</u>	<u>Cybertrust</u>
1403	<u>Rich Salz</u>	<u>DataPower</u>
1404	<u>Sam Wei</u>	<u>Documentum</u>
1405	<u>Tim Moses</u>	<u>Entrust</u>
1406	<u>Carolina Canales-Valenzuela</u>	<u>Ericsson</u>
1407	<u>Dana Kaufman</u>	<u>Forum Systems, Inc.</u>
1408	<u>Toshihiro Nishimura</u>	<u>Fujitsu</u>
1409	<u>Kefeng Chen</u>	<u>GeoTrust</u>
1410	<u>Irving Reid</u>	<u>Hewlett-Packard</u>
1411	<u>Kojiro Nakayama</u>	<u>Hitachi</u>
1412	<u>Paula Austel</u>	<u>IBM</u>
1413	<u>Derek Fu</u>	<u>IBM</u>
1414	<u>Maryann Hondo</u>	<u>IBM</u>
1415	<u>Kelvin Lawrence</u>	<u>IBM</u>
1416	<u>Hiroshi Maruyama</u>	<u>IBM</u>
1417	<u>Michael McIntosh</u>	<u>IBM</u>
1418	<u>Anthony Nadalin</u>	<u>IBM</u>
1419	<u>Nataraj Nagaratnam</u>	<u>IBM</u>
1420	<u>Ron Williams</u>	<u>IBM</u>
1421	<u>Don Flinn</u>	<u>Individual</u>
1422	<u>Jerry Schwarz</u>	<u>Individual</u>
1423	<u>Bob Morgan</u>	<u>Internet2</u>
1424	<u>Kate Cherry</u>	<u>Lockheed Martin</u>
1425	<u>Paul Cotton</u>	<u>Microsoft Corporation</u>
1426	<u>Vijay Gajjala</u>	<u>Microsoft Corporation</u>
1427	<u>Alan Geller</u>	<u>Microsoft Corporation</u>
1428	<u>Chris Kaler</u>	<u>Microsoft Corporation</u>
1429	<u>Jeff Hodges</u>	<u>Neustar</u>
1430	<u>Frederick Hirsch</u>	<u>Nokia</u>
1431	<u>Senthil Sengodan</u>	<u>Nokia</u>
1432	<u>Abbie Barbir</u>	<u>Nortel Networks</u>
1433	<u>Lloyd Burch</u>	<u>Novell</u>
1434	<u>Charles Knouse</u>	<u>Oblix</u>
1435	<u>Vamsi Motukuru</u>	<u>Oracle</u>
1436	<u>Ramana Turlapati</u>	<u>Oracle</u>
1437	<u>Prateek Mishra</u>	<u>Principal Identity</u>
1438	<u>Andrew Nash</u>	<u>Reactivity</u>

1439	Ben Hammond	RSA Security
1440	Rob Philpott	RSA Security
1441	Martijn de Boer	SAP
1442	Blake Dournaee	Sarvega
1443	Coumara Radja	Sarvega
1444	Pete Wenzel	SeeBeyond Technology Corporation
1445	Manveen Kaur	Sun Microsystems
1446	Eve Maler	Sun Microsystems
1447	Ronald Monzillo	Sun Microsystems
1448	Jan Alexander	Systinet
1449	Symon Chang	Tibco
1450	J Weiland	US Dept of the Navy
1451	Hans Granqvist	VeriSign
1452	Phillip Hallam-Baker	Verisign
1453	Hemma Prafullchandra	VeriSign

Appendix A: Appendix B: Revision History

Rev	Date	What
00	07-Oct-04	Initial draft produced from cd-03 of version 1.0 of the profile. Version 1.1 was created to add support for SAML V2.0 Assertions.
01	19-Jan-05	Expert group draft submitted to TC.
02	17-May-2005	<ol style="list-style-type: none"> 1. Designated as V1.1 profile. 2. Incorporated resolution to issue 250 (which created the <code>TokenType</code> attribute). 3. Began transition of compatibility requirements to allow an implementation to support V1.1, V2.0, or both versions of SAML assertions. 4. Added footnote to clarify processing of bearer confirmation mechanism, and also depicted use of bearer in example.
<u>03</u>	<u>31-May-2005</u>	<ol style="list-style-type: none"> 1. <u>Applied Version 1.0 Errata</u> 2. <u>Applied comments from review.</u> 3. <u>Added section on version support and backward compatibility.</u> 4. <u>Clarified requirements with respect to bearer confirmed assertions.</u>
<u>04</u>	<u>13-June-2005</u>	<ol style="list-style-type: none"> 1. <u>Applied revised document template.</u> 2. <u>Updated contributor list (in Acknowledgements)</u>

1455

Appendix B: Notices

1456

1457

1458

1459

1460

1461

1462

1463

1464

1465

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

1466

1467

1468

1469

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

1470

~~Copyright © OASIS Open 2003-2004. All Rights Reserved.~~

1471

1472

1473

1474

1475

1476

1477

1478

1479

1480

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

1481

1482

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

1483

1484

1485

1486

1487

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