

# Web Services Security: SAML Token Profile 1.1

## OASIS Public Review Draft 01, 28 June 2005

### Document Identifier:

[wss-v1.1-spec-pr-SAMLTokenProfile-01](#)

### OASIS Identifier:

{*WSS: SOAP Message Security*}-{SAMLTokenProfile}-{1.1} (OpenOffice) (PDF) (HTML)

### Location:

Persistent: [\[persistent location\]](#)

This Version: <http://docs.oasis-open.org/wss/oasis-wss-SAMLTokenProfile-1.1>

Previous Version:<http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.0>

### Technical Committee:

OASIS Web Services Security (WSS) TC

### Chairs:

Kelvin Lawrence, IBM  
Chris kaler, Microsoft

### Editors:

Ronald Monzillo, Sun  
Chris kaler, Microsoft  
Anthony Nadalin, IBM  
Phillip Hallam-Baker, Verisign

### Abstract:

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

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

### Status:

This document was last revised or approved by the membership of the Web Services Security TC on the above date. The level of approval is also listed above. Check the current location noted above for possible later revisions of this document. This document is updated periodically on no particular schedule.

Technical Committee members should send comments on this specification to the Technical Committee's email list. Others should send comments to the Technical Committee by using the "Send A Comment" button on the Technical Committee's web page at [www.oasis-open.org/committees/wss](http://www.oasis-open.org/committees/wss).

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the Technical Committee web page ([www.oasis-open.org/committees/wss/ipr.php](http://www.oasis-open.org/committees/wss/ipr.php)).

The non-normative errata for this specification is located at [www.oasis-open.org/committees/wss](http://www.oasis-open.org/committees/wss).

## Notices

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

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

47 Copyright © The Organization for the Advancement of Structured Information Standards [OASIS] 2002-  
48 2005. All Rights Reserved.

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

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

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

## Table of Contents

64	1 Introduction.....	4
65	1.1 Goals.....	4
66	1.1.1 Non-Goals.....	4
67	2 Notations and Terminology.....	5
68	2.1 Notational Conventions.....	5
69	2.2 Namespaces.....	5
70	2.3 Terminology.....	5
71	3 Usage.....	7
72	3.1 Processing Model.....	7
73	3.2 SAML Version Differences.....	7
74	3.2.1 Assertion Identifier.....	7
75	3.2.2 Relationship of Subjects to Statements.....	7
76	3.2.3 Assertion URI Reference Replaces AuthorityBinding.....	9
77	3.2.4 Attesting Entity Identifier.....	9
78	3.3 Attaching Security Tokens.....	9
79	3.4 Identifying and Referencing Security Tokens.....	10
80	3.4.1 SAML Assertion Referenced from Header or Element.....	12
81	3.4.2 SAML Assertion Referenced from KeyInfo.....	13
82	3.4.3 SAML Assertion Referenced from SignedInfo.....	15
83	3.4.4 SAML Assertion Referenced from Encrypted Data Reference.....	16
84	3.4.5 SAML Version Support and Backward Compatability.....	16
85	3.5 Subject Confirmation of SAML Assertions.....	16
86	3.5.1 Holder-of-key Subject Confirmation Method.....	17
87	3.5.2 Sender-vouches Subject Confirmation Method.....	20
88	3.5.3 Bearer Confirmation Method.....	24
89	3.6 Error Codes.....	24
90	4 Threat Model and Countermeasures (non-normative).....	26
91	4.1 Eavesdropping.....	26
92	4.2 Replay.....	26
93	4.3 Message Insertion.....	26
94	4.4 Message Deletion.....	26
95	4.5 Message Modification.....	26
96	4.6 Man-in-the-Middle.....	27
97	5 References .....	28
98	Appendix A. Acknowledgements.....	29
99	Appendix B. Revision History.....	31
100		

---

# 101 1 Introduction

102 The [WSS: SOAP Message Security](#) specification defines a standard set of [SOAP](#) extensions that  
103 implement SOAP message authentication and encryption. This specification defines the use of Security  
104 Assertion Markup Language (SAML) assertions as security tokens from the `<wsse:Security>` header  
105 block defined by the [WSS: SOAP Message Security](#) specification.

## 106 1.1 Goals

107 The goal of this specification is to define the use of SAML V1.1 and V2.0 assertions in the context of  
108 [WSS: SOAP Message Security](#) including for the purpose of securing [SOAP](#) messages and [SOAP](#)  
109 message exchanges. To achieve this goal, this profile describes how:

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

### 113 1.1.1 Non-Goals

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

- 115 1. Defining SAML statement syntax or semantics.
- 116 2. Describing the use of SAML assertions other than for SOAP Message Security.
- 117 3. Describing the use of SAML V1.0 assertions with the [Web Services Security \(WSS\): SOAP Message](#)  
118 [Security](#) specification.

---

## 2 Notations and Terminology

119

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

120

### 2.1 Notational Conventions

121

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119](#).

122

123

124

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

125

126

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

127

128

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

129

130

131

132

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

133

### 2.2 Namespaces

134

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

135

136

137

Prefix	Namespace
s11	<a href="http://schemas.xmlsoap.org/soap/envelope/">http://schemas.xmlsoap.org/soap/envelope/</a>
s12	<a href="http://www.w3.org/2003/05/soap-envelope">http://www.w3.org/2003/05/soap-envelope</a>
ds	<a href="http://www.w3.org/2000/09/xmldsig#">http://www.w3.org/2000/09/xmldsig#</a>
xenc	<a href="http://www.w3.org/2001/04/xmlenc">http://www.w3.org/2001/04/xmlenc</a>
wsse	<a href="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-01.xsd">http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-01.xsd</a>
wsse11	TBD
wsu	<a href="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd">http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd</a>
saml	urn: oasis:names:tc:SAML:1.0:assertion
saml2	urn: oasis:names:tc:SAML:2.0:assertion
samlp	urn: oasis:names:tc:SAML:1.0:protocol

138 Table-1 Namespace Prefixes

139

### 2.3 Terminology

139

This specification employs the terminology defined in the [WSS: SOAP Message Security](#) specification. The definitions for additional terminology used in this specification appear below.

140

141

- 142 Attesting Entity – the entity that provides the confirmation evidence that will be used to establish the  
143 correspondence between the subjects and claims of SAML statements (in SAML assertions) and SOAP  
144 message content.
- 145 Confirmation Method Identifier – the value within a SAML SubjectConfirmation element that identifies the  
146 subject confirmation process to be used with the corresponding statements.
- 147 Subject Confirmation – the process of establishing the correspondence between the subject and claims of  
148 SAML statements (in SAML assertions) and SOAP message content by verifying the confirmation  
149 evidence provided by an attesting entity.
- 150 SAML Assertion Authority - A *system entity* that issues *assertions*.
- 151 Subject – A representation of the entity to which the claims in one or more SAML statements apply.

---

## 3 Usage

152

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

### 3.1 Processing Model

155

156 This specification extends the token-independent processing model defined by the [WSS: SOAP Message](#)  
157 [Security](#) specification.

158 When a receiver processes a `<wsse:Security>` header containing or referencing SAML assertions, it  
159 selects, based on its policy, the signatures and assertions that it will process. It is assumed that a  
160 receiver's signature selection policy MAY rely on semantic labeling<sup>1</sup> of  
161 `<wsse:SecurityTokenReference>` elements occurring in the `<ds:KeyInfo>` elements within the  
162 signatures. It is also assumed that the assertions selected for validation and processing will include those  
163 referenced from the `<ds:KeyInfo>` and `<ds:SignedInfo>` elements of the selected signatures.

164 As part of its validation and processing of the selected assertions, the receiver MUST<sup>2</sup> establish the  
165 relationship between the subject and claims of the SAML statements (of the referenced SAML assertions)  
166 and the entity providing the evidence to satisfy the confirmation method defined for the statements (i.e.,  
167 the attesting entity). Two methods for establishing this correspondence, `holder-of-key` and `sender-`  
168 `vouches` are described below. Systems implementing this specification MUST implement the processing  
169 necessary to support both of these subject confirmation methods.

### 3.2 SAML Version Differences

170

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

#### 3.2.1 Assertion Identifier

173

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

#### 3.2.2 Relationship of Subjects to Statements

176

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

182 A SAML V1.1 statement that contains a `<saml:Subject>` element (i.e., a subject statement) may  
183 contain a `<saml:SubjectConfirmation>` element that defines the rules for confirming the subject and  
184 claims of the statement. If present, the `<saml:SubjectConfirmation>` element occurs within the  
185 subject element, and defines one or more methods (i.e., `<saml:ConfirmationMethod>` elements) by  
186 which the statement may be confirmed and will include a `<ds:KeyInfo>`<sup>3</sup> element when any of the

---

<sup>1</sup> The optional `Usage` attribute of the `<wsse:SecurityTokenReference>` element MAY be used to associate one of 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.

<sup>2</sup> 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.

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

187 specified methods are based on demonstration of a confirmation key. The  
 188 <saml:SubjectConfirmation> element also provides for the inclusion of additional information to be  
 189 applied in the confirmation method processing via the optional <saml:SubjectConfirmationData>  
 190 element. The following example depicts a SAML V1.1 assertion containing two subject statements with  
 191 different subjects and different subject confirmation elements.

```

192 <saml:Assertion
193   ...
194   <saml:SubjectStatement>
195     <saml:Subject>
196       <saml:NameIdentifier
197         ...
198       </saml:NameIdentifier>
199       <saml:SubjectConfirmation>
200         <saml:ConfirmationMethod>
201           urn:oasis:names:tc:SAML:1.0:cm:sender-vouches
202         </saml:ConfirmationMethod>
203         <saml:ConfirmationMethod>
204           urn:oasis:names:tc:SAML:1.0:cm:holder-of-key
205         </saml:ConfirmationMethod>
206         <ds:KeyInfo>
207           <ds:KeyValue>...</ds:KeyValue>
208         </ds:KeyInfo>
209       </saml:SubjectConfirmation>
210     </saml:Subject>
211     ...
212   </saml:SubjectStatement>
213   <saml:SubjectStatement>
214     <saml:Subject>
215       <saml:NameIdentifier
216         ...
217       </saml:NameIdentifier>
218       <saml:SubjectConfirmation>
219         <saml:ConfirmationMethod>
220           urn:oasis:names:tc:SAML:1.0:cm:sender-vouches
221         </saml:ConfirmationMethod>
222       </saml:SubjectConfirmation>
223     </saml:Subject>
224     ...
225   </saml:SubjectStatement>
226   ...
227 </saml:Assertion>

```

228 A SAML V2.0 assertion may contain a single <saml2:Subject> that applies to all the statements of the  
 229 assertion. When a subject is included in A SAML V2.0 assertion, it may contain any number of  
 230 <saml2:SubjectConfirmation> elements, satisfying any of which is sufficient to confirm the subject  
 231 and all the statements of the assertion. Each <saml2:SubjectConfirmation> element identifies a  
 232 single confirmation method (by attribute value) and may include an optional  
 233 <saml2:SubjectConfirmationData> element that is used to specify optional confirmation method  
 234 independent condition attributes and to define additional method specific confirmation data. In the case of  
 235 a key dependent confirmation method, a <saml2:KeyInfoConfirmationDataType> that includes 1  
 236 or more <ds:KeyInfo> elements is included as <saml2:SubjectConfirmationData>. In this case,  
 237 each <ds:KeyInfo> element identifies a key that may be demonstrated to confirm the assertion. The  
 238 following example depicts a SAML V2.0 assertion containing a subject with multiple confirmation elements  
 239 that apply to all the statements of the assertion.

```

240 <saml2:Assertion
241   ...
242   <saml2:Subject>
243     <saml2:NameID>
244       ...
245     </saml2:NameID>
246     <saml2:SubjectConfirmation
247       Method="urn:oasis:names:tc:SAML:2.0:cm:sender-vouches">

```

```

248     <saml2:SubjectConfirmationData>
249       Address="129.148.9.42"
250     </saml2:SubjectConfirmationData>
251   </saml2:SubjectConfirmation>
252   <saml2:SubjectConfirmation
253     Method="urn:oasis:names:tc:SAML:2.0:cm:holder-of-key">
254     <saml2:KeyInfoSubjectConfirmationData>
255       <ds:KeyInfo>
256         <ds:KeyValue>...</ds:KeyValue>
257       </ds:KeyInfo>
258     </saml2:KeyInfoSubjectConfirmationData>
259   </saml2:SubjectConfirmation>
260 </saml2:Subject>
261   ...
262 <saml2:Statement>
263   ...
264 </saml2:Statement>
265
266 <saml2:Statement>
267   ...
268 </saml2:Statement>
269   ...
270
271 </saml2:Assertion>

```

### 272 3.2.3 Assertion URI Reference Replaces AuthorityBinding

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

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

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

279 <https://saml.example.edu/assertion-authority>

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

281 <https://saml.example.edu/assertion-authority?ID=abcde>

### 282 3.2.4 Attesting Entity Identifier

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

```

285 <saml2:SubjectConfirmation
286   Method="urn:oasis:names:tc:SAML:2.0:cm:sender-vouches">
287   <NameID>
288     gateway
289   </NameID>
290   <saml2:SubjectConfirmationData>
291     Address="129.148.9.42"
292   </saml2:SubjectConfirmationData>
293 </saml2:SubjectConfirmation>

```

## 294 3.3 Attaching Security Tokens

295 SAML assertions are attached to SOAP messages using [WSS: SOAP Message Security](#) by placing  
 296 assertion elements or references to assertions inside a `<wsse:Security>` header. The following  
 297 example illustrates a SOAP message containing a bearer confirmed SAML V1.1 assertion in a  
 298 `<wsse:Security>` header.

```

299 <S12:Envelope>
300   <S12:Header>
301     <wsse:Security>
302
303       <saml:Assertion
304         AssertionID="_a75adf55-01d7-40cc-929f-dbd8372ebdfc"
305         IssueInstant="2003-04-17T00:46:02Z"
306         Issuer="www.opensaml.org"
307         MajorVersion="1"
308         MinorVersion="1"
309         . . .
310       <saml:AuthenticationStatement>
311         <saml:Subject>
312           <saml:NameIdentifier
313             NameQualifier="www.example.com"
314             Format="urn:oasis:names:tc:SAML:1.1:nameid-
315 format:X509SubjectName">
316             uid=joe,ou=people,ou=saml-demo,o=baltimore.com
317           </saml:NameIdentifier>
318           <saml:SubjectConfirmation>
319             <saml:ConfirmationMethod>
320               urn:oasis:names:tc:SAML:1.0:cm:bearer
321             </saml:ConfirmationMethod>
322           </saml:SubjectConfirmation>
323         </saml:Subject>
324       </saml:AuthenticationStatement>
325
326     </saml:Assertion>
327
328   </wsse:Security>
329 </S12:Header>
330 <S12:Body>
331   . . .
332 </S12:Body>
333 </S12:Envelope>

```

### 334 3.4 Identifying and Referencing Security Tokens

335 The **WSS: SOAP Message Security** specification defines the `<wsse:SecurityTokenReference>`  
336 element for referencing security tokens. Three forms of token references are defined by this element and  
337 the element schema includes provision for defining additional reference forms should they be necessary.  
338 The three forms of token references defined by the `<wsse:SecurityTokenReference>` element are  
339 defined as follows:

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

344 When a key identifier is used to reference a SAML assertion, it MUST contain as its element value the  
345 corresponding SAML assertion identifier. The key identifier MUST also contain a `ValueType`  
346 attribute and the value of this attribute MUST be the value from Table 2 corresponding to the version  
347 of the referenced assertion. The key identifier MUST NOT include an `EncodingType`<sup>4</sup> attribute and  
348 the element content of the key identifier MUST be encoded as `xsi:string`.

349 When a key identifier is used to reference a V1.1 SAML assertion that is not contained in the same  
350 message as the key identifier, a `<saml:AuthorityBinding>` element MUST be contained in the

---

<sup>4</sup> "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).

351 <wsse:SecurityTokenReference> element containing the key identifier. The contents of the  
352 <saml:AuthorityBinding> element MUST contain values sufficient for the intended recipients of  
353 the <wsse:SecurityTokenReference> to acquire the identified assertion from the intended  
354 Authority. To this end, the value of the AuthorityKind attribute of the  
355 <saml:AuthorityBinding> element MUST be "samlp:AssertionIdReference".

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

359 A key identifier MUST NOT be used to reference a SAML V2.0 assertion if the assertion is NOT  
360 contained in the same message as the key identifier.

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

365 A reference to a SAML V2.0 assertion that is NOT contained in the same message MUST be a Direct  
366 or URI reference. In this case, the value of the URI attribute must conform to the URI syntax defined in  
367 section 3.7.5.1 of [SAMLBindV2]. That is, an HTTP or HTTPS request with a single query string  
368 parameter named ID. The reference MUST also contain a wss11:TokenType attribute and the  
369 value of this attribute MUST be the value from Table 3 identifying the assertion as a SAML V2.0  
370 security token. When a Direct reference is made to a SAML V2.0 Assertion, the Direct reference  
371 SHOULD NOT contain a ValueType attribute.

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

- 373 • An Embedded reference – a reference that encapsulates a security token.

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

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

- 378 • A SAML assertion may be referenced directly from a <wsse:Security> header element. In this  
379 case, the assertion is being conveyed by reference in the message.
- 380 • A SAML assertion may be referenced from a <ds:KeyInfo> element of a <ds:Signature>  
381 element in a <wsse:Security> header. In this case, the assertion contains a  
382 SubjectConfirmation element that identifies the key used in the signature calculation.
- 383 • A SAML assertion reference may be referenced from a <ds:Reference> element within the  
384 <ds:SignedInfo> element of a <ds:Signature> element in a <wsse:Security> header. In this  
385 case, the doubly-referenced assertion is signed by the containing signature.
- 386 • A SAML assertion reference may occur as encrypted content within an <xenc:EncryptedData>  
387 element referenced from a <xenc:DataReference> element within an <xenc:ReferenceList>  
388 element. In this case, the assertion reference (which may contain an embedded assertion) is  
389 encrypted.

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

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

395 A SAML key identifier reference MUST be used for all (local and remote) references to SAML 1.1  
396 assertions. All (local and remote) references to SAML V2.0 assertions SHOULD be by Direct reference  
397 and all remote references to V2.0 assertions MUST be by Direct reference URI. A key identifier reference  
398 MAY be used to reference a local V2.0 assertion. To maintain compatibility with [Web Services Security:  
399 SOAP Message Security 1.0](#), the practice of referencing local SAML 1.1 assertions by Direct  
400 <wsse:SecurityTokenReference> reference is not defined by this profile.

401 Every key identifier, direct, or embedded reference to a SAML assertion SHOULD contain a  
 402 `wsse11:TokenType` attribute and the value of this attribute MUST be the value from Table 3 that  
 403 identifies the type and version of the referenced security token. When the referenced assertion is a SAML  
 404 V2.0 Assertion the reference MUST contain a `wsse11:TokenType` attribute (as described above).

Assertion Version	Value
V1.1	<a href="http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.0#SAMLAssertionID">http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.0#SAMLAssertionID</a>
V2.0	<a href="http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLID">http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLID</a>

405 Table-2 Key Identifier ValueType Attribute Values

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

406 Table-3 TokenType Attribute Values

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

### 410 3.4.1 SAML Assertion Referenced from Header or Element

411 All conformant implementations MUST be able to process SAML assertion references occurring in a  
 412 `<wsse:Security>` header or in a header element other than a signature to acquire the corresponding  
 413 assertion. A conformant implementation MUST be able to process any such reference independent of the  
 414 confirmation method of the referenced assertion.

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

```

418 <S12:Envelope>
419   <S12:Header>
420     <wsse:Security>
421       <saml:Assertion
422         AssertionID="_a75adf55-01d7-40cc-929f-dbd8372ebdfc"
423         IssueInstant="2003-04-17T00:46:02Z"
424         Issuer="www.opensaml.org"
425         MajorVersion="1"
426         MinorVersion="1"
427         . . .
428       </saml:Assertion>
429       <wsse:SecurityTokenReference wsu:Id="STR1"
430         wsse11:TokenType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-
431 profile-1.1#SAMLV1.1">
432         <wsse:KeyIdentifier wsu:Id="..."
433           ValueType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-
434 profile-1.0#SAMLAssertionID">
435           _a75adf55-01d7-40cc-929f-dbd8372ebdfc
436         </wsse:KeyIdentifier>
437       </wsse:SecurityTokenReference>
438     </wsse:Security>
439   </S12:Header>
440   <S12:Body>
441     . . .
  
```

```
442     </S12:Body>
443 </S12:Envelope>
```

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

```
446 <wsse:SecurityTokenReference
447   wsu:Id="STR1"
448   wssell:TokenType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-
449   1.1#SAMLV2.0">
450   <wsse:KeyIdentifier wsu:Id="..."
451     ValueType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-
452     1.1#SAMLID">
453     _a75adf55-01d7-40cc-929f-dbd8372ebdfc
454   </wsse:KeyIdentifier>
455 </wsse:SecurityTokenReference>
```

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

```
460 <wsse:SecurityTokenReference wsu:Id="STR1"
461   wssell:TokenType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-
462   profile-1.1#SAMLV1.1">
463   <saml:AuthorityBinding>
464     Binding="urn:oasis:names:tc:SAML:1.0:bindings:SOAP-binding"
465     Location="http://www.opensaml.org/SAML-Authority"
466     AuthorityKind="samlp:AssertionIdReference"
467   </saml:AuthorityBinding>
468   <wsse:KeyIdentifier
469     wsu:Id="..."
470     ValueType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-
471     1.0#SAMLAssertionID">
472     _a75adf55-01d7-40cc-929f-dbd8372ebdfc
473   </wsse:KeyIdentifier>
474 </wsse:SecurityTokenReference>
```

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

```
477 </wsse:SecurityTokenReference
478   wsu:Id="..."
479   wssell:TokenType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-
480   profile-1.1#SAMLV2.0">
481   <wsse:Reference
482     wsu:Id="..."
483     URI="https://saml.example.edu/assertion-authority?ID=abcde">
484   </wsse:Reference>
485 </wsse:SecurityTokenReference>
```

### 486 3.4.2 SAML Assertion Referenced from KeyInfo

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

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

```
492 <ds:KeyInfo>
493   <wsse:SecurityTokenReference wsu:Id="STR1"
494     wssell:TokenType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-
495     profile-1.1#SAMLV1.1">
496     <wsse:KeyIdentifier wsu:Id="..."
497       ValueType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-
498       1.0#SAMLAssertionID">
```

```

499     _a75adf55-01d7-40cc-929f-dbd8372ebdfc
500     </wsse:KeIdentifier>
501     </wsse:SecurityTokenReference>
502 </ds:KeyInfo>

```

503 A local, V2.0 assertion may be referenced by replacing the values of the Key Identifier `ValueType` and  
504 reference `TokenType` attributes with the values defined in tables 2 and 3 (respectively) for SAML V2.0 as  
505 follows:

```

506 <ds:KeyInfo>
507   <wsse:SecurityTokenReference wsu:Id="STR1"
508     wsse11:TokenType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-
509     profile-1.1#SAMLV2.0">
510     <wsse:KeyIdentifier wsu:Id="..."
511       ValueType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-
512       1.0#SAMLID">
513       _a75adf55-01d7-40cc-929f-dbd8372ebdfc
514     </wsse:KeIdentifier>
515   </wsse:SecurityTokenReference>
516 </ds:KeyInfo>

```

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

```

521 <ds:KeyInfo>
522   <wsse:SecurityTokenReference wsu:Id="STR1"
523     wsse11:TokenType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-
524     profile-1.1#SAMLV1.1">
525     <saml:AuthorityBinding>
526       Binding="urn:oasis:names:tc:SAML:1.0:bindings:SOAP-binding"
527       Location="http://www.opensaml.org/SAML-Authority"
528       AuthorityKind="samlp:AssertionIdReference"
529     </saml:AuthorityBinding>
530     <wsse:KeyIdentifier wsu:Id="..."
531       ValueType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-
532       1.0#SAMLAssertionID">
533       _a75adf55-01d7-40cc-929f-dbd8372ebdfc
534     </wsse:KeyIdentifier>
535   </wsse:SecurityTokenReference>
536 </ds:KeyInfo>

```

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

```

539 <ds:KeyInfo>
540   <wsse:SecurityTokenReference
541     wsu:id="STR1"
542     wsse11:TokenType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-
543     profile-1.1#SAMLV2.0">
544     <wsse:Reference
545       wsu:id="..."
546       URI="https://saml.example.edu/assertion-authority?ID=abcde">
547     </wsse:Reference>
548   </wsse:SecurityTokenReference>
549 </ds:KeyInfo>

```

550 `<ds:KeyInfo>` elements may also occur in `<xenc:EncryptedData>` and `<xenc:EncryptedKey>`  
551 elements where they serve to identify the encryption key. `<ds:KeyInfo>` elements may also occur in  
552 SAML SubjectConfirmation elements where they identify a key that MUST be demonstrated to  
553 confirm the subject of the corresponding statement(s).

554 Conformant implementations of this profile are NOT required to process SAML assertion references  
555 occurring within the `<ds:KeyInfo>` elements within `<xenc:EncryptedData>`,  
556 `<xenc:EncryptedKey>`, or SAML SubjectConfirmation elements.

### 557 3.4.3 SAML Assertion Referenced from SignedInfo

558 Independent of the confirmation method of the referenced assertion, all conformant implementations  
559 MUST be able to process SAML assertions referenced by <wsse:SecurityTokenReference> from  
560 <ds:Reference> elements within the <ds:SignedInfo> element of a <ds:Signature> element in a  
561 <wsse:Security> header. Embedded references may be digested directly, thus effectively digesting the  
562 encapsulated assertion. Other <wsse:SecurityTokenReference> forms must be dereferenced for  
563 the referenced assertion to be digested.

564 The core specification, [WSS: SOAP Message Security](#), defines the STR Dereference transform to cause  
565 the replacement (in the digest stream) of a <wsse:SecurityTokenReference> with the contents of  
566 the referenced token. The STR Dereference transform MUST be specified and applied to digest any  
567 SAML assertion that is referenced by a <wsse:SecurityTokenReference> that is not an embedded  
568 reference. The STR Dereference transform SHOULD NOT be applied to an embedded reference.

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

```
572 <wsse:SecurityTokenReference wsu:Id="STR1"  
573   wss11:TokenType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-  
574   profile-1.1#SAMLV1.1">  
575   <saml:AuthorityBinding>  
576     Binding="urn:oasis:names:tc:SAML:1.0:bindings:SOAP-binding"  
577     Location="http://www.opensaml.org/SAML-Authority"  
578     AuthorityKind="samlp:AssertionIdReference"  
579   </saml:AuthorityBinding>  
580   <wsse:KeyIdentifier wsu:Id="..."  
581     ValueType="http://docs.oasis-open.org/wss/oasis-2004XX-wss-saml-token-  
582   profile-1.0#SAMLAssertionID">  
583     a75adf55-01d7-40cc-929f-dbd8372ebdfc  
584   </wsse:KeyIdentifier>  
585 </wsse:SecurityTokenReference>  
586 . . .  
587 <ds:SignedInfo>  
588   <ds:CanonicalizationMethod  
589     Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />  
590   <ds:SignatureMethod  
591     Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />  
592   <ds:Reference URI="#STR1">  
593     <Transforms>  
594       <ds:Transform  
595         Algorithm="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-  
596   soap-message-security-1.0#STR-Transform" />  
597       <wsse:TransformationParameters>  
598         <ds:CanonicalizationMethod  
599           Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />  
600       </wsse:TransformationParameters>  
601     </ds:Transform>  
602   </Transforms>  
603   <ds:DigestMethod  
604     Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />  
605   <ds:DigestValue>...</ds:DigestValue>  
606 </ds:Reference>  
607 </ds:SignedInfo>
```

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

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

### 617 **3.4.4 SAML Assertion Referenced from Encrypted Data Reference**

618 Independent of the confirmation method of the referenced assertion, all conformant implementations  
619 MUST be able to process SAML assertion references occurring as encrypted content within the  
620 <xenc:EncryptedData> elements referenced by Id from the <xenc:DataReference> elements of  
621 <xenc:ReferenceList> elements. An <xenc:ReferenceList> element may occur either as a top-  
622 level element in a <wsse:Security> header, or embedded within an <xenc:EncryptedKey>  
623 element. In either case, the <xenc:ReferenceList> identifies the encrypted content.

624 Such references are similar in format to the references that MAY appear in the <ds:Reference>  
625 element within <ds:SignedInfo>, except the STR Dereference transform does not apply. As shown in  
626 the following example, an encrypted <wsse:SecurityTokenReference> (which may contain an  
627 embedded assertion) is referenced from an <xenc:DataReference> by including the identifier of the  
628 <xenc:EncryptedData> element that contains the encrypted <wsse:SecurityTokenReference>  
629 in the <xenc:DataReference>.

```
630 <xenc:EncryptedData Id="EncryptedSTR1">  
631 <ds:KeyInfo>  
632 . . .  
633 </ds:KeyInfo>  
634 <xenc:CipherData>  
635 <xenc:CipherValue>...</xenc:CipherValue>  
636 </xenc:CipherData>  
637 /xenc:EncryptedData>  
638 <xenc:ReferenceList>  
639 <xenc:DataReference URI="#EncryptedSTR1"/>  
640 </xenc:ReferenceList>
```

### 641 **3.4.5 SAML Version Support and Backward Compatibility**

642 An implementation of this profile MUST satisfy all of its requirements with respect to either or both SAML  
643 V1.1 or SAML V2.0 Assertions. An implementation that satisfies the requirements of this profile with  
644 respect to SAML V1.1 assertions MUST be able to fully interoperate with any fully compatible  
645 implementation of version 1.0 of this profile.

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

## 651 **3.5 Subject Confirmation of SAML Assertions**

652 The SAML profile of [WSS: SOAP Message Security](#) requires that systems support the holder-of-key and  
653 sender-vouches methods of subject confirmation. It is strongly RECOMMENDED that an XML signature  
654 be used to establish the relationship between the message and the statements of the attached assertions.  
655 This is especially RECOMMENDED whenever the SOAP message exchange is conducted over an  
656 unprotected transport.

657 Any processor of SAML assertions MUST conform to the required validation and processing rules defined  
658 in the corresponding SAML specification including the validation of assertion signatures, the processing of  
659 <saml:Condition> elements within assertions, and the processing of  
660 <saml2:SubjectConfirmationData> attributes. [\[SAMLCoreV1\]](#) defines the validation and  
661 processing rules for V1.1 assertions, while [\[SAMLCoreV2\]](#) is authoritative for V2.0 assertions.

662 The following table enumerates the mandatory subject confirmation methods and summarizes their  
663 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.

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

### 669 3.5.1 Holder-of-key Subject Confirmation Method

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

#### 673 3.5.1.1 Attesting Entity

674 An attesting entity demonstrates that it is authorized to act as the subject of a holder-of-key confirmed  
 675 SAML statement by demonstrating knowledge of any key identified in a holder-of-key  
 676 `SubjectConfirmation` element associated with the statement by the assertion containing the  
 677 statement. Statements attested for by the holder-of-key method MUST be associated, within their  
 678 containing assertion, with one or more holder-of-key `SubjectConfirmation` elements.

679 The `SubjectConfirmation` elements MUST include a `<ds:KeyInfo>` element that identifies a public  
 680 or secret key<sup>5</sup> that can be used to confirm the identity of the subject.

681 To satisfy the associated confirmation method processing to be performed by the message receiver, the  
 682 attesting entity MUST demonstrate knowledge of the confirmation key. The attesting entity MAY  
 683 accomplish this by using the confirmation key to sign content within the message and by including the  
 684 resulting `<ds:Signature>` element in the `<wsse:Security>` header. `<ds:Signature>` elements  
 685 produced for this purpose MUST conform to the `canonicalization` and token pre-pending rules  
 686 defined in the [WSS: SOAP Message Security](#) specification.

---

<sup>5</sup>[\[SAMLCoreV1\]](#) 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)`.

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

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

### 694 3.5.1.2 Receiver

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

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

### 703 3.5.1.3 Example V1.1

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

```
707 <?xml version="1.0" encoding="UTF-8"?>
708 <S12:Envelope>
709   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
710   xmlns:xsd="http://www.w3.org/2001/XMLSchema">
711   <S12:Header>
712     <wsse:Security>
713       <saml:Assertion
714         AssertionID="_a75adf55-01d7-40cc-929f-dbd8372ebdfc"
715         IssueInstant="2005-05-27T16:53:33.173Z"
716         Issuer="www.opensaml.org"
717         MajorVersion="1"
718         MinorVersion="1"
719         xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion">
720         <saml:Conditions>
721           NotBefore="2005-05-27T16:53:33.173Z"
722           NotOnOrAfter="2005-05-27T16:58:33.17302Z"/>
723         <saml:AttributeStatement>
724           <saml:Subject>
725             <saml:NameIdentifier
726               NameQualifier="www.example.com"
727               Format="urn:oasis:names:tc:SAML:1.1:nameid-
728 format:X509SubjectName">
729               uid=joe,ou=people,ou=saml-demo,o=baltimore.com
730             </saml:NameIdentifier>
731             <saml:SubjectConfirmation>
732               <saml:ConfirmationMethod>
733                 urn:oasis:names:tc:SAML:1.0:cm:holder-of-key
734               </saml:ConfirmationMethod>
735               <ds:KeyInfo>
736                 <ds:KeyValue>...</ds:KeyValue>
737               </ds:KeyInfo>
738             </saml:SubjectConfirmation>
739           </saml:Subject>
740           <saml:Attribute
741             AttributeName="MemberLevel"
742
```

```

743         AttributeNamespace="http://www.oasis-
744 open.org/Catalyst2002/attributes">
745         <saml:AttributeValue>gold</saml:AttributeValue>
746     </saml:Attribute>
747     <saml:Attribute
748         AttributeName="E-mail"
749         AttributeNamespace="http://www.oasis-
750 open.org/Catalyst2002/attributes">
751         <saml:AttributeValue>joe@yahoo.com</saml:AttributeValue>
752     </saml:Attribute>
753 </saml:AttributeStatement>
754 <ds:Signature>...</ds:Signature>
755 </saml:Assertion>
756
757 <ds:Signature>
758     <ds:SignedInfo>
759         <ds:CanonicalizationMethod
760             Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
761         <ds:SignatureMethod
762             Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
763         <ds:Reference
764             URI="#MsgBody">
765             <ds:DigestMethod
766                 Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
767             <ds:DigestValue>GyGsF0Pi4xPU...</ds:DigestValue>
768         </ds:Reference>
769     </ds:SignedInfo>
770     <ds:SignatureValue>HJJWbvqW9E84vJVQk...</ds:SignatureValue>
771     <ds:KeyInfo>
772         <wsse:SecurityTokenReference wsu:Id="STR1"
773             wss11:TokenType="http://docs.oasis-open.org/wss/oasis-wss-saml-
774 token-profile-1.1#SAMLV1.1">
775             <wsse:KeyIdentifier wsu:Id="..."
776                 ValueType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-
777 profile-1.0#SAMLAssertionID">
778                 _a75adf55-01d7-40cc-929f-dbd8372ebdfc
779             </wsse:KeyIdentifier>
780         </wsse:SecurityTokenReference>
781     </ds:KeyInfo>
782 </ds:Signature>
783 </wsse:Security>
784 </S12:Header>
785
786 <S12:Body wsu:Id="MsgBody">
787     <ReportRequest>
788         <TickerSymbol>SUNW</TickerSymbol>
789     </ReportRequest>
790 </S12:Body>
791 </S12:Envelope>

```

### 792 3.5.1.4 Example V2.0

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

```

796 <?xml version="1.0" encoding="UTF-8"?>
797 <S12:Envelope>
798     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
799     xmlns:xsd="http://www.w3.org/2001/XMLSchema">
800     <S12:Header>
801
802         <wsse:Security>
803             <saml2:Assertion
804                 ...

```

```

805         ID="_a75adf55-01d7-40cc-929f-dbd8372ebdfc"
806         ...>
807     <saml2:subject>
808     <saml2:NameID>
809     ...
810     </saml2:NameID>
811     <saml2:SubjectConfirmation
812     Method="urn:oasis:names:tc:SAML:2.0:cm:holder-of-key">
813     <saml2:KeyInfoSubjectConfirmationData>
814     <ds:KeyInfo>
815     <ds:KeyValue>...</ds:KeyValue>
816     </ds:KeyInfo>
817     </saml2:KeyInfoSubjectConfirmationData>
818     <saml2:SubjectConfirmation>
819     </saml2:Subject>
820     <saml2:Statement>
821     ...
822     </saml2:Statement>
823     <ds:Signature>...</ds:Signature>
824     </saml2:Assertion>
825
826     <ds:Signature>
827     <ds:SignedInfo>
828     <ds:CanonicalizationMethod
829     Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
830     <ds:SignatureMethod
831     Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
832     <ds:Reference
833     URI="#MsgBody">
834     <ds:DigestMethod
835     Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
836     <ds:DigestValue>GyGsF0Pi4xPU...</ds:DigestValue>
837     </ds:Reference>
838     </ds:SignedInfo>
839     <ds:SignatureValue>HJJWbvqW9E84vJVQk...</ds:SignatureValue>
840     <ds:KeyInfo>
841     <wsse:SecurityTokenReference wsu:Id="STR1"
842     wsse11:TokenType="http://docs.oasis-open.org/wss/oasis-wss-saml-
843 token-profile-1.1#SAMLV2.0">
844     <wsse:KeyIdentifier wsu:Id="..."
845     ValueType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-
846 profile-1.1#SAMLID">
847     _a75adf55-01d7-40cc-929f-dbd8372ebdfc
848     </wsse:KeyIdentifier>
849     </wsse:SecurityTokenReference>
850     </ds:KeyInfo>
851     </ds:Signature>
852     </wsse:Security>
853     </S12:Header>
854
855     <S12:Body wsu:Id="MsgBody">
856     <ReportRequest>
857     <TickerSymbol>SUNW</TickerSymbol>
858     </ReportRequest>
859     </S12:Body>
860 </S12:Envelope>

```

### 861 3.5.2 Sender-vouches Subject Confirmation Method

862 The following sections describe the sender-vouches method of establishing the correspondence between  
863 a SOAP message and the SAML assertions added to the SOAP message according to the SAML profile  
864 of [WSS: SOAP Message Security](#).

### 865 3.5.2.1 Attesting Entity

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

870 To satisfy the associated confirmation method processing of the receiver, the attesting entity MUST  
871 protect the vouched for SOAP message content such that the receiver can determine when it has been  
872 altered by another party. The attesting entity MUST also cause the vouched for statements (as necessary)  
873 and their binding to the message contents to be protected such that unauthorized modification can be  
874 detected. The attesting entity MAY satisfy these requirements by including in the corresponding  
875 `<wsse:Security>` header a `<ds:Signature>` element that it prepares by using its key to sign the  
876 relevant message content and assertions. As defined by the [XML Signature](#) specification, the attesting  
877 entity MAY identify its key by including a `<ds:KeyInfo>` element within the `<ds:Signature>` element.

878 A `<ds:Signature>` element produced for this purpose MUST conform to the canonicalization and  
879 token pre-pending rules defined in the [WSS: SOAP Message Security](#) specification.

### 880 3.5.2.2 Receiver

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

### 886 3.5.2.3 Example V1.1

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

890 The assertion referenced by "STR1" is not included in the message. "STR1" is referenced by  
891 `<ds:Reference>` from `<ds:SignedInfo>`. The `ds:Reference` includes the STR-transform to  
892 cause the assertion, not the `<SecurityTokenReference>` to be included in the digest calculation.  
893 "STR1" includes a `<saml:AuthorityBinding>` element that utilizes the remote assertion referencing  
894 technique depicted in the example of section 3.3.3.

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

```
898 <?xml version="1.0" encoding="UTF-8"?>
899 <S12:Envelope>
900   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
901   xmlns:xsd="http://www.w3.org/2001/XMLSchema">
902   <S12:Header>
903     <wsse:Security>
904
905       <saml:Assertion
906         AssertionID="_a75adf55-01d7-40cc-929f-dbd8372ebdfc"
907         IssueInstant="2005-05-27T16:53:33.173Z"
908         Issuer="www.opensaml.org"
909         MajorVersion="1"
910         MinorVersion="1"
911         xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion">
912       <saml:Conditions>
913         NotBefore="2005-05-27T16:53:33.173Z"
914         NotOnOrAfter="2005-05-27T16:58:33.173Z"/>
915       <saml:AttributeStatement>
916         <saml:Subject>
917           <saml:NameIdentifier
```

```

918         NameQualifier="www.example.com"
919         Format="urn:oasis:names:tc:SAML:1.1:nameid-
format:X509SubjectName">
920             uid=proxy,ou=system,ou=saml-demo,o=baltimore.com
921         </saml:NameIdentifier>
922         <saml:SubjectConfirmation>
923             <saml:ConfirmationMethod>
924                 urn:oasis:names:tc:SAML:1.0:cm:holder-of-key
925             </saml:ConfirmationMethod>
926             <ds:KeyInfo>
927                 <ds:KeyValue>...</ds:KeyValue>
928             </ds:KeyInfo>
929         </saml:SubjectConfirmation>
930     </saml:Subject>
931     <saml:Attribute
932     . . .
933     </saml:Attribute>
934     . . .
935     </saml:AttributeStatement>
936 </saml:Assertion>
937
938     <wsse:SecurityTokenReference wsu:Id="STR1">
939         wssell:TokenType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-
profile-1.1#SAMLV1.1">
940     <saml:AuthorityBinding>
941         Binding="urn:oasis:names:tc:SAML:1.0:bindings:SOAP-binding"
942         Location="http://www.opensaml.org/SAML-Authority"
943         AuthorityKind="samlp:AssertionIdReference"
944     </saml:AuthorityBinding>
945     <wsse:KeyIdentifier wsu:Id="..."
946         ValueType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-
profile-1.0#SAMLAssertionID">
947         a75adf55-01d7-40cc-929f-dbd8372ebdbe
948     </wsse:KeyIdentifier>
949 </wsse:SecurityTokenReference>
950
951     <ds:Signature>
952     <ds:SignedInfo>
953         <ds:CanonicalizationMethod
954             Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
955     <ds:SignatureMethod
956         Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
957     <ds:Reference URI="#STR1">
958         <Transforms>
959             <ds:Transform
960                 Algorithm="http://docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-soap-message-security-1.0#STR-Transform" />
961             <wsse:TransformationParameters>
962                 <ds:CanonicalizationMethod
963                     Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
964             </wsse:TransformationParameters>
965         </ds:Transform>
966     </Transforms>
967     <ds:DigestMethod
968         Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
969     <ds:DigestValue>...</ds:DigestValue>
970 </ds:Reference>
971     <ds:Reference URI="#MsgBody">
972         <ds:DigestMethod
973             Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
974         <ds:DigestValue>...</ds:DigestValue>
975     </ds:Reference>
976 </ds:SignedInfo>
977     <ds:SignatureValue>HJJWbvqW9E84vJVQk...</ds:SignatureValue>
978     <ds:KeyInfo>
979         <wsse:SecurityTokenReference wsu:Id="STR2"

```

```

984         wss11:TokenType="http://docs.oasis-open.org/wss/oasis-wss-saml-
985 token-profile-1.1#SAMLV1.1">
986         <wsse:KeyIdentifier wsu:Id="..."
987           ValueType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-
988 profile-1.0#SAMLAssertionID">
989           _a75adf55-01d7-40cc-929f-dbd8372ebdfc
990         </wsse:KeyIdentifier>
991       </wsse:SecurityTokenReference>
992     </ds:KeyInfo>
993   </ds:Signature>
994 </wsse:Security>
995 </S12:Header>
996
997 <S12:Body wsu:Id="MsgBody">
998   <ReportRequest>
999     <TickerSymbol>SUNW</TickerSymbol>
1000   </ReportRequest>
1001 </S12:Body>
1002 </S12:Envelope>

```

### 1003 3.5.2.4 Example V2.0

1004 The following example illustrates the mapping of the preceding example to SAML V2.0 assertions.

```

1005 <?xml version="1.0" encoding="UTF-8"?>
1006 <S12:Envelope>
1007   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
1008   xmlns:xsd="http://www.w3.org/2001/XMLSchema">
1009   <S12:Header>
1010
1011     <wsse:Security>
1012       <saml2:Assertion
1013         ...
1014         ID="_a75adf55-01d7-40cc-929f-dbd8372ebdfc"
1015         ...>
1016         <saml2:subject>
1017           <saml2:NameID>
1018             ...
1019           </saml2:NameID>
1020           <saml2:SubjectConfirmation
1021             Method="urn:oasis:names:tc:SAML:2.0:cm:holder-of-key">
1022             <saml2:KeyInfoSubjectConfirmationData>
1023               <ds:KeyInfo>
1024                 <ds:KeyValue>...</ds:KeyValue>
1025               </ds:KeyInfo>
1026             </saml2:KeyInfoSubjectConfirmationData>
1027           </saml2:SubjectConfirmation>
1028         </saml2:Subject>
1029         <saml2:Statement>
1030           ...
1031         </saml2:Statement>
1032         <ds:Signature>...</ds:Signature>
1033       </saml2:Assertion>
1034
1035       <wsse:SecurityTokenReference wsu:Id="STR1"
1036         wss11:TokenType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-
1037 profile-1.1#SAMLV2.0">
1038         <wsse:Reference wsu:Id="..."
1039           URI="https://www.opensaml.org?_a75adf55-01d7-40cc-929f-
1040 dbd8372ebdfc">
1041         </wsse:Reference>
1042       </wsse:SecurityTokenReference>
1043
1044     <ds:Signature>
1045       <ds:SignedInfo>
1046         <ds:CanonicalizationMethod

```

```

1047     Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
1048   <ds:SignatureMethod
1049     Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
1050   <ds:Reference URI="#STR1">
1051     <Transforms>
1052       <ds:Transform
1053         Algorithm="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
1054 wss-soap-message-security-1.0#STR-Transform"/>
1055       <wsse:TransformationParameters>
1056         <ds:CanonicalizationMethod
1057           Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
1058         </wsse:TransformationParameters>
1059       </ds:Transform>
1060     </Transforms>
1061   </ds:SignatureMethod>
1062   <ds:DigestMethod
1063     Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
1064   <ds:DigestValue>...</ds:DigestValue>
1065 </ds:Reference>
1066 <ds:Reference URI="#MsgBody">
1067   <ds:DigestMethod
1068     Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
1069   <ds:DigestValue>...</ds:DigestValue>
1070 </ds:Reference>
1071 </ds:SignedInfo>
1072 <ds:SignatureValue>HJJWbvqW9E84vJVQk...</ds:SignatureValue>
1073 <ds:KeyInfo>
1074   <wsse:SecurityTokenReference wsu:Id="STR2"
1075     wss11:TokenType="http://docs.oasis-open.org/wss/oasis-wss-saml-
1076 token-profile-1.1#SAMLV2.0">
1077     <wsse:KeyIdentifier wsu:Id="..."
1078       ValueType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-
1079 profile-1.1#SAMLID">
1080       a75adf55-01d7-40cc-929f-dbd8372ebdfc
1081     </wsse:KeyIdentifier>
1082   </wsse:SecurityTokenReference>
1083 </ds:KeyInfo>
1084 </ds:Signature>
1085 </wsse:Security>
1086 </S12:Header>
1087
1088 <S12:Body wsu:Id="MsgBody">
1089   <ReportRequest>
1090     <TickerSymbol>SUNW</TickerSymbol>
1091   </ReportRequest>
1092 </S12:Body>
1093 </S12:Envelope>

```

### 1094 3.5.3 Bearer Confirmation Method

1095 This profile does NOT require message receivers to establish the relationship between a received  
1096 message and the statements of any bearer confirmed (i.e., confirmation method  
1097 urn:oasis:names:tc:SAML:1.0:cm:bearer) assertions conveyed or referenced from the message.  
1098 Conformant implementations of this profile MUST be able to process references and convey bearer  
1099 assertions within <wsse:Security> headers. Any additional processing requirements that pertain  
1100 specifically to bearer confirmed assertions are outside the scope of this profile.

### 1101 3.6 Error Codes

1102 When a system that implements the SAML token profile of [WSS: SOAP Message Security](#) does not  
1103 perform its normal processing because of an error detected during the processing of a security header, it  
1104 MAY choose to report the cause of the error using the SOAP fault mechanism. The SAML token profile of  
1105 [WSS: SOAP Message Security](#) does not require that SOAP faults be returned for such errors, and

1106 systems that choose to return faults SHOULD take care not to introduce any security vulnerabilities as a  
1107 result of the information returned in error responses.

1108 Systems that choose to return faults SHOULD respond with the error codes and fault strings defined in the  
1109 [WSS: SOAP Message Security](#) specification. The RECOMMENDED correspondence between the  
1110 common assertion processing failures and the error codes defined in [WSS: SOAP Message Security](#) are  
1111 defined in the following table:

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

1112 The preceding table defines fault codes in a form suitable for use with SOAP 1.1. The [WSS: SOAP](#)  
1113 [Message Security](#) specification describes how to map SOAP 1.1 fault constructs to the SOAP 1.2 fault  
1114 constructs.

---

## 1115 4 Threat Model and Countermeasures (non- 1116 normative)

1117 This document defines the mechanisms and procedures for securely attaching SAML assertions to SOAP  
1118 messages. SOAP messages are used in multiple contexts, specifically including cases where the  
1119 message is transported without an active session, the message is persisted, or the message is routed  
1120 through a number of intermediaries. Such a general context of use suggests that users of this profile must  
1121 be concerned with a variety of threats.

1122 In general, the use of SAML assertions with [WSS: SOAP Message Security](#) introduces no new threats  
1123 beyond those identified for SAML or by the [WSS: SOAP Message Security](#) specification. The following  
1124 sections provide an overview of the characteristics of the threat model, and the countermeasures that  
1125 SHOULD be adopted for each perceived threat.

### 1126 4.1 Eavesdropping

1127 Eavesdropping is a threat to the SAML token profile of [WSS: SOAP Message Security](#) in the same  
1128 manner as it is a threat to any network protocol. The routing of SOAP messages through intermediaries  
1129 increases the potential incidences of eavesdropping. Additional opportunities for eavesdropping exist  
1130 when SOAP messages are persisted.

1131 To provide maximum protection from eavesdropping, assertions, assertion references, and sensitive  
1132 message content SHOULD be encrypted such that only the intended audiences can view their content.  
1133 This approach removes threats of eavesdropping in transit, but MAY not remove risks associated with  
1134 storage or poor handling by the receiver.

1135 Transport-layer security MAY be used to protect the message and contained SAML assertions and/or  
1136 references from eavesdropping while in transport, but message content MUST be encrypted above the  
1137 transport if it is to be protected from eavesdropping by intermediaries.

### 1138 4.2 Replay

1139 Reliance on authority-protected (e.g., signed) assertions with a holder-of-key subject confirmation  
1140 mechanism precludes all but a holder of the key from binding the assertions to a SOAP message.  
1141 Although this mechanism effectively restricts data origin to a holder of the confirmation key, it does not, by  
1142 itself, provide the means to detect the capture and resubmission of the message by other parties.

1143 Assertions that contain a sender-vouches confirmation mechanism introduce another dimension to replay  
1144 vulnerability if the assertions impose no restriction on the entities that may use or reuse the assertions.

1145 Replay attacks can be detected by receivers if message senders include additional message identifying  
1146 information (e.g., timestamps, nonces, and or recipient identifiers) within origin-protected message  
1147 content and receivers check this information against previously received values.

### 1148 4.3 Message Insertion

1149 The SAML token profile of [WSS: SOAP Message Security](#) is not vulnerable to message insertion attacks.

### 1150 4.4 Message Deletion

1151 The SAML token profile of [WSS: SOAP Message Security](#) is not vulnerable to message deletion attacks.

### 1152 4.5 Message Modification

1153 Messages constructed according to this specification are protected from message modification if receivers  
1154 can detect unauthorized modification of relevant message content. Therefore, it is strongly  
1155 RECOMMENDED that all relevant and immutable message content be signed by an attesting entity.  
1156 Receivers SHOULD only consider the correspondence between the subject of the SAML assertions and

1157 the SOAP message content to have been established for those portions of the message that are protected  
1158 by the attesting entity against modification by another entity.

1159 To ensure that message receivers can have confidence that received assertions have not been forged or  
1160 altered since their issuance, SAML assertions appearing in or referenced from `<wsse:Security>`  
1161 header elements MUST be protected against unauthorized modification (e.g., signed) by their issuing  
1162 authority or the attesting entity (as the case warrants). It is strongly RECOMMENDED that an attesting  
1163 entity sign any `<saml:Assertion>` elements that it is attesting for and that are not signed by their  
1164 issuing authority.

1165 Transport-layer security MAY be used to protect the message and contained SAML assertions and/or  
1166 assertion references from modification while in transport, but signatures are required to extend such  
1167 protection through intermediaries.

## 1168 **4.6 Man-in-the-Middle**

1169 Assertions with a holder-of-key subject confirmation method are not vulnerable to a MITM attack.  
1170 Assertions with a sender-vouches subject confirmation method are vulnerable to MITM attacks to the  
1171 degree that the receiver does not have a trusted binding of key to the attesting entity's identity.

---

## 5 References

1172

- 1173 **[GLOSSARY]** Informational RFC 2828, "[Internet Security Glossary](#)," May 2000.
- 1174 **[KEYWORDS]** S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels," [RFC 2119](#), Harvard University, March 1997
- 1175
- 1176 **[SAMLBindV1]** Oasis Standard, E. Maler, P.Mishra, and R. Philpott (Editors), [Bindings and Profiles for the OASIS Security Assertion Markup Language \(SAML\) V1.1](#), September 2003.
- 1177
- 1178
- 1179 **[SAMLBindV2]** Oasis Standard, S. Cantor, F. Hirsch, J. Kemp, R. Philpott, E. Maler (Editors), [Bindings for the OASIS Security Assertion Markup Language \(SAML\) V2.0](#), March 2005.
- 1180
- 1181
- 1182 **[SAMLCoreV1]** Oasis Standard, E. Maler, P.Mishra, and R. Philpott (Editors), [Assertions and Protocols for the OASIS Security Assertion Markup Language \(SAML\) V1.1](#), September 2003.
- 1183
- 1184
- 1185 **[SAMLCoreV2]** Oasis Standard, S. Cantor, J. Kemp, R. Philpott, E. Maler (Editors), [Assertions and Protocol for the OASIS Security Assertion Markup Language \(SAML\) V2.0](#), March 2005.
- 1186
- 1187
- 1188 **[SOAP]** W3C Note, "[SOAP: Simple Object Access Protocol 1.1](#)," 08 May 2000.
- 1189 W3C Working Draft, Nilo Mitra (Editor), [SOAP Version 1.2 Part 0: Primer](#), June 2002.
- 1190
- 1191 W3C Working Draft, Martin Gudgin, Marc Hadley, Noah Mendelsohn, Jean-
- 1192 Jacques Moreau, Henrik Frystyk Nielsen (Editors), [SOAP Version 1.2 Part 1: Messaging Framework](#), June 2002.
- 1193
- 1194 W3C Working Draft, Martin Gudgin, Marc Hadley, Noah Mendelsohn, Jean-
- 1195 Jacques Moreau, Henrik Frystyk Nielsen (Editors), [SOAP Version 1.2 Part 2: Adjuncts](#), June 2002.
- 1196
- 1197 **[URI]** T. Berners-Lee, R. Fielding, L. Masinter, "Uniform Resource Identifiers (URI): Generic Syntax," [RFC 2396](#), MIT/LCS, U.C. Irvine, Xerox Corporation, August 1998.
- 1198
- 1199
- 1200 **[WS-SAML]** Contribution to the WSS TC, P. Mishra (Editor), [WS-Security Profile of the Security Assertion Markup Language \(SAML\) Working Draft 04](#), Sept 2002.
- 1201
- 1202 **[WSS: SAML Token Profile]** Oasis Standard, P. Hallem-Baker, A. Nadalin, C. Kaler, R. Monzillo (Editors), [Web Services Security: SAML Token Profile 1.0](#), December 2004.
- 1203
- 1204 **[WSS: SOAP Message Security]** Oasis Standard, A. Nadalin, C.Kaler, P. Hallem-Baker, R. Monzillo (Editors), [Web Services Security: SOAP Message Security 1.0 \(WS-Security 2004\)](#), August 2003.
- 1205
- 1206
- 1207 **[XML-ns]** W3C Recommendation, "[Namespaces in XML](#)," 14 January 1999.
- 1208 **[XML Signature]** W3C Recommendation, "[XML Signature Syntax and Processing](#)," 12 February 2002.
- 1209
- 1210 **[XML Token]** Contribution to the WSS TC, Chris Kaler (Editor),
- 1211 [WS-Security Profile for XML-based Tokens](#), August 2002.

---

## Appendix A. Acknowledgements

Maneesh Sahu	Actional Corp
Gene Thurston	AmberPoint
Frank Siebenlist	Argonne National Laboratory
Hal Lockhart	BEA Systems, Inc.
Corinna Witt	BEA Systems, Inc.
Steve Anderson	BMC Software
Davanum Srinivas	Computer Associates
Rich Levinson	Computer Associates
Thomas DeMartini	ContentGuard
Guillermo Lao	ContentGuard
Merlin Hughes	Cybertrust
Rich Salz	DataPower
Sam Wei	Documentum
Tim Moses	Entrust
Carolina Canales-Valenzuela	Ericsson
Dana Kaufman	Forum Systems, Inc.
Toshihiro Nishimura	Fujitsu
Kefeng Chen	GeoTrust
Irving Reid	Hewlett-Packard
Kojiro Nakayama	Hitachi
Paula Austel	IBM
Derek Fu	IBM
Maryann Hondo	IBM
Kelvin Lawrence	IBM
Hiroshi Maruyama	IBM
Michael McIntosh	IBM
Anthony Nadalin	IBM
Nataraj Nagaratnam	IBM
Ron Williams	IBM
Don Flinn	Individual
Jerry Schwarz	Individual
Bob Morgan	Internet2
Kate Cherry	Lockheed Martin
Paul Cotton	Microsoft Corporation
Vijay Gajjala	Microsoft Corporation
Alan Geller	Microsoft Corporation
Chris Kaler	Microsoft Corporation
Jeff Hodges	Neustar
Frederick Hirsch	Nokia

Senthil Sengodan	Nokia
Abbie Barbir	Nortel Networks
Lloyd Burch	Novell
Charles Knouse	Oblix
Vamsi Motukuru	Oracle
Ramana Turlapati	Oracle
Prateek Mishra	Principal Identity
Andrew Nash	Reactivity
Ben Hammond	RSA Security
Rob Philpott	RSA Security
Martijn de Boer	SAP
Blake Dournaee	Sarvega
Coumara Radja	Sarvega
Pete Wenzel	SeeBeyond Technology Corporation
Manveen Kaur	Sun Microsystems
Eve Maler	Sun Microsystems
Ronald Monzillo	Sun Microsystems
Jan Alexander	Systinet
Symon Chang	Tibco
J Weiland	US Dept of the Navy
Hans Granqvist	VeriSign
Phillip Hallam-Baker	VeriSign
Hemma Prafullchandra	VeriSign

1213

## Appendix B. Revision History

Rev	Date	What
00	07-Oct-2004	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"> <li>1. Designated as V1.1 profile.</li> <li>2. Incorporated resolution to issue 250 (which created the <code>TokenType</code> attribute).</li> <li>3. Began transition of compatibility requirements to allow an implementation to support V1.1, V2.0, or both versions of SAML assertions.</li> <li>4. Added footnote to clarify processing of bearer confirmation mechanism, and also depicted use of bearer in example.</li> </ol>
03	31-May-2005	<ol style="list-style-type: none"> <li>1. Applied Version 1.0 Errata</li> <li>2. Applied comments from review.</li> <li>3. Added section on version support and backward compatibility.</li> <li>4. Clarified requirements with respect to bearer confirmed assertions.</li> </ol>
04	13-June-2005	<ol style="list-style-type: none"> <li>1. Applied revised document template.</li> <li>2. Updated contributor list (in Acknowledgements)</li> </ol>
CD-01	14-June-2005	Designated as Committee Draft
PR-01	28-June-2005	<ol style="list-style-type: none"> <li>1. Transitioned source to OpenOffice.</li> <li>2. Imported styles from OASIS template.</li> <li>3. Designated as Public Review Draft 01</li> <li>4. Named document according to OASIS naming conventions</li> <li>5. Modified front page to conform to template</li> <li>6. Reformatted contributor list as table (for html export)</li> </ol>