



# Web Services Security: SAML Token Profile

## Committee Draft 03, 21 Sept. 2004

### Document identifier:

wss-saml-token-profile-1.0-cd-03(PDF)(Word)

### Location:

<http://docs.oasis-open.org/wss/2004/XX/oasis-2004XX-wss-saml-token-profile-1.0>

<http://www.oasis-open.org/committees/documents.php>

### Errata Location:

<http://www.oasis-open.org/committees/wss>

### Editors:

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

### 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
Tim Moses	Entrust
Dana Kaufman	Forum Systems, Inc.
Toshihiro Nishimura	Fujitsu
Kefeng Chen	GeoTrust
Irving Reid	Hewlett-Packard
Kojiro Nakayama	Hitachi
Paula Austel	IBM
Derek Fu	IBM

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

78 **Abstract:**

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

82 **Status:**

83 This is a committee draft. Please send comments to the editors.

84

85 Committee members should send comments on this specification to  
86 [wss@lists.oasis-open.org](mailto:wss@lists.oasis-open.org) list. Others should subscribe to and send comments  
87 to the [wss-comment@lists.oasis-open.org](mailto:wss-comment@lists.oasis-open.org) list. To subscribe, visit  
88 <http://lists.oasis-open.org/ob/adm.pl>.

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

94	<b>Table of Contents</b>	
95	1 Introduction.....	5
96	1.1 Goals.....	5
97	1.1.1 Non-Goals.....	5
98	2 Notations and Terminology.....	6
99	2.1 Notational Conventions.....	6
100	2.2 Namespaces.....	6
101	2.3 Terminology.....	7
102	3 Usage.....	8
103	3.1 Processing Model.....	8
104	3.2 Attaching Security Tokens.....	8
105	3.3 Identifying and Referencing Security Tokens.....	9
106	3.3.1 SAML Assertion Referenced from Header or Element.....	11
107	3.3.2 SAML Assertion Referenced from KeyInfo.....	12
108	3.3.3 SAML Assertion Referenced from SignedInfo.....	13
109	3.3.4 SAML Assertion Referenced from Encrypted Data Reference.....	14
110	3.4 Subject Confirmation of SAML Assertions.....	14
111	3.4.1 Holder-of-key Subject Confirmation Method.....	15
112	3.4.2 Sender-vouches Subject Confirmation Method.....	18
113	3.5 Error Codes.....	21
114	4 Threat Model and Countermeasures (Non-Normative).....	23
115	4.1 Eavesdropping.....	23
116	4.2 Replay.....	23
117	4.3 Message Insertion.....	24
118	4.4 Message Deletion.....	24
119	4.5 Message Modification.....	24
120	4.6 Man-in-the-Middle.....	24
121	5 References.....	25
122	Appendix A: Revision History.....	26
123	Appendix B: Notices.....	31
124		

---

# 125 **1 Introduction**

126 The [WSS: SOAP Message Security](#) specification defines a standard set of [SOAP](#)  
127 extensions that implement message level integrity and confidentiality. This  
128 specification defines the use of Security Assertion Markup Language (SAML)  
129 assertions as security tokens from the `<wsse:Security>` header block defined by the  
130 [WSS: SOAP Message Security](#) specification.

## 131 **1.1 Goals**

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

- 136 1. SAML assertions are carried in and referenced from `<wsse:security>` Headers.
- 137 2. SAML assertions are used with XML signature to bind the statements of the  
138 assertions (i.e. the claims) to a SOAP message.

### 139 **1.1.1 Non-Goals**

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

- 141 3. Defining SAML statement syntax or semantics.
- 142 4. Describing the use of SAML assertions other than for SOAP Message Security.
- 143 5. Describing the use of SAML V1.0 assertions with the [Web Services Security](#)  
144 ([WSS](#)): [SOAP Message Security](#) specification.

---

## 145 2 Notations and Terminology

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

### 148 2.1 Notational Conventions

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

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

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

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

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

### 163 2.2 Namespaces

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

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>
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
samlp	Urn: oasis:names:tc:SAML:1.0:protocol

168 **Table-1 Namespace Prefixes**

## 169 **2.3 Terminology**

170 This specification employs the terminology defined in the [WSS: SOAP Message](#)  
 171 [Security](#) specification. Defined below are the definitions for additional terminology  
 172 used in this specification.

173

174 Attesting Entity – the entity that provides the confirmation evidence that will be used  
 175 to establish the correspondence between the subject of SAML subject statements (in  
 176 SAML assertions) and SOAP message content.

177

178 Confirmation Method Identifier – the value within the `<saml:SubjectConfirmation>`  
 179 element of a SAML subject statement that identifies the confirmation method to be  
 180 used with the statement.

181

182 Subject Confirmation – the method used to establish the correspondence between  
 183 the subject of SAML subject statements (in SAML assertions) and SOAP message  
 184 content by verifying the confirmation evidence provided by an attesting entity.

185

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

187

188 Subject – A representation of the entity to which the claims in a SAML subject  
 189 statement apply.

---

## 190 3 Usage

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

### 193 3.1 Processing Model

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

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

203 As part of its validation and processing of the selected assertions, the receiver MUST  
204 establish the relationship between the subject of each SAML subject statement (of  
205 the referenced SAML assertions) and the entity providing the evidence to satisfy the  
206 confirmation method defined for the statements (i.e. the attesting entity). Two  
207 methods for establishing this correspondence, `holder-of-key` and `sender-vouches`  
208 are described below. Systems implementing this specification MUST implement the  
209 processing necessary to support both of these subject confirmation methods.

### 210 3.2 Attaching Security Tokens

211 SAML assertions are attached to SOAP messages using [WSS: SOAP Message Security](#)  
212 by placing assertion elements or references to assertions inside a `<wsse:Security>`  
213 header. The following example illustrates a SOAP message containing a SAML  
214 assertion in a `<wsse:Security>` header.

```
215 <S12:Envelope>  
216   <S12:Header>  
217     <wsse:Security>  
218       <saml:Assertion  
219         AssertionID="_a75adf55-01d7-40cc-929f-dbd8372ebdfc"  
220         IssueInstant="2003-04-17T00:46:02Z"  
221         Issuer="www.opensaml.org"  
222         MajorVersion="1"  
223         MinorVersion="1"  
224         . . .
```

---

<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.



225  
226  
227  
228  
229  
230  
231  
232

```
</saml:Assertion>
. . .
</wsse:Security>
</S12:Header>
<S12:Body>
. . .
</S12:Body>
</S12:Envelope>
```

### 233 3.3 Identifying and Referencing Security Tokens

234 The [WSS: SOAP Message Security](#) specification defines the  
235 `<wsse:SecurityTokenReference>` element for referencing security tokens. Three  
236 forms of token references are defined by this element and the element schema  
237 includes provision for defining additional reference forms should they be necessary.  
238 The three forms of token references defined by the  
239 `<wsse:SecurityTokenReference>` element are defined as follows:

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

245 When a key identifier is used to reference a SAML assertion, it MUST contain as  
246 its element value the corresponding SAML assertion identifier. The key identifier  
247 MUST also contain a `ValueType` attribute and the value of this attribute MUST be  
248 the `wsse:KeyIdentifier/@ValueType` from Table 2. The key identifier MUST  
249 NOT include an `EncodingType`<sup>2</sup> attribute and the element content of the key  
250 identifier MUST be encoded as `xsi:string`.

251 When a key identifier is used to reference a V1.1 SAML Assertion that is not  
252 contained in the same message as the key identifier, a  
253 `<saml:AuthorityBinding>` element MUST be contained in the  
254 `<wsse:SecurityTokenReference>` element containing the key identifier. The  
255 contents of the `<saml:AuthorityBinding>` element MUST contain values  
256 sufficient for the intended recipients of the `<wsse:SecurityTokenReference>` to  
257 acquire the identified assertion from the intended Authority. To this end, the  
258 value of the `AuthorityKind` attribute of the `<saml:AuthorityBinding>` element  
259 MUST be `"samlp:AssertionIdReference"`. When a key Identifier is used to  
260 reference a V1.1 SAML Assertion contained in the same message as the key  
261 identifier, a `<saml:AuthorityBinding>` element MUST NOT be included in the  
262 `<wsse:SecurityTokenReference>` containing the key identifier.

---

<sup>2</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 errata.

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

268 This profile does not describe the use of Direct or URI references to reference  
269 V1.1 SAML Assertions.

270 • An Embedded reference – a reference that encapsulates a security token.  
271 When an Embedded reference is used to encapsulate a SAML assertion, the SAML  
272 assertion MUST be included as a contained element within a `<wsse:Embedded>`  
273 element within a `<wsse:SecurityTokenReference>`.

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

275 • A SAML assertion may be referenced directly from a `<wsse:Security>` header  
276 element. In this case, the assertion is being conveyed by reference in the  
277 message.

278 • A SAML assertion may be referenced from a `<ds:KeyInfo>` element of a  
279 `<ds:Signature>` element in a `<wsse:Security>` header. In this case, the  
280 assertion contains a subject statement with a `<saml:SubjectConfirmation>`  
281 element that identifies the key used in the signature calculation.

282 • A SAML assertion reference may be referenced from a `<ds:Reference>` element  
283 within the `<ds:SignedInfo>` element of a `<ds:Signature>` element in a  
284 `<wsse:Security>` header. In this case, the doubly-referenced assertion is signed  
285 by the containing signature.

286 • A SAML assertion reference may occur as encrypted content within an  
287 `<xenc:EncryptedData>` element referenced from a `<xenc:DataReference>`  
288 element within an `<xenc:ReferenceList>` element. In this case, the assertion  
289 reference (which may contain an embedded assertion) is encrypted.

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

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

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

297 SAML key identifier references, with (in the case of remote references) a supporting  
298 `<saml:AuthorityBinding>` element are currently the best suited, of the  
299 `<wsse:SecurityTokenReference>` forms, for expressing references to SAML  
300 assertions. A future version of [SAMLCore] is expected to facilitate remote references  
301 by Direct reference URI. The practice of referencing local SAML Assertions by Direct  
302 `<wsse:SecurityTokenReference>` reference is not included in this profile because  
303 doing so would require recognition of the `<saml:AssertionID>` attribute as an  
304 identifier which would impose token dependent processing on the interpretation of  
305 local Direct references.

Attribute	Value
wsse:KeyIdentifier/@ValueType	<a href="http://docs.oasis-open.org/wss/2004/XX/oasis-2004XX-wss-saml-token-profile-1.0#SAMLAssertionID">http://docs.oasis-open.org/wss/2004/XX/oasis-2004XX-wss-saml-token-profile-1.0#SAMLAssertionID</a>

306 Table-2 ValueType Attribute Values

### 307 3.3.1 SAML Assertion Referenced from Header or Element

308 All conformant implementations MUST be able to process SAML assertion references  
 309 occurring in a <wsse:Security> header or in a header element other than a  
 310 signature to acquire the corresponding assertion. A conformant implementation  
 311 MUST be able to process any such reference independent of the confirmation method  
 312 of the referenced assertion.

313 A SAML assertion may be referenced from a <wsse:Security> header or from an  
 314 element (other than a signature) in the header. The following example demonstrates  
 315 the use of a key identifier in a <wsse:Security> header to reference a local SAML  
 316 assertion.

```

317 <S12:Envelope>
318   <S12:Header>
319     <wsse:Security>
320       <saml:Assertion
321         AssertionID="_a75adf55-01d7-40cc-929f-dbd8372ebdfc"
322         IssueInstant="2003-04-17T00:46:02Z"
323         Issuer="www.opensaml.org"
324         MajorVersion="1"
325         MinorVersion="1"
326         . . .
327       </saml:Assertion>
328       <wsse:SecurityTokenReference wsu:Id="STR1">
329         <wsse:KeyIdentifier wsu:Id="..."
330           ValueType="http://docs.oasis-open.org/wss/2004/XX/oasis-
331 2004XX-wss-saml-token-profile-1.0#SAMLAssertionID">
332           _a75adf55-01d7-40cc-929f-dbd8372ebdfc
333         </wsse:KeyIdentifier>
334       </wsse:SecurityTokenReference>
335     </wsse:Security>
336   </S12:Header>
337   <S12:Body>
338     . . .
339   </S12:Body>
340 </S12:Envelope>

```

341 A SAML assertion that exists outside of a <wsse:Security> header may be  
 342 referenced from the <wsse:Security> header element by including (in the  
 343 <wsse:SecurityTokenReference>) a <saml:AuthorityBinding> element that  
 344 defines the location, binding, and query that may be used to acquire the identified  
 345 assertion at a SAML assertion authority or responder.

```

346 <wsse:SecurityTokenReference wsu:Id="STR1">
347   <saml:AuthorityBinding
348     Binding="urn:oasis:names:tc:SAML:1.0:bindings:SOAP-binding"
349     Location="http://www.opensaml.org/SAML-Authority"
350     AuthorityKind="samlp:AssertionIdReference"

```

```

351 </saml:AuthorityBinding>
352 <wsse:KeyIdentifier
353   wsu:Id="..."
354   ValueType="http://docs.oasis-open.org/wss/2004/XX/oasis-2004XX-wss-
355 saml-token-profile-1.0#SAMLAssertionID">
356   _a75adf55-01d7-40cc-929f-dbd8372ebdfc
357 </wsse:KeyIdentifier>
358 </wsse:SecurityTokenReference>

```

### 359 **3.3.2 SAML Assertion Referenced from KeyInfo**

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

363 The following example depicts the use of a key identifier to reference a local  
364 assertion from <ds:KeyInfo>.

```

365 <ds:KeyInfo>
366   <wsse:SecurityTokenReference wsu:Id="STR1">>
367     <wsse:KeyIdentifier wsu:Id="..."
368       ValueType="http://docs.oasis-open.org/wss/2004/XX/oasis-2004XX-
369 wss-saml-token-profile-1.0#SAMLAssertionID">
370       _a75adf55-01d7-40cc-929f-dbd8372ebdfc
371     </wsse:KeyIdentifier>
372   </wsse:SecurityTokenReference>
373 </ds:KeyInfo>

```

374 The following example demonstrates the use of a <wsse:SecurityTokenReference>  
375 containing a key identifier and a <saml:AuthorityBinding> to communicate  
376 information (location, binding, and query) sufficient to acquire the identified  
377 assertion at an identified SAML assertion authority or responder.

```

378 <ds:KeyInfo>
379   <wsse:SecurityTokenReference wsu:Id="STR1">
380     <saml:AuthorityBinding>
381       Binding="urn:oasis:names:tc:SAML:1.0:bindings:SOAP-binding"
382       Location="http://www.opensaml.org/SAML-Authority"
383       AuthorityKind="samlp:AssertionIdReference"
384     </saml:AuthorityBinding>
385     <wsse:KeyIdentifier wsu:Id="..."
386       ValueType="http://docs.oasis-open.org/wss/2004/XX/oasis-2004XX-
387 wss-saml-token-profile-1.0#SAMLAssertionID">
388       _a75adf55-01d7-40cc-929f-dbd8372ebdfc
389     </wsse:KeyIdentifier>
390   </wsse:SecurityTokenReference>
391 </ds:KeyInfo>

```

392 <ds:KeyInfo> elements may also occur in <xenc:EncryptedData> and  
393 <xenc:EncryptedKey> elements where they serve to identify the encryption key.  
394 <ds:KeyInfo> elements may also occur in <saml:SubjectConfirmation> elements  
395 where they identify a key that MUST be demonstrated to confirm the subject of the  
396 corresponding subject statement(s). Conformant implementations of this profile are  
397 not required to process SAML assertion references occurring within the

398 <ds:keyInfo> elements within <xenc:EncryptedData>, <xenc:EncryptedKey>, or  
399 <saml:SubjectConfirmation><sup>3</sup> elements.

### 400 3.3.3 SAML Assertion Referenced from SignedInfo

401 Independent of the confirmation method of the referenced assertion, all conformant  
402 implementations MUST be able to process SAML assertions referenced by  
403 <wsse:SecurityTokenReference> from <ds:Reference> elements within the  
404 <ds:SignedInfo> element of a <ds:Signature> element in a <wsse:Security>  
405 header. Embedded references may be digested directly, thus effectively digesting the  
406 encapsulated assertion. Other <wsse:SecurityTokenReference> forms must be  
407 dereferenced for the referenced assertion to be digested.

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

415 The following example demonstrates the use of the STR Dereference transform to  
416 dereference a reference to a SAML Assertion (i.e. Security Token) such that the  
417 digest operation is performed on the security token not its reference.

```
418 <wsse:SecurityTokenReference wsu:Id="STR1">  
419 <saml:AuthorityBinding>  
420 Binding="urn:oasis:names:tc:SAML:1.0:bindings:SOAP-binding"  
421 Location="http://www.opensaml.org/SAML-Authority"  
422 AuthorityKind= "sampl:AssertionIdReference"  
423 </saml:AuthorityBinding>  
424 <wsse:KeyIdentifier wsu:Id="...">  
425 ValueType="http://docs.oasis-open.org/wss/2004/XX/oasis-2004XX-wss-  
426 saml-token-profile-1.0#SAMLAssertionID">  
427 _a75adf55-01d7-40cc-929f-dbd8372ebdfc  
428 </wsse:KeyIdentifier>  
429 </wsse:SecurityTokenReference>  
430 . . .  
431 <ds:SignedInfo>  
432 <ds:CanonicalizationMethod>  
433 Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>  
434 <ds:SignatureMethod>  
435 Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>  
436 <ds:Reference URI="#STR1">  
437 <Transforms>  
438 <ds:Transform>  
439 Algorithm="http://docs.oasis-open.org/wss/2004/01/oasis-200401-  
440 wss-soap-message-security-1.0#STR-Transform"/>  
441 <wsse:TransformationParameters>
```

---

<sup>3</sup> A SAML Assertion referenced from the <ds:KeyInfo> element within a <saml:SubjectConfirmation> element MUST contain one or more holder-of-key confirmed subject statements each of which identifies a key that MAY be used to confirm the subject and any other claims of the referencing statement.

```

442     <ds:CanonicalizationMethod
443         Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
444     </wsse:TransformationParameters>
445 </ds:Transform>
446 </Transforms>
447 <ds:DigestMethod
448     Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
449 <ds:DigestValue>...</ds:DigestValue>
450 </ds:Reference>
451 </ds:SignedInfo>

```

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

### 457 3.3.4 SAML Assertion Referenced from Encrypted Data 458 Reference

459 Independent of the confirmation method of the referenced assertion, all conformant  
460 implementations MUST be able to process SAML assertion references occurring as  
461 encrypted content within the `<xenc:EncryptedData>` elements referenced by Id  
462 from the `<xenc:DataReference>` elements of `<xenc:ReferenceList>` elements. An  
463 `<xenc:ReferenceList>` element may occur either as a top-level element in a  
464 Security header, or embedded within an `<xenc:EncryptedKey>` element. In either  
465 case, the `<xenc:ReferenceList>` identifies the encrypted content.

466 Such references are similar in format to the references that MAY appear in the  
467 `<ds:Reference>` element within `<ds:SignedInfo>`, except the STR Dereference  
468 transform does not apply. As shown in the following example, an encrypted  
469 `<wsse:SecurityTokenReference>` (which may contain an embedded assertion) is  
470 referenced from an `<xenc:DataReference>` by including the identifier of the  
471 `<xenc:EncryptedData>` element that contains the encrypted  
472 `<wsse:SecurityTokenReference>` in the `<xenc:DataReference>`.

```

473 <xenc:EncryptedData Id="EncryptedSTR1">
474   <ds:keyInfo>
475     . . .
476   </ds:KeyInfo>
477   <xenc:CipherData>
478     <xenc:CipherValue>...</xenc:CipherValue>
479   </xenc:CipherData>
480 </xenc:EncryptedData>
481 <xenc:ReferenceList>
482   <xenc:DataReference URI="#EncryptedSTR1" />
483 </xenc:ReferenceList>

```

## 484 3.4 Subject Confirmation of SAML Assertions

485 The SAML profile of [WSS: SOAP Message Security](#) requires that systems support the  
486 holder-of-key and sender-vouches methods of subject confirmation. It is strongly  
487 RECOMMENDED that an XML signature be used to establish the relationship between  
488 the message and the subject statements of the attached assertions. This is especially

489 RECOMMENDED whenever the SOAP message exchange is conducted over an  
490 unprotected transport.

491 Any processor of SAML assertions MUST conform to the required validation and  
492 processing rules defined in the SAML specification [SAMLCore] including the  
493 validation of assertion signatures, and the processing of <saml:Condition> elements  
494 within Assertions.

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

<b>Mechanism</b>	<b>RECOMMENDED Processing Rules</b>
urn:oasis:names:tc:SAML:1.0:cm:holder-of-key	The attesting entity includes an XML Signature that can be verified with the key information in the <saml:ConfirmationMethod> of the subject statements of the SAML assertion referenced for keyInfo by the Signature.
urn:oasis:names:tc:SAML:1.0: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 (containing the subject statements) in combination with the message content against modification by another party. See also section 4.

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

### 503 **3.4.1 Holder-of-key Subject Confirmation Method**

504 The following sections describe the holder-of-key method of establishing the  
505 correspondence between a SOAP message and the subject of SAML assertions added  
506 to the SOAP message according to this specification.



### 507 **3.4.1.1 Attesting Entity**

508 An attesting entity uses the holder-of-key confirmation method to demonstrate that  
509 it is authorized to act as the subject of the SAML subject statements containing the  
510 holder-of-key `<saml:SubjectConfirmation>` element. The subject statements that  
511 will be confirmed by the holder-of-key method MUST include the following  
512 `<saml:SubjectConfirmation>` element:

```
513 <saml:SubjectConfirmation>  
514   <saml:ConfirmationMethod>  
515     urn:oasis:names:tc:SAML:1.0:cm:holder-of-key  
516   </saml:ConfirmationMethod>  
517   <ds:KeyInfo>...</ds:KeyInfo>  
518 </saml:SubjectConfirmation>
```

519 The `<saml:SubjectConfirmation>` element MUST include a `<ds:KeyInfo>` element  
520 that identifies the public or secret key<sup>4</sup> to be used to confirm the identity of the  
521 subject.

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

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

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

### 536 **3.4.1.2 Receiver**

537 Of the SAML assertions it selects for processing, a message receiver MUST NOT  
538 accept assertions containing a holder-of-key `<saml:ConfirmationMethod>`, unless  
539 the receiver has validated the integrity of the assertions and the attesting entity has  
540 demonstrated knowledge of the key identified by the `<ds:keyInfo>` element of the  
541 `<saml:SubjectConfirmation>` element.

---

<sup>4</sup>[\[SAMLCore\]](#) 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. The OASIS Security Services Technical Committee has resolved to remove the phrase "held by the subject" from the definition of KeyInfo of SubjectConfirmation.



542 If the receiver determines that the attesting entity has demonstrated knowledge of a  
543 subject confirmation key, then the SAML assertions containing the confirmation key  
544 MAY be attributed to the attesting entity and any elements of the message whose  
545 integrity is protected by the subject confirmation key MAY be considered to have  
546 been provided by the subject.

### 547 3.4.1.3 Example

548 The following example illustrates the use of the holder-of-key subject confirmation  
549 method to establish the correspondence between the SOAP message and the subject  
550 of the SAML assertions in the <wsse:Security> header:

```
551 <?xml:version="1.0" encoding="UTF-8"?>
552 <S12:Envelope>
553   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
554   xmlns:xsd="http://www.w3.org/2001/XMLSchema">
555   <S12:Header>
556
557     <wsse:Security>
558       <saml:Assertion
559         AssertionID="_a75adf55-01d7-40cc-929f-dbd8372ebdfc"
560         IssueInstant="2003-04-17T00:46:02Z"
561         Issuer="www.opensaml.org"
562         MajorVersion="1"
563         MinorVersion="1"
564         xmlns="urn:oasis:names:tc:SAML:1.0:assertion">
565         <saml:Conditions>
566           NotBefore="2002-06-19T16:53:33.173Z"
567           NotOnOrAfter="2002-06-19T17:08:33.173Z"/>
568         <saml:AttributeStatement>
569           <saml:Subject>
570             <saml:NameIdentifier
571               NameQualifier="www.example.com"
572               Format="...">
573               uid=joe,ou=people,ou=saml-demo,o=baltimore.com
574             </saml:NameIdentifier>
575             <saml:SubjectConfirmation>
576               <saml:ConfirmationMethod>
577                 urn:oasis:names:tc:SAML:1.0:cm:holder-of-key
578               </saml:ConfirmationMethod>
579               <ds:KeyInfo>
580                 <ds:KeyValue>...</ds:KeyValue>
581               </ds:KeyInfo>
582             </saml:SubjectConfirmation>
583           </saml:Subject>
584           <saml:Attribute
585             AttributeName="MemberLevel"
586             AttributeNamespace="http://www.oasis.open.
587             org/Catalyst2002/attributes">
588             <saml:AttributeValue>gold</saml:AttributeValue>
589           </saml:Attribute>
590           <saml:Attribute
591             AttributeName="E-mail"
592             AttributeNamespace="http://www.oasis.open.
593             org/Catalyst2002/attributes">
594             <saml:AttributeValue>joe@yahoo.com</saml:AttributeValue>
595           </saml:Attribute>
596         </saml:AttributeStatement>
597         <ds:Signature>...</ds:Signature>
```

```

598     </saml:Assertion>
599
600     <ds:Signature>
601       <ds:SignedInfo>
602         <ds:CanonicalizationMethod
603           Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
604         <ds:SignatureMethod
605           Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1" />
606         <ds:Reference
607           URI="#MsgBody">
608           <ds:DigestMethod
609             Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
610           <ds:DigestValue>GyGsF0Pi4xPU...</ds:DigestValue>
611         </ds:Reference>
612       </ds:SignedInfo>
613       <ds:SignatureValue>HJJWbvqW9E84vJVQk...</ds:SignatureValue>
614       <ds:KeyInfo>
615         <wsse:SecurityTokenReference wsu:Id="STR1">
616           <wsse:KeyIdentifier wsu:Id="..."
617             ValueType="http://docs.oasis-open.org/wss/2004/XX/oasis-
618 2004XX-wss-saml-token-profile-1.0#SAMLAssertionID">
619             _a75adf55-01d7-40cc-929f-dbd8372ebdfc
620           </wsse:KeyIdentifier>
621         </wsse:SecurityTokenReference>
622       </ds:KeyInfo>
623     </ds:Signature>
624   </wsse:Security>
625 </S12:Header>
626
627   <S12:Body wsu:Id="MsgBody">
628     <ReportRequest>
629       <TickerSymbol>SUNW</TickerSymbol>
630     </ReportRequest>
631   </S12:Body>
632 </S12:Envelope>

```

### 633 3.4.2 Sender-vouches Subject Confirmation Method

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

#### 637 3.4.2.1 Attesting Entity

638 An attesting entity uses the sender-vouches confirmation method to assert that it is  
639 acting on behalf of the subject of SAML subject statements containing a sender-  
640 vouches `<saml:SubjectConfirmation>` element. The subject statements that the  
641 attesting entity will confirm by the sender-vouches method MUST include the  
642 following `<saml:SubjectConfirmation>` element:

```

643   <saml:SubjectConfirmation>
644     <saml:ConfirmationMethod>
645       urn:oasis:names:tc:SAML:1.0:cm:sender-vouches
646     </saml:ConfirmationMethod>
647   </saml:SubjectConfirmation>

```

648 To satisfy the associated confirmation method processing of the receiver, the  
649 attesting entity MUST protect the vouched for SOAP message content such that the

650 receiver can determine when it has been altered by another party. The attesting  
651 entity MUST also cause the vouched for subject statements (as necessary) and their  
652 binding to the message contents to be protected such that unauthorized modification  
653 can be detected. The attesting entity MAY satisfy these requirements by including in  
654 the corresponding `<wsse:Security>` header a `<ds:Signature>` element that it  
655 prepares by using its key to sign the relevant message content and assertions. As  
656 defined by the [XML Signature](#) specification, the attesting entity MAY identify its key  
657 by including a `<ds:KeyInfo>` element within the `<ds:Signature>` element.

658 A `<ds:Signature>` element produced for this purpose MUST conform to the  
659 canonicalization and token prepending rules defined in the [WSS: SOAP Message  
660 Security](#) specification.

### 661 **3.4.2.2 Receiver**

662 Of the SAML assertions it selects for processing, a message receiver MUST NOT  
663 accept assertions containing a sender-vouches `<saml:ConfirmationMethod>` unless  
664 the assertions and SOAP message content being vouched for are protected (as  
665 described above) by an attesting entity who is trusted by the receiver to act on  
666 behalf of the subject of the assertions.

### 667 **3.4.2.3 Example**

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

672 The assertion referenced by "STR1" is not included in the message. "STR1" is  
673 referenced by `<ds:reference>` from `<ds:SignedInfo>`. The `ds:reference>`  
674 includes the STR-transform to cause the assertion, not the  
675 `<SecurityTokenReference>` to be included in the digest calculation. "STR1" includes  
676 an `<AuthorityBinding>` element that utilizes the remote assertion referencing  
677 technique depicted in the example of section 3.3.3.

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

```
682 <?xml:version="1.0" encoding="UTF-8"?>
683 <S12:Envelope>
684   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
685   xmlns:xsd="http://www.w3.org/2001/XMLSchema">
686   <S12:Header>
687     <wsse:Security>
688
689     <saml:Assertion
690       AssertionID="_a75adf55-01d7-40cc-929f-dbd8372ebdfc"
691       IssueInstant="2003-04-17T00:46:02Z"
692       Issuer="www.opensaml.org"
693       MajorVersion="1"
694       MinorVersion="1"
695       xmlns="urn:oasis:names:tc:SAML:1.0:assertion">
696     <saml:Conditions>
```

```

697     NotBefore="2002-06-19T16:53:33.173Z"
698     NotOnOrAfter="2002-06-19T17:08:33.173Z"/>
699     <saml:AttributeStatement>
700         <saml:Subject>
701             <saml:NameIdentifier
702                 NameQualifier="www.example.com"
703                 Format="...">
704                 uid=proxy,ou=system,ou=saml-demo,o=baltimore.com
705             </saml:NameIdentifier>
706             <saml:SubjectConfirmation>
707                 <saml:ConfirmationMethod>
708                     urn:oasis:names:tc:SAML:1.0:cm:holder-of-key
709                 </saml:ConfirmationMethod>
710                 <ds:KeyInfo>
711                     <ds:KeyValue>...</ds:KeyValue>
712                 </ds:KeyInfo>
713             </saml:SubjectConfirmation>
714         </saml:Subject>
715         <saml:Attribute
716             . . .
717         </saml:Attribute>
718         . . .
719     </saml:AttributeStatement>
720 </saml:Assertion>
721
722     <wsse:SecurityTokenReference wsu:Id="STR1">
723         <saml:AuthorityBinding>
724             saml:Binding="urn:oasis:names:tc:SAML:1.0:bindings:SOAP-
725 binding"
726             saml:Location="http://www.opensaml.org/SAML-Authority"
727             saml:AuthorityKind="samlp:AssertionIdReference"
728         </saml:AuthorityBinding>
729         <wsse:KeyIdentifier wsu:Id="..."
730             ValueType="http://docs.oasis-open.org/wss/2004/XX/oasis-
731 2004XX-wss-saml-token-profile-1.0#SAMLAssertionID">
732             _a75adf55-01d7-40cc-929f-dbd8372ebdbe
733         </wsse:KeyIdentifier>
734     </wsse:SecurityTokenReference>
735
736     <ds:Signature>
737         <ds:SignedInfo>
738             <ds:CanonicalizationMethod
739                 Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
740             <ds:SignatureMethod
741                 Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
742             <ds:Reference URI="#STR1">
743                 <Transforms>
744                     <ds:Transform
745                         Algorithm="http://docs.oasis-
746 open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#STR-
747 Transform"/>
748                         <wsse:TransformationParameters>
749                             <ds:CanonicalizationMethod
750                                 Algorithm="http://www.w3.org/2001/10/xml-exc-
751 c14n#"/>
752                             </wsse:TransformationParameters>
753                         </ds:Transform>
754                     </Transforms>
755                 <ds:DigestMethod
756                     Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
757                 <ds:DigestValue>...</ds:DigestValue>

```

```

758     </ds:Reference>
759     <ds:Reference URI="#MsgBody">
760         <ds:DigestMethod
761             Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
762         <ds:DigestValue>...</ds:DigestValue>
763     </ds:Reference>
764 </ds:SignedInfo>
765 <ds:SignatureValue>HJJWbvqW9E84vJVQk...</ds:SignatureValue>
766 <ds:KeyInfo>
767     <wsse:SecurityTokenReference wsu:Id="STR2">
768         <wsse:KeyIdentifier wsu:Id="..."
769             ValueType="http://docs.oasis-open.org/wss/2004/XX/oasis-
770 2004XX-wss-saml-token-profile-1.0#SAMLAssertion-1.1">
771             _a75adf55-01d7-40cc-929f-dbd8372ebdfc
772         </wsse:KeyIdentifier>
773     </wsse:SecurityTokenReference>
774 </ds:KeyInfo>
775 </ds:Signature>
776 </wsse:Security>
777 </S12:Header>
778
779 <S12:Body wsu:Id="MsgBody">
780     <ReportRequest>
781         <TickerSymbol>SUNW</TickerSymbol>
782     </ReportRequest>
783 </S12:Body>
784 </S12:Envelope>

```

### 785 3.5 Error Codes

786 When a system that implements the SAML token profile of [WSS: SOAP Message Security](#) does not perform its normal processing because of an error detected during  
787 the processing of a security header, it MAY choose to report the cause of the error  
788 using the SOAP fault mechanism. The SAML token profile of [WSS: SOAP Message Security](#)  
789 does not require that SOAP faults be returned for such errors, and systems  
790 that choose to return faults SHOULD take care not to introduce any security  
791 vulnerabilities as a result of the information returned in error responses.  
792

793 Systems that choose to return faults SHOULD respond with the error codes defined  
794 in the [WSS: SOAP Message Security](#) specification. The RECOMMENDED  
795 correspondence between the common assertion processing failures and the error  
796 codes defined in [WSS: SOAP Message Security](#) are defined in the following table:

Assertion Processing Error (faultString)	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

797 The preceding table defines fault strings and codes in a form suitable to be used with  
798 SOAP 1.1. The [WSS: SOAP Message Security](#) specification describes how to map  
799 SOAP 1.1 fault constructs to the SOAP 1.2 fault constructs.

---

## 800 **4 Threat Model and Countermeasures** 801 **(Non-Normative)**

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

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

### 813 **4.1 Eavesdropping**

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

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

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

### 828 **4.2 Replay**

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

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

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

### 841 **4.3 Message Insertion**

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

### 844 **4.4 Message Deletion**

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

### 847 **4.5 Message Modification**

848 Messages constructed according to this specification are protected from message  
849 modification if receivers can detect unauthorized modification of relevant message  
850 content. Therefore, it is strongly RECOMMENDED that all relevant and immutable  
851 message content be signed by an attesting entity. Receivers SHOULD only consider  
852 the correspondence between the subject of the SAML assertions and the SOAP  
853 message content to have been established for those portions of the message that are  
854 protected by the attesting entity against modification by another entity.

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

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

### 865 **4.6 Man-in-the-Middle**

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



---

## 5 References

- 870
- 871     **[GLOSSARY]**     Informational RFC 2828, "[Internet Security Glossary](#)," May  
872                             2000.
- 873     **[KEYWORDS]**     S. Bradner, "Key words for use in RFCs to Indicate Requirement  
874                             Levels," [RFC 2119](#), Harvard University, March 1997
- 875     **[SAMLBind]**     Oasis Committee Specification 01, E. Maler, P.Mishra, and R.  
876                             Philpott (Editors), [Bindings and Profiles for the OASIS Security  
877                             Assertion Markup Language \(SAML\) V1.1](#), September 2003.
- 878     **[SAMLCore]**     Oasis Committee Specification 01, E. Maler, P.Mishra, and R.  
879                             Philpott (Editors), [Assertions and Protocol for the OASIS  
880                             Security Assertion Markup Language \(SAML\) V1.1](#), September  
881                             2003.
- 882     **[SOAP]**         W3C Note, "[SOAP: Simple Object Access Protocol 1.1](#)," 08 May  
883                             2000.
- 884                             W3C Working Draft, Nilo Mitra (Editor), [SOAP Version 1.2 Part  
885                             0: Primer](#), June 2002.
- 886                             W3C Working Draft, Martin Gudgin, Marc Hadley, Noah  
887                             Mendelsohn, Jean-Jacques Moreau, Henrik Frystyk Nielsen  
888                             (Editors), [SOAP Version 1.2 Part 1: Messaging Framework](#), June  
889                             2002.
- 890                             W3C Working Draft, Martin Gudgin, Marc Hadley, Noah  
891                             Mendelsohn, Jean-Jacques Moreau, Henrik Frystyk Nielsen  
892                             (Editors), [SOAP Version 1.2 Part 2: Adjuncts](#), June 2002.
- 893     **[URI]**         T. Berners-Lee, R. Fielding, L. Masinter, "Uniform Resource  
894                             Identifiers (URI): Generic Syntax," [RFC 2396](#), MIT/LCS, U.C.  
895                             Irvine, Xerox Corporation, August 1998.
- 896     **[WS-SAML]**     Contribution to the WSS TC, P. Mishra (Editor), [WS-Security  
897                             Profile of the Security Assertion Markup Language \(SAML\)  
898                             Working Draft 04](#), Sept 2002.
- 899     **[WSS: SOAP Message Security]** Oasis Standard, A. Nadalin, C.Kaler, P.  
900                             Hallem-Baker, R. Monzillo (Editors), [Web Services Security:  
901                             SOAP Message Security 1.0 \(WS-Security 2004\)](#), August 2003.
- 902     **[XML-ns]**         W3C Recommendation, "[Namespaces in XML](#)," 14 January  
903                             1999.
- 904     **[XML Signature]**W3C Recommendation, "[XML Signature Syntax and  
905                             Processing](#)," 12 February 2002.
- 906     **[XML Token]**     Contribution to the WSS TC, Chris Kaler (Editor),  
907                             WS-Security Profile for XML-based Tokens, August 2002.

## Appendix A: Revision History

Rev	Date	What
01	19-Sep-02	Initial draft produced by extracting SAML related content from [XML token]
02	23-Sep-02	Merged in content from SS TC submission
03	18-Nov-02	Resolved issues raised by TC
04	09-Dec-02	Refined confirmation mechanisms, and added signing example
05	15-Dec-02	Results of Baltimore F2F
06	21-Feb-03	Changed name to profile
07	05-May-03	Acknowledged contributors
07	05-May-03	Throughout document, Refined terminology to distinguish attesting entity from subject and sender, and to distinguish assertions from statements within assertions. Also modified sender-vouches to support traced vouching (by allowing for the use of a confirmation key)
08	09-Jun-03	Indicated reliance on conventions of core in "Notational Conventions"
08	09-Jun-03	In "Terminology", added definitions of new terms (attesting entity and confirmation method identifier), edited definition of Subject Confirmation, and replaced definition of sender with subject.
08	09-Jun-03	In "Subject Confirmation of SAML Assertions", added requirement that an attesting entity must protect unsigned sender-vouches confirmed assertions.
08	25-Nov-03	Added SAM v1.1 version distinction to "Abstract"
08	25-Nov-03	Editorial changes to "Introduction"
08	25-Nov-03	Reorganized non-normative text of requirements and goals sections
08	25-Nov-03	Removed Identification, Contact Information, Description, and Updates from "Usage".
08	25-Nov-03	Updated schema URIs and corrected namespace prefixes in "Namespaces"
08	25-Nov-03	Updated SAML document references in "References" to point to v1.1. specs.

Rev	Date	What
08	25-Nov-03	In Error codes, changed error processing such that it is optional and consistent with the recommendations in core.
08	25-Nov-03	Qualified "Threat Model and Counter-measures" as non-normative.
08	30-Nov-03	In "Identifying and Referencing Security Tokens", removed keyname references and added embedded references. Also removed editorial comment regarding using artifacts to reference assertions.
08	30-Nov-03	Made editorial changes to "Processing Model", including clarification (by footnote) of "semantic labeling"
08	30-Nov-03	Removed "Acknowledgments" as it duplicated preceding sections of the document
08	12-15-03	Added high level goals and non-goals
08	12-15-03	Added support for the use of (fragment) URI references to section 3.3
08	12-15-03	Specified default encoding type for SAML and fragment UR references to be xsi:string
08	12-15-03	Added two more contexts in which SAML assertions may be referenced; from within SubjectConfirmation elements and as encrypted data.
08	12-15-03	Made it a requirement of conformant implementations that they support the various methods of referencing SAML assertions
08	12-15-03	Added new sections to describe SAML assertion referenced from SubjectConfirmation and SAML assertion referenced from Encrypted Data reference.
09	01-27-04	Changed document identifier and location
09	01-27-04	Modified namespace table of section 2.2 to differentiate SOAP 1.1 and SOAP 1.2
10	02-05-04	Changed all instances of wsu:id to wsu:Id
10	02-05-04	In section 3.4.2.1 beginning around line 705, removed the distinction of the "typical case where the assertion authority has NOT securely bound a key..." because we no longer expect sender-vouches to use a confirmation key.
10	3-29-04	Corrected STR transform URL to match change

Rev	Date	What
		in core.
10	3-29-04	Removed from section 3.3.2 mention of use of KeyInfo with sender-vouches confirmation method.
10	3-29-04	Modified footnote in section 3.2 regarding usage attribute to reflect change from QNAMES to URIs.
10	3-29-04	Corrected signature algorithm in examples.
10	3-29-04	Corrected transforms syntax of example in section 3.3.3.
10	3-29-04	In section 3.3.3 recommended that STR dereference transform not be applied to embedded token references.
10	3-29-04	Removed requirement (from section 4.5 of Security Considerations) that assertion references be protected from unauthorized modification.
10	4-02-04	Removed namespace qualification from ValueType, URI, EncodingType, and Usage Attributes (mostly in examples). Also removed angle brackets.
10	4-05-04	Reworded initial paragraph of section 2.2 Namespaces such that it is not normative, and affords more flexibility in the form of the examples.
10	4-05-04	Removed namespace declarations from examples.
10	4-05-04	Corrected misspelling of "Authorty" in examples.
10	4-05-04	Modified processing rule for sender-vouches in Table of section 3.4 (to allow sender to vouch for itself).
10	4-05-04	Editing changes to the error codes section. In particular, replaced the word "generated" with "returned", and rewrote the description of the mapping to 1.2 constructs.
10	4-05-04	Removed unused SAMLreqs and SAMLSecure from the references section.
10	4-06-04	Added footnote to explain optional support for SAML V1.0 assertions.
10	4-06-04	Removed section 3.3.4 "SAML Assertion referenced from SubjectConfirmation", as

Rev	Date	What
		SAML is evolving in a manner that will make it unlikely that authorities will need to produce such assertions. Moved the description of SAML Assertions references occurring within KeyInfo of SubjectConfirmation to section 3.3.2 "SAML assertion referenced from KeyInfo"
10	4-06-04	From Section 3.3 "Identifying and referencing Security Tokens", removed referencing a SAML assertion from KeyInfo of SubjectConfirmation from the five contexts in which SAML assertions may be referenced.
10	4-06-04	Moved description of SAML Assertion references occurring within KeyInfo of SubjectConfirmation to section 3.3.2.
10	4-06-04	Added footnote to description of holder-of-key semantics in section 3.4.1.1 to describe interpretation of "held by the subject" phrase appearing in definition in [SAMLCore].
10	4-06-04	Updated contributors list
11	5-21-04	Moved " <a href="http://...documents.php">http://...documents.php</a> " URL from "Location" to "Document Repository (temporary):" which will be removed when document is available from "Location".
11	5-21-04	In section "1.1.1 Non-Goals", added new bullet to indicate that describing support for V1.0 assertions is outside the scope of the profile.
11	5-21-04	Changed SAMLAssertion-1.0 wsse:Reference/@ValueType to SAMLAssertion-1.1 in examples (lines 366, 611, and 752)
11	5-21-04	Updated document, specification, and schema URL's to accommodate change to OASIS document URLs (i.e. <a href="http://www.docs.oasis-open.org">www.docs.oasis-open.org</a> changed to docs.oasis-open.org)
11	5-21-04	Removed SAMLAssertion-1.0 wsse:Reference/@ValueType from "Table-2 ValueType Attribute Values." Also removed footnote on table title.
11	5-21-04	Editorial correction made to the attributes of the NameIdentifier element in the examples (see lines 564 and 684).
11	5-21-04	In section 3.4, "Subject Confirmation of SAML Assertions" (line 485), changed the reference to be to [SAMLCore] for the definition of the validation and processing rules that apply to

Rev	Date	What
		SAML assertions. Also (as the resolution to issue 275), extended the stated reliance (on [SAMLCore]) with "including the validation of assertion signatures, and the processing of <saml:Condition> elements within Assertions"
12	6-25-04	In section 3.4.2.3, clarified the description of the sender-vouches example.
13	6-30-04	Modified section 3.3 to describe the use of KeyIdentifiers as apposed to Direct references to reference SAML assertions.
13	6-30-04	In section 3.3 and 3.3.4 clarified the use of STRs from <xenc:DataReference>
13	6-3--04	Removed wsse:Reference/@ValueType from Table 2 of section 3.3, as the change to KeyIdentifiers rendered the ValueType unnecessary.
13	6-30-04	Changed the examples in sections 3.3.1, 3.3.2, 3.3.4, 3.4.1.3, and 3.4.2.3 to reflect the change from Direct references to KeyIdentifiers.
14	7-12-04	Corrected KeyIdentifier syntax of examples at lines 338, 376, 627, and 780.
15	7-19-04	Added clarification to sections 3.3.1, 3.3.2, and 3.3.4 to address issue 295b; that the profile include provision for the use of "Bearer" confirmed assertions.
CD 02	9-08-04	Renamed as committee draft, added reference to errata, updated contributor lists, modified status to CD, and added footnote to description of KeyIdentifier to direct reader to clarification in errata.
CD 03	9-21-04	Removed version qualification (i.e. "Version 2 of ") from the reference to the Errata occurring in the footnote (of section 3.3).

909

---

## Appendix B: Notices

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

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

924 Copyright © OASIS Open 2003. *All Rights Reserved.*

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

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

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