



ID-WSF 2.0 SecMech SAML Profile

Version: v2.0

Editors:

Frederick Hirsch, Nokia Corporation

Contributors:

Robert Aarts, Hewlett-Packard
Conor Cahill, Intel Corporation, formerly America Online, Inc.
Carolina Canales-Valenzuela, Ericsson
Scott Cantor, Internet2, The Ohio State University
Gary Ellison, Sun Microsystems, Inc.
Jeff Hodges, Neustar
John Kemp, Nokia Corporation
John Linn, RSA Security Inc.
Paul Madsen, NTT, formerly Entrust
Jonathan Sergent, Sun Microsystems, Inc.
Greg Whitehead, Hewlett-Packard

Abstract:

Security Mechanism profile of the SAML assertions and WSS SAML Token Profile v1.1 in conjunction with the Liberty ID-WSF 2.0 Security Mechanisms specification.

Filename: liberty-idwsf-security-mechanisms-saml-profile-v2.0.pdf

1

Notice

2 This document has been prepared by Sponsors of the Liberty Alliance. Permission is hereby granted to use the
3 document solely for the purpose of implementing the Specification. No rights are granted to prepare derivative works
4 of this Specification. Entities seeking permission to reproduce portions of this document for other uses must contact
5 the Liberty Alliance to determine whether an appropriate license for such use is available.

6 Implementation of certain elements of this document may require licenses under third party intellectual property
7 rights, including without limitation, patent rights. The Sponsors of and any other contributors to the Specification are
8 not and shall not be held responsible in any manner for identifying or failing to identify any or all such third party
9 intellectual property rights. **This Specification is provided "AS IS", and no participant in the Liberty Alliance
10 makes any warranty of any kind, express or implied, including any implied warranties of merchantability,
11 non-infringement of third party intellectual property rights, and fitness for a particular purpose.** Implementers
12 of this Specification are advised to review the Liberty Alliance Project's website (<http://www.projectliberty.org/>) for
13 information concerning any Necessary Claims Disclosure Notices that have been received by the Liberty Alliance
14 Management Board.

15 Copyright © 2006 Adobe Systems; America Online, Inc.; American Express Company; Amsoft Systems Pvt Ltd.;
16 Avatier Corporation; Axalto; Bank of America Corporation; BIPAC; BMC Software, Inc.; Computer Associates
17 International, Inc.; DataPower Technology, Inc.; Diversinet Corp.; Enosis Group LLC; Entrust, Inc.; Epok, Inc.;
18 Ericsson; Fidelity Investments; Forum Systems, Inc.; France Télécom; French Government Agence pour le
19 développement de l'administration électronique (ADAE); Gamefederation; Gemplus; General Motors; Giesecke &
20 Devrient GmbH; GSA Office of Governmentwide Policy; Hewlett-Packard Company; IBM Corporation; Intel
21 Corporation; Intuit Inc.; Kantega; Kayak Interactive; MasterCard International; Mobile Telephone Networks (Pty)
22 Ltd; NEC Corporation; Netegrity, Inc.; NeuStar, Inc.; Nippon Telegraph and Telephone Corporation; Nokia
23 Corporation; Novell, Inc.; NTT DoCoMo, Inc.; OpenNetwork; Oracle Corporation; Ping Identity Corporation;
24 Reactivity Inc.; Royal Mail Group plc; RSA Security Inc.; SAP AG; Senforce; Sharp Laboratories of America;
25 Sigaba; SmartTrust; Sony Corporation; Sun Microsystems, Inc.; Supremacy Financial Corporation; Symlabs, Inc.;
26 Telecom Italia S.p.A.; Telefónica Móviles, S.A.; Trusted Network Technologies; UTI; VeriSign, Inc.; Vodafone
27 Group Plc.; Wave Systems Corp. All rights reserved.

28 Liberty Alliance Project
29 Licensing Administrator
30 c/o IEEE-ISTO
31 445 Hoes Lane
32 Piscataway, NJ 08855-1331, USA
33 info@projectliberty.org

34 Contents

35	1. Introduction	4
36	2. Notation, Terminology, Namespaces and typographical conventions	5
37	3. Identifier Privacy Protection	6
38	3.1. Encrypted Name Identifiers	6
39	4. Authentication Mechanisms	7
40	4.1. SAML Assertion Message Authentication	7
41	4.1.1. Sender Processing Rules	7
42	4.1.2. Recipient Processing Rules	8
43	4.2. Bearer Token Authentication	8
44	4.2.1. Processing Rules	8
45	4.2.2. SAML Bearer Token Example	8
46	5. Message Authorization	12
47	5.1. Authorization Data Generation	12
48	5.1.1. Processing Rules	12
49	5.1.2. Consuming Authorization Data	13
50	6. Provider Chaining	14
51	6.1. Provider Chaining Example (Informative)	14
52	7. Identity Token	19
53	7.1. Identity Token Requirements	19
54	8. Examples (Informative)	20
55	8.1. Fragmentary Examples	20
56	8.1.1. Sender as Invocation Identity	20
57	8.1.2. Sender as Transited Provider Identity	20
58	8.1.3. Invoking Identity Authentication	21
59	8.1.4. Resource as an Attribute	22
60	8.2. Proxying with Authentication Context of the Invoking Identity	22
61	8.3. Conveyance of Sender as Invocation Identity	25
62	References	29

63 **1. Introduction**

64 This document specifies specific normative requirements on the use of SAML assertions and/or the WSS SAML Token
65 profile in conjunction with the ID-WSF 2.0 Security Mechanisms specification ([[wss-saml11](#)], [[LibertySecMech20](#)],
66 [[SAMLCore2](#)], [[SAMLBind2](#)]).

67 This document assumes familiarity with the Security Mechanisms core specification and does not replicate the general
68 discussion or normative requirements from that specification.

69 **2. Notation, Terminology, Namespaces and typographical**
70 **conventions**

71 Please refer to the Security Mechanisms core for specification of notations, namespaces and terminology used
72 throughout this specification, as well as typographical conventions.

73 **3. Identifier Privacy Protection**

74 **3.1. Encrypted Name Identifiers**

75 To securely protect the privacy of the identifier as the message passes through intermediaries, the `<saml2:Subject>`
76 **MUST** contain a `<saml2:EncryptedID>` where a privacy risk due to provider collaboration based on iden-
77 tity is a concern. In general the `<saml2:Subject>` **SHOULD** contain a `<saml2:EncryptedID>`. Use of
78 `<saml2:EncryptedID>` **MUST** follow the processing rules and recommendations specified in [[SAMLCore2](#)].

79 4. Authentication Mechanisms

80 This section outlines specific normative requirements for using SAML 2.0 assertions for message authentication.
81 General normative requirements are specified in the Security Mechanisms core [[LibertySecMech20](#)].

82 4.1. SAML Assertion Message Authentication

83 The semantics and processing rules for the following URIs are described in this profile. These URIs indicate unilateral
84 SAML-based message authentication, i.e. authentication of the invoker, using SAML 2.0:

- 85 • *urn:liberty:security:2006-08:null:SAMLV2*
- 86 • *urn:liberty:security:2006-08:TLS:SAMLV2*
- 87 • *urn:liberty:security:2006-08:ClientTLS:SAMLV2*
- 88 • *urn:liberty:security:2006-08:ClientTLS:peerSAMLV2*

89 These mechanisms utilize the OASIS Web Services Security SAML Token Profile v1.1 [[wss-saml11](#)] as the means
90 by which the message sender authenticates to the recipient. In general these mechanisms assume that an Identity
91 Provider issues an assertion that includes an `<saml2:AuthnStatement>` and other statements applicable to the
92 `<saml2:Subject>` entity and contained within the `<saml2:Subject>` element.

93 The `<saml2:AuthnStatement>` describes the authentication event of the subject to the issuing authority. For this
94 and any other statements in the assertion to be considered trustworthy, the subject confirmation obligations specified
95 in the `<saml2:Subject>` element must be met by the sender.

96 As a security precaution, the issuer of the assertion MUST include a `<saml2:AudienceRestriction>` element
97 that specifies the intended consumer(s) of the assertion. One `<saml2:Audience>` element MUST be set
98 to contain the unique identifier of the intended recipient, as described by the name identifier Format URI of
99 *urn:oasis:names:tc:SAML2:2.0:nameid-format:entity* as specified in [[SAMLCore2](#)].

100 The recipient MUST validate that it is an intended consumer of the assertion before relying upon it. The assertion
101 MAY contain additional `<saml2:Audience>` elements that specify other intended consumers of the assertion.

102 These message authentication mechanisms are unilateral. That is, only the sender of the message is authenticated. It
103 is not in the scope of this specification to suggest when response messages should be authenticated, but it is worth
104 noting that the mechanisms defined in Security Mechanisms core regarding WSS X.509 token authentication could
105 be relied upon to authenticate any response message as well. Deployers should recognize, however, that independent
106 authentication of response messages does not provide the same message stream protection semantics as a mutual peer
107 entity authentication mechanism.

108 For deployment settings that require message authentication independent of peer entity authentication, then the sending
109 peer MUST perform message authentication by confirming in accordance with the obligations described by the
110 `<saml2:SubjectConfirmation>` element.

111 When the sender wields the subject confirmation key to sign portions of the message the signature ensures the
112 authenticity and integrity of the portions covered by the signature. However, this alone does not mitigate the threat of
113 replay, insertion and certain classes of message modification attacks. To secure the message from such threats, one of
114 the mechanisms which support peer entity authentication (see the Peer Entity Authentication section in the Security
115 Mechanisms core) MAY be used or the underlying SOAP binding request processing model MUST address these
116 threats.

117 4.1.1. Sender Processing Rules

118 The core specification lists generic processing rules, which are to be augmented by the following SAML 2.0 specific
119 rules:

- 120 • The construction and decoration of the `<wsse:Security>` header element MUST adhere to the rules specified in
121 the [wss-sms11] and [wss-saml11].
- 122 • The sender MUST present the `<saml2:Assertion>` (as security token) by inserting it as a child of the
123 `<wsse:Security>` element.
- 124 • The sender MUST adhere to its subject confirmation obligation in accordance with the semantics of the confir-
125 mation method. This is described by one of the `<saml2:SubjectConfirmation>` elements carried within the
126 `<saml2:Subject>`
- 127 For deployment settings which REQUIRE independent message authentication, the obligation MUST be accom-
128 plished by signing elements of the message and decorating the `<wsse:Security>` element with the signature.
- 129 For deployment settings which DO NOT REQUIRE independent message authentication then the subject confirma-
130 tion obligation may be accomplished by correlating the certificate and key used to affect peer entity authentication
131 with the certificate and key described by the subject confirmation element. To accommodate this, the assertion
132 issuing authority MUST construct the assertion such that the confirmation key can be unambiguously verified to
133 be the same certificate and key used in establishing peer entity authentication. This is necessary to mitigate the
134 threat of a certificate substitution attack. It is RECOMMENDED that the certificate or certificate chain be bound
135 to the subject confirmation key.

136 4.1.2. Recipient Processing Rules

- 137 The core specification lists generic processing rules, which are to be augmented by the following SAML 2.0 specific
138 rules:
- 139 • The recipient MUST locate the `<saml2:Assertion>` (security token) and the recipient MUST determine that it
140 trusts the authority which issued the `<saml2:Assertion>`.
 - 141 • The recipient MUST validate the issuer's signature over the `<saml2:Assertion>`. The recipient SHOULD
142 validate the trust semantics of the signing key, as appropriate to the risk of incorrect authentication.
 - 143 • The recipient SHOULD verify that at least one of the confirmation obligations specified in the
144 `<saml2:SubjectConfirmation>` element has been met.
 - 145 • If the validation policy regards peer entity authentication sufficient for purposes of message authentication then
146 the recipient MUST locate the `<ds:KeyInfo>` element within `<saml2:SubjectConfirmation>` element. This
147 key MUST be unambiguously verified to be referring to the same certificate and key used in establishing peer
148 entity authentication.
 - 149 • The recipient MUST determine that it trusts the key used to sign the message.
 - 150 • When an OASIS X.509 token is used to convey key information, the recipient SHOULD validate the sender's
151 certificate and verify the certificate revocation status, as appropriate to the risk of incorrect authentication.

152 4.2. Bearer Token Authentication

153 A SAML 2.0 assertion may be used as a bearer token when the SubjectConfirmation element's Method attribute has
154 the value `urn:oasis:names:tc:SAML:2.0:cm:bearer`. Normative rules on the use of SAML 2.0 assertions as
155 SOAP Message Security tokens are provided in the OASIS WSS SAML Token Profile v1.1 [[wss-saml11](#)].

156 Particular attention must be paid to the proper validation of the `<saml2:AudienceRestriction>` element which
157 specifies the intended consumer(s) of the assertion. In this case the assertion construction guidance in [Section 4.1](#)
158 would apply.

159 4.2.1. Processing Rules

160 The bearer sender and receiver processing rules specified in core must be observed.

161 4.2.2. SAML Bearer Token Example

162 The following example demonstrates the Bearer message authentication mechanism by supplying a SAML bearer
163 token [[wss-saml11](#)] in the security header.

```
164 <?xml version="1.0" encoding="UTF-8"?>
165 <s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
166     xmlns:sb="urn:liberty:sb:2006-08"
167     xmlns:pp="urn:liberty:id-sis-pp:2003-08"
168     xmlns:sec="urn:liberty:security:2006-08"
169     xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecur
170 ity-secext-1.0.xsd"
171     xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-
172 1.0.xsd"
173     xmlns:wsa="http://www.w3.org/2005/08/addressing"
174     xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
175     xmlns:xenc="http://www.w3.org/2001/04/xmlenc#">
176
177   <s:Header>
178
179     <!-- see Liberty SOAP Binding Specification for which headers
180      are required and optional -->
181
182     <wsa:MessageID wsu:Id="mid">...</wsa:MessageID>
183
184     <wsa:To wsu:Id="to">...</wsa:To>
185
186     <wsa:Action wsu:Id="action">...</wsa:Action>
187
188     <wsse:Security mustUnderstand="1">
189
190       <wsu:Timestamp wsu:Id="ts">
191         <wsu:Created>2005-06-17T04:49:17Z</wsu:Created >
192       </wsu:Timestamp>
193
194       <!-- this is the bearer token -->
195       <saml2:Assertion
196         xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion"
197         Version="2.0"
198         ID="sxJu9g/vvLG9sAN9bKp/8q0NKU="
199         IssueInstant="2005-04-01T16:58:33.173Z">
200
201         <saml2:Issuer>http://authority.example.com/</Saml2:Issuer>
202
203         <!-- signature by the issuer over the assertion -->
204         <ds:Signature>...</ds:Signature>
205
206         <saml2:Subject>
207           <saml2:EncryptedID>
208             <xenc:EncryptedData>U2XTcNvRx7B11NK182nmY00TEk==</xenc:EncryptedData>
```

```
209     <xenc:EncryptedKey>...</xenc:EncryptedKey>
210   </saml2:EncryptedID>
211
212   <saml2:SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
213   </saml2:SubjectConfirmation>
214 </saml2:Subject>
215
216 <!-- By placing an audience restriction on the assertion we
217      can limit the scope of which entity should consume
218      the information in the assertion. -->
219
220 <saml2:Conditions
221   NotBefore="2005-04-01T16:57:20Z"
222   NotOnOrAfter="2005-04-01T21:42:43Z">
223
224   <saml2:AudienceRestrictionCondition>
225     <saml2:Audience>http://wsp.example.com</saml2:Audience>
226   </saml2:AudienceRestrictionCondition>
227 </saml2:Conditions>
228
229 <!-- The AuthnStatement carries information
230      that describes the authentication event
231      of the Subject to an Authentication Authority -->
232 <saml2:AuthnStatement
233   AuthnInstant="2005-04-01T16:57:30.000Z"
234   SessionIndex="6345789">
235   <saml2:AuthnContext>
236     <saml2:AuthnContextClassRef>
237       urn:oasis:names:tc:SAML:2.0:ac:classes>PasswordProtectedTransport
238     </saml2:AuthnContextClassRef>
239   </saml2:AuthnContext>
240 </saml2:AuthnStatement>
241
242 <!-- This AttributeStatement carries an EncryptedAttribute.
243      Once this element is decrypted with the supplied key
244      an <Attribute> element bearing an endpoint reference
245      can be found, specifying resources which the invoker may
246      access. Details on this element can be found in the
247      discovery service specification. -->
248
249 <saml2:AttributeStatement>
250   <saml2:EncryptedAttribute>
251     <xenc:EncryptedData Type="http://www.w3.org/2001/04/xmlenc#Element" >
252       mQEMAzRniWkAAAEH9RWirOeKDKyFAB7PoFazx3ftp0vWwbbzqXdgcX8fpEqSrlv4
253       YqUc70MiJcBtKbp3+jlD4HPUaurIqHA0vrDmMpM+sF2BnpND118f/mXCv3XbWhiL
254       ...
255       hg6nZ5c0I6L6Gn9A
256       =HCQY
257     </xenc:EncryptedData>
258     <xenc:EncryptedKey> ... </xenc:EncryptedKey>
259   </saml2:EncryptedAttribute>
260 </saml2:AttributeStatement>
261
262 </saml2:Assertion>
263
264 <!-- This SecurityTokenReference is used to reference the SAML
265      Assertion from a ds:Reference -->
266
267 <wsse:SecurityTokenReference
268   xmlns:wsse="..." xmlns:wsu="..." xmlns:wssell="..."
269   wsu:Id="str1"
270   wssell:TokenType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-p
271   rofile-1.1#SAMLV2.0">
272   <wsse:KeyIdentifier
273     ValueType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLID">
274     sxJu9g/vvLG9sAN9bKp/8q0NKU=
275   </wsse:KeyIdentifier>
```

```
276     </wsse:SecurityTokenReference>
277
278     <ds:Signature>
279       <ds:SignedInfo>
280         <!-- in general include a ds:Reference for each wsa: header
281             added according to SOAP binding -->
282
283         <!-- include the MessageID in the signature -->
284         <ds:Reference URI="#mid">...</ds:Reference>
285
286         <!-- include the To in the signature -->
287         <ds:Reference URI="#to">...</ds:Reference>
288
289         <!-- include the Action in the signature -->
290         <ds:Reference URI="#action">...</ds:Reference>
291
292         <!-- include the MessageID in the signature -->
293         <ds:Reference URI="#mid">...</ds:Reference>
294
295         <!-- include the Timestamp in the signature -->
296         <ds:Reference URI="#ts">...</ds:Reference>
297
298         <!-- include the SAML Assertion in the signature to avoid
299             token substitution attacks -->
300         <ds:Reference URI="#Str1">
301           <ds:Transform Algorithm="...#STR-Transform">
302             <wsse:TransformationParameters>
303               <ds:CanonicalizationMethod
304                 Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315" />
305             </wsse:TransformationParameters>
306           </ds:Transform>
307         </ds:Reference>
308
309         <!-- include the message body -->
310         <ds:Reference URI="#MsgBody">
311           <!-- bind to the body -->
312         </ds:Reference>
313       </ds:SignedInfo>
314       ...
315     </ds:Signature>
316   </wsse:Security>
317 </s:Header>
318 <s:Body wsu:Id="MsgBody">
319   <pp:Modify>
320     <!-- this is an ID-SIS-PP Modify message -->
321   </pp:Modify>
322 </s:Body>
323 </s:Envelope>
324
325
326
```

327 5. Message Authorization

328 5.1. Authorization Data Generation

329 The following mechanism description assumes that the Web Services Security SAML Token Profile [wss-saml11]
330 is utilized as the means by which the message sender authenticates to the message recipient. Each communicating
331 peer performs message level authentication by fulfilling the subject confirmation obligation. Typically this is
332 by demonstrating proof of possession of a subject confirmation key, where the assertion issuer binds the subject
333 confirmation key to the assertion by signing the assertion. This attestation provides assurance to the consumer of the
334 assertion that the subject confirmation key is that of the intended sender. Thus the sender's subject confirmation key
335 can be recognized by the recipient as belonging to the confirming peer. The assertion issuer should also bind a name
336 identifier to the subject confirmation element. This name binding would serve as an aid in associating the sender
337 with its confirmation key. Subsequent to the authentication of the sender the recipient can leverage this knowledge in
338 support of the authorization model described below.

339 The following processing rules are in addition to the processing rules specified in core and are specific to the use of
340 SAML 2.0 assertions.

341 5.1.1. Processing Rules

342 The assertion issuing authority constructs the assertion in accordance with the following rules:

- 343 • The assertion **MUST** indicate the invocation identity within the `<saml2:Subject>` element of the assertion.
344 The `<saml2:Subject>` element **MUST** include at least one `<saml2:SubjectConfirmation>` element. This
345 element **MUST** have a `Method` attribute with a value of `urn:oasis:names:tc:SAML2:2.0:cm:holder-of-key`.
346 (This requirement enables a proof of possession and binding to the message on behalf of the invoker).
347 The subject confirmation element **MUST** be specified with a `<saml2:SubjectConfirmationData>` element
348 qualified with an `xsi:type` of `saml2:KeyInfoConfirmationDataType` as specified in [SAMLCore2].
- 349 • When the invocation identity represents the identity of the sender, the `<saml2:Subject>` element is decorated as
350 follows. Refer to [Section 8.1.1](#) for an informative example.
351 The name identifier element **SHOULD** include a `<saml2:NameID>` element and the `Format` attribute value
352 **SHOULD** be `urn:oasis:names:tc:SAML2:2.0:nameid-format:entity`. Note: This identifier might
353 assist the relying party in locating metadata concerning the subject of the assertion.
354 The `<saml2:SubjectConfirmation>` element **SHOULD NOT** be decorated with a `<saml2:NameID>` element.
355 The reason is that the presence of the `<saml2:NameID>` is used to indicate that the sender is not the same as the
356 invoker, but acting on behalf of the invoker.
- 357 • When the invocation identity is **NOT** that of the sender (i.e., the sender is acting on behalf of the subject) the
358 `<saml2:Subject>` element is decorated as follows:
359 In an operational setting where the invocation identity (the subject) is only to be released to the relying party (the
360 audience) then the name identifier element **SHOULD** be of type `<saml2:EncryptedID>` and conform to the
361 guidance in [SAMLCore2]. Refer to [Section 8.1.2.2](#) for an informative example.
362 In settings where the invocation identity does not call for privacy protections then the name identifier element
363 **SHOULD** be conveyed using a `<saml2:NameID>` element with a `Format` attribute which is appropriate for the
364 operational setting. Refer to [Section 8.1.2.1](#) for an informative example.
365 To identify the confirming entity the `<saml2:SubjectConfirmation>` element **SHOULD** contain a
366 `<saml2:NameID>` element with a `Format` attribute value of `urn:oasis:names:tc:SAML2:2.0:nameid-format:entity`.
367 Note: This identifier might assist the relying party in locating metadata concerning the confirming entity as well
368 as help associate the name of the confirming entity in the application domain namespace with the key used for
369 subject confirmation.

- 370 • The assertion issuing authority MAY describe the authentication status of the interacting party by including
371 a `<saml2:AuthnStatement>` element which MUST include a `<saml2:AuthnContext>` element. Refer to
372 [Section 8.1.3](#) for an informative example.
- 373 • The assertion issuing authority MAY limit the resource which the invoker may access at the relying party by
374 describing the relevant resources in the `<saml2:AttributeStatement>`. This may be done by explicitly listing
375 endpoint references of the resources that the invoker may access.
- 376 In an operational setting where the value of the attribute requires confidentiality protections then the attribute
377 element SHOULD be of type `<saml2:EncryptedAttribute>` and conform to the guidance in [[SAMLCore2](#)].
- 378 If the confidentiality of the attribute is not a concern then the element SHOULD be conveyed using a
379 `<saml2:Attribute>`.
- 380 • OPTIONALLY, the assertion issuer MAY include information that assists in building a chain of transited providers.
381 How this is done is defined in the [Provider Chaining](#) section ([Section 6](#)).
- 382 • The assertion MUST be signed by the assertion issuing authority in accordance with the signing requirements
383 specified in [[SAMLCore2](#)].

384 5.1.2. Consuming Authorization Data

385 A recipient that exposes a resource typically makes access control decisions based on the invocation identity.
386 Additionally the recipient may also predicate access control policies upon the sender identity. The semantics of
387 resource access authorization are described in the Security Mechanisms core.

388 The recipient of an authorization assertion based on SAML 2.0 assertions determines the invocation identity by
389 inspecting the `<saml2:Subject>` element. If a proxy is involved in the communication then it's identity is carried
390 within the `<saml2:NameID>` element of the `<saml2:SubjectConfirmation>` element in effect. Providing both
391 the invocation identity and the proxy identity enables the recipient to tailor authorization policy to a finer degree
392 of granularity. That is, the recipient generally uses the invocation identity to make its authorization decisions and
393 potentially determine whether the proxy is permitted to access the resource on behalf of said invocation identity.

394 5.1.2.1. Processing Rules

395 The following processing rules are in addition to those specified in SecMech core.

- 396 • The recipient MUST locate the `<saml2:Assertion>` (security token) which conferred the subject confirmation
397 key relied upon for sender authentication.
- 398 The recipient MUST corroborate that the bound subject confirmation key is the same key used to authenticate the
399 communicating peer.
- 400 • The recipient MUST determine that it trusts the authority which signed the `<saml2:Assertion>`.
- 401 The recipient MUST validate the signature of the `<saml2:Assertion>`. The recipient SHOULD validate the
402 trust semantics of the signing key, as appropriate to the risk of incorrect authentication.

403 6. Provider Chaining

404 This profile defines how transited provider information should be recorded when a SAML 2.0 assertion is used as a
405 security token to convey provider chaining information. General discussion and overall normative requirements related
406 to provider chaining are in the Security Mechanisms core specification [[LibertySecMech20](#)].

407 When a Discovery Service issues a SAML 2.0 token to be used in provider chaining, the general structure of the
408 assertion may be informatively described as follows:

- 409 • Issuer
- 410 • Signature of entire assertion
- 411 • Provider Chaining (if needed)
- 412 • Audience Restriction Condition
- 413 • Subject of assertion (with corresponding confirmation method information)
- 414 • AuthnStatement (convey information about authentication of the subject)
- 415 • Endpoint reference information

416 To convey the provider chaining information, the SAML assertion SHOULD include a `<saml2:Advice>` el-
417 ement containing a single `<TransitedProviderPath>` element. This `<TransitedProviderPath>` MUST
418 contain a `<TransitedProvider>` element for each provider that has been transited. General use of the
419 `<TransitedProviderPath>` element is defined in the Security Mechanisms core specification [[Liberty-](#)
420 [SecMech20](#)].

421 Each `<TransitedProvider>` element MUST contain one URI element content value. This is used to enable the
422 recipient to verify the provider identity and will typically be the `ProviderID` of the transited provider. The
423 `ProviderID` is defined in the Discovery Specification. Each `<TransitedProvider>` element may also include
424 the confirmation URI indicating the form of confirmation the transited provider used to authenticate to the Discovery
425 Service and a timestamp for the interaction.

426 The following example shows a `<saml2:Assertion>` carrying a `<TransitedProviderPath>` with multiple
427 `<TransitedProvider>` elements.

428 6.1. Provider Chaining Example (Informative)

429 The following example demonstrates using SAML 2.0 assertions to convey provider chaining information, in
430 particular:

- 431 • Provider Chain captured in a single `<TransitedProviderPath>` with multiple `<TransitedProvider>` ele-
432 ments. Two different transited providers distinct from the sender are listed.
- 433 • Encrypted Name Identifier.
- 434 • Authentication status of Invoking Identity.

```

435 <?xml version="1.0" encoding="UTF-8"?>
436 <s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
437     xmlns:sb="urn:liberty:sb:2006-08"
438     xmlns:pp="urn:liberty:id-sis-pp:2003-08"
439     xmlns:sec="urn:liberty:security:2006-08"
440     xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecur
441 ity-secext-1.0.xsd"
442     xmlns:wssell="http://docs.oasis-open.org/wss/2005/xx/oasis-2005xx-wss-wssecurity-secex
443 t-1.1.xsd"
444     xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-ws
445 s-wssecurity-utility-1.0.xsd"
446     xmlns:wsa="http://www.w3.org/2005/08/addressing"
447     xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
448     xmlns:xenc="http://www.w3.org/2001/04/xmlenc#">
449
450 <s:Header>
451 <!-- see Liberty SOAP Binding Specification for which headers
452     are required and optional -->
453
454 <wsa:MessageID wsu:Id="mid">...</wsa:MessageID>
455
456 <wsa:To wsu:Id="to">...</wsa:To>
457
458 <wsa:Action wsu:Id="action">...</wsa:Action>
459
460 <wsse:Security mustUnderstand="1">
461
462 <wsu:Timestamp wsu:Id="ts">
463 <wsu:Created>2005-06-17T04:49:17Z</wsu:Created >
464 </wsu:Timestamp>
465
466 <saml2:Assertion
467     xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion"
468     Version="2.0"
469     ID="sxJu9g/vvLG9sAN9bKp/8q0NKU="
470     IssueInstant="2005-04-01T16:58:33.173Z">
471
472 <saml2:Issuer>http://authority.example.com/</saml2:Issuer>
473
474 <!-- signature by the issuer over the assertion -->
475 <ds:Signature>...</ds:Signature>
476
477 <saml2:Advice>
478 <sec:TransitedProviderPath>
479 <TransitedProvider>http://www.example.com/one</TransitedProvider>
480 <TransitedProvider>http://www.example.com/two</TransitedProvider>
481 </sec:TransitedProviderPath>
482 </saml2:Advice>
483
484 <!-- By placing an audience restriction on the assertion we
485     can limit the scope of which entity should consume
486     the information in the assertion. -->
487
488 <saml2:Conditions
489     NotBefore="2005-04-01T16:57:20Z"
490     NotOnOrAfter="2005-04-01T21:42:43Z">
491
492 <saml2:AudienceRestrictionCondition>
493 <saml2:Audience>http://wsp.example.com</saml2:Audience>
494 </saml2:AudienceRestrictionCondition>
495 </saml2:Conditions>
496
497 <saml2:Subject>
498 <saml2:EncryptedID>
499 <xenc:EncryptedData>U2XTCNvRx7BllNK182nmY00TEk==</xenc:EncryptedData>
500 <xenc:EncryptedKey>...</xenc:EncryptedKey>
501 </saml2:EncryptedID>

```

```

502
503     <saml2:SubjectConfirmation
504         Method="urn:oasis:names:tc:SAML:2.0:cm:holder-of-key">
505     <saml2:NameID format="urn:oasis:names:tc:SAML:2.0:nameid-format:entity">
506         http://third.example.com/
507     </saml2:NameID>
508     <saml2:SubjectConfirmationData xsi:type="saml2:KeyInfoConfirmationDataType">
509
510         <!-- This keyinfo is the key by which the sender must
511             prove possession in order for the relying party to
512             accept the Statements in this assertion. -->
513         <ds:KeyInfo>
514             <ds:KeyName>
515                 CN=third.example.com,OU=Client Services R US,O=Service Station,...
516             </ds:KeyName>
517             <ds:KeyValue>...</ds:KeyValue>
518         </ds:KeyInfo>
519     </saml2:SubjectConfirmationData>
520 </saml2:SubjectConfirmation>
521 </saml2:Subject>
522
523 <!-- The AuthnStatement carries information
524     that describes the authentication event
525     of the Subject to an Authentication Authority -->
526 <saml2:AuthnStatement
527     AuthnInstant="2005-04-01T16:57:30.000Z"
528     SessionIndex="6345789">
529     <saml2:AuthnContext>
530         <saml2:AuthnContextClassRef>
531             urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport
532         </saml2:AuthnContextClassRef>
533     </saml2:AuthnContext>
534 </saml2:AuthnStatement>
535
536 <!-- The AttributeStatement carries an EncryptedAttribute.
537     Once this element is decrypted with the supplied key
538     an <Attribute> element bearing an endpoint reference
539     can be found. Details on this element can be found in the
540     discovery service specification. -->
541
542 <saml2:AttributeStatement>
543     <saml2:EncryptedAttribute>
544         <xenc:EncryptedData Type="http://www.w3.org/2001/04/xmlenc#Element">
545             mQEMAzRniWkAAAEH9RWir0eKDkyFAB7P oFazx3ftp0vWwbbzqXdgcX8fpEqSrl v4
546             YqUc7OMiJcBtKBp3+jlD4HPUaurIqHA0vrDmMpM+sF2 BnpND118f/mXCv3XbWhiL
547             xj1/M4y0CMAM/wBHT3xa17tWJwsZkDRLWxXP7wSlTXNjCThHzBL8gBKZRqNBcZlU
548             ...
549             VRu9BpYBD4Y/98y1jtX9Pm898+zzketoc4ZvhCgh9P0arVK 1B3cKxB87bKiDDWAU
550             hg6nZ5c0I6L6Gn9A
551             =HCQY
552         </xenc:EncryptedData>
553         <xenc:EncryptedKey> ... </xenc:EncryptedKey>
554     </saml2:EncryptedAttribute>
555 </saml2:AttributeStatement>
556 </saml2:Assertion>
557
558 <!-- This SecurityTokenReference is used to reference the SAML
559     Assertion from a ds:Reference -->
560
561 <wsse:SecurityTokenReference
562     xmlns:wsse="..." xmlns:wsu="..." xmlns:wssell="..."
563     wsu:Id="str1"
564     wssell:TokenType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#S
565     AMLV2.0">
566     <wsse:KeyIdentifier
567         ValueType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLID">
568         sxJu9g/vvLG9sAN9bKp/8q0NKU=

```



```

569     </wsse:KeyIdentifier>
570 </wsse:SecurityTokenReference>
571
572 <!-- this is the signature the sender generated to demonstrate
573 holder-of-key -->
574
575 <ds:Signature>
576   <ds:SignedInfo>
577     <!-- in general include a ds:Reference for each wsa: header
578          added according to SOAP binding -->
579
580     <!-- include the MessageID in the signature -->
581     <ds:Reference URI="#mid">...</ds:Reference>
582
583     <!-- include the To in the signature -->
584     <ds:Reference URI="#to">...</ds:Reference>
585
586     <!-- include the Action in the signature -->
587     <ds:Reference URI="#action">...</ds:Reference>
588
589     <!-- include the MessageID in the signature -->
590     <ds:Reference URI="#mid">...</ds:Reference>
591
592     <!-- include the Timestamp in the signature -->
593     <ds:Reference URI="#ts">...</ds:Reference>
594
595     <!-- include the SAML Assertion in the signature to avoid
596          token substitution attacks -->
597     <ds:Reference URI="#Str1">
598       <ds:Transform Algorithm="...#STR-Transform">
599         <wsse:TransformationParameters>
600           <ds:CanonicalizationMethod
601             Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315" />
602           </wsse:TransformationParameters>
603         </ds:Transform>
604       </ds:Reference>
605
606     <!-- include the message body -->
607     <ds:Reference URI="#MsgBody">
608       <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
609       <ds:DigestValue>YgGfs0pi56pu...</ds:DigestValue>
610     </ds:Reference>
611   </ds:SignedInfo>
612
613   <ds:SignatureValue>
614     HJJWbvqW9E84vJVQkjJLLA6nNvBX7mY00TZhWbDFNDElGscSXZ5Ekw==
615   </ds:SignatureValue>
616
617   <ds:KeyInfo>
618     <wsse:SecurityTokenReference
619       wssell:TokenType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1
620 #SAMLV2.0">
621       <wsse:KeyIdentifier
622         ValueType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLID">
623         2sxJu9g/vvLG9sAN9bKp/8q0NKU=
624       </wsse:KeyIdentifier>
625     </wsse:SecurityTokenReference>
626   </ds:KeyInfo>
627
628 </ds:Signature>
629 </wsse:Security>
630
631 </s:Header>
632 <s:Body id="MsgBody">
633   <pp:Modify>
634     <!-- this is an ID-SIS-PP Modify message -->
635   </pp:Modify>

```

```
636 </s:Body>
637 </s:Envelope>
638
639
```

640 **7. Identity Token**

641 Identity tokens are used to identify parties in flows where the identity of a party related to a use case is distinct from
642 an authenticated invoker.

643 **7.1. Identity Token Requirements**

644 Identity tokens that are implemented using SAML 2.0 assertions must meet the following requirements:

- 645 1. The subject of the identity token **MUST** represent the identity to be associated with the token.
- 646 2. The identity token **SHOULD** contain an attribute containing the endpoint reference for the Discovery Service
647 associated with the subject identity. The bootstrap attribute is defined in the ID-WSF 2.0 Discovery Service
648 Specification [[LibertyDisco](#)].
- 649 3. The Identity token **SHOULD** have an AudienceRestrictionCondition as part of the SAML assertion Condition
650 element.

651 8. Examples (Informative)

652 These examples demonstrate SAML 2.0 assertions.

653 8.1. Fragmentary Examples

654 The examples in this section are fragments of full assertions - they are intended to demonstrate a particular aspect of
655 the message syntax.

656 8.1.1. Sender as Invocation Identity

657 In the simplest of settings the sender of a message is acting on its own behalf. The assertion issuing authority identifies
658 the sender as the subject of the assertion.

```
659 001 <saml2:Subject xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion" >
660 002   <saml2:NameID format="urn:oasis:names:tc:SAML:2.0:nameid-format:entity">
661 003     http://example.com/
662 004   </saml2:NameID>
663 005   <saml2:SubjectConfirmation
664 006     Method="urn:oasis:names:tc:SAML:2.0:cm:holder-of-key">
665 007     <saml2:SubjectConfirmationData xsi:type="saml2:KeyInfoConfirmationDataType">
666 008       <!-- This keyinfo is the key by which the sender must
667 009         prove possession in order for the relying party to
668 010         accept the Statements in this assertion. -->
669 011       <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#" >
670 012         <ds:KeyName>
671 013           CN=example.com,OU=SomeDepartment,O=SomeOrganization,...
672 014         </ds:KeyName>
673 015         <ds:KeyValue>...</ds:KeyValue>
674 016       </ds:KeyInfo>
675 017     </saml2:SubjectConfirmationData>
676 018   </saml2:SubjectConfirmation>
677 019 </saml2:Subject>
678
```

679 Contents in the above example worth particular mention include lines 002-004 which specify the identifier is an entity
680 id and the name of the sender. Lines 005-018 describe the confirmation requirements that the sender must uphold
681 to be confirmed as the subject of the assertion. Line 006 mandates that the sender demonstrate possession of the
682 confirmation key described in lines 011-016.

683 8.1.2. Sender as Transited Provider Identity

684 At times it is necessary to convey multiple identities to a relying party. One identity is the invoking identity, the
685 subject of the assertion. The other is that of a transited provider, a sender which is acting on behalf of the subject
686 whose identity needs to be distinguished from that of the subject. To accomplish this the assertion issuer specifies the
687 sender identity with a `saml2:NameID` element within the `saml2:SubjectConfirmation` element of the assertion.

688 8.1.2.1. Transparent Subject Identifier

689 In the following example the identity of the subject is transparent to the transited provider and the transited provider
690 is identified as the confirming entity. The presence of the name identifier in the `saml2:SubjectConfirmation`
691 element indicates that a transited provider is used.

```
692 001 <saml2:Subject xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion">
693 002   <saml2:NameID Format="urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress">
694 003     somebody@someplace.example.com
695 004   </saml2:NameID>
696 005   <saml2:SubjectConfirmation
697 006     Method="urn:oasis:names:tc:SAML:2.0:cm:holder-of-key">
698 007     <saml2:NameID format="urn:oasis:names:tc:SAML:2.0:nameid-format:entity">
699 008       http://somemailhost.example.com/
```

```
700 009     </saml2:NameID>
701 010 <saml2:SubjectConfirmationData xsi:type="saml:KeyInfoConfirmationDataType">
702 011 <!-- This keyinfo is the key by which the sender (aka proxy) must
703 012     prove possession in order for the relying party to
704 013     accept the Statements in this assertion. -->
705 014 <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
706 015     <ds:KeyName>
707 016         CN=somemailhost.example.com,OU=SomePlace,O=ExampleOrg,...
708 017     </ds:KeyName>
709 018     <ds:KeyValue>...</ds:KeyValue>
710 019 </ds:KeyInfo>
711 020 </saml2:SubjectConfirmationData>
712 021 </saml2:SubjectConfirmation>
713 022 </saml2:Subject>
714
715
```

716 In the above example the noteworthy elements are described. Lines 002-004 describe the identity of the subject, aka the
717 invocation identity. Lines 005-020 describe the confirmation requirements that the sender must uphold to be confirmed
718 as the subject of the assertion. Line 006 mandates that the sender demonstrate possession of the confirmation key
719 described in lines 010-020. Lines 007-009 identify the name of the proxy.

720 8.1.2.2. Opaque Subject Identifier

721 In the following example, the identity of the subject is made opaque to the proxy through encryption and the proxy is
722 identified as the confirming entity.

```
723 001 <saml2:Subject xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion">
724 002 <saml2:EncryptedID xmlns:xenc="http://www.w3.org/2001/04/xmlenc#">
725 003 <xenc:EncryptedData>U2XTCNvRX7B1lNK182nmY00TEk==</xenc:EncryptedData>
726 004 <xenc:EncryptedKey>...</xenc:EncryptedKey>
727 005 </saml2:EncryptedID>
728 006 <saml2:SubjectConfirmation
729 007     Method="urn:oasis:names:tc:SAML:2.0:cm:holder-of-key">
730 008 <saml2:NameID format="urn:oasis:names:tc:SAML:2.0:nameid-format:entity">
731 009     http://somemailhost.example.com/
732 010 </saml2:NameID>
733 011 <saml2:SubjectConfirmationData xsi:type="saml:KeyInfoConfirmationDataType">
734 012 <!-- This keyinfo is the key by which the sender (aka proxy) must
735 013     possession in order for the relying party to
736 014     the Statements in this assertion. -->
737 015 <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
738 016     <ds:KeyName>
739 017         CN=somemailhost.example.com,OU=SomePlace,O=ExampleOrg,...
740 018     </ds:KeyName>
741 019     <ds:KeyValue>...</ds:KeyValue>
742 020 </ds:KeyInfo>
743 021 </saml2:SubjectConfirmationData>
744 022 </saml2:SubjectConfirmation>
745 023 </saml2:Subject>
746
747
```

748 This example is very similar to the previous. The difference is that the name identifier for the subject of the assertion
749 is encrypted, lines 002-005.

750 8.1.3. Invoking Identity Authentication

751 The relying party may need information regarding the authentication of the subject (aka invocation identity.) To
752 accommodate this the assertion issuer includes a <saml2:AuthnStatement> as part of the assertion, providing
753 additional information about the invoker specified in the Subject.

```
754 001 <!-- The saml2:AuthnStatement carries information that
755 002     describes the authentication event of the subject
```

```
756 003     to an authenticating authority -->
757 004 <saml2:AuthnStatement
758 005     xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion"
759 006     AuthnInstant="2005-04-01T16:57:30.000Z"
760 007     SessionIndex="6345789">
761 008 <saml2:AuthnContext>
762 009     <saml2:AuthnContextClassRef>
763 010     urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport
764 011 </saml2:AuthnContextClassRef>
765 012 </saml2:AuthnContext>
766 013 </saml2:AuthnStatement>
767
768
```

769 Lines 006-007 describe attributes of the authentication event. Line 006 indicates the time at which authentication
770 occurred. The session index between the subject and the authentication authority is on line 007. Lines 008-012
771 provide the technical details of the authentication action itself.

772 8.1.4. Resource as an Attribute

773 The assertion issuer may make coarse-grained authorization decisions and in so doing specify precisely the resource
774 for which the assertion is targeted. By identifying the resource in an attribute statement and binding the statement to
775 the assertion the relying party can base its authorization decision on the bound attribute and the actual resource being
776 accessed. However, applications that use this specification may have alternative methods of referring to resources and
777 thus disseminating this information in an attribute statement may be redundant.

778 8.2. Proxying with Authentication Context of the Invoking Identity

779 Access to resources exposed by a service instance is nominally restricted by access control policy enforced by the
780 entity hosting the resource. Additionally, the policy information, enforcement and decision points may be distributed
781 across multiple system entities. Authorization to access a resource may require that the entity interacting (e.g. browser
782 principal) with another entity (e.g. service consumer) have an active authenticated session.

783 To facilitate this scenario the trusted authority may supply authorization data that conveys the session status of the
784 interacting entity. This is accomplished by including a `<saml2:AuthnStatement>` in the assertion.

785 The following example demonstrates:

- 786 • Proxying
- 787 • Encrypted Name Identifier
- 788 • Encrypted Endpoint Reference conveyed as Attribute

```
789 <?xml version="1.0" encoding="UTF-8"?>
790 <s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
791     xmlns:sb="urn:liberty:sb:2006-08"
792     xmlns:pp="urn:liberty:id-sis-pp:2003-08"
793     xmlns:sec="urn:liberty:security:2006-08"
794     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance#"
795     xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss
796 -wssecurity-secext-1.0.xsd"
797     xmlns:wss11="http://docs.oasis-open.org/wss/2005/xx/oasis-2005xx-wss-wssecuri
798 ty-secext-1.1.xsd"
799     xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1
800 .0.xsd"
801     xmlns:wsa="http://www.w3.org/2005/08/addressing"
802     xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
803     xmlns:xenc="http://www.w3.org/2001/04/xmlenc#">
804
805 <s:Header>
```

```
806
807 <!-- see Liberty SOAP Binding Specification for which headers
808 are required and optional -->
809
810 <wsa:MessageID wsu:Id="mid">...</wsa:MessageID>
811
812 <wsa:To wsu:Id="to">...</wsa:To>
813
814 <wsa:Action wsu:Id="action">...</wsa:Action>
815
816 <wsse:Security mustUnderstand="1">
817
818 <wsu:Timestamp wsu:Id="ts">
819 <wsu:Created>2005-06-17T04:49:17Z</wsu:Created >
820 </wsu:Timestamp>
821
822 <saml2:Assertion
823 xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion"
824 Version="2.0"
825 ID="sxJu9g/vvLG9sAN9bKp/8q0NKU="
826 IssueInstant="2005-04-01T16:58:33.173Z">
827
828 <saml2:Issuer>http://authority.example.com</saml2:Issuer>
829
830 <!-- signature by the issuer over the assertion -->
831 <ds:Signature>...</ds:Signature >
832
833 <!-- By placing an audience restriction on the assertion we
834 can limit the scope of which entity should consume
835 the information in the assertion. -->
836
837 <saml2:Conditions
838 NotBefore="2005-04-01T16:57:20Z"
839 NotOnOrAfter="2005-04-01T21:42:43Z">
840
841 <saml2:AudienceRestrictionCondition>
842 <saml2:Audience>http://wsp.example.com</saml2:Audience>
843 </saml2:AudienceRestrictionCondition>
844 </saml2:Conditions>
845
846 <saml2:Subject>
847 <saml2:EncryptedID>
848 <xenc:EncryptedData>U2XTCNvRX7BllNK182nmY00TEk==</xenc:EncryptedData>
849 <xenc:EncryptedKey>...</xenc:EncryptedKey>
850 </saml2:EncryptedID>
851
852 <saml2:SubjectConfirmation
853 Method="urn:oasis:names:tc:SAML:2.0:cm:holder-of-key">
854 <saml2:NameID format="urn:oasis:names:tc:SAML:2.0:nameid-format:entity">
855 http://wsc.example.com/
856 </saml2:NameID>
857 <saml2:SubjectConfirmationData xsi:type="saml2:KeyInfoConfirmationDataType">
858
859 <!-- This keyinfo is the key by which the sender must
860 prove possession in order for the relying party to
861 accept the Statements in this assertion. -->
862 <ds:KeyInfo>
863 <ds:KeyName>
864 CN=wsc.example.com,OU=Client Services R US,O=Service Station,...
865 </ds:KeyName>
866 <ds:KeyValue>...</ds:KeyValue>
867 </ds:KeyInfo>
868 </saml2:SubjectConfirmationData>
869 </saml2:SubjectConfirmation>
870 </saml2:Subject>
871
872 <!-- The AuthnStatement carries information
```

```

873         that describes the authentication event
874         of the Subject to an Authentication Authority -->
875     <saml2:AuthnStatement
876         AuthnInstant="2005-04-01T16:57:30.000Z"
877         SessionIndex="6345789">
878         <saml2:AuthnContext>
879             <saml2:AuthnContextClassRef>
880                 urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport
881             </saml2:AuthnContextClassRef>
882         </saml2:AuthnContext>
883     </saml2:AuthnStatement>
884
885     <!-- The AttributeStatement carries an EncryptedAttribute.
886         Once this element is decrypted with the supplied key
887         an <Attribute> element bearing an endpoint reference
888         can be found. Details on this element can be found in the
889         discovery service specification. -->
890
891     <saml2:AttributeStatement>
892     <saml2:EncryptedAttribute>
893         <xenc:EncryptedData Type="http://www.w3.org/2001/04/xmlenc#Element">
894             mQEMAZRniWkAAAEH9RWir0eKDkyFAB7PoFazx3ftp0vWwbbzqXdg cX8fpEqSr1v4
895             YqUc70MiJcBtKbp3+jlD4HPUaurIqHA0v rdmMpM+sF2BnpND118f/mXCv3XbWhi L
896             xj1/M4y0CMAM/wBHT3xa17tWJwsZkDRLWxXP7wSlTXNj CThHzBL8gBKZRqNBcZlU
897             ...
898             VRu9BpYBD4Y/98y1jtX9Pm898+zxketoc4Zvh Cgh9P0arVK1B3cKxB87bKiDDWAU
899             hg6nZ5c0I6L6Gn9A
900             =HCQY
901         </xenc:EncryptedData>
902         <xenc:EncryptedKey> ... </xenc:EncryptedKey>
903     </saml2:EncryptedAttribute>
904 </saml2:AttributeStatement>
905
906 </saml2:Assertion>
907
908 <!-- This SecurityTokenReference is used to reference the SAML
909 Assertion from a ds:Reference -->
910
911 <wsse:SecurityTokenReference
912     xmlns:wsse="..." xmlns:wsu="..." xmlns:wssell="..."
913     wsu:Id="str1"
914     wssell:TokenType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-pr
915 ofile-1.1#SAMLV2.0">
916     <wsse:KeyIdentifier
917         ValueType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLID">
918         sxJu9g/vvLG9sAN9bKp/8q0NKU=
919     </wsse:KeyIdentifier>
920 </wsse:SecurityTokenReference>
921
922 <!-- this is the signature the sender generated to demonstrate
923 holder-of-key -->
924
925 <ds:Signature>
926 <ds:SignedInfo>
927     <!-- in general include a ds:Reference for each wsa: header
928         added according to SOAP binding -->
929
930     <!-- include the MessageID in the signature -->
931     <ds:Reference URI="#mid">...</ds:Reference>
932
933     <!-- include the To in the signature -->
934     <ds:Reference URI="#to">...</ds:Reference>
935
936     <!-- include the Action in the signature -->
937     <ds:Reference URI="#action">...</ds:Reference>
938
939     <!-- include the MessageID in the signature -->

```



```
940     <ds:Reference URI="#mid">...</ds:Reference>
941
942     <!-- include the Timestamp in the signature -->
943     <ds:Reference URI="#ts">...</ds:Reference>
944
945     <!-- include the SAML Assertion in the signature to avoid
946           token substitution attacks -->
947     <ds:Reference URI="#Str1">
948       <ds:Transform Algorithm="...#STR-Transform">
949         <wsse:TransformationParameters>
950           <ds:CanonicalizationMethod
951             Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315" />
952         </wsse:TransformationParameters>
953       </ds:Transform>
954     </ds:Reference>
955
956     <!-- include the message body -->
957     <ds:Reference URI="#MsgBody">
958       <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
959       <ds:DigestValue>YgGfS0pi56pu...</ds:DigestValue>
960     </ds:Reference>
961   </ds:SignedInfo>
962
963   <ds:SignatureValue>
964     HJJWbvqW9E84vJVQkjjLLA6nNvBX7mY00TZhwBdFNDElgsScSXZ5Ekw==
965   </ds:SignatureValue>
966
967   <ds:KeyInfo>
968     <wsse:SecurityTokenReference
969       wsse11:TokenType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-
970 profile-1.1#SAMLV2.0">
971       <wsse:KeyIdentifier
972         ValueType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLID">
973         2sxJu9g/vvLG9sAN9bKp/8q0NKU=
974       </wsse:KeyIdentifier>
975     </wsse:SecurityTokenReference>
976   </ds:KeyInfo>
977 </ds:Signature>
978
979 </wsse:Security>
980
981 </s:Header>
982 <s:Body wsu:Id="MsgBody">
983   <pp:Modify>
984     <!-- this is an ID-SIS-PP Modify message -->
985   </pp:Modify>
986 </s:Body>
987 </s:Envelope>
988
```

989 8.3. Conveyance of Sender as Invocation Identity

990 This example depicts a request to access an identity-based web service in which the sender identity and the invocation
991 identity are the same (i.e. non-proxying). The resource which the sender is attempting to access is described in an
992 <AttributeStatement> within the assertion.

993 Note that, while the assertion associates a subject's name with a key, this association is made as a means to indicate
994 the authorization of that subject, acting with that key, to invoke a service. This facility, incorporated for authorization
995 purposes, serves a distinct and complementary function to the binding between subject and key, which the subject's
996 certificate accomplishes for authentication purposes.

997 The example demonstrates:

- 998 • Sender is Invocation Identity.
- 999 • Endpoint Reference conveyed as attribute without encryption.

```
1000
1001 <?xml version="1.0" encoding="UTF-8"?>
1002 <s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
1003     xmlns:sb="urn:liberty:sb:2006-08"
1004     xmlns:pp="urn:liberty:id-sis-pp:2003-08"
1005     xmlns:sec="urn:liberty:security:2006-08"
1006     xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssec
1007     urity-secext-1.0.xsd"
1008     xmlns:wssell="http://docs.oasis-open.org/wss/2005/xx/oasis-2005xx-wss-wssecurity-sec
1009     ext-1.1.xsd"
1010     xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
1011     wss-wssecurity-utility-1.0.xsd"
1012     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance#"
1013     xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
1014     xmlns:wsa="http://www.w3.org/2005/03/addressing">
1015
1016   <s:Header>
1017
1018     <!-- see Liberty SOAP Binding Specification for which headers
1019     are required and optional -->
1020
1021     <wsa:MessageID wsu:Id="mid">...</wsa:MessageID>
1022
1023     <wsa:To wsu:Id="to">...</wsa:To>
1024
1025     <wsa:Action wsu:Id="action">...</wsa:Action>
1026
1027     <wsse:Security mustUnderstand="1">
1028
1029       <wsu:Timestamp wsu:Id="ts">
1030         <wsu:Created>2005-06-17T04:49:17Z</wsu:Created >
1031       </wsu:Timestamp>
1032
1033       <saml2:Assertion
1034         xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion"
1035         Version="2.0"
1036         ID="sxJu9g/vvLG9sAN9bKp/8q0NKU="
1037         IssueInstant="2005-04-01T16:58:33.173Z">
1038
1039         <saml2:Issuer>http://authority.example.com/</saml2:Issuer>
1040
1041         <!-- signature by the issuer over the assertion -->
1042         <ds:Signature>...</ds:Signature>
1043
1044         <!-- By placing an audience restriction on the assertion we
1045         can limit the scope of which entity should consume
1046         the information in the assertion. -->
1047
1048         <saml2:Conditions
1049           NotBefore="2005-04-01T16:57:20Z"
1050           NotOnOrAfter="2005-04-01T21:42:43Z">
1051
1052           <saml2:AudienceRestrictionCondition>
1053             <saml2:Audience>http://wsp.example.com</saml2:Audience>
1054           </saml2:AudienceRestrictionCondition>
1055         </saml2:Conditions>
1056
1057         <saml2:Subject>
1058           <saml2:NameID format="urn:oasis:names:tc:SAML:2.0:nameid-format:entity">
1059             http://example.com/</saml2:NameID>
1060           <saml2:SubjectConfirmation
1061             Method="urn:oasis:names:tc:SAML:2.0:cm:holder-of-key">
1062             <saml2:SubjectConfirmationData xsi:type="saml2:KeyInfoConfirmationData Type">
```

```

1063         <!-- This keyinfo is the key by which the sender must
1064             prove possession in order for the relying party to
1065             accept the Statements in this assertion. -->
1066         <ds:KeyInfo>
1067             <ds:KeyName>
1068                 CN=example.com,OU=SomeDivision,O=SomeOrganization,...
1069             </ds:KeyName>
1070             <ds:KeyValue>...</ds:KeyValue>
1071         </ds:KeyInfo>
1072     </saml2:SubjectConfirmationData>
1073 </saml2:SubjectConfirmation>
1074 </saml2:Subject>
1075
1076 <!-- For details on the contents of the Endpoint Reference see the
1077     discovery service specification which has details -->
1078 <saml2:AttributeStatement>
1079     <saml2:Attribute NameFormat="urn:liberty:disco:2005-06"
1080         Name="IDWSFEPR">
1081         <saml2:AttributeValue>
1082             <wsa:EndpointReference>
1083                 ...
1084             </wsa:EndpointReference>
1085         </saml2:AttributeValue>
1086     </saml2:Attribute>
1087 </saml2:AttributeStatement>
1088 </saml2:Assertion>
1089
1090 <!-- This SecurityTokenReference is used to reference the SAML
1091     Assertion from a ds:Reference -->
1092
1093 <wsse:SecurityTokenReference
1094     xmlns:wsse="..." xmlns:wsu="..." xmlns:wssell="..."
1095     wsu:Id="str1"
1096     wssell:TokenType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLV2
1097 .0">
1098     <wsse:KeyIdentifier
1099         ValueType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLID">
1100         sxJu9g/vvLG9sAN9bKp/8q0NKU=
1101     </wsse:KeyIdentifier>
1102 </wsse:SecurityTokenReference>
1103
1104 <!-- this is the signature the sender generated to demonstrate
1105     holder-of-key the signature should cover the isf header and body-->
1106
1107 <ds:Signature>
1108     <ds:SignedInfo>
1109         <!-- in general include a ds:Reference for each wsa: header
1110             added according to SOAP binding -->
1111
1112         <!-- include the MessageID in the signature -->
1113         <ds:Reference URI="#mid">...</ds:Reference>
1114
1115         <!-- include the To in the signature -->
1116         <ds:Reference URI="#to">...</ds:Reference>
1117
1118         <!-- include the Action in the signature -->
1119         <ds:Reference URI="#action">...</ds:Reference>
1120
1121         <!-- include the MessageID in the signature -->
1122         <ds:Reference URI="#mid">...</ds:Reference>
1123
1124         <!-- include the Timestamp in the signature -->
1125         <ds:Reference URI="#ts">...</ds:Reference>
1126
1127         <!-- include the SAML Assertion in the signature to avoid
1128             token substitution attacks -->
1129         <ds:Reference URI="#Str1">

```

```
1130     <ds:Transform Algorithm="...#STR-Transform">
1131     <wsse:TransformationParameters>
1132     <ds:CanonicalizationMethod
1133     Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315" />
1134     </wsse:TransformationParameters>
1135     </ds:Transform>
1136 </ds:Reference>
1137
1138 <!-- include the message body -->
1139 <ds:Reference URI="#MsgBody">
1140 <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
1141 <ds:DigestValue>YgGfS0pi56pu...</ds:DigestValue>
1142 </ds:Reference>
1143 </ds:SignedInfo>
1144
1145 <ds:SignatureValue>
1146 HJJWbvqW9E84vJVQkjjLLA6nNvBX7mY00TzhwBdFNDElgsc SXZ5Ekw==
1147 </ds:SignatureValue>
1148
1149 <ds:KeyInfo>
1150 </ds:KeyInfo>
1151
1152 </ds:Signature>
1153 </wsse:Security>
1154 </s:Header>
1155 <s:Body wsu:Id="MsgBody">
1156 <pp:Modify>
1157 <!-- this is an ID-SIS-PP Modify message -->
1158 </pp:Modify>
1159 </s:Body>
1160 </s:Envelope>
1161
1162
```

1163 Details on the use of Endpoint References can be found in the discovery service specification.

1164 **References**

1165 **Normative**

- 1166 [LibertyDisco] Hodges, Jeff, Cahill, Conor, eds. "Liberty ID-WSF Discovery Service Specification," Version 2.0,
1167 Liberty Alliance Project (30 July, 2006). <http://www.projectliberty.org/specs>
- 1168 [LibertySecMech20] Hirsch, Frederick, eds. "Liberty ID-WSF Security Mechanisms Core," Version v2.0, Liberty
1169 Alliance Project (30 July, 2006). <http://www.projectliberty.org/specs>
- 1170 [SAMLCore2] Cantor, Scott, Kemp, John, Philpott, Rob, Maler, Eve, eds. (15 March 2005). "Assertions
1171 and Protocol for the OASIS Security Assertion Markup Language (SAML) V2.0," SAML V2.0, OA-
1172 SIS Standard, Organization for the Advancement of Structured Information Standards [http://docs.oasis-
open.org/security/saml/v2.0/saml-core-2.0-os.pdf](http://docs.oasis-
1173 open.org/security/saml/v2.0/saml-core-2.0-os.pdf)
- 1174 [SAMLBind2] Cantor, Scott, Hirsch, Frederick, Kemp, John, Philpott, Rob, Maler, Eve, eds. (15 March
1175 2005). "Bindings for the OASIS Security Assertion Markup Language (SAML) V2.0," SAML V2.0, OA-
1176 SIS Standard, Organization for the Advancement of Structured Information Standards [http://docs.oasis-
open.org/security/saml/v2.0/saml-bindings-2.0-os.pdf](http://docs.oasis-
1177 open.org/security/saml/v2.0/saml-bindings-2.0-os.pdf)
- 1178 [wss-sms11] Hallam-Baker, Phillip, Kaler, Chris, Monzillo, Ronald, Nadalin, Anthony, eds. (June 28, 2005).
1179 "Web Services Security: SOAP Message Security 1.1 (WS-Security 2004)," Public Review Draft - 28
1180 June 2005, Organization for the Advancement of Structured Information Standards [http://www.oasis-
open.org/committees/download.php/13397/wss-v1.1-spec-pr-SOAPMessageSecurity-01.pdf](http://www.oasis-
1181 open.org/committees/download.php/13397/wss-v1.1-spec-pr-SOAPMessageSecurity-01.pdf)
- 1182 [wss-saml11] Monzillo, Ronald, Kaler, Chris, Nadalin, Anthony, Hallam-Baker, Phillip, eds. (June 28,
1183 2005). Organization for the Advancement of Structured Information Standards [http://www.oasis-
1185 open.org/committees/download.php/13405/wss-v1.1-spec-pr-SAMLTokenProfile-01.pdf](http://www.oasis-
1184 open.org/committees/download.php/13405/wss-v1.1-spec-pr-SAMLTokenProfile-01.pdf) "Web Services
Security: SAML Token Profile 1.1," OASIS Public Review Draft 01,
- 1186 [wss-x509] Hallam-Baker, Phillip, Kaler, Chris, Monzillo, Ronald, Nadalin, Anthony, eds. (March, 2004). Organiza-
1187 tion for the Advancement of Structured Information Standards [http://docs.oasis-open.org/wss/2004/01/oasis-
1189 200401-wss-x509-token-profile-1.0.pdf](http://docs.oasis-open.org/wss/2004/01/oasis-
1188 200401-wss-x509-token-profile-1.0.pdf) "Web Services Security: X509 Certificate Token Profile," OASIS
Standard V1.0 [OASIS 200401],
- 1190 [XMLDsig] Eastlake, Donald, Reagle, Joseph, Solo, David, eds. (12 Feb 2002). "XML-Signature Syntax and
1191 Processing," Recommendation, World Wide Web Consortium <http://www.w3.org/TR/xmlsig-core>
- 1192 [xmlenc-core] Eastlake, Donald, Reagle, Joseph, eds. (10 December 2002). "XML Encryption Syntax and Process-
1193 ing," W3C Recommendation, World Wide Web Consortium <http://www.w3.org/TR/xmlenc-core/>
- 1194 [RFC3268] Chown, P., eds. (June 2002). "Advanced Encryption Standard (AES) Ciphersuites for Transport Layer
1195 Security (TLS)," RFC 3268., Internet Engineering Task Force <http://www.ietf.org/rfc/rfc3268.txt>