



ID-WSF 2.0 SecMech SAML Profile

Version: 2.0-errata-v1.0

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
Darryl Champagne, IEEE-ISTO
Gary Ellison, Sun Microsystems, Inc.
Jeff Hodges, Neustar
John Kemp, Nokia Corporation
John Linn, RSA Security Inc.
Rob Lockhart, IEEE-ISTO
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-2.0-errata-v1.0.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 © 2007 2FA Technology; Adobe Systems; Agencia Catalana De Certificacio; America Online, Inc.;
16 American Express Company; Amssoft Systems Pvt Ltd.; Avatier Corporation; BIPAC; BMC Software, Inc.; Bank of
17 America Corporation; Beta Systems Software AG; British Telecommunications plc; Computer Associates
18 International, Inc.; Credentica; DataPower Technology, Inc.; Deutsche Telekom AG, T-Com; Diamelle Technologies,
19 Inc.; Diversinet Corp.; Drummond Group Inc.; Enosis Group LLC; Entrust, Inc.; Epok, Inc.; Ericsson; Falkin
20 Systems LLC; Fidelity Investments; Forum Systems, Inc.; France Télécom; French Government Agence pour le
21 développement de l'administration électronique (ADAE); Fugen Solutions, Inc; Fulvens Ltd.; GSA Office of
22 Governmentwide Policy; Gamefederation; Gemalto; General Motors; GeoFederation; Giesecke & Devrient GmbH;
23 Hewlett-Packard Company; Hochhauser & Co., LLC; IBM Corporation; Intel Corporation; Intuit Inc.; Kantega;
24 Kayak Interactive; Livo Technologies; Luminance Consulting Services; MasterCard International; MedCommons
25 Inc.; Mobile Telephone Networks (Pty) Ltd; NEC Corporation; NTT DoCoMo, Inc.; Netegrity, Inc.; Neustar, Inc.;
26 New Zealand Government State Services Commission; Nippon Telegraph and Telephone Corporation; Nokia
27 Corporation; Novell, Inc.; OpenNetwork; Oracle Corporation; Ping Identity Corporation; RSA Security Inc.;
28 Reactivity Inc.; Royal Mail Group plc; SAP AG; Senforce; Sharp Laboratories of America; Sigaba; SmartTrust; Sony
29 Corporation; Sun Microsystems, Inc.; Supremacy Financial Corporation; Symlabs, Inc.; Telecom Italia S.p.A.;
30 Telefónica Móviles, S.A.; Telenor R&D; Thales e-Security; Trusted Network Technologies; UNINETT AS; UTI;
31 VeriSign, Inc.; Vodafone Group Plc.; Wave Systems Corp. All rights reserved.

32 Liberty Alliance Project
33 Licensing Administrator
34 c/o IEEE-ISTO
35 445 Hoes Lane
36 Piscataway, NJ 08855-1331, USA
37 info@projectliberty.org

Contents

38		
39	1. Introduction	4
40	2. Notation, Terminology, Namespaces and typographical conventions	5
41	3. Identifier Privacy Protection	6
42	3.1. Encrypted Name Identifiers	6
43	4. Authentication Mechanisms	7
44	4.1. SAML Assertion Message Authentication	7
45	4.1.1. Sender Processing Rules	7
46	4.1.2. Recipient Processing Rules	8
47	4.2. Bearer Token Authentication	8
48	4.2.1. Processing Rules	8
49	4.2.2. SAML Bearer Token Example	8
50	5. Message Authorization	12
51	5.1. Authorization Data Generation	12
52	5.1.1. Processing Rules	12
53	5.1.2. Consuming Authorization Data	13
54	6. Provider Chaining	14
55	6.1. Provider Chaining Example (Informative)	14
56	7. Identity Token	19
57	7.1. Identity Token Requirements	19
58	8. Examples (Informative)	20
59	8.1. Fragmentary Examples	20
60	8.1.1. Sender as Invocation Identity	20
61	8.1.2. Sender as Transited Provider Identity	20
62	8.1.3. Invoking Identity Authentication	21
63	8.1.4. Resource as an Attribute	22
64	8.2. Proxying with Authentication Context of the Invoking Identity	22
65	8.3. Conveyance of Sender as Invocation Identity	25
66	References	29

67 **1. Introduction**

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

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

73 **2. Notation, Terminology, Namespaces and typographical** 74 **conventions**

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

77 **3. Identifier Privacy Protection**

78 **3.1. Encrypted Name Identifiers**

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

4. Authentication Mechanisms

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

4.1. SAML Assertion Message Authentication

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

- *urn:liberty:security:2006-08:null:SAMLV2*
- *urn:liberty:security:2006-08:TLS:SAMLV2*
- *urn:liberty:security:2006-08:ClientTLS:SAMLV2*
- *urn:liberty:security:2006-08:ClientTLS:peerSAMLV2*

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

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

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

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

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

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

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

4.1.1. Sender Processing Rules

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

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

140 4.1.2. Recipient Processing Rules

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

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

4.2. Bearer Token Authentication

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

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

4.2.1. Processing Rules

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

4.2.2. SAML Bearer Token Example

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

```
<?xml version="1.0" encoding="UTF-8"?>
<s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:sb="urn:liberty:sb:2006-08"
  xmlns:pp="urn:liberty:id-sis-pp:2003-08"
  xmlns:sec="urn:liberty:security:2006-08"
  xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
  xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
  xmlns:wsa="http://www.w3.org/2005/08/addressing"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:xenc="http://www.w3.org/2001/04/xmenc#">
  <s:Header>
    <!-- see Liberty SOAP Binding Specification for which headers
      are required and optional -->
    <wsa:MessageID wsu:Id="mid">...</wsa:MessageID>
    <wsa:To wsu:Id="to">...</wsa:To>
    <wsa:Action wsu:Id="action">...</wsa:Action>
    <wsse:Security mustUnderstand="1">
      <wsu:Timestamp wsu:Id="ts">
        <wsu:Created>2005-06-17T04:49:17Z</wsu:Created >
      </wsu:Timestamp>
      <!-- this is the bearer token -->
      <saml2:Assertion
        xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion"
        Version="2.0"
        ID="sxJu9g/vvLG9sAN9bKp/8q0NKU="
        IssueInstant="2005-04-01T16:58:33.173Z">
        <saml2:Issuer>http://authority.example.com/</Saml2:Issuer>
        <!-- signature by the issuer over the assertion -->
        <ds:Signature>...</ds:Signature>
        <saml2:Subject>
          <saml2:EncryptedID>
            <xenc:EncryptedData>U2XTCNvRX7B1lNK182nmY00TEk==</xenc:EncryptedData>
            <xenc:EncryptedKey>...</xenc:EncryptedKey>
          </saml2:EncryptedID>
```

```
213
214     <saml2:SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm: bearer">
215     </saml2:SubjectConfirmation>
216 </saml2:Subject>
217
218 <!-- By placing an audience restriction on the assertion we
219      can limit the scope of which entity should consume
220      the information in the assertion. -->
221
222 <saml2:Conditions
223     NotBefore="2005-04-01T16:57:20Z"
224     NotOnOrAfter="2005-04-01T21:42:43Z">
225
226     <saml2:AudienceRestrictionCondition>
227     <saml2:Audience>http://wsp.example.com</saml2: Audience>
228     </saml2:AudienceRestrictionCondition>
229 </saml2:Conditions>
230
231 <!-- The AuthnStatement carries information
232      that describes the authentication event
233      of the Subject to an Authentication Authority -->
234 <saml2:AuthnStatement
235     AuthnInstant="2005-04-01T16:57:30.000Z"
236     SessionIndex="6345789">
237     <saml2:AuthnContext>
238     <saml2:AuthnContextClassRef
239     urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport
240     </saml2:AuthnContextClassRef>
241     </saml2:AuthnContext>
242 </saml2:AuthnStatement>
243
244 <!-- This AttributeStatement carries an EncryptedAttribute.
245      Once this element is decrypted with the supplied key
246      an <Attribute> element bearing an endpoint reference
247      can be found, specifying resources which the invoker may
248      access. Details on this element can be found in the
249      discovery service specification. -->
250
251 <saml2:AttributeStatement>
252     <saml2:EncryptedAttribute>
253     <xenc:EncryptedData Type="http://www.w3.org/2001/04/xmlenc#Element">
254     mQEMAzRniWkAAAEH9RWir0eKDkyFAB7PoFazx3ftp0vWwbbzqXdgcX8fpEqSrlv4
255     YqUc70MiJcBtKBp3+jlD4HPUaurIqHA0vrDmMpM+sF2BnpND118f/mXCv3XbWhiL
256     ...
257     hg6nZ5c0I6L6Gn9A
258     =HCQY
259     </xenc:EncryptedData>
260     <xenc:EncryptedKey> ... </xenc:EncryptedKey>
261     </saml2:EncryptedAttribute>
262 </saml2:AttributeStatement>
263
264 </saml2:Assertion>
265
266 <!-- This SecurityTokenReference is used to reference the SAML
267      Assertion from a ds:Reference -->
268
269 <wsse:SecurityTokenReference
270     xmlns:wsse="..." xmlns:wsu="..." xmlns:wssell="..."
271     wsu:Id="str1"
272     wssell:TokenType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-profi
273 le-1.1#SAMLV2.0">
274     <wsse:KeyIdentifier
275     ValueType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLID">
276     sxJu9g/vvLG9sAN9bKp/8q0NKU=
277     </wsse:KeyIdentifier>
278 </wsse:SecurityTokenReference>
279
```

```
280 <ds:Signature>
281 <ds:SignedInfo>
282 <!-- in general include a ds:Reference for each wsa: header
283 added according to SOAP binding -->
284
285 <!-- include the MessageID in the signature -->
286 <ds:Reference URI="#mid">...</ds:Reference>
287
288 <!-- include the To in the signature -->
289 <ds:Reference URI="#to">...</ds:Reference>
290
291 <!-- include the Action in the signature -->
292 <ds:Reference URI="#action">...</ds:Reference>
293
294 <!-- include the MessageID in the signature -->
295 <ds:Reference URI="#mid">...</ds:Reference>
296
297 <!-- include the Timestamp in the signature -->
298 <ds:Reference URI="#ts">...</ds:Reference>
299
300 <!-- include the SAML Assertion in the signature to avoid
301 token substitution attacks -->
302 <ds:Reference URI="#Str1">
303 <ds:Transform Algorithm="...#STR-Transform">
304 <wsse:TransformationParameters>
305 <ds:CanonicalizationMethod
306 Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315" />
307 </wsse:TransformationParameters>
308 </ds:Transform>
309 </ds:Reference>
310
311 <!-- include the message body -->
312 <ds:Reference URI="#MsgBody">
313 <!-- bind to the body -->
314 </ds:Reference>
315 </ds:SignedInfo>
316 ...
317
318 </ds:Signature>
319 </wsse:Security>
320 </s:Header>
321 <s:Body wsu:Id="MsgBody">
322 <pp:Modify>
323 <!-- this is an ID-SIS-PP Modify message -->
324 </pp:Modify>
325 </s:Body>
326 </s:Envelope>
327
328
```

5. Message Authorization

5.1. Authorization Data Generation

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

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

5.1.1. Processing Rules

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

- The assertion MUST indicate the invocation identity within the `<saml2:Subject>` element of the assertion. The `<saml2:Subject>` element MUST include at least one `<saml2:SubjectConfirmation>` element. This element MUST have a `Method` attribute with a value of `urn:oasis:names:tc:SAML2:2.0:cm:holder-of-key`. (This requirement enables a proof of possession and binding to the message on behalf of the invoker).

The subject confirmation element MUST be specified with a `<saml2:SubjectConfirmationData>` element qualified with an `xsi:type` of `saml2:KeyInfoConfirmationDataType` as specified in [SAMLCore2].

- When the invocation identity represents the identity of the sender, the `<saml2:Subject>` element is decorated as follows. Refer to [Section 8.1.1](#) for an informative example.

The name identifier element SHOULD include a `<saml2:NameID>` element and the `Format` attribute value SHOULD be `urn:oasis:names:tc:SAML2:2.0:nameid-format:entity`. Note: This identifier might assist the relying party in locating metadata concerning the subject of the assertion.

The `<saml2:SubjectConfirmation>` element SHOULD NOT be decorated with a `<saml2:NameID>` element. The reason is that the presence of the `<saml2:NameID>` is used to indicate that the sender is not the same as the invoker, but acting on behalf of the invoker.

- When the invocation identity is NOT that of the sender (i.e., the sender is acting on behalf of the subject) the `<saml2:Subject>` element is decorated as follows:

In an operational setting where the invocation identity (the subject) is only to be released to the relying party (the audience) then the name identifier element SHOULD be of type `<saml2:EncryptedID>` and conform to the guidance in [SAMLCore2]. Refer to [Section 8.1.2.2](#) for an informative example.

In settings where the invocation identity does not call for privacy protections then the name identifier element SHOULD be conveyed using a `<saml2:NameID>` element with a `Format` attribute which is appropriate for the operational setting. Refer to [Section 8.1.2.1](#) for an informative example.

To identify the confirming entity the `<saml2:SubjectConfirmation>` element SHOULD contain a `<saml2:NameID>` element with a `Format` attribute value of `urn:oasis:names:tc:SAML2:2.0:nameid-format:entity`. Note: This identifier might assist the relying party in locating metadata concerning the confirming entity as well as help associate the name of the confirming entity in the application domain namespace with the key used for subject confirmation.

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

386 5.1.2. Consuming Authorization Data

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

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

396 5.1.2.1. Processing Rules

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

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

6. Provider Chaining

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

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

- Issuer
- Signature of entire assertion
- Provider Chaining (if needed)
- Audience Restriction Condition
- Subject of assertion (with corresponding confirmation method information)
- AuthnStatement (convey information about authentication of the subject)
- Endpoint reference information

To convey the provider chaining information, the SAML assertion SHOULD include a `<saml2:Advice>` element containing a single `<TransitedProviderPath>` element. This `<TransitedProviderPath>` MUST contain a `<TransitedProvider>` element for each provider that has been transited. General use of the `<TransitedProviderPath>` element is defined in the Security Mechanisms core specification [[LibertySecMech20](#)].

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

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

6.1. Provider Chaining Example (Informative)

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

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

```
437 <?xml version="1.0" encoding="UTF-8"?>
438 <s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
439   xmlns:sb="urn:liberty:sb:2006-08"
440   xmlns:pp="urn:liberty:id-sis-pp:2003-08"
441   xmlns:sec="urn:liberty:security:2006-08"
442   xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
443   xmlns:wssell="http://docs.oasis-open.org/wss/2005/xx/oasis-2005xx-
444 wss-wssecurity-secext-1.1.xsd"
445   xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-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>U2XTCNvRX7B1lNK182nmY00TEk==</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             mQEMAzRniWkAAAEH9RWir0eKDkyFAB7PoFazx3ftp0vWwbbzqXdgcX8fpEqSrlv4
546             YqUc70MiJcBtKBp3+jlD4HPUaurIqHA0vrDmMpM+sF2BnpND118f/mX Cv3XbWhiL
547             xj1/M4y0CMAM/wBHT3xa17tWJwsZkDRLWxXP 7wSlTXNjCThHzBL8gBKZRqNBcZlU
548             ...
549             VRu9BpYBD4Y/98y1jtX9Pm898+xzketoc4ZvhCgh9P0arVK1B3cKxB87bKi DDWAU
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-t
565     oken-profile-1.1#SAMLV2.0">
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       HJJWbvqW9E84vJVQkjjLLA6nNvBX7mY00TzhwBdFNDElgsCSXZ5 Ekw==
615     </ds:SignatureValue>
616
617     <ds:KeyInfo>
618       <wsse:SecurityTokenReference
619         wss11:TokenType="http://docs.oasis-open.org/wss/oasis-wss-saml
620 -token-profile-1.1#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
650 `Condition` element.
- 651 4. When Holder-of-Key Subject Confirmation is used, SOAP Message Security for integrity **SHOULD** be used to
652 protect the identity token when conveyed in a SOAP message.
- 653 5. Identity assertions **SHOULD** indicate a limit to the lifetime of the assertion. With SAML 2.0 assertions, the
654 `Conditions` element `notOnOrAfter` attribute **SHOULD** be set to specify an expiration of the identity assertion.

8. Examples (Informative)

These examples demonstrate SAML 2.0 assertions.

8.1. Fragmentary Examples

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

8.1.1. Sender as Invocation Identity

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

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

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

8.1.2. Sender as Transited Provider Identity

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

8.1.2.1. Transparent Subject Identifier

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

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

```
704 009     </saml2:NameID>
705 010 <saml2:SubjectConfirmationData xsi:type="saml:KeyInfoConfirmationDataType">
706 011   <!-- This keyinfo is the key by which the sender (aka proxy) must
707 012     prove possession in order for the relying party to
708 013     accept the Statements in this assertion. -->
709 014   <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
710 015     <ds:KeyName>
711 016       CN=somemailhost.example.com,OU=SomePlace,O=ExampleOrg,...
712 017     </ds:KeyName>
713 018     <ds:KeyValue>...</ds:KeyValue>
714 019   </ds:KeyInfo>
715 020 </saml2:SubjectConfirmationData>
716 021 </saml2:SubjectConfirmation>
717 022 </saml2:Subject>
718
719
```

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

724 8.1.2.2. Opaque Subject Identifier

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

```
727 001 <saml2:Subject xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion">
728 002 <saml2:EncryptedID xmlns:xenc="http://www.w3.org/2001/04/xmlenc#">
729 003   <xenc:EncryptedData>U2XTCNvRX7B1lNK182nmY00TEk==</xenc:EncryptedData>
730 004   <xenc:EncryptedKey>...</xenc:EncryptedKey>
731 005 </saml2:EncryptedID>
732 006 <saml2:SubjectConfirmation
733 007   Method="urn:oasis:names:tc:SAML:2.0:cm:holder-of-key">
734 008   <saml2:NameID format="urn:oasis:names:tc:SAML:2.0:nameid-format:entity">
735 009     http://somemailhost.example.com/
736 010   </saml2:NameID>
737 011   <saml2:SubjectConfirmationData xsi:type="saml:KeyInfoConfirmationDataType">
738 012     <!-- This keyinfo is the key by which the sender (aka proxy) must
739 013     possession in order for the relying party to
740 014     the Statements in this assertion. -->
741 015     <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
742 016       <ds:KeyName>
743 017         CN=somemailhost.example.com,OU=SomePlace,O=ExampleOrg,...
744 018       </ds:KeyName>
745 019       <ds:KeyValue>...</ds:KeyValue>
746 020     </ds:KeyInfo>
747 021   </saml2:SubjectConfirmationData>
748 022 </saml2:SubjectConfirmation>
749 023 </saml2:Subject>
750
751
```

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

754 8.1.3. Invoking Identity Authentication

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

```
758 001 <!-- The saml2:AuthnStatement carries information that
759 002   describes the authentication event of the subject
```

```
760 003     to an authenticating authority -->
761 004 <saml2:AuthnStatement
762 005     xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion"
763 006     AuthnInstant="2005-04-01T16:57:30.000Z"
764 007     SessionIndex="6345789">
765 008 <saml2:AuthnContext>
766 009     <saml2:AuthnContextClassRef>
767 010         urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport
768 011     </saml2:AuthnContextClassRef>
769 012 </saml2:AuthnContext>
770 013 </saml2:AuthnStatement>
771
772
```

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

776 8.1.4. Resource as an Attribute

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

782 8.2. Proxying with Authentication Context of the Invoking Identity

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

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

789 The following example demonstrates:

- 790 • Proxying
- 791 • Encrypted Name Identifier
- 792 • Encrypted Endpoint Reference conveyed as Attribute

```
793 <?xml version="1.0" encoding="UTF-8"?>
794 <s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
795     xmlns:sb="urn:liberty:sb:2006-08"
796     xmlns:pp="urn:liberty:id-sis-pp:2003-08"
797     xmlns:sec="urn:liberty:security:2006-08"
798     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance#"
799     xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
800     xmlns:wssell="http://docs.oasis-open.org/wss/2005/xx/oasis-2005xx-
801 wss-wssecurity-secext-1.1.xsd"
802     xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd"
803     xmlns:wsa="http://www.w3.org/2005/08/addressing"
804     xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
805     xmlns:xenc="http://www.w3.org/2001/04/xmenc#">
806
807 <s:Header>
808
809 <!-- see Liberty SOAP Binding Specification for which headers
```

```
810         are required and optional -->
811
812     <wsa:MessageID wsu:Id="mid">...</wsa:MessageID>
813
814     <wsa:To wsu:Id="to">...</wsa:To>
815
816     <wsa:Action wsu:Id="action">...</wsa:Action>
817
818     <wsse:Security mustUnderstand="1">
819
820         <wsu:Timestamp wsu:Id="ts">
821             <wsu:Created>2005-06-17T04:49:17Z</wsu:Created >
822         </wsu:Timestamp>
823
824     <saml2:Assertion
825         xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion"
826         Version="2.0"
827         ID="sxJu9g/vvLG9sAN9bKp/8q0NKU="
828         IssueInstant="2005-04-01T16:58:33.173Z">
829
830         <saml2:Issuer>http://authority.example.com/</saml2:Issuer>
831
832         <!-- signature by the issuer over the assertion -->
833         <ds:Signature>...</ds:Signature>
834
835         <!-- By placing an audience restriction on the assertion we
836              can limit the scope of which entity should consume
837              the information in the assertion. -->
838
839         <saml2:Conditions
840             NotBefore="2005-04-01T16:57:20Z"
841             NotOnOrAfter="2005-04-01T21:42:43Z">
842
843             <saml2:AudienceRestrictionCondition>
844                 <saml2:Audience>http://wsp.example.com</saml2:Audience>
845             </saml2:AudienceRestrictionCondition>
846         </saml2:Conditions>
847
848         <saml2:Subject>
849             <saml2:EncryptedID>
850                 <xenc:EncryptedData>U2XTCNvRX7B1lNK182nmY00TEk==</xenc:EncryptedData>
851                 <xenc:EncryptedKey>...</xenc:EncryptedKey>
852             </saml2:EncryptedID>
853
854             <saml2:SubjectConfirmation
855                 Method="urn:oasis:names:tc:SAML:2.0:cm:holder-of-key">
856                 <saml2:NameID format="urn:oasis:names:tc:SAML:2.0:nameid-format:entity">
857                     http://wsc.example.com/
858                 </saml2:NameID>
859                 <saml2:SubjectConfirmationData xsi:type="saml2:KeyInfoConfirmationDataType">
860
861                     <!-- This keyinfo is the key by which the sender must
862                          prove possession in order for the relying party to
863                          accept the Statements in this assertion. -->
864                     <ds:KeyInfo>
865                         <ds:KeyName>
866                             CN=wsc.example.com,OU=Client Services R US,O=Service Station,...
867                         </ds:KeyName>
868                         <ds:KeyValue>...</ds:KeyValue>
869                     </ds:KeyInfo>
870                 </saml2:SubjectConfirmationData>
871             </saml2:SubjectConfirmation>
872         </saml2:Subject>
873
874         <!-- The AuthnStatement carries information
875              that describes the authentication event
876              of the Subject to an Authentication Authority -->
```

```
877     <saml2:AuthnStatement
878       AuthnInstant="2005-04-01T16:57:30.000Z"
879       SessionIndex="6345789">
880     <saml2:AuthnContext>
881       <saml2:AuthnContextClassRef>
882         urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport
883       </saml2:AuthnContextClassRef>
884     </saml2:AuthnContext>
885   </saml2:AuthnStatement>
886
887   <!-- The AttributeStatement carries an EncryptedAttribute.
888        Once this element is decrypted with the supplied key
889        an <Attribute> element bearing an endpoint reference
890        can be found. Details on this element can be found in the
891        discovery service specification. -->
892
893   <saml2:AttributeStatement>
894     <saml2:EncryptedAttribute>
895       <xenc:EncryptedData Type="http://www.w3.org/2001/04/xmenc#Element">
896         mQEMAzRniWkAAAEH9RWir0eKDkyFAB7PoFazx3ftp0vWwbbzqXdgcX8fpEqSrlv4
897         YqUc70MiJcBtKBp3+jlD4HPUaurIqHA0vrDmMpM+sF2BnpND118f/mXCv3XbWhiL
898         xjl/M4yOCMAM/wBHT3xa17tWJwsZkDRLWxXP7wSlTXNjCThHzBL8gBKZRqNBcZ1U
899         ...
900         VRu9BpYBD4Y/98y1jtX9Pm898+xzketoc4ZvhCgh9P0arVK1B3cKxB87bKiDDWAU
901         hg6nZ5c0I6L6Gn9A
902         =HCQY
903       </xenc:EncryptedData>
904       <xenc:EncryptedKey> ... </xenc:EncryptedKey>
905     </saml2:EncryptedAttribute>
906   </saml2:AttributeStatement>
907
908 </saml2:Assertion>
909
910 <!-- This SecurityTokenReference is used to reference the SAML
911 Assertion from a ds:Reference -->
912
913 <wsse:SecurityTokenReference
914   xmlns:wsse="..." xmlns:wsu="..." xmlns:wssell="..."
915   wsu:Id="str1"
916   wssell:TokenType="http://docs.oasis-open.org/wss/oasis-wss-sam
917 l-token-profile-1.1#SAMLV2.0">
918   <wsse:KeyIdentifier
919     ValueType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLID">
920     sxJu9g/vvLG9sAN9bKp/8q0NKU=
921   </wsse:KeyIdentifier>
922 </wsse:SecurityTokenReference>
923
924 <!-- this is the signature the sender generated to demonstrate
925 holder-of-key -->
926
927 <ds:Signature>
928   <ds:SignedInfo>
929     <!-- in general include a ds:Reference for each wsa: header
930          added according to SOAP binding -->
931
932     <!-- include the MessageID in the signature -->
933     <ds:Reference URI="#mid">...</ds:Reference>
934
935     <!-- include the To in the signature -->
936     <ds:Reference URI="#to">...</ds:Reference>
937
938     <!-- include the Action in the signature -->
939     <ds:Reference URI="#action">...</ds:Reference>
940
941     <!-- include the MessageID in the signature -->
942     <ds:Reference URI="#mid">...</ds:Reference>
943
```

```
944     <!-- include the Timestamp in the signature -->
945     <ds:Reference URI="#ts">...</ds:Reference>
946
947     <!-- include the SAML Assertion in the signature to avoid
948         token substitution attacks -->
949     <ds:Reference URI="#Str1">
950         <ds:Transform Algorithm="...#STR-Transform">
951             <wsse:TransformationParameters>
952                 <ds:CanonicalizationMethod
953                     Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315" />
954             </wsse:TransformationParameters>
955         </ds:Transform>
956     </ds:Reference>
957
958     <!-- include the message body -->
959     <ds:Reference URI="#MsgBody">
960         <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmlsig#sha1"/>
961         <ds:DigestValue>YgGfS0pi56pu...</ds:DigestValue>
962     </ds:Reference>
963 </ds:SignedInfo>
964
965 <ds:SignatureValue>
966     HJJWbvqW9E84vJVQkjJLLA6nNvBX7mY00TZhwBdFNDElgsC SXZ5Ekw==
967 </ds:SignatureValue>
968
969 <ds:KeyInfo>
970     <wsse:SecurityTokenReference
971         wsse11:TokenType="http://docs.oasis-open.org/wss/oasis-wss-s
972 aml-token-profile-1.1#SAMLV2.0">
973         <wsse:KeyIdentifier
974             ValueType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLID">
975             2sxJu9g/vvLG9sAN9bKp/8q0NKU=
976         </wsse:KeyIdentifier>
977         </wsse:SecurityTokenReference>
978     </ds:KeyInfo>
979 </ds:Signature>
980
981 </wsse:Security>
982
983 </s:Header>
984 <s:Body wsu:Id="MsgBody">
985     <pp:Modify>
986         <!-- this is an ID-SIS-PP Modify message -->
987     </pp:Modify>
988 </s:Body>
989 </s:Envelope>
990
```

8.3. Conveyance of Sender as Invocation Identity

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

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

The example demonstrates:

- Sender is Invocation Identity.

- 1001 • Endpoint Reference conveyed as attribute without encryption.

```
1002
1003 <?xml version="1.0" encoding="UTF-8"?>
1004 <s:Envelope xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
1005     xmlns:sb="urn:liberty:sb:2006-08"
1006     xmlns:pp="urn:liberty:id-sis-pp:2003-08"
1007     xmlns:sec="urn:liberty:security:2006-08"
1008     xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
1009     wss-wsse" xmlns:wss11="http://docs.oasis-open.org/wss/2005/xx/oasis-2005xx-wss-wssecurity-sec
1010     ext-1.1.xsd"
1011     xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
1012     wss-wssecurity-utility-1.0.xsd"
1013     xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
1014     xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
1015     xmlns:wsa="http://www.w3.org/2005/03/addressing">
1016
1017   <s:Header>
1018
1019     <!-- see Liberty SOAP Binding Specification for which headers
1020     are required and optional -->
1021
1022     <wsa:MessageID wsu:Id="mid">...</wsa:MessageID>
1023
1024     <wsa:To wsu:Id="to">...</wsa:To>
1025
1026     <wsa:Action wsu:Id="action">...</wsa:Action>
1027
1028     <wsse:Security mustUnderstand="1">
1029
1030       <wsu:Timestamp wsu:Id="ts">
1031         <wsu:Created>2005-06-17T04:49:17Z</wsu:Created >
1032         </wsu:Timestamp>
1033
1034       <saml2:Assertion
1035         xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion"
1036         Version="2.0"
1037         ID="sxJu9g/vvLG9sAN9bKp/8q0NKU="
1038         IssueInstant="2005-04-01T16:58:33.173Z">
1039
1040         <saml2:Issuer>http://authority.example.com/</saml2:Issuer>
1041
1042         <!-- signature by the issuer over the assertion -->
1043         <ds:Signature>...</ds:Signature>
1044
1045         <!-- By placing an audience restriction on the assertion we
1046         can limit the scope of which entity should consume
1047         the information in the assertion. -->
1048
1049         <saml2:Conditions
1050           NotBefore="2005-04-01T16:57:20Z"
1051           NotOnOrAfter="2005-04-01T21:42:43Z">
1052
1053           <saml2:AudienceRestrictionCondition>
1054             <saml2:Audience>http://wsp.example.com</saml2:Audience>
1055           </saml2:AudienceRestrictionCondition>
1056         </saml2:Conditions>
1057
1058         <saml2:Subject>
1059           <saml2:NameID format="urn:oasis:names:tc:SAML:2.0:nameid-format:entity">
1060             http://example.com/</saml2:NameID>
1061           <saml2:SubjectConfirmation
1062             Method="urn:oasis:names:tc:SAML:2.0:cm:holder-of-key">
1063             <saml2:SubjectConfirmationData xsi:type="saml2:KeyInfoConfirmationData Type">
1064               <!-- This keyinfo is the key by which the sender must
1065               prove possession in order for the relying party to
```

```

1067         accept the Statements in this assertion. -->
1068     <ds:KeyInfo>
1069         <ds:KeyName>
1070             CN=example.com,OU=SomeDivision,O=SomeOrganization,...
1071         </ds:KeyName>
1072         <ds:KeyValue>...</ds:KeyValue>
1073     </ds:KeyInfo>
1074     </saml2:SubjectConfirmationData>
1075 </saml2:SubjectConfirmation>
1076 </saml2:Subject>
1077
1078 <!-- For details on the contents of the Endpoint Reference see the
1079      discovery service specification which has details -->
1080 <saml2:AttributeStatement>
1081     <saml2:Attribute NameFormat="urn:liberty:disco:2005-06"
1082         Name="IDWSFEPR">
1083         <saml2:AttributeValue>
1084             <wsa:EndpointReference>
1085                 ...
1086             </wsa:EndpointReference>
1087         </saml2:AttributeValue>
1088     </saml2:Attribute>
1089 </saml2:AttributeStatement>
1090 </saml2:Assertion>
1091
1092 <!-- This SecurityTokenReference is used to reference the SAML
1093      Assertion from a ds:Reference -->
1094
1095 <wsse:SecurityTokenReference
1096     xmlns:wsse="..." xmlns:wsu="..." xmlns:wssell="..."
1097     wsu:Id="str1"
1098     wssell:TokenType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLV2
1099 .0">
1100     <wsse:KeyIdentifier
1101         ValueType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLID">
1102         sxJu9g/vvLG9sAN9bKp/8q0NKU=
1103     </wsse:KeyIdentifier>
1104 </wsse:SecurityTokenReference>
1105
1106 <!-- this is the signature the sender generated to demonstrate
1107      holder-of-key the signature should cover the isf header and body-->
1108
1109 <ds:Signature>
1110     <ds:SignedInfo>
1111         <!-- in general include a ds:Reference for each wsa: header
1112              added according to SOAP binding -->
1113
1114         <!-- include the MessageID in the signature -->
1115         <ds:Reference URI="#mid">...</ds:Reference>
1116
1117         <!-- include the To in the signature -->
1118         <ds:Reference URI="#to">...</ds:Reference>
1119
1120         <!-- include the Action in the signature -->
1121         <ds:Reference URI="#action">...</ds:Reference>
1122
1123         <!-- include the MessageID in the signature -->
1124         <ds:Reference URI="#mid">...</ds:Reference>
1125
1126         <!-- include the Timestamp in the signature -->
1127         <ds:Reference URI="#ts">...</ds:Reference>
1128
1129         <!-- include the SAML Assertion in the signature to avoid
1130              token substitution attacks -->
1131         <ds:Reference URI="#Str1">
1132             <ds:Transform Algorithm="...#STR-Transform">
1133                 <wsse:TransformationParameters>

```

```
1134         <ds:CanonicalizationMethod
1135             Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315" />
1136         </wsse:TransformationParameters>
1137     </ds:Transform>
1138 </ds:Reference>
1139
1140     <!-- include the message body -->
1141     <ds:Reference URI="#MsgBody">
1142         <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
1143         <ds:DigestValue>YgGfS0pi56pu...</ds:DigestValue>
1144     </ds:Reference>
1145 </ds:SignedInfo>
1146
1147 <ds:SignatureValue>
1148     HJJWbvqW9E84vJVQkjLLA6nNvBX7mY00TzhwBdFNDElgscSXZ5Ekw==
1149 </ds:SignatureValue>
1150
1151 <ds:KeyInfo>
1152 </ds:KeyInfo>
1153
1154 </ds:Signature>
1155 </wsse:Security>
1156 </s:Header>
1157 <s:Body wsu:Id="MsgBody">
1158     <pp:Modify>
1159         <!-- this is an ID-SIS-PP Modify message -->
1160     </pp:Modify>
1161 </s:Body>
1162 </s:Envelope>
1163
1164
```

1165 Details on the use of Endpoint References can be found in the discovery service specification.

References

Normative

- 1166 [LibertyDisco] Cahill, Conor, Hodges, Jeff, eds. "Liberty ID-WSF Discovery Service Specification," Version 2.0-errata-v1.0, Liberty Alliance Project (29 November, 2006). <http://www.projectliberty.org/specs>
- 1170 [LibertyIDWSFv20Errata] Champagne, Darryl, Lockhart, Rob, Tiffany, Eric, eds. "Liberty ID-WSF 2.0 Errata," Version 1.0, Liberty Alliance Project (13 April, 2007). <http://www.projectliberty.org/specs>
- 1172 [LibertySecMech20] Hirsch, Frederick, eds. "Liberty ID-WSF Security Mechanisms Core," Version 2.0-errata-v1.0, Liberty Alliance Project (21 April, 2007). <http://www.projectliberty.org/specs>
- 1174 [SAMLCore2] Cantor, Scott, Kemp, John, Philpott, Rob, Maler, Eve, eds. (15 March 2005). "Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) V2.0," SAML V2.0, OASIS Standard, Organization for the Advancement of Structured Information Standards <http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf>
- 1178 [SAMLBind2] Cantor, Scott, Hirsch, Frederick, Kemp, John, Philpott, Rob, Maler, Eve, eds. (15 March 2005). "Bindings for the OASIS Security Assertion Markup Language (SAML) V2.0," SAML V2.0, OASIS Standard, Organization for the Advancement of Structured Information Standards <http://docs.oasis-open.org/security/saml/v2.0/saml-bindings-2.0-os.pdf>
- 1182 [wss-sms11] Hallam-Baker, Phillip, Kaler, Chris, Monzillo, Ronald, Nadalin, Anthony, eds. (June 28, 2005). "Web Services Security: SOAP Message Security 1.1 (WS-Security 2004)," Public Review Draft - 28 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>
- 1186 [wss-saml11] Monzillo, Ronald, Kaler, Chris, Nadalin, Anthony, Hallam-Baker, Phillip, eds. (June 28, 2005). Organization for the Advancement of Structured Information Standards <http://www.oasis-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,
- 1190 [wss-x509] Hallam-Baker, Phillip, Kaler, Chris, Monzillo, Ronald, Nadalin, Anthony, eds. (March, 2004). Organization for the Advancement of Structured Information Standards <http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-x509-token-profile-1.0.pdf> "Web Services Security: X509 Certificate Token Profile," OASIS Standard V1.0 [OASIS 200401],
- 1194 [XMLDsig] Eastlake, Donald, Reagle, Joseph, Solo, David, eds. (12 Feb 2002). "XML-Signature Syntax and Processing," Recommendation, World Wide Web Consortium <http://www.w3.org/TR/xmlsig-core>
- 1196 [xmlenc-core] Eastlake, Donald, Reagle, Joseph, eds. (10 December 2002). "XML Encryption Syntax and Processing," W3C Recommendation, World Wide Web Consortium <http://www.w3.org/TR/xmlenc-core/>
- 1198 [RFC3268] Chown, P., eds. (June 2002). "Advanced Encryption Standard (AES) Ciphersuites for Transport Layer Security (TLS)," RFC 3268., Internet Engineering Task Force <http://www.ietf.org/rfc/rfc3268.txt>