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

Notice

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

Contents

1. Introduction . . . 4
2. Notation, Terminology, Namespaces and typographical conventions . . . 5
3. Identifier Privacy Protection . . . 6
    3.1. Encrypted Name Identifiers . . . 6
4. Authentication Mechanisms . . . 7
    4.1. SAML Assertion Message Authentication . . . 7
        4.1.1. Sender Processing Rules . . . 7
        4.1.2. Recipient Processing Rules . . . 8
    4.2. Bearer Token Authentication . . . 8
        4.2.1. Processing Rules . . . 8
        4.2.2. SAML Bearer Token Example . . . 8
5. Message Authorization . . . 12
    5.1. Authorization Data Generation . . . 12
        5.1.1. Processing Rules . . . 12
        5.1.2. Consuming Authorization Data . . . 13
6. Provider Chaining . . . 14
    6.1. Provider Chaining Example (Informative) . . . 14
7. Identity Token . . . 19
    7.1. Identity Token Requirements . . . 19
8. Examples (Informative) . . . 20
    8.1. Fragmentary Examples . . . 20
        8.1.1. Sender as Invocation Identity . . . 20
        8.1.2. Sender as Transited Provider Identity . . . 20
        8.1.3. Invoking Identity Authentication . . . 21
        8.1.4. Resource as an Attribute . . . 22
    8.2. Proxying with Authentication Context of the Invoking Identity . . . 22
    8.3. Conveyance of Sender as Invocation Identity . . . 25
References . . . 29

1. Introduction

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

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

2. Notation, Terminology, Namespaces and typographical conventions

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

3. Identifier Privacy Protection

3.1. Encrypted Name Identifiers

To securely protect the privacy of the identifier as the message passes through intermediaries, the `<saml2:Subject>` MUST contain a `<saml2:EncryptedID>` where a privacy risk due to provider collaboration based on identity is a concern. In general the `<saml2:Subject>` SHOULD contain a `<saml2:EncryptedID>`. Use of `<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:SAML2:2.0: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:

- The construction and decoration of the `<wsse:Security>` header element MUST adhere to the rules specified in the [wss-sms11] and [wss-saml11].

- The sender MUST present the `<saml2:Assertion>` (as security token) by inserting it as a child of the `<wsse:Security>` element.

- The sender MUST adhere to its subject confirmation obligation in accordance with the semantics of the confirmation method. This is described by one of the `<saml2:SubjectConfirmation>` elements carried within the `<saml2:Subject>`

  For deployment settings which REQUIRE independent message authentication, the obligation MUST be accomplished by signing elements of the message and decorating the `<wsse:Security>` element with the signature.

  For deployment settings which DO NOT REQUIRE independent message authentication then the subject confirmation obligation may be accomplished by correlating the certificate and key used to affect peer entity authentication with the certificate and key described by the subject confirmation element. To accommodate this, the assertion issuing authority MUST construct the assertion such that the confirmation key can be unambiguously verified to be the same certificate and key used in establishing peer entity authentication. This is necessary to mitigate the threat of a certificate substitution attack. It is RECOMMENDED that the certificate or certificate chain be bound to the subject confirmation key.

4.1.2. Recipient Processing Rules

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

- The recipient MUST locate the `<saml2:Assertion>` (security token) and the recipient MUST determine that it trusts the authority which issued the `<saml2:Assertion>`.

- The recipient MUST validate the issuer's signature over the `<saml2:Assertion>`. The recipient SHOULD validate the trust semantics of the signing key, as appropriate to the risk of incorrect authentication.

- The recipient SHOULD verify that at least one of the confirmation obligations specified in the `<saml2:SubjectConfirmation>` element has been met.

- If the validation policy regards peer entity authentication sufficient for purposes of message authentication then the recipient MUST locate the `<ds:KeyInfo>` element within `<saml2:SubjectConfirmation>` element. This key MUST be unambiguously verified to be referring to the same certificate and key used in establishing peer entity authentication.

- The recipient MUST determine that it trusts the key used to sign the message.

- When an OASIS X.509 token is used to convey key information, the recipient SHOULD validate the sender's 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/xmlenc#">

 <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>U2XTCNvRX7Bl1NK182nmY00TEk==</xenc:EncryptedData>
      <xenc:EncryptedKey>...</xenc:EncryptedKey>
     </saml2:EncryptedID>
```

```

        <saml2:SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
        </saml2:SubjectConfirmation>
      </saml2:Subject>

      <!-- By placing an audience restriction on the assertion we
           can limit the scope of which entity should consume
           the information in the assertion. -->

      <saml2:Conditions
          NotBefore="2005-04-01T16:57:20Z"
          NotOnOrAfter="2005-04-01T21:42:43Z">

        <saml2:AudienceRestrictionCondition>
          <saml2:Audience>http://wsp.example.com</saml2:Audience>
        </saml2:AudienceRestrictionCondition>
      </saml2:Conditions>

      <!-- The AuthnStatement carries information
           that describes the authentication event
           of the Subject to an Authentication Authority -->
      <saml2:AuthnStatement
          AuthnInstant="2005-04-01T16:57:30.000Z"
          SessionIndex="6345789">
        <saml2:AuthnContext>
          <saml2:AuthnContextClassRef>
            urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport
          </saml2:AuthnContextClassRef>
        </saml2:AuthnContext>
      </saml2:AuthnStatement>

      <!-- This AttributeStatement carries an EncryptedAttribute.
           Once this element is decrypted with the supplied key
           an <Attribute> element bearing an endpoint reference
           can be found, specifying resources which the invoker may
           access.  Details on this element can be found in the
           discovery service specification. -->

      <saml2:AttributeStatement>
        <saml2:EncryptedAttribute>
          <xenc:EncryptedData Type="http://www.w3.org/2001/04/xmlenc#Element">
            mQEMAzRniWkAAAEH9RWir0eKDkyFAB7PoFazx3ftp0vWwbbzqXdgcX8fpEqSr1v4
            YqUc7OMiJcBtKBp3+jlD4HPUaurIqHA0vrdmMpM+sF2BnpND118f/mXCv3XbWhiL
            ...
            hg6nZ5c0I6L6Gn9A
            =HCQY
          </xenc:EncryptedData>
          <xenc:EncryptedKey> ... </xenc:EncryptedKey>
        </saml2:EncryptedAttribute>
      </saml2:AttributeStatement>

    </saml2:Assertion>

    <!-- This SecurityTokenReference is used to reference the SAML
    Assertion from a ds:Reference -->

    <wsse:SecurityTokenReference
       xmlns:wsse="..." xmlns:wsu="..." xmlns:wsse11="..."
       wsu:Id="str1"
       wsse11:TokenType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-profi
le-1.1#SAMLV2.0">
      <wsse:KeyIdentifier
         ValueType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLID">
        sxJu9g/vvLG9sAN9bKp/8q0NKU=
      </wsse:KeyIdentifier>
    </wsse:SecurityTokenReference>

```

```
      <ds:Signature>
        <ds:SignedInfo>
          <!-- in general include a ds:Reference for each wsa: header
               added according to SOAP binding -->

          <!-- include the MessageID in the signature -->
          <ds:Reference URI="#mid">...</ds:Reference>

          <!-- include the To in the signature -->
          <ds:Reference URI="#to">...</ds:Reference>

          <!-- include the Action in the signature -->
          <ds:Reference URI="#action">...</ds:Reference>

          <!-- include the MessageID in the signature -->
          <ds:Reference URI="#mid">...</ds:Reference>

          <!-- include the Timestamp in the signature -->
          <ds:Reference URI="#ts">...</ds:Reference>

          <!-- include the SAML Assertion in the signature to avoid
               token substitution attacks -->
          <ds:Reference URI="#Str1">
            <ds:Transform Algorithm="...#STR-Transform">
              <wsse:TransformationParameters>
                <ds:CanonicalizationMethod
                   Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n- 20010315" />
              </wsse:TransformationParameters>
            </ds:Transform>
          </ds:Reference>

          <!-- include the message body -->
          <ds:Reference URI="#MsgBody">
            <!-- bind to the body -->
          </ds:Reference>
        </ds:SignedInfo>
        ...

      </ds:Signature>
    </wsse:Security>
  </s:Header>
  <s:Body wsu:Id="MsgBody">
    <pp:Modify>
      <!-- this is an ID-SIS-PP Modify message -->
    </pp:Modify>
  </s:Body>
</s:Envelope>
```

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.

- The assertion issuing authority MAY describe the authentication status of the interacting party by including a `<saml2:AuthnStatement>` element which MUST include a `<saml2:AuthnContext>` element. Refer to Section 8.1.3 for an informative example.

- The assertion issuing authority MAY limit the resource which the invoker may access at the relying party by describing the relevant resources in the `<saml2:AttributeStatement>`. This may be done by explicitly listing endpoint references of the resources that the invoker may access.

  In an operational setting where the value of the attribute requires confidentiality protections then the attribute element SHOULD be of type `<saml2:EncryptedAttribute>` and conform to the guidance in [SAMLCore2].

  If the confidentiality of the attribute is not a concern then the element SHOULD be conveyed using a `<saml2:Attribute>`.

- OPTIONALLY, the assertion issuer MAY include information that assists in building a chain of transited providers. How this is done is defined in the Provider Chaining section (Section 6).

- The assertion MUST be signed by the assertion issuing authority in accordance with the signing requirements specified in [SAMLCore2].

5.1.2. Consuming Authorization Data

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

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

5.1.2.1. Processing Rules

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

- The recipient MUST locate the `<saml2:Assertion>` (security token) which conferred the subject confirmation key relied upon for sender authentication.

  The recipient MUST corroborate that the bound subject confirmation key is the same key used to authenticate the communicating peer.

- The recipient MUST determine that it trusts the authority which signed the `<saml2:Assertion>`.

  The recipient MUST validate the signature of the `<saml2:Assertion>`. The recipient SHOULD validate the 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.

```
 <?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:wsse11="http://docs.oasis-open.org/wss/2005/xx/oasis-2005xx-
wss-wssecurity-secext-1.1.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/xmlenc#">

 <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>

    <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:Advice>
       <sec:TransitedProviderPath>
         <TransitedProvider>http://www.example.com/one</TransitedProvider>
         <TransitedProvider>http://www.example.com/two</TransitedProvider>
       </sec:TransitedProviderPath>
      </saml2:Advice>

      <!-- By placing an audience restriction on the assertion we
          can limit the scope of which entity should consume
          the information in the assertion. -->

      <saml2:Conditions
         NotBefore="2005-04-01T16:57:20Z"
         NotOnOrAfter="2005-04-01T21:42:43Z">

       <saml2:AudienceRestrictionCondition>
         <saml2:Audience>http://wsp.example.com</saml2:Audience>
       </saml2:AudienceRestrictionCondition>
      </saml2:Conditions>

      <saml2:Subject>
       <saml2:EncryptedID>
         <xenc:EncryptedData>U2XTCNvRX7Bl1NK182nmY00TEk==</xenc:EncryptedData>
         <xenc:EncryptedKey>...</xenc:EncryptedKey>
       </saml2:EncryptedID>

       <saml2:SubjectConfirmation
```

```
            Method="urn:oasis:names:tc:SAML:2.0:cm:holder-of-key">
          <saml2:NameID format="urn:oasis:names:tc:SAML:2.0:nameid-format:entity">
            http://third.example.com/
          </saml2:NameID>
          <saml2:SubjectConfirmationData xsi:type="saml2:KeyInfoConfirmationDataType">

            <!-- This keyinfo is the key by which the sender must
                 prove possession in order for the relying party to
                 accept the Statements in this assertion.  -->
            <ds:KeyInfo>
              <ds:KeyName>
                CN=third.example.com,OU=Client Services R US,O=Service Station,...
              </ds:KeyName>
              <ds:KeyValue>...</ds:KeyValue>
            </ds:KeyInfo>
          </saml2:SubjectConfirmationData>
        </saml2:SubjectConfirmation>
      </saml2:Subject>

      <!-- The AuthnStatement carries information
           that describes the authentication event
           of the Subject to an Authentication Authority -->
      <saml2:AuthnStatement
          AuthnInstant="2005-04-01T16:57:30.000Z"
          SessionIndex="6345789">
        <saml2:AuthnContext>
          <saml2:AuthnContextClassRef>
            urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport
          </saml2:AuthnContextClassRef>
        </saml2:AuthnContext>
      </saml2:AuthnStatement>

      <!-- The AttributeStatement carries an EncryptedAttribute.
           Once this element is decrypted with the supplied key
           an <Attribute> element bearing an endpoint reference
           can be found. Details on this element can be found in the
           discovery service specification. -->

      <saml2:AttributeStatement>
        <saml2:EncryptedAttribute>
          <xenc:EncryptedData Type="http://www.w3.org/2001/04/xmlenc#Element">
            mQEMAzRniWkAAAEH9RWir0eKDkyFAB7PoFazx3ftp0vWwbbzqXdgcX8fpEqSr1v4
            YqUc7OMiJcBtKBp3+jlD4HPUaurIqHA0vrdmMpM+sF2BnpND118f/mXCv3XbWhiL
            xj1/M4y0CMAM/wBHT3xa17tWJwsZkDRLWxXP7wSlTXNjCThHzBL8gBKZRqNBcZlU
            ...
            VRu9BpYBD4Y/98y1jtX9Pm898+xzketoc4ZvhCgh9P0arVK1B3cKxB87bKiDDWAU
            hg6nZ5c0I6L6Gn9A
            =HCQY
          </xenc:EncryptedData>
          <xenc:EncryptedKey> ... </xenc:EncryptedKey>
        </saml2:EncryptedAttribute>
      </saml2:AttributeStatement>
    </saml2:Assertion>

    <!-- This SecurityTokenReference is used to reference the SAML
    Assertion from a ds:Reference -->

    <wsse:SecurityTokenReference
       xmlns:wsse="..." xmlns:wsu="..." xmlns:wsse11="..."
       wsu:Id="str1"
       wsse11:TokenType="http://docs.oasis-open.org/wss/oasis-wss-saml-t
oken-profile-1.1#SAMLV2.0">
      <wsse:KeyIdentifier
         ValueType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLID">
        sxJu9g/vvLG9sAN9bKp/8q0NKU=
      </wsse:KeyIdentifier>
    </wsse:SecurityTokenReference>
```

```

      <!-- this is the signature the sender generated to demonstrate
      holder-of-key -->

      <ds:Signature>
        <ds:SignedInfo>
          <!-- in general include a ds:Reference for each wsa: header
               added according to SOAP binding -->

          <!-- include the MessageID in the signature -->
          <ds:Reference URI="#mid">...</ds:Reference>

          <!-- include the To in the signature -->
          <ds:Reference URI="#to">...</ds:Reference>

          <!-- include the Action in the signature -->
          <ds:Reference URI="#action">...</ds:Reference>

          <!-- include the MessageID in the signature -->
          <ds:Reference URI="#mid">...</ds:Reference>

          <!-- include the Timestamp in the signature -->
          <ds:Reference URI="#ts">...</ds:Reference>

          <!-- include the SAML Assertion in the signature to avoid
               token substitution attacks -->
          <ds:Reference URI="#Str1">
            <ds:Transform Algorithm="...#STR-Transform">
              <wsse:TransformationParameters>
                <ds:CanonicalizationMethod
                    Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315" />
              </wsse:TransformationParameters>
            </ds:Transform>
          </ds:Reference>

          <!-- include the message body -->
          <ds:Reference URI="#MsgBody">
            <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
            <ds:DigestValue>YgGfS0pi56pu...</ds:DigestValue>
          </ds:Reference>
        </ds:SignedInfo>

        <ds:SignatureValue>
          HJJWbvqW9E84vJVQkjjLLA6nNvBX7mY00TZhwBdFNDElgscSXZ5Ekw==
        </ds:SignatureValue>

        <ds:KeyInfo>
          <wsse:SecurityTokenReference
              wsse11:TokenType="http://docs.oasis-open.org/wss/oasis-wss-saml
-token-profile-1.1#SAMLV2.0">
            <wsse:KeyIdentifier
                ValueType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLID">
              2sxJu9g/vvLG9sAN9bKp/8q0NKU=
            </wsse:KeyIdentifier>
          </wsse:SecurityTokenReference>
        </ds:KeyInfo>

      </ds:Signature>
    </wsse:Security>

  </s:Header>
  <s:Body id="MsgBody">
    <pp:Modify>
      <!-- this is an ID-SIS-PP Modify message -->
    </pp:Modify>
  </s:Body>
</s:Envelope>
```

7. Identity Token

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

7.1. Identity Token Requirements

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

1. The subject of the identity token MUST represent the identity to be associated with the token.
2. The identity token SHOULD contain an attribute containing the endpoint reference for the Discovery Service associated with the subject identity. The bootstrap attribute is defined in the ID-WSF 2.0 Discovery Service Specification [LibertyDisco].
3. The Identity token SHOULD have an `AudienceRestrictionCondition` as part of the SAML assertion `Condition` element.
4. When Holder-of-Key Subject Confirmation is used, SOAP Message Security for integrity SHOULD be used to protect the identity token when conveyed in a SOAP message.
5. Identity assertions SHOULD indicate a limit to the lifetime of the assertion. With SAML 2.0 assertions, the 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://somemailhost.example.com/
```

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

In the above example the noteworthy elements are described. Lines 002-004 describe the identity of the subject, aka the invocation identity. Lines 005-020 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 010-020. Lines 007-009 identify the name of the proxy.

8.1.2.2. Opaque Subject Identifier

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

```
 001 <saml2:Subject xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion">
002   <saml2:EncryptedID xmlns:xenc="http://www.w3.org/2001/04/xmlenc#">
003     <xenc:EncryptedData>U2XTCNvRX7Bl1NK182nmY00TEk==</xenc:EncryptedData>
004     <xenc:EncryptedKey>...</xenc:EncryptedKey>
005   </saml2:EncryptedID>
006   <saml2:SubjectConfirmation
007       Method="urn:oasis:names:tc:SAML:2.0:cm:holder-of-key">
008     <saml2:NameID format="urn:oasis:names:tc:SAML:2.0:nameid-format:entity">
009       http://somemailhost.example.com/
010     </saml2:NameID>
011     <saml2:SubjectConfirmationData xsi:type="saml2:KeyInfoConfirmationDataType">
012       <!-- This keyinfo is the key by which the sender (aka proxy) must
013            possession in order for the relying party to
014            the Statements in this assertion.  -->
015       <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
016         <ds:KeyName>
017           CN=somemailhost.example.com,OU=SomePlace,O=ExampleOrg,...
018         </ds:KeyName>
019         <ds:KeyValue>...</ds:KeyValue>
020       </ds:KeyInfo>
021     </saml2:SubjectConfirmationData>
022   </saml2:SubjectConfirmation>
023 </saml2:Subject>
```

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

8.1.3. Invoking Identity Authentication

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

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

```
003       to an authenticating authority -->
004   <saml2:AuthnStatement
005      xmlns:saml2="urn:oasis:names:tc:SAML:2.0:assertion"
006      AuthnInstant="2005-04-01T16:57:30.000Z"
007      SessionIndex="6345789">
008     <saml2:AuthnContext>
009      <saml2:AuthnContextClassRef>
010       urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport
011      </saml2:AuthnContextClassRef>
012     </saml2:AuthnContext>
013   </saml2:AuthnStatement>
```

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

8.1.4. Resource as an Attribute

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

8.2. Proxying with Authentication Context of the Invoking Identity

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

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

The following example demonstrates:

- Proxying
- Encrypted Name Identifier
- Encrypted Endpoint Reference conveyed as Attribute

```
<?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:xsi="http://www.w3.org/2001/XMLSchema-instance#"
   xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd"
   xmlns:wsse11="http://docs.oasis-open.org/wss/2005/xx/oasis-2005xx-
wss-wssecurity-secext-1.1.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/xmlenc#">

  <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>

  <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>

    <!-- By placing an audience restriction on the assertion we
         can limit the scope of which entity should consume
         the information in the assertion. -->

    <saml2:Conditions
        NotBefore="2005-04-01T16:57:20Z"
        NotOnOrAfter="2005-04-01T21:42:43Z">

      <saml2:AudienceRestrictionCondition>
        <saml2:Audience>http://wsp.example.com</saml2:Audience>
      </saml2:AudienceRestrictionCondition>
    </saml2:Conditions>

    <saml2:Subject>
      <saml2:EncryptedID>
        <xenc:EncryptedData>U2XTCNvRX7Bl1NK182nmY00TEk==</xenc:EncryptedData>
        <xenc:EncryptedKey>...</xenc:EncryptedKey>
      </saml2:EncryptedID>

      <saml2:SubjectConfirmation
          Method="urn:oasis:names:tc:SAML:2.0:cm:holder-of-key">
        <saml2:NameID format="urn:oasis:names:tc:SAML:2.0:nameid-format:entity">
          http://wsc.example.com/
        </saml2:NameID>
        <saml2:SubjectConfirmationData xsi:type="saml2:KeyInfoConfirmationDataType">

          <!-- This keyinfo is the key by which the sender must
               prove possession in order for the relying party to
               accept the Statements in this assertion.  -->
          <ds:KeyInfo>
            <ds:KeyName>
              CN=wsc.example.com,OU=Client Services R US,O=Service Station,...
            </ds:KeyName>
            <ds:KeyValue>...</ds:KeyValue>
          </ds:KeyInfo>
        </saml2:SubjectConfirmationData>
      </saml2:SubjectConfirmation>
    </saml2:Subject>

    <!-- The AuthnStatement carries information
         that describes the authentication event
         of the Subject to an Authentication Authority -->
      <saml2:AuthnStatement
         AuthnInstant="2005-04-01T16:57:30.000Z"
         SessionIndex="6345789">
        <saml2:AuthnContext>
          <saml2:AuthnContextClassRef>
            urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport
          </saml2:AuthnContextClassRef>
        </saml2:AuthnContext>
      </saml2:AuthnStatement>

      <!-- The AttributeStatement carries an EncrpytedAttribute.
           Once this element is decrypted with the supplied key
           an <Attribute> element bearing an endpoint reference
           can be found. Details on this element can be found in the
           discovery service specification. -->

      <saml2:AttributeStatement>
        <saml2:EncryptedAttribute>
          <xenc:EncryptedData Type="http://www.w3.org/2001/04/xmlenc#Element">
            mQEMAzRniWkAAAEH9RWir0eKDkyFAB7PoFazx3ftp0vWwbbzqXdgcX8fpEqSr1v4
            YqUc7OMiJcBtKBp3+jlD4HPUaurIqHA0vrdmMpM+sF2BnpND118f/mXCv3XbWhiL
            xj1/M4y0CMAM/wBHT3xa17tWJwsZkDRLWxXP7wSlTXNjCThHzBL8gBKZRqNBcZlU
            ...
            VRu9BpYBD4Y/98y1jtX9Pm898+xzketoc4ZvhCgh9P0arVK1B3cKxB87bKiDDWAU
            hg6nZ5c0I6L6Gn9A
            =HCQY
          </xenc:EncryptedData>
          <xenc:EncryptedKey> ... </xenc:EncryptedKey>
        </saml2:EncryptedAttribute>
      </saml2:AttributeStatement>

    </saml2:Assertion>

    <!-- This SecurityTokenReference is used to reference the SAML
    Assertion from a ds:Reference -->

    <wsse:SecurityTokenReference
       xmlns:wsse="..." xmlns:wsu="..." xmlns:wsse11="..."
       wsu:Id="str1"
       wsse11:TokenType="http://docs.oasis-open.org/wss/oasis-wss-sam
l-token-profile-1.1#SAMLV2.0">
      <wsse:KeyIdentifier
         ValueType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLID">
        sxJu9g/vvLG9sAN9bKp/8q0NKU=
      </wsse:KeyIdentifier>
    </wsse:SecurityTokenReference>

    <!-- this is the signature the sender generated to demonstrate
    holder-of-key -->

    <ds:Signature>
      <ds:SignedInfo>
        <!-- in general include a ds:Reference for each wsa: header
             added according to SOAP binding -->

        <!-- include the MessageID in the signature -->
        <ds:Reference URI="#mid">...</ds:Reference>

        <!-- include the To in the signature -->
        <ds:Reference URI="#to">...</ds:Reference>

        <!-- include the Action in the signature -->
        <ds:Reference URI="#action">...</ds:Reference>

        <!-- include the MessageID in the signature -->
        <ds:Reference URI="#mid">...</ds:Reference>

```

```
     <!-- include the Timestamp in the signature -->
     <ds:Reference URI="#ts">...</ds:Reference>

     <!-- include the SAML Assertion in the signature to avoid
        token substitution attacks -->
     <ds:Reference URI="#Str1">
      <ds:Transform Algorithm="...#STR-Transform">
       <wsse:TransformationParameters>
        <ds:CanonicalizationMethod
          Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315" />
       </wsse:TransformationParameters>
      </ds:Transform>
     </ds:Reference>

     <!-- include the message body -->
     <ds:Reference URI="#MsgBody">
      <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
      <ds:DigestValue>YgGfS0pi56pu...</ds:DigestValue>
     </ds:Reference>
    </ds:SignedInfo>

    <ds:SignatureValue>
     HJJWbvqW9E84vJVQkjjLLA6nNvBX7mY00TZhwBdFNDElgscSXZ5Ekw==
    </ds:SignatureValue>

    <ds:KeyInfo>
     <wsse:SecurityTokenReference
       wsse11:TokenType="http://docs.oasis-open.org/wss/oasis-wss-s
aml-token-profile-1.1#SAMLV2.0">
      <wsse:KeyIdentifier
        ValueType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLID">
       2sxJu9g/vvLG9sAN9bKp/8q0NKU=
      </wsse:KeyIdentifier>
     </wsse:SecurityTokenReference>
    </ds:KeyInfo>
   </ds:Signature>

  </wsse:Security>

 </s:Header>
 <s:Body wsu:Id="MsgBody">
  <pp:Modify>
   <!-- this is an ID-SIS-PP Modify message -->
  </pp:Modify>
 </s:Body>
</s:Envelope>
```

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.

- Endpoint Reference conveyed as attribute without encryption.

```
 <?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-wssec
urity-secext-1.0.xsd"
        xmlns:wsse11="http://docs.oasis-open.org/wss/2005/xx/oasis-2005xx-wss-wssecurity-sec
ext-1.1.xsd"
        xmlns:wsu="http://docs.oasis-open.org/wss/2004/01/oasis-200401-
wss-wssecurity-utility-1.0.xsd"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance#"
        xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
        xmlns:wsa="http://www.w3.org/2005/03/addressing">

 <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>

    <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>

      <!-- By placing an audience restriction on the assertion we
         can limit the scope of which entity should consume
         the information in the assertion. -->

      <saml2:Conditions
        NotBefore="2005-04-01T16:57:20Z"
        NotOnOrAfter="2005-04-01T21:42:43Z">

       <saml2:AudienceRestrictionCondition>
        <saml2:Audience>http://wsp.example.com</saml2:Audience>
       </saml2:AudienceRestrictionCondition>
      </saml2:Conditions>

      <saml2:Subject>
       <saml2:NameID format="urn:oasis:names:tc:SAML:2.0:nameid-format:entity">
       http://example.com/</saml2:NameID>
       <saml2:SubjectConfirmation
          Method="urn:oasis:names:tc:SAML:2.0:cm:holder-of-key">
        <saml2:SubjectConfirmationData xsi:type="saml2:KeyInfoConfirmationDataType">
          <!-- This keyinfo is the key by which the sender must
             prove possession in order for the relying party to
```

```
                 accept the Statements in this assertion.  -->
            <ds:KeyInfo>
              <ds:KeyName>
                CN=example.com,OU=SomeDivision,O=SomeOrganization,...
              </ds:KeyName>
              <ds:KeyValue>...</ds:KeyValue>
            </ds:KeyInfo>
          </saml2:SubjectConfirmationData>
        </saml2:SubjectConfirmation>
      </saml2:Subject>

      <!-- For details on the contents of the Endpoint Reference see the
           discovery service specification which has details -->
      <saml2:AttributeStatement>
        <saml2:Attribute NameFormat="urn:liberty:disco:2005-06"
                         Name="IDWSFEPR">
          <saml2:AttributeValue>
            <wsa:EndpointReference>
              ...
            </wsa:EndpointReference>
          </saml2:AttributeValue>
        </saml2:Attribute>
      </saml2:AttributeStatement>
    </saml2:Assertion>

    <!-- This SecurityTokenReference is used to reference the SAML
    Assertion from a ds:Reference -->

    <wsse:SecurityTokenReference
       xmlns:wsse="..." xmlns:wsu="..." xmlns:wsse11="..."
       wsu:Id="str1"
       wsse11:TokenType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLV2
.0">
      <wsse:KeyIdentifier
         ValueType="http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.1#SAMLID">
        sxJu9g/vvLG9sAN9bKp/8q0NKU=
      </wsse:KeyIdentifier>
    </wsse:SecurityTokenReference>

    <!-- this is the signature the sender generated to demonstrate
    holder-of-key the signature should cover the isf header and body-->

    <ds:Signature>
      <ds:SignedInfo>
        <!-- in general include a ds:Reference for each wsa: header
             added according to SOAP binding -->

        <!-- include the MessageID in the signature -->
        <ds:Reference URI="#mid">...</ds:Reference>

        <!-- include the To in the signature -->
        <ds:Reference URI="#to">...</ds:Reference>

        <!-- include the Action in the signature -->
        <ds:Reference URI="#action">...</ds:Reference>

        <!-- include the MessageID in the signature -->
        <ds:Reference URI="#mid">...</ds:Reference>

        <!-- include the Timestamp in the signature -->
        <ds:Reference URI="#ts">...</ds:Reference>

        <!-- include the SAML Assertion in the signature to avoid
             token substitution attacks -->
        <ds:Reference URI="#Str1">
          <ds:Transform Algorithm="...#STR-Transform">
            <wsse:TransformationParameters>
        <ds:CanonicalizationMethod
          Algorithm="http://www.w3.org/TR/2001/REC-xml-c14n-20010315" />
       </wsse:TransformationParameters>
      </ds:Transform>
     </ds:Reference>

     <!-- include the message body -->
     <ds:Reference URI="#MsgBody">
      <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
      <ds:DigestValue>YgGfS0pi56pu...</ds:DigestValue>
     </ds:Reference>
    </ds:SignedInfo>

    <ds:SignatureValue>
     HJJWbvqW9E84vJVQkjjLLA6nNvBX7mY00TZhwBdFNDElgscSXZ5Ekw==
    </ds:SignatureValue>

    <ds:KeyInfo>
    </ds:KeyInfo>

   </ds:Signature>
  </wsse:Security>
 </s:Header>
 <s:Body wsu:Id="MsgBody">
  <pp:Modify>
   <!-- this is an ID-SIS-PP Modify message -->
  </pp:Modify>
 </s:Body>
</s:Envelope>
```

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

References

Normative

[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*

[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*

[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*

[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*

[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*

[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*

[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,

[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],

[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/xmldsig-core*

[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/*

[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*