Web Services Security SAML Token Binding

Working Draft 04~~2~~, 9~~23~~ December~~September~~ 2002

Document identifier:

WSS-SAML-04~~1~~

Location:

TBD

Editors:

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

Contributors:

TBD – Revise this list to include WSS TC contributors

Phillip Hallam-Baker, VeriSign
Jeff Hodges, Sun Microsystems
Maryann Hondo, IBM
Chris Kaler, Microsoft
Eve Maler, Sun Microsystems
Hiroshi Maruyama, IBM
Chris McLaren, Netegrity
Prateek Mishra, Netegrity
Anthony Nadalin, IBM
Nataraj Nagaratnam, IBM
Hemma Prafullchandra, VeriSign
Irving Reid, Baltimore
Krishna Sankar, Cisco
John Shewchuk, Microsoft

Abstract:

This document describes how to use Security Assertion Markup Language (SAML) assertions with the WS-Security specification.

Status:

This is an interim draft. Please send comments to the editors.

Committee members should send comments on this specification to ~~the mailto:~~wss@lists.oasis-open.org list. Others should subscribe to and send comments to the wss-comment@lists.oasis-open.org list. To subscribe, visit http://lists.oasis-open.org/ob/adm.pl.

For information on the disclosure of Intellectual Property Rights or licensing terms related to the work of the Web Services Security TC ~~whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the Security Services TC web page (http://www.oasis-open.org/who/intellectualproperty.shtml).~~please refer to the Intellectual Property Rights section of the TC web page at http://www.oasis-open.org/committees/wss/. The OASIS policy on Intellectual Property Rights is described at http://www.oasis-open.org/who/intellectualproperty.shtml.

Table of Contents

1 Introduction

The WS-Security specification proposes a standard set of SOAP extensions that can be used when building secure Web services to implement message level integrity and confidentiality. This specification describes the use of Security Assertion Markup Language (SAML) assertions from the `<wsse:Security>` header block defined by the ~~with respect to the~~ WS-Security specification.

1.1 Goals and Requirements

The goal of this specification is to define the use of SAML assertions in the context of WS-Security including for the purpose of securing SOAP message exchanges.

The requirements to be satisfied by this specification are listed below.

1.1.1 Requirements

TBS

~~□~~

1.1.2 Non-Goals

The following topics are outside the scope of this document:

~~□~~TBS

2 Notations and Terminology

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

2.1 Notational Conventions

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

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

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

Readers are presumed to be familiar with the terms in the Internet Security Glossary.

2.2 Namespaces

The XML namespace URIs that MUST be used by implementations of this specification are as follows (note that different elements in this specification are from different namespaces):

```
        http://schemas.xmlsoap.org/ws/2002/xx/secext
        http://schemas.xmlsoap.org/ws/2002/xx/utility
```

The following namespaces are used in this document:

Prefix	Namespace
S	http://www.w3.org/2001/12/soap-envelope
ds	http://www.w3.org/2000/09/xmldsig#
xenc	http://www.w3.org/2001/04/xmlenc#
wsse	http://schemas.xmlsoap.org/ws/2002/xx/secext
wsu	http://schemas.xmlsoap.org/ws/2002/xx/utility
saml	urn: oasis:names:tc:SAML:1.0:assertion

samlp	urn: oasis:names:tc:SAML:1.0:protocol

2.3 Terminology

This specification employs the terminology defined in the WS-Security Core Specification.

Defined below are the basic definitions for additional terminology used in this specification.

[TBS]

3 Usage

This section describes the specific mechanisms and procedures for the SAML binding of WS-Security.

Identification: urn:oasis:names:tc:WSS:1.0:bindings:WSS-SAML-binding

Contact information: TBD

Description: Given below.

Updates: None.

3.1 Processing Model

The SAML binding of WS-Security extends the token-independent processing model defined by the core WS-Security specification.

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

As part of its validation and processing of the selected assertions, the receiver MUST make ~~make~~ an explicit determination of the relationship between the subject of each~~each~~ assertion and the sender of the message. Two methods for establishing this correspondence, `holder-of-key` and `sender-vouches` are described below. Senders and receivers implementing the SAML binding of WS-Security MUST implement the processing necessary to support both of these subject confirmation methods.

3.2 Attaching Security Tokens

SAML assertions are attached to SOAP messages using WS-Security by placing assertion elements or references to assertions inside a `<wsse:Security>` header. The following example illustrates a SOAP message containing a SAML assertion in a `<wsse:Security>` header.

```
<S:Envelope xmlns:S="...">
    <S:Header>
        <wsse:Security xmlns:wsse="...">
            <saml:Assertion
                      MajorVersion="1"
                      MinorVersion="0"
                      AssertionID="SecurityToken-ef375268"
                      Issuer="elliotw1"
                      IssueInstant="2002-07-23T11:32:05.6228146-07:00"
                    xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion">
                ...
            </saml:Assertion>
        </wsse:Security>
    </S:Header>
    <S:Body>
        ...
    </S:Body>
</S:Envelope>
```

3.3 Identifying and Referencing Security Tokens

The WS-Security specification defines the <wsse:SecurityTokenReference> element for referencing security tokens. Three forms of token references are defined:

- An element reference – a security token specific XML element that contains an identifier and perhaps locator of a security token within the message or at some external location.
- A URI reference – a generic element that conveys in its attributes, the security token URI and token type value (i.e. ValueType) that define the location and perhaps identifier of a security token occurring either within the message or at some external location. A URI containing only a fragment identifier is interpreted as identifying the corresponding security token within the message in which the fragment identifier occurs.
- A key identifier reference – a generic element that conveys in its attributes, the security token identifier (i.e. wsu:id) and token type value (i.e. ValueType) that identifies a security token with matching wsu:id and ValueType occurring within a <wsse:Security> header of the message. Identifier references may only be used to reference security tokens that carry matching attributes, which approximately restricts their use to Binary Security Tokens attributed as a result of their encapsulation in XML.

A URI reference containing a URL may be combined with a token specific element reference to yield a location qualified reference.

In The SAML binding of WS-security, a referenced SAML assertion is identified by a <saml:AssertionIDReference> occurring either as an element reference or as a String value fragment identifier in a URI reference.

3.3.1 SAML Assertion Reference Elements

A < wsse:SecurityTokenReference> containing a <saml:AssertionIDReference> element containing a SAML assertion identifier may be used to reference a SAML assertion occurring within the <wsse:Security> header of the SOAP message in which the reference occurs. The following example illustrates the use of a <wsse:securityTokenReference> containing a <saml:AssertionIDReference> within the <keyInfo> of an XML Signature element to reference the SAML assertion (in the <wsse:Security> header) that contains the key used to compute the signature. ~~wsu:Id attribute as the common mechanism for referencing security tokens by "Id". Because the <saml:AssertionIDReference> element does not provide for attribute extensibility, this binding encapsulates <saml:AssertionIDReference> elements in the <wsse:SecurityTokenReference> element such that the wsu:id attribute of the encapsulating element can be used to~~

~~identify assertions according to the common WS-Security mechanism. When this element is encountered within a reference, the recipient, if it supports the SAML binding of WS-Security, MUST interpret the contained element as a <saml:AssertionIDReference>.~~

~~The following example illustrates a message with an XML Signature that references a SAML assertion token.~~

```
<S:Envelope xmlns:S="...">
    <S:Header>
        <wsse:Security xmlns:wsse="...">
            <saml:Assertion
                      MajorVersion="1"
                      MinorVersion="0"
                      AssertionID="SecurityToken-ef375268"
                      Issuer="elliotw1"
                      IssueInstant="2002-07-23T11:32:05.6228146-07:00"
                    xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion">
                ...
            </saml:Assertion>
            <ds:Signature xmlns:ds="...">
                ...
                <ds:KeyInfo>
                    <wsse:SecurityTokenReference>
                        <saml:AssertionIDReference>
                            SecurityToken-ef375268
                        </saml:AssertionIDReference>
                    </wsse:SecurityTokenReference>
                </ds:KeyInfo>
            </ds:Signature>
            ...
        </wsse:Security>
    </S:Header>
    <S:Body>
        ...
    </S:Body>
</S:Envelope>

```

3.3.2 URI References to SAML assertions

As depicted in the following example, a URI reference containing only a fragment identifier consisting of a <saml:AssertionIDReference> may be used to reference a SAML assertion occurring within the <wsseSecurity> header of the SOAP message in which the reference occurs. A URI reference containing an XML path expression can be used to reference a SAML assertion occurring anywhere within the containing SOAP message.

```
<wsse:SecurityTokenReference>
   <wsse:Reference URI="#SecurityToken-ef375268"
                      ValueType="saml:IDReferenceType">
   </wsse:Reference>
</wsse:SecurityTokenReference>
```

The following example demonstrates the use of a URI reference in conjunction with a <saml:AssertionIDReference> to define the location of the SAML responder at which the identified assertion may be obtained.

```
<wsse:SecurityTokenReference>
   <saml:AssertionIDReference>SecurityToken-ef375268
```

```
    </saml:AssertionIDReference>
    <wsse:Reference URI="http://www.fabrikam123.com/elliotw1"
    </wsse:Reference>
</wsse:SecurityTokenReference>
```

3.3.3 Identifier References to SAML Assertions

SAML assertions may not be referenced by identifier references because the `<saml:Assertion>` element schema does not include the `wsu:id` and `ValueType` attributes.

3.4 Proof-of-Possession of Security Tokens

T~~As previously stated, t~~he SAML binding of WS-Security requires that message senders and receivers support the holder-of-key and sender-vouches methods of subject confirmation. ~~Additional subject confirmation mechanisms may also be supported.~~ It is strongly RECOMMENDED that an XML signature be used to establish the relationship between the message sender and the attached assertions. This is especially RECOMMENDED whenever the SOAP message exchange is conducted over an unprotected transport.

Any processor of SAML assertions MUST conform to the required validation and processing rules defined in the SAML specification.

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

Mechanism	RECOMMENDED Processing Rules
`urn:oasis:names:tc:SAML:1.0:cm:holder-of-key`	The requestor ~~(the subject)~~ includes an XML Signature that can be verified with the key information in the <saml:ConfimationMethod> of the SAML assertion referenced by the Signature.~~referenced security token.~~
`Urn:oasis:names:tc:SAML:1.0:cm:sender-vouches`	The requestor (the sender, different from the subject) vouches for the verification of the subject. The receiver MUST have an existing trust relationship with the requestor to accept this. It is RECOMMENDED that the requestor sign the token and the message or use a secure transport.

Note that the high level processing model described in the following sections does not differentiate between message author and message sender as would be necessary to guard against replay attacks. The high-level processing model also does not take into account requirements for authentication of receiver by sender, or for message or assertion confidentiality. These concerns must be addressed by means other than those described in the high-level processing model.

3.4.1 Holder-of-key Subject Confirmation Method

The following sections describe the holder-of-key method of establishing the correspondence between a SOAP message sender and the subject of SAML assertions added to the SOAP message according to the SAML binding of WS-Security.

3.4.1.1 Sender

A message sender uses the holder-of-key confirmation method to demonstrate that it is authorized to act~~is~~ as the subject of the assertions in the message. The assertions included in a message that the sender will confirm by the holder-of-key method MUST include the following `<saml:SubjectConfirmation>` element:

```
<saml:SubjectConfirmation>
  <saml:ConfirmationMethod>
    urn:oasis:names:tc:SAML:1.0:cm:holder-of-key
  </saml:ConfirmationMethod>
  <ds:KeyInfo>…</ds:KeyInfo>
</saml:SubjectConfirmation>
```

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

To satisfy the associated confirmation method processing of the message receiver, the sender MUST demonstrate knowledge of the confirmation key~~key of the subject~~. The sender MAY accomplish this by using the confirmation key ~~of the subject~~ to sign content within the message and by including the resulting `<ds:Signature>` element in the `<wsse:Security>` header.

`<ds:Signature>` elements produced for this purpose MUST conform to the `canonicalization` and token inclusion rules defined in the core WS-Security specification.

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

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

3.4.1.2 Receiver

Of the SAML assertions it selects for processing, a message receiver ~~A message receiver~~ SHOULD NOT accept assertions containing a holder-of-key `<saml:ConfirmationMethod>`, unless the assertions are signed and validated as described above and the message sender has demonstrated knowledge of the key identified by the `<ds:keyInfo>` element of the `<saml:SubjectConfirmation>`

element.~~:~~ If the receiver determines that the sender has demonstrated knowledge of a subject confirmation key, then the SAML assertions containing the confirmation key MAY be attributed to the sender and any elements of the message whose integrity is protected by the subject confirmation key MAY be considered to have been authored by the subject.

3.4.1.3 Example

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

```
<?xml:version="1.0" encoding="UTF-8"?>
<S~~OAP-ENV~~:Envelope
xmlns:S~~OAP-ENV~~="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<S~~OAP-ENV~~:Header>
<wsse:Security>
  <saml:Assertion
    xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"
    MajorVersion="1" MinorVersion="0"
    AssertionID="2sxJu9g/vvLG9sAN9bKp/8q0NKU="
    Issuer="www.example.com"
    IssueInstant="2002-06-19T16:58:33.173Z">
    <saml:Conditions
      NotBefore="2002-06-19T16:53:33.173Z"
      NotOnOrAfter="2002-06-19T17:08:33.173Z"/>

    <saml:AuthenticationStatement
      AuthenticationMethod="urn:oasis:names:tc:SAML:1.0:am:password"
      AuthenticationInstant="2002-06-19T16:57:30.000Z">
      <saml:Subject>
        <saml:NameIdentifier
          NameQualifier="www.example.com"
          Format="">
                uid=joe,ou=people,ou=saml-demo,o=example.com
        </saml:NameIdentifier>
        <saml:SubjectConfirmation>
          <saml:ConfirmationMethod>
                urn:oasis:names:tc:SAML:1.0:cm:holder-of-key
          </saml:ConfirmationMethod>
          <ds:KeyInfo>
            <ds:KeyValue>…</ds:KeyValue>
          </ds:KeyInfo>
        </saml:SubjectConfirmation>
      </saml:Subject>
    </saml:AuthenticationStatement>

    <saml:AttributeStatement>
      <saml:Subject>
        <saml:NameIdentifier
          NameQualifier="www.example.com"
          Format="">
                uid=joe,ou=people,ou=saml-demo,o=baltimore.com
        </saml:NameIdentifier>
        <saml:SubjectConfirmation>
          <saml:ConfirmationMethod>
                urn:oasis:names:tc:SAML:1.0:cm:holder-of-key
          </saml:ConfirmationMethod>
          <ds:KeyInfo>
            <ds:KeyValue>…</ds:KeyValue>
          </ds:KeyInfo>
        </saml:SubjectConfirmation>
      </saml:Subject>

      <saml:Attribute
        AttributeName="MemberLevel"
        AttributeNamespace="http://www.oasis-open.org/Catalyst2002/attributes">
          <saml:AttributeValue>gold</saml:AttributeValue>
      </saml:Attribute>
      <saml:Attribute
        AttributeName="E-mail"
        AttributeNamespace="http://www.oasis-open.org/Catalyst2002/attributes">
        <saml:AttributeValue>joe@yahoo.com</saml:AttributeValue>
      </saml:Attribute>
    </saml:AttributeStatement>
    <ds:Signature>…</ds:Signature>
  </saml:Assertion>
  <ds:Signature>
    <ds:SignedInfo>…</ds:SignedInfo>
    <ds:SignatureValue>
HJJWbvqW9E84vJVQk…~~jjLLA6nNvBX7mY00TZhwBdFNDElgscSXZ5Ekw==~~
    </ds:SignatureValue>
    <ds:KeyInfo>
      <wsse:SecurityTokenReference>
        <saml:AssertionIDReference>"2sxJu9g/vvLG9sAN9bKp/8q0NKU="
        </saml:AssertionIDReference>
      </wsse:SecurityTokenReference>
    </ds:KeyInfo>
  </ds:Signature>
</wsse:Security>
</~~SOAP-EN~~S~~V~~:Header>

<~~SOAP-ENV~~S:Body>
  <ReportRequest>
    <TickerSymbol>SUNW</TickerSymbol>
  </ReportRequest>
</~~SOAP-ENV~~S:Body>
</S~~OAP-ENV~~:Envelope>
```

3.4.2 Sender-vouches Subject Confirmation Method

The following sections describe the sender-vouches method of establishing the correspondence between a SOAP message sender and the SAML assertions added to the SOAP message according to the SAML binding of WS-Security.

3.4.2.1 Sender

A message sender uses the sender-vouches confirmation method to assert that it is acting on behalf of the subjects of the assertions in the message. The assertions included in a message that the sender will confirm by the sender-vouches method MUST include the following `<saml:SubjectConfirmation>` element:

```
<saml:SubjectConfirmation>
  <saml:ConfirmationMethod>
         urn:oasis:names:tc:SAML:1.0:cm:sender-vouches
  </saml:ConfirmationMethod>
```

```
</saml:SubjectConfirmation>
```

To satisfy the associated confirmation method processing of the receiver, the sender MUST use its key to integrity protect the assertions and those elements of the SOAP message that the sender is vouching for. The sender MAY accomplish this by including in the corresponding `<wsse:Security>` header a `<ds:Signature>` element that the sender prepares by using its key to sign the assertions and relevant message content. As defined by the XML Signature Specification, the sender MAY identify its key by including a `<ds:KeyInfo>` element within the `<ds:Signature>` element.

A `<ds:Signature>` element produced for this purpose MUST conform to the `canonicalization` and token inclusion rules defined in the core WS-Security specification.

3.4.2.2 Receiver

Of the SAML assertions it selects for processing, a~~A~~ message receiver SHOULD NOT accept assertions containing a sender-vouches `<saml:ConfirmationMethod>` unless the assertions and SOAP message content being vouched for by the sender are integrity protected by a sender who is trusted by the receiver to act on behalf of the subject of the assertions.

3.4.2.3 Example

The following example illustrates a sender's use of the sender-vouches subject confirmation method with an associated `<ds:Signature>` element to establish its identity and to assert that it has sent message elements on behalf of the subjects of the contained assertions:

```
<?xml:version="1.0" encoding="UTF-8"?>
<S:Envelope xmlns:S="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">

<S:Header>
<wsse:Security>
  <saml:Assertion
    xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"
    MajorVersion="1" MinorVersion="0"
    AssertionID="2sxJu9g/vvLG9sAN9bKp/8q0NKU="
    Issuer="www.example.com"
    IssueInstant="2002-06-19T16:58:33.173Z">
    <saml:Conditions
      NotBefore="2002-06-19T16:53:33.173Z"
      NotOnOrAfter="2002-06-19T17:08:33.173Z"/>

    <saml:AuthenticationStatement
      AuthenticationMethod="urn:oasis:names:tc:SAML:1.0:am:password"
      AuthenticationInstant="2002-06-19T16:57:30.000Z">
      <saml:Subject>
        <saml:NameIdentifier
          NameQualifier="www.example.com"
          Format="">
                uid=joe,ou=people,ou=saml-demo,o=example.com
        </saml:NameIdentifier>
        <saml:SubjectConfirmation>
```

```
          <saml:ConfirmationMethod>
                urn:oasis:names:tc:SAML:1.0:cm:sender-vouches
          </saml:ConfirmationMethod>
        </saml:SubjectConfirmation>
      </saml:Subject>
    </saml:AuthenticationStatement>

    <saml:AttributeStatement>
      <saml:Subject>
        <saml:NameIdentifier
          NameQualifier="www.example.com"
          Format="">
                uid=joe,ou=people,ou=saml-demo,o=baltimore.com
        </saml:NameIdentifier>
        <saml:SubjectConfirmation>
          <saml:ConfirmationMethod>
                urn:oasis:names:tc:SAML:1.0:cm:sender-vouches
          </saml:ConfirmationMethod>
        </saml:SubjectConfirmation>
      </saml:Subject>

      <saml:Attribute
        AttributeName="MemberLevel"
        AttributeNamespace="http://www.oasis-
open.org/Catalyst2002/attributes">
          <saml:AttributeValue>gold</saml:AttributeValue>
      </saml:Attribute>
      <saml:Attribute
        AttributeName="E-mail"
        AttributeNamespace="http://www.oasis-
open.org/Catalyst2002/attributes">
        <saml:AttributeValue>joe@yahoo.com</saml:AttributeValue>
      </saml:Attribute>
    </saml:AttributeStatement>
  </saml:Assertion>
  <ds:Signature>
    <ds:SignedInfo>
      <ds:CanonicalizationMethod Algorithm=
        "http://www.w3.org/2001/10/xml-exc-c14n#"/>
      <ds:SignatureMethod Algorithm=
       "http://www.w3.org/2000/09/xmldsig#hmac-sha1"/>
      <ds:Reference URI="#2sxJu9g/vvLG9sAN9bKp/8q0NKU="
                    Type= “saml:IDReferenceType”>
        <ds:DigestMethod Algorithm=
          "http://www.w3.org/2000/09/xmldsig#sha1"/>
        <ds:DigestValue>GyGsF0Pi4xPU...</ds:DigestValue>
      </ds:Reference>
      <ds:Reference URI="#MsgBody">
        <ds:DigestMethod Algorithm=
          "http://www.w3.org/2000/09/xmldsig#sha1"/>
        <ds:DigestValue>GyGsF0Pi4xPU...</ds:DigestValue>
      </ds:Reference>
     </ds:SignedInfo>
    <ds:SignatureValue>JWbvqW94vJVQkA…</ds:SignatureValue>
    <ds:KeyInfo>
      <X509Data>
        <X509SubjectName>portal@yahoo.com</X509SubjectName>
      </X509Data>
    </ds:KeyInfo>
  </ds:Signature>
</wsse:Security>
</S:Header>
```

```
<S:Body wsu:Id="MsgBody">
  <ReportRequest>
    <TickerSymbol>SUNW</TickerSymbol>
  </ReportRequest>
</S:Body>

</S:Envelope>~~<SOAP-ENV:Envelope~~
~~  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">~~
~~  <SOAP-ENV:Header~~
~~    xmlns:saml="…"~~
~~    <wsse:Security>~~
~~      <wsse:SecurityTokenReference>~~
~~      <saml:AssertionIDReference>XVB12#$21abc</AssertionIDReference>~~
~~      <wsse:Reference URI=http://www.example.com/SAMLservice”/>~~
~~      </wsse:SecurityTokenReference>~~
~~      <saml:Assertion>…</saml:Assertion>~~
~~      <ds:Signature>…~~
~~        <ds:KeyInfo>…</ds:KeyInfo>~~
~~      </ds:Signature>~~
~~    </wsse:Security>~~
~~  </SOAP-ENV:Header>~~
~~  <SOAP-ENV:Body>~~
~~    …~~
~~  </SOAP-ENV:Body>~~
~~</SOAP-ENV:Envelope>~~
```

3.5 Error Codes

It is RECOMMENDED that systems that implement~~ing~~ the SAML binding of WS-Security respond with the error codes defined in the core WS-Security specification. Implementations that chose to respond with custom errors, defined in private namespaces, SHOULD take care not to introduce any security vulnerabilities as a result of the information returned in their error responses.

A receiver that is unable to process the SAML assertions contained in a `<wsse:Security>` header SHOULD use one of the fault codes listed in the core WS-Security specification to report the error. The RECOMMENDED correspondence between the common assertion processing failures and the error codes defined in the core WS-security specification are defined in the following table:

Assertion Processing Error	RECOMMENDED Error
A referenced SAML assertion could not be retrieved.	`Wsse:SecurityTokenUnavailable`
An assertion contains a `<saml:Condition>` element that the receiver does not understand.	`Wsse:UnsupportedSecurityToken`
A signature within an assertion or referencing~~including~~ an assertion is invalid.	`Wsse:FailedCheck`
The issuer of an assertion is not acceptable to the receiver.	`Wsse:InvalidSecurityToken`

The receiver does not understand the extension schema used in an assertion.	Wsse:UnsupportedSecurityToken

3.6 Threat Model and Countermeasures

This document defines the mechanisms and procedures for securely attaching SAML assertions to SOAP messages. SOAP messages are used in multiple contexts, specifically including cases where the message is transported without an active session, the message is persisted, or the message is routed through a number of intermediaries. Such a general context of use suggests that users of this binding must be concerned with a variety of threats. The following sections describe the vulnerability of the SAML token binding of WS-Security~~ to a variety of threats~~. In general, the use of SAML assertions with WS-Security introduces no new threats beyond those identified for SAML or by the core WS-Security specification.

The following sections provide an overview of the characteristics of the threat model, and the countermeasures that SHOULD be adopted for each perceived threat.

3.6.1 Eavesdropping

Eavesdropping is a threat to the SAML token binding of WS-Security in the same manner as it is a threat to any network protocol. The routing of SOAP messages through intermediaries increases the potential incidences of eavesdropping. Additional opportunities for eavesdropping exist when SOAP messages are persisted.

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

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

3.6.2 Replay

The reliance on authority signed assertions with a holder~~s~~-of-key subject confirmation mechanism precludes all but a holder of the key from binding the assertions to a SOAP message. Although this mechanism affectively restricts message authorship to the holder of the confirmation~~subject~~ key, it does not preclude the capture and resubmission of the message by other parties.

Assertions that contain a sender-vouches confirmation mechanism introduce another dimension to replay vulnerability because the assertions impose no restriction on the senders who may use or reuse the assertions. Any entity coming into contact with such assertions could use them in a message in which they use their identity to vouch for the subject of the assertions.

Replay attacks can be addressed by using message timestamps and caching, as well as by using other application-specific tracking mechanisms.

3.6.3 Message Insertion

The SAML token binding of WS-Security is not vulnerable to message insertion attacks.

3.6.4 Message Deletion

The SAML token binding of WS-Security is not vulnerable to message deletion~~insertion~~ attacks.

3.6.5 Message Modification

The SAML token binding of WS-Security is protected from message modification if the relevant message content is signed by the holder of the key or by the vouching sender. It is strongly RECOMMENDED that all relevant and immutable message content be signed by the sender. Receivers SHOULD only consider those portions of the document that are covered by the sender's signature as being subject to the assertions in the message.

SAML assertions appearing in <wsse:Security> header elements SHOULD be signed by their issuing a~~A~~uthority so~~uch~~ that message receivers can have confidence that the assertions have not been forged or altered since their issuance. It is strongly RECOMMENDED that a~~the~~ message sender ~~also~~ sign any ~~the~~ <saml:Assertion> elements that it is confirming and that ~~(either within the token, as part of the message,~~ are not signed by their issuing authority.~~or both).~~

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

3.6.6 Man-in-the-Middle

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

4 Acknowledgements

This specification was developed as a result of joint work of many individuals from the WSS TC including:

TBD

5 References

[DIGSIG] Informational RFC 2828, "Internet Security Glossary," May 2000.

[KEYWORDS] S. Bradner, "Key words for use in RFCs to Indicate Requirement Levels," RFC 2119, Harvard University, March 1997

[SAMLBind] Oasis Committee Specification 01, P. Mishra (Editor) Bindings and Profiles for the OASIS *Security Assertion Markup Language (SAML)*, May 2002.

[SAMLCore] Oasis Committee Specification 01, P. Hallem-Baker, and E. Maler, (Editors), *Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML)*, May 2002.

[SAMLReqs] OASIS Committee Consensus Draft, D. Platt, Evan Prodromou (Editors), SAML Requirements and Use Cases, OASIS, December 2001.

[SAMLSecure] OASIS Committee Specification 01, C. McLaren (Editor), Security and Privacy Considerations for the OASIS Security Assertion Markup Language (SAML) , May 2002.

[SOAP] W3C Note, "SOAP: Simple Object Access Protocol 1.1," 08 May 2000.

W3C Working Draft, Nilo Mitra (Editor), SOAP Version 1.2 Part 0: Primer, June 2002.

W3C Working Draft, Martin Gudgin, Marc Hadley, Noah Mendelsohn, Jean-Jacques Moreau, Henrik Frystyk Nielsen (Editors), SOAP Version 1.2 Part 1: Messaging Framework, June 2002.

W3C Working Draft, Martin Gudgin, Marc Hadley, Noah Mendelsohn, Jean-Jacques Moreau, Henrik Frystyk Nielsen (Editors), SOAP Version 1.2 Part 2: Adjuncts, June 2002.

[URI] T. Berners-Lee, R. Fielding, L. Masinter, "Uniform Resource Identifiers (URI): Generic Syntax," RFC 2396, MIT/LCS, U.C. Irvine, Xerox Corporation, August 1998.

[WS-SAML] Contribution to the WSS TC, P. Mishra (Editor), WS-Security Profile of the Security Assertion Markup Language (SAML) Working Draft 04, Sept 2002.

[WS-Security] TBS – point to the OASIS core draft

[XML-ns] W3C Recommendation, "Namespaces in XML," 14 January 1999.

[XML Signature] W3C Recommendation, "XML Signature Syntax and Processing," 12 February 2002.

[XML Token] Contribution to the WSS TC, Chris Kaler (Editor), WS-Security Profile for XML-based Tokens, August 2002.

Appendix A: Revision History

Rev	Date	What
01	19-Sep-02	Initial draft produced by extracting SAML related content from [XML token]
02	23-Sep-02	Merged in content from SS TC submission
03	18-Nov-02	Resolved issues raised by TC
04	09-Dec-02	Refined confirmation mechanisms, and added signing example

Appendix B: Notices

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

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

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

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

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