Web Services Security: SAML Token Profile

Working Draft 10, 06 April 2004

Document identifier:

{WSS : SOAP Message Security}–{SAML Token Profile}-{1.0}(Word)(PDF)

Location:

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

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

Editors:

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

Contributors (voting members of the WSS TC as of July 1st 2003)

Note: It is assumed that this list will be updated to be current on the date of Committee Spec.

Gene Thurston	AmberPoint
Frank Siebenlist	Argonne National Lab
Merlin Hughes	Baltimore Technologies
Irving Reid	Baltimore Technologies
Peter Dapkus	BEA
Hal Lockhart	BEA
Symon Chang	CommerceOne
Thomas DeMartini	ContentGuard
Guillermo Lao	ContentGuard
TJ Pannu	ContentGuard
Shawn Sharp	Cyclone Commerce
Ganesh Vaideeswaran	Documentum

Sam Wei	Documentum
John Hughes	Entegrity
Tim Moses	Entrust
Toshihiro Nishimura	Fujitsu
Tom Rutt	Fujitsu
Jason Rouault	HP
Yutaka Kudo	Hitachi
Maryann Hondo	IBM
Kelvin Lawrence	IBM (co-Chair)
Anthony Nadalin	IBM
Nataraj Nagaratnam	IBM
Don Flinn	Individual
Bob Morgan	Individual
Paul Cotton	Microsoft
Vijay Gajjala	Microsoft
Chris Kaler	Microsoft (co-Chair)
Chris Kurt	Microsoft
John Shewchuk	Microsoft
Prateek Mishra	Netegrity
Richard Levinson	Netegrity
Frederick Hirsch	Nokia
Senthil Sengodan	Nokia
Lloyd Burch	Novell
Ed Reed	Novell
Charles Knouse	Oblix
Steve Anderson	OpenNetwork (Secretary)
Vipin Samar	Oracle
Jerry Schwarz	Oracle
Eric Gravengaard	Reactivity
Stuart King	Reed Elsevier
Andrew Nash	RSA Security
Rob Philpott	RSA Security
Peter Rostin	RSA Security
Martijn de Boer	SAP
Pete Wenzel	SeeBeyond
Jonathan Tourzan	Sony
Yassir Elley	Sun Microsystems
Jeff Hodges	Sun Microsystems
Ronald Monzillo	Sun Microsystems
Jan Alexander	Systinet
Michael Nguyen	The IDA of Singapore
Don Adams	TIBCO
John Weiland	US Navy
Phillip Hallam-Baker	VeriSign
Morten Jorgensen	Vordel
Maneesh Satu	Westbridge

Contributors of input Documents (if not already listed above):

Hiroshi Maruyama	IBM
Chris McLaren	Netegrity
Eve Maler	Sun Microsystems
Hemma Prafullchandra	VeriSign

Abstract:

This document describes how to use Security Assertion Markup Language (SAML) V1.1 assertions with the Web Services Security (WSS): SOAP Message Security specification.

Status:

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

Committee members should send comments on this specification to 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 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 WSS: SOAP Message Security specification defines a standard set of SOAP extensions that implement message level integrity and confidentiality. This specification defines the use of Security Assertion Markup Language (SAML) assertions as security tokens from the `<wsse:Security>` header block defined by the WSS: SOAP Message Security specification.

1.1 Goals

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

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

1.1.1 Non-Goals

The following topics are outside the scope of this document:

3. Defining SAML statement syntax or semantics.
4. Describing the use of SAML assertions other than for SOAP Message Security.

Copyright © OASIS Open 2004. All Rights Reserved.

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.

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

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

Prefix	Namespace
S11	http://schemas.xmlsoap.org/soap/envelope/
S12	http://www.w3.org/2003/05/soap-envelope
ds	http://www.w3.org/2000/09/xmldsig#
xenc	http://www.w3.org/2001/04/xmlenc

Copyright © OASIS Open 2004. All Rights Reserved.

wsse	http://www.docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-01.xsd
wsu	http://www.docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd
saml	**`Urn: oasis:names:tc:SAML:1.0:assertion`**
samlp	**`Urn: oasis:names:tc:SAML:1.0:protocol`**

Table-1 Namespace Prefixes

2.3 Terminology

This specification employs the terminology defined in the WSS: SOAP Message Security specification. Defined below are the definitions for additional terminology used in this specification.

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

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

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

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

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

Copyright © OASIS Open 2004. All Rights Reserved.

3 Usage

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

3.1 Processing Model

This specification extends the token-independent processing model defined by the WSS: SOAP Message Security specification.

When a receiver processes a `<wsse:Security>` header containing or referencing SAML assertions, it selects, 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[1] 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 establish the relationship between the subject of each SAML subject statement (of the referenced SAML assertions) and the entity providing the evidence to satisfy the confirmation method defined for the statements (i.e. the attesting entity). Two methods for establishing this correspondence, `holder-of-key` and `sender-vouches` are described below. Systems implementing this specification 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 WSS: SOAP Message 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.

```
<S12:Envelope>
  <S12:Header>
    <wsse:Security>
      <saml:Assertion
        AssertionID="_a75adf55-01d7-40cc-929f-dbd8372ebdfc"
```

[1] The optional `Usage` attribute of the `<wsse:SecurityTokenReference>` element MAY be used to associate one of more semantic usage labels (as URIs) with a reference and thus use of a Security Token. Please refer to WSS: SOAP Message Security for the details of this attribute.

Copyright © OASIS Open 2004. All Rights Reserved.

```
        IssueInstant="2003-04-17T00:46:02Z"
        Issuer=”www.opensaml.org”
        MajorVersion="1"
        MinorVersion="1"
          . . .
      </saml:Assertion>
        . . .
    </wsse:Security>
  </S12:Header>
  <S12:Body>
    . . .
  </S12:Body>
</S12:Envelope>
```

3.3 Identifying and Referencing Security Tokens

The WSS: SOAP Message Security specification defines the `<wsse:SecurityTokenReference>` element for referencing security tokens. Three forms of token references are defined by this element and the element schema includes provision for defining additional reference forms should they be necessary. The three forms of token references defined by the `<wsse:SecurityTokenReference>` element are defined as follows:

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

  When a key identifier is used to reference a SAML assertion, it MUST contain as its element value the corresponding SAML assertion identifier. The key identifier MUST also contain a `ValueType` attribute and the value of this attribute MUST be the `wsse:KeyIdentifier/@ValueType` from Table 2. When the `EncodingType` attribute is not specified, the element content of the key identifier MUST be encoded as `xsi:string`.

  When a key identifier is used to reference a V1.1 SAML Assertion, a `<saml:AuthorityBinding>` element MUST be contained in the `<wsse:SecurityTokenReference>` element containing the key identifier. The contents of the `<saml:AuthorityBinding>` element MUST contain values sufficient for the intended recipients of the `<wsse:SecurityTokenReference>` to acquire the identified assertion from the intended Authority. To this end, the value of the `AuthorityKind` attribute of the `<saml:AuthorityBinding>` element MUST be "`samlp:AssertionIdReference`”.

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

When a Direct or URI reference is used to reference a SAML assertion within the document, the value of the URI attribute of the reference MAY be a fragment identifier containing the SAML assertion identifier (i.e. the value of the AssertionID attribute of the referenced assertion. Independent of whether a fragment identifier or full URI is specified, The reference MUST contain a ValueType attribute and the value of this attribute MUST be the wsse:Reference/@ValueType from Table 2 that corresponds to the version of the SAML Assertion being referenced.

- An Embedded reference – a reference that encapsulates a security token.

  When an Embedded reference is used to encapsulate a SAML assertion the SAML assertion MUST be included as a contained element within a <wsse:Embedded> element within a <wsse:SecurityTokenReference>.

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

- A SAML assertion may be referenced directly from a <wsse:Security> header element. In this case, the assertion is being conveyed by reference in the message.
- A SAML assertion may be referenced from a <ds:KeyInfo> element of a <ds:Signature> element in a <wsse:Security> header. In this case, the assertion contains a subject statement with a <saml:SubjectConfirmation> element that identifies the key used in the signature calculation.
- A SAML assertion reference may be referenced from a <ds:Reference> element within the <ds:SignedInfo> element of a <ds:Signature> element in a <wsse:Security> header. In this case, the doubly referenced assertion is signed by the containing signature.
- A SAML assertion may be referenced from a <xenc:DataReference> element within an <xenc:ReferenceList> element. In this case, the referenced assertion is encrypted.

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

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

SAML key identifier references, with a supporting <saml:AuthorityBinding> element are currently the best suited, of the <wsse:SecurityTokenReference>

Copyright © OASIS Open 2004. All Rights Reserved.

forms, for expressing remote references to SAML assertions. A future version of [SAMLCore] is expected to facilitate remote references by URI.

Attribute	Value
wsse:Reference/@ValueType	http://www.docs.oasis-open.org/wss/2004/XX/oasis-2004XX-wss-saml-token-profile-1.0#SAMLAssertion-1.0
wsse:Reference/@ValueType	http://www.docs.oasis-open.org/wss/2004/XX/oasis-2004XX-wss-saml-token-profile-1.0#SAMLAssertion-1.1
wsse:KeyIdentifier/@ValueType	http://www.docs.oasis-open.org/wss/2004/XX/oasis-2004XX-wss-saml-token-profile-1.0#SAMLAssertionID

Table-2 ValueType Attribute Values[2]

3.3.1 SAML Assertion Referenced from Header or Element

All conformant implementations MUST be able to process SAML assertion references occurring in a `<wsse:Security>` header or in a header element other than a signature to acquire the corresponding assertion.

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

```
<S12:Envelope>
  <S12:Header>
    <wsse:Security>
      <saml:Assertion
        AssertionID="_a75adf55-01d7-40cc-929f-dbd8372ebdfc"
        IssueInstant="2003-04-17T00:46:02Z"
        Issuer="www.opensaml.org"
        MajorVersion="1"
        MinorVersion="1"
          . . .
      </saml:Assertion>
    <wsse:SecurityTokenReference wsu:Id="STR1">
      <wsse:Reference wsu:Id="…"
        ValueType="http://www.docs.oasis-open.org/wss/2004/XX/oasis-
2004XX-wss-saml-token-profile-1.0#SAMLAssertion-1.1"
        URI="#_a75adf55-01d7-40cc-929f-dbd8372ebdfc"/>
      </wsse:SecurityTokenReference>
    </wsse:Security>
```

[2] This profile defines the use of SAML V1.1 assertions to secure SOAP messages. The profile also accommodates the use of SAML V1.0 assertions, although support for V1.0 assertions is optional.

```
  </S12:Header>
  <S12:Body>
    . . .
  </S12:Body>
</S12:Envelope>
```

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

```
<wsse:SecurityTokenReference wsu:Id="STR1">
  <saml:AuthorityBinding>
    Binding="urn:oasis:names:tc:SAML:1.0:bindings:SOAP-binding"
    Location="http://www.opensaml.org/SAML-Authority"
    AuthorityKind= "samlp:AssertionIdReference"
  </saml:AuthorityBinding>
  <wsse:KeyIdentifier
    wsu:Id="…"
    ValueType="http://www.docs.oasis-open.org/wss/2004/XX/oasis-2004XX-
wss-saml-token-profile-1.0#SAMLAssertionID">
    _a75adf55-01d7-40cc-929f-dbd8372ebdfc
  </wsse:KeyIdentifier>
</wsse:SecurityTokenReference>
```

3.3.2 SAML Assertion Referenced from KeyInfo

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

The following example depicts the use of a direct reference to a local assertion from <ds:KeyInfo>.

```
<ds:KeyInfo>
  <wsse:SecurityTokenReference wsu:Id="STR1">>
    <wsse:Reference wsu:Id="…"
      ValueType="http://www.docs.oasis-open.org/wss/2004/XX/oasis-
2004XX-wss-saml-token-profile-1.0#SAMLAssertion-1.0"
      URI="#_a75adf55-01d7-40cc-929f-dbd8372ebdfc"/>
  </wsse:SecurityTokenReference>
</ds:KeyInfo>
```

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

```
<ds:KeyInfo>
  <wsse:SecurityTokenReference wsu:Id="STR1">
    <saml:AuthorityBinding>
     Binding="urn:oasis:names:tc:SAML:1.0:bindings:SOAP-binding"
      Location="http://www.opensaml.org/SAML-Authority"
      AuthorityKind= "samlp:AssertionIdReference"
    </saml:AuthorityBinding>
    <wsse:KeyIdentifier wsu:Id="…"
      ValueType="http://www.docs.oasis-open.org/wss/2004/XX/oasis-
2004XX-wss-saml-token-profile-1.0#SAMLAssertionID">
   _a75adf55-01d7-40cc-929f-dbd8372ebdfc
    </wsse:KeyIdentifier>
  </wsse:SecurityTokenReference>
</ds:KeyInfo>
```

`<ds:KeyInfo>` elements may also occur in `<xenc:EncryptedData>` and `<xenc:EncryptedKey>` elements where they serve to identify the encryption key. `<ds:KeyInfo>` elements may also occur in <`saml:SubjectConfirmation>` elements where they identify a key that MUST be demonstrated to confirm the subject of the corresponding subject statement(s). Conformant implementations of this profile are not required to process SAML assertion references occurring within the `<ds:keyInfo>` elements within `<xenc:EncryptedData>`, `<xenc:EncryptedKey>`, or `<saml:SubjectConfirmation>`[3] elements.

3.3.3 SAML Assertion Referenced from SignedInfo

All conformant implementations MUST be able to process SAML assertions referenced by `<wsse:SecurityTokenReference>` from `<ds:Reference>` elements within the `<ds:SignedInfo>` element of a `<ds:Signature>` element in a `<wsse:Security>` header. Embedded references may be digested directly, thus affectively digesting the encapsulated assertion. Other `<wsse:SecurityTokenReference>` forms must be dereferenced for the referenced assertion to be digested.

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

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

```
<wsse:SecurityTokenReference wsu:Id="STR1">
  <saml:AuthorityBinding>
    Binding="urn:oasis:names:tc:SAML:1.0:bindings:SOAP-binding"
```

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

Copyright © OASIS Open 2004. All Rights Reserved.

```
    Location=”http://www.opensaml.org/SAML-Authority”
    AuthorityKind= “samlp:AssertionIdReference”
  </saml:AuthorityBinding>
  <wsse:KeyIdentifier wsu:Id=”…”
    ValueType=”http://www.docs.oasis-open.org/wss/2004/XX/oasis-2004XX-
wss-saml-token-profile-1.0#SAMLAssertionID”>
    _a75adf55-01d7-40cc-929f-dbd8372ebdfc
  </wsse:KeyIdentifier>
</wsse:SecurityTokenReference>
   . . .
<ds:SignedInfo>
  <ds:CanonicalizationMethod
    Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
  <ds:SignatureMethod
    Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
  <ds:Reference URI="#STR1">
    <Transforms>
      <ds:Transform
        Algorithm="http://www.docs.oasis-open.org/wss/2004/01/oasis-
200401-wss-soap-message-security-1.0#STR-Transform”/>
        <wsse:TransformationParameters>
          <ds:CanonicalizationMethod
            Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
        </wsse:TransformationParameters>
      </ds:Transform>
    </Transforms>
    <ds:DigestMethod
      Algorithm= "http://www.w3.org/2000/09/xmldsig#sha1"/>
    <ds:DigestValue>...</ds:DigestValue>
  </ds:Reference>
</ds:SignedInfo>
```

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

3.3.4 SAML Assertion Referenced from Encrypted Data Reference

All conformant implementations MUST be able to process SAML assertion references occurring in the `<xenc:DataReference>` element of a `<xenc:ReferenceList>` element. An `<xenc:ReferenceList>` element may occur either as a top level element in a Security header, or embedded within an `<xenc:EncryptedKey>` element. In either case, the `<xenc:ReferenceList>` identifies the encrypted content.

Such references are similar in format to the references that MAY appear in the `<ds:Reference>` element within `<ds:SignedInfo>`, except the STR Dereference transform does not apply. As shown in the following example, an encrypted assertion or an encrypted `<wsse:SecurityTokenReference>` is referenced from an

<xenc:DataReference> by a direct (i.e. URI) reference, where the URI appearing in the <xenc:DataReference> element identifies the encrypted (within the message) <wsse:SecurityTokenReference> element by its wsu:Id value.

```
<xenc:EncryptedData Id="STR1">
  <ds:KeyInfo>
     . . .
  </ds:KeyInfo>
  <xenc:CipherData>
    <xenc:CipherValue>...</xenc:CipherValue>
  </xenc:CipherData>
/xenc:EncryptedData>
<xenc:ReferenceList>
  <xenc:DataReference URI="#STR1"/>
</xenc:ReferenceList>
```

3.4 Subject Confirmation of SAML Assertions

The SAML profile of WSS: SOAP Message Security requires that systems support the holder-of-key and sender-vouches methods of subject confirmation. It is strongly RECOMMENDED that an XML signature be used to establish the relationship between the message and the subject statements of 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 [SAMLBind].

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 attesting entity includes an XML Signature that can be verified with the key information in the <saml:ConfimationMethod> of the subject statements of the SAML assertion referenced for keyInfo by the Signature.
urn:oasis:names:tc:SAML:1.0:cm:sender-vouches	The attesting entity, (presumed to be) different from the subject, vouches for the verification of the subject. The receiver MUST have an existing trust relationship with the

Copyright © OASIS Open 2004. All Rights Reserved.

	attesting entity. The attesting entity MUST protect the Assertion (containing the subject statements) in combination with the message content against modification by another party. See also section 4.

Note that the high level processing model described in the following sections does not differentiate between the attesting entity and the 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 (i.e. section 3.1).

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 and the subject of SAML assertions added to the SOAP message according to this specification.

3.4.1.1 Attesting Entity

An attesting entity uses the holder-of-key confirmation method to demonstrate that it is authorized to act as the subject of the SAML subject statements containing the holder-of-key `<saml:SubjectConfirmation>` element. The subject statements that will be confirmed 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[4] to be used to confirm the identity of the subject.

[4] [SAMLCore] defines KeyInfo of SubjectConfirmation as containing a “cryptographic key held by the subject”. Demonstration of this key is sufficient to establish who is (or may act as the) subject. Moreover, since it cannot be proven that a confirmation key is known (or known only) by the subject whose identity it establishes, requiring that the key be held by the subject is an untestable requirement that adds nothing to the strength of the confirmation mechanism. The OASIS Security Services Technical

To satisfy the associated confirmation method processing to be performed by the message receiver, the attesting entity MUST demonstrate knowledge of the confirmation key. The attesting entity MAY accomplish this by using the confirmation key 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 pre-pending rules defined in the WSS: SOAP Message 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 MUST NOT accept assertions containing a holder-of-key `<saml:ConfirmationMethod>`, unless the receiver has validated the integrity of the assertions and the attesting entity has demonstrated knowledge of the key identified by the `<ds:keyInfo>` element of the `<saml:SubjectConfirmation>` element.

If the receiver determines that the attesting entity has demonstrated knowledge of a subject confirmation key, then the SAML assertions containing the confirmation key MAY be attributed to the attesting entity and any elements of the message whose integrity is protected by the subject confirmation key MAY be considered to have been provided 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 and the subject of the SAML assertions in the `<wsse:Security>` header:

```
<?xml:version="1.0" encoding="UTF-8"?>
<S12:Envelope>
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <S12:Header>

    <wsse:Security>
      <saml:Assertion
```

Committee has resolved to remove the phrase “held by the subject” from the definition of KeyInfo of SubjectConfirmation.

Copyright © OASIS Open 2004. All Rights Reserved.

```
        AssertionID="_a75adf55-01d7-40cc-929f-dbd8372ebdfc"
        IssueInstant="2003-04-17T00:46:02Z"
        Issuer=”www.opensaml.org”
        MajorVersion="1"
        MinorVersion="1"
        xmlns="urn:oasis:names:tc:SAML:1.0:assertion">
        <saml:Conditions>
          NotBefore="2002-06-19T16:53:33.173Z"
          NotOnOrAfter="2002-06-19T17:08:33.173Z"/>
        <saml:AttributeStatement>
          <saml:Subject>
            <saml:NameIdentifier
              NameQualifier="www.example.com"
              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:CanonicalizationMethod
            Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
          <ds:SignatureMethod
            Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
          <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>HJJWbvqW9E84vJVQk…</ds:SignatureValue>
        <ds:KeyInfo>
          <wsse:SecurityTokenReference wsu:Id=”STR1”>
```

```
            <wsse:Reference wsu:Id=”…”
              ValueType=”http://www.docs.oasis-
open.org/wss/2004/XX/oasis-2004XX-wss-saml-token-profile-
1.0#SAMLAssertion-1.0”
              URI=”#_a75adf55-01d7-40cc-929f-dbd8372ebdfc”/>
          </wsse:SecurityTokenReference>
        </ds:KeyInfo>
      </ds:Signature>
    </wsse:Security>
  </S12:Header>

  <S12:Body wsu:Id="MsgBody">
    <ReportRequest>
      <TickerSymbol>SUNW</TickerSymbol>
    </ReportRequest>
  </S12:Body>
</S12: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 and the SAML assertions added to the SOAP message according to the SAML profile of WSS: SOAP Message Security.

3.4.2.1 Attesting Entity

An attesting entity uses the sender-vouches confirmation method to assert that it is acting on behalf of the subject of SAML subject statements containing a sender-vouches `<saml:SubjectConfirmation>` element. The subject statements that the attesting entity 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 attesting entity MUST protect the vouched for SOAP message content such that the receiver can determine when it has been altered by another party. The attesting entity MUST also cause the vouched for subject statements (as necessary) and their binding to the message contents to be protected such that unauthorized modification can be detected. The attesting entity MAY satisfy these requirements by including in the corresponding `<wsse:Security>` header a `<ds:Signature>` element that it prepares by using its key to sign the relevant message content and assertions. As defined by the XML Signature specification, the attesting entity 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 prepending rules defined in the WSS: SOAP Message Security specification.

3.4.2.2 Receiver

Of the SAML assertions it selects for processing, a message receiver MUST NOT accept assertions containing a sender-vouches <saml:ConfirmationMethod> unless the assertions and SOAP message content being vouched for are protected (as described above) by an attesting entity 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 an attesting entity'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 assertion (i.e., the assertion referenced by "STR1"):

```
<?xml:version="1.0" encoding="UTF-8"?>
<S12:Envelope>
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <S12:Header>
    <wsse:Security>

      <saml:Assertion
        AssertionID="_a75adf55-01d7-40cc-929f-dbd8372ebdfc"
        IssueInstant="2003-04-17T00:46:02Z"
        Issuer="www.opensaml.org"
        MajorVersion="1"
        MinorVersion="1"
        xmlns="urn:oasis:names:tc:SAML:1.0:assertion">
        <saml:Conditions>
          NotBefore="2002-06-19T16:53:33.173Z"
          NotOnOrAfter="2002-06-19T17:08:33.173Z"/>
        <saml:AttributeStatement>
          <saml:Subject>
            <saml:NameIdentifier
              NameQualifier="www.example.com"
              Format="">
              uid=proxy,ou=system,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
            . . .
          </saml:Attribute>
            . . .
        </saml:AttributeStatement>
      </saml:Assertion>

      <wsse:SecurityTokenReference wsu:Id=”STR1”>
        <saml:AuthorityBinding>
          saml:Binding=”urn:oasis:names:tc:SAML:1.0:bindings:SOAP-
binding”
          saml:Location=”http://www.opensaml.org/SAML-Authority”
          saml:AuthorityKind= “samlp:AssertionIdReference”
        </saml:AuthorityBinding>
        <wsse:KeyIdentifier wsu:Id=”…”
          ValueType=”http://www.docs.oasis-open.org/wss/2004/XX/oasis-
2004XX-wss-saml-token-profile-1.0#SAMLAssertionID”>
          _a75adf55-01d7-40cc-929f-dbd8372ebdbe
        </wsse:KeyIdentifier>
      </wsse:SecurityTokenReference>

      <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#rsa-sha1"/>
          <ds:Reference URI="#STR1">
            <Transforms>
              <ds:Transform
                Algorithm="http://www.docs.oasis-
open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0#STR-
Transform"/>
                <wsse:TransformationParameters>
                  <ds:CanonicalizationMethod
                    Algorithm="http://www.w3.org/2001/10/xml-exc-
c14n#"/>
                </wsse:TransformationParameters>
              </ds:Transform>
            </Transforms>
            <ds:DigestMethod
              Algorithm= "http://www.w3.org/2000/09/xmldsig#sha1"/>
            <ds:DigestValue>...</ds:DigestValue>
          </ds:Reference>
          <ds:Reference URI="#MsgBody">
            <ds:DigestMethod
              Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
            <ds:DigestValue>...</ds:DigestValue>
          </ds:Reference>
        </ds:SignedInfo>
        <ds:SignatureValue>HJJWbvqW9E84vJVQk…</ds:SignatureValue>
        <ds:KeyInfo>
          <wsse:SecurityTokenReference wsu:Id=”STR2”>
            <wsse:Reference wsu:Id=”…”
```

```
              ValueType=”http://www.docs.oasis-
open.org/wss/2004/XX/oasis-2004XX-wss-saml-token-profile-
1.0#SAMLAssertion-1.0”
              URI=”#_a75adf55-01d7-40cc-929f-dbd8372ebdfc”/>
          </wsse:SecurityTokenReference>
        </ds:KeyInfo>
      </ds:Signature>
    </wsse:Security>
  </S12:Header>

  <S12:Body wsu:Id="MsgBody">
    <ReportRequest>
      <TickerSymbol>SUNW</TickerSymbol>
    </ReportRequest>
  </S12:Body>
</S12:Envelope>
```

3.5 Error Codes

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

Systems that choose to return faults SHOULD respond with the error codes defined in the WSS: SOAP Message Security specification. The RECOMMENDED correspondence between the common assertion processing failures and the error codes defined in WSS: SOAP Message Security are defined in the following table:

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

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

The preceding table defines fault strings and codes in a form suitable to be used with SOAP 1.1. The WSS: SOAP Message Security specification describes how to map SOAP 1.1 fault constructs to the SOAP 1.2 fault constructs.

Copyright © OASIS Open 2004. All Rights Reserved.

4 Threat Model and Countermeasures (Non-Normative)

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 profile must be concerned with a variety of threats.

In general, the use of SAML assertions with WSS: SOAP Message Security introduces no new threats beyond those identified for SAML or by the WSS: SOAP Message 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.

4.1 Eavesdropping

Eavesdropping is a threat to the SAML token profile of WSS: SOAP Message 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, assertion references, and sensitive message content SHOULD be encrypted such that only the intended audiences can view their content. This approach removes threats of eavesdropping in transit, but MAY not remove risks associated with storage or poor handling by the receiver.

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

4.2 Replay

Reliance on authority protected (e.g. signed) assertions with a holder-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 data origin to a holder of the confirmation key, it does not, by itself, provide the means to detect the capture and resubmission of the message by other parties.

Copyright © OASIS Open 2004. All Rights Reserved.

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

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

4.3 Message Insertion

The SAML token profile of WSS: SOAP Message Security is not vulnerable to message insertion attacks.

4.4 Message Deletion

The SAML token profile of WSS: SOAP Message Security is not vulnerable to message deletion attacks.

4.5 Message Modification

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

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

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

4.6 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

Copyright © OASIS Open 2004. All Rights Reserved.

vulnerable to MITM attacks to the degree that the receiver does not have a trusted binding of key to the attesting entity’s identity.

5 References

[GLOSSARY] 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, E. Maler, P.Mishra, and R. Philpott (Editors), *Bindings and Profiles for the OASIS Security Assertion Markup Language (SAML) V1.1*, September 2003.

[SAMLCore] Oasis Committee Specification 01, E. Maler, P.Mishra, and R. Philpott (Editors), *Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) V1.1*, September 2003.

[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.

[WSS: SOAP Message Security] Oasis Standard, A. Nadalin, C.Kaler, P. Hallem-Baker, R. Monzillo (Editors), Web Services Security: SOAP Message Security 1.0 (WS-Security 2004), August 2003.

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

Copyright © OASIS Open 2004. All Rights Reserved.

[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
05	15-Dec-02	Results of Baltimore F2F
06	21-Feb-03	Changed name to profile
07	05-May-03	Acknowledged contributors
07	05-May-03	Throughout document, Refined terminology to distinguish attesting entity from subject and sender, and to distinguish assertions from statements within assertions. Also modified sender-vouches to support traced vouching (by allowing for the use of a confirmation key)
08	09-Jun-03	Indicated reliance on conventions of core in “Notational Conventions”
08	09-Jun-03	In “Terminology”, added definitions of new terms (attesting entity and confirmation method identifier), edited definition of Subject Confirmation, and replaced definition of sender with subject.
08	09-Jun-03	In “Subject Confirmation of SAML Assertions”, added requirement that an attesting entity must protect unsigned sender-vouches confirmed assertions.
08	25-Nov-03	Added SAM v1.1 version distinction to “Abstract”
08	25-Nov-03	Editorial changes to “Introduction”
08	25-Nov-03	Reorganized non-normative text of requirements and goals sections
08	25-Nov-03	Removed Identification, Contact Information, Description, and Updates from “Usage”.
08	25-Nov-03	Updated schema URIs and corrected

Rev	Date	What
		namespace prefixes in “Namespaces”
08	25-Nov-03	Updated SAML document references in “References” to point to v1.1. specs.
08	25-Nov-03	In Error codes, changed error processing such that it is optional and consistent with the recommendations in core.
08	25-Nov-03	Qualified “Threat Model and Counter-measures” as non-normative.
08	30-Nov-03	In “Identifying and Referencing Security Tokens”, removed keyname references and added embedded references. Also removed editorial comment regarding using artifacts to reference assertions.
08	30-Nov-03	Made editorial changes to “Processing Model”, including clarification (by footnote) of “semantic labeling”
08	30-Nov-03	Removed “Acknowledgments” as it duplicated preceding sections of the document
08	12-15-03	Added high level goals and non-goals
08	12-15-03	Added support for the use of (fragment) URI references to section 3.3
08	12-15-03	Specified default encoding type for SAML and fragment UR references to be xsi:string
08	12-15-03	Added two more contexts in which SAML assertions may be referenced; from within SubjectConfirmation elements and as encrypted data.
08	12-15-03	Made it a requirement of conformant implementations that they support the various methods of referencing SAML assertions
08	12-15-03	Added new sections to describe SAML assertion referenced from SubjectConfirmation and SAML assertion referenced from Encrypted Data reference.
09	01-27-04	Changed document identifier and location
09	01-27-04	Modified namespace table of section 2.2 to differentiate SOAP 1.1 and SOAP 1.2

Copyright © OASIS Open 2004. All Rights Reserved.

Rev	Date	What
10	02-05-04	Changed all instances of wsu:id to wsu:Id
10	02-05-04	In section 3.4.2.1 beginning around line 705, removed the distinction of the "typical case where the assertion authority has NOT securely bound a key…" because we no longer expect sender-vouches to use a confirmation key.
10	3-29-04	Corrected STR transform URL to match change in core.
10	3-29-04	Removed from section 3.3.2 mention of use of KeyInfo with sender-vouches confirmation method.
10	3-29-04	Modified footnote in section 3.2 regarding usage attribute to reflect change from QNAMES to URIs.
10	3-29-04	Corrected signature algorithm in examples.
10	3-29-04	Corrected transforms syntax of example in section 3.3.3.
10	3-29-04	In section 3.3.3 recommended that STR dereference transform not be applied to embedded token references.
10	3-29-04	Removed requirement (from section 4.5 of Security Considerations) that assertion references be protected from unauthorized modification.
10	4-02-04	Removed namespace qualification from ValueType, URI, EncodingType, and Usage Attributes (mostly in examples). Also removed angle brackets.
10	4-05-04	Reworded initial paragraph of section 2.2 Namespaces such that it is not normative, and affords more flexibility in the form of the examples.
10	4-05-04	Removed namespace declarations from examples.
10	4-05-04	Corrected misspelling of "Authorty" in examples.
10	4-05-04	Modified processing rule for sender-vouches in Table of section 3.4 (to allow sender to vouch

Copyright © OASIS Open 2004. All Rights Reserved.

Rev	Date	What
		for itself).
10	4-05-04	Editing changes to the error codes section. In particular, replaced the word “generated” with “returned”, and rewrote the description of the mapping to 1.2 constructs.
10	4-05-04	Removed unused SAMLreqs and SAMLSecure from the references section.
10	4-06-04	Added footnote to explain optional support for SAML V1.0 assertions.
10	4-06-04	Removed section 3.3.4 “SAML Assertion referenced from SubjectConfirmation”, as SAML is evolving in a manner that will make it unlikely that authorities will need to produce such assertions. Moved the description of SAML Assertions references occurring within KeyInfo of SubjectConfirmation to section 3.3.2 “SAML assertion referenced from KeyInfo”
10	4-06-04	From Section 3.3 “Identifying and referencing Security Tokens”, removed referencing a SAML assertion from KeyInfo of SubjectConfirmation from the five contexts in which SAML assertions may be referenced.
10	4-06-04	Moved description of SAML Assertion references occurring within KeyInfo of SubjectConfirmation to section 3.3.2.
10	4-06-04	Added footnote to description of holder-of-key semantics in section 3.4.1.1 to describe interpretation of “held by the subject” phrase appearing in definition in [SAMLCore].
10	4-06-04	Updated contributors list

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.

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

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.