WS-Security Profile for XML-based Tokens

Version 1.0

August 28, 2002

Authors

Phillip Hallam-Baker, VeriSign
Maryann Hondo, IBM
Chris Kaler (Editor), Microsoft
Hiroshi Maruyama, IBM
Anthony Nadalin, IBM
Nataraj Nagaratnam, IBM
Hemma Prafullchandra, VeriSign
John Shewchuk, Microsoft

Copyright Notice

AGREEMENT RELATING TO THIS DOCUMENT, WHETHER OR NOT SUCH PARTY HAD ADVANCE NOTICE OF THE POSSIBILITY OF SUCH DAMAGES.

Abstract

This document describes a general framework to enable XML-based security tokens to be used with WS-Security. Two profiles that use this general framework are provided: one for the Security Assertion Markup Language (SAML) and another for the eXtensible rights Markup Language (XrML).

Status

This document is provided as-is and for review and evaluation only. IBM and Microsoft and VeriSign hope to solicit your contributions and suggestions in the near future. IBM and Microsoft and VeriSign make no warrantees or representations regarding the specifications in any manner whatsoever.

Table of Contents

1. Introduction

- 1.1. Notational Conventions
- 1.2. Namespaces

2. General Principles

- 2.1. Attaching Security Tokens
- 2.2. Identifying and Referencing Security Tokens
- 2.3. Subject Confimation
- 2.4. Processing Rules

3. Security Assertion Markup Language (SAML) Usage

- 3.1. Processing Model
- 3.2. Attaching Security Tokens
- 3.3. Identifying and Referencing Security Tokens
- 3.4. Subject Confirmation
- 3.5. Error Codes
- 3.6. Threat Model and Countermeasures

4. eXtensible rights Markup Language (XrML) Usage

- 4.1. Processing Model
- 4.2. Attaching Security Tokens
- 4.3. Identifying and Referencing Security Tokens
- 4.4. Subject Confirmation
- 4.5. Error Codes
- 4.6. Threat Model and Countermeasures

5. Security Considerations

6. References

1. Introduction

There is a growing popularity in XML-based security tokens. Two well-known formats are the Security Assertion Markup Language [SAML] and the eXtensible rights Markup Language [XrML]. Since these formats are described in standalone specifications, not unlike X.509 and Kerberos, this document describes their usage with respect to the WS-Security specification.

This document describes a general framework to enable XML-based security tokens to be used with WS-Security. Two profiles that use this general framework are provided: one for the Security Assertion Markup Language (SAML) and another for the eXtensible rights Markup Language (XrML). Note that this specification does not endorse any particular XML security token standard – the description of SAML and XrML are provided to show the mechanisms by which the bindings should be performed. Additional XML token formats may be added to this specification in future revisions as needed.

1.1. 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.

1.2. Namespaces

The following namespaces are used in this document:

Prefix	Namespace
S	http://www.w3.org/2002/06/soap-envelope
ds	http://www.w3.org/2000/09/xmldsig#
saml	urn:oasis:names:tc:SAML:1.0:assertion
xenc	http://www.w3.org/2001/04/xmlenc#
wsse	http://schemas.xmlsoap.org/ws/2002/07/secext
xmltok	http://schemas.xmlsoap.org/ws/2002/08/xmltok
wsu	http://schemas.xmlsoap.org/ws/2002/07/utility

xrml	http://www.xrml.org/schema/2001/11/xrml2core

2. General Principles

This section presents the basic principals around using WS-Security with security tokens. Later sections describe rules and processes specific to certain XML-based security token formats.

2.1. Attaching Security Tokens

The WS-Security specification defines the <wsse:Security> header as a mechanism for conveying security information with and about a SOAP message. This header is, by design, extensible to support many types of security information.

The specification defines the <wsse:BinarySecurityToken> element as a mechanism for attaching security tokens that are represented by binary octet streams and therefore do not naturally lend themselves to XML.

For security tokens based on XML, the extensibility of the <wsse:Security> header allows for these security tokens to be directly inserted into the header.

2.2. Identifying and Referencing Security Tokens

The WS-Security specification defines multiple mechanisms for identifying and referencing security tokens using the *wsu:Id* attribute and the <wsse:SecurityTokenReference> element (as well as some additional mechanisms).

2.3. Subject Confirmation

The WS-Security specification does not dictate how subject confirmation must be done, however, it does define how signatures can be used and associated with security tokens (by referencing them in the signature) towards this end.

2.4. Processing Rules

The WS-Security specification describes the processing rules for using and processing XML Signature and XML Encryption. These rules MUST be followed when using any type of security token including XML-based tokens. Note that this does NOT mean that XML-based tokens MUST be signed or encrypted – only that if signature or encryption is used in conjunction with XML-based tokens, they MUST be used in a way that conforms to the processing rules defined by the WS-Security specification.

3. Security Assertion Markup Language (SAML) Usage

This section describes the profile (specific mechanisms and procedures) for the WS-Security profile of SAML.

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 processing model for WS-Security with SAML assertion tokens is no different from that of WS-Security with other token formats as described in WS-Security.

3.2. Attaching Security Tokens

SAML assertions are attached to SOAP messages using WS-Security by placing assertion elements inside the <wsse:Security> header. The following example illustrates a SOAP message with 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>
            ...
        </wsse:Security>
    </S:Header>
    <S:Body>
        ...
    </S:Body>
</S:Envelope>
```

3.3. Identifying and Referencing Security Tokens

The WS-Security specification defines the *wsu:Id* attribute as the common mechanism for referencing security tokens by "Id" (the specification describes the reasons for this). Since the SAML specification does not allow attribute extensibility on the <saml:Assertion> element, this specification allows the <saml:AssertionIDReference> element to be placed inside of a <wsse:SecurityTokenReference> element. When this element is encountered within a reference, the recipient, if it supports SAML assertion tokens, MUST know to de-reference the SAML Assertion ID reference to identify the correct SAML assertion to use as the security token.

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>
```

3.4. Subject Confirmation

As previously stated, the WS-Security specification does not dictate how subject confirmation must be performed. As well, the SAML specification allows for multiple types of confirmation. If a secure transport is not used, it is strongly RECOMMENDED that a key-based confirmation mechanism be used.

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

The following table illustrates how several different confirmation mechanisms are processed:

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 referenced security token.
`urn:ietf:rfc:3075`	The requestor (the subject) includes an XML Signature that can be verified with the key information in the 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.

3.5. Error Codes

When using SAML assertion tokens, it is RECOMMENDED to use the error codes defined in the WS-Security specification. However, implementations MAY use custom errors, defined in private namespaces if they desire. Care should be taken not to introduce security vulnerabilities in the errors returned.

3.6. Threat Model and Countermeasures

The use of SAML assertion tokens with WS-Security introduces no new threats beyond those identified for SAML or WS-Security with other types of security tokens.

Message alteration and eavesdropping can be addressed by using the integrity and confidentiality mechanisms described in WS-Security. Replay attacks can be addressed by using message timestamps and caching, as well as other application-specific tracking mechanisms. For SAML assertion tokens whose ownership is verified by use of keys, man-in-the-middle attacks are generally mitigated by the use of subject confirmation.

It is strongly RECOMMENDED that all relevant and immutable message data be signed.

It should be noted that transport-level security MAY be used to protect the message and the security token.

4. eXtensible rights Markup Language (XrML) Usage

This section describes the profile (specific mechanisms and procedures) for the WS-Security profile of XrML.

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

Contact information: TBD

Description: Given below.

Updates: None.

4.1. Processing Model

The processing model for WS-Security with XrML tokens is no different from that of WS-Security with other token formats as described in WS-Security.

4.2. Attaching Security Tokens

XrML licenses are attached to SOAP messages using WS-Security by placing the license element inside the <wsse:Security> header. The following example illustrates a SOAP message with an XrML license token.

```
<S:Envelope xmlns:S="...">
    <S:Header>
        <wsse:Security xmlns:wsse="...">
            <xrml:license xmlns:xrml="...">
                ...
            </xrml:license>
            ...
        </wsse:Security>
    </S:Header>
    <S:Body>
        ...
    </S:Body>
</S:Envelope>
```

4.3. Identifying and Referencing Security Tokens

The WS-Security specification defines the *wsu:Id* attribute as the common mechanism for referencing security tokens by "Id" (the specification describes the reasons for this). Since the XrML specification does not allow attribute extensibility of the <xrml:license> element, this specification defines a separate mechanism for referencing licenses. The XrML specification allows licenses to be named using a URI with the *licenseId* attribute. Consequently, this specification defines the global namespace qualifier attribute *xmltok:RefType* for use with the <wsse:Reference> element (used within a <wsse:SecurityTokenReference> element). Using this attribute, references can specify the type of token desired thereby allowing the token-specific matching rules to be processed. Specifically, when the *xmktok:RefType* attribute's value is "xrml:license", then the *URI* attribute refers to an <xrml:license> element whose *licenseId* attribute is specified by the URI attribute.

The following example illustrates a message with an XML Signature that references an XrML token.

```
<S:Envelope xmlns:S="...">
```

```
    <S:Header>
        <wsse:Security xmlns:wsse="...">
            <xrml:license xmlns:xrml="..."
                          licenseId="urn:SecurityToken-ef375268"/>

                ...
            </xrml:license>
            <ds:Signature xmlns:ds="...">
                ...
                <ds:KeyInfo>
                   <wsse:SecurityTokenReference>
                     <wsse:Reference URI="urn:SecurityToken-ef375268"
                                    xmltok:RefType="xrml:license"
                                    xmlns:xmltok="..."/>
                   </wsse:SecurityTokenReference>
                </ds:KeyInfo>
            </ds:Signature>
            ...
        </wsse:Security>
    </S:Header>
    <S:Body>
        ...
    </S:Body>
</S:Envelope>
```

4.4. Subject Confirmation

As previously stated, the WS-Security specification does not dictate how subject confirmation must be performed. As well, the XrML specification allows for multiple types of confirmation. If a secure transport is not used, it is strongly RECOMMENDED that a key-based confirmation mechanism be used.

Any processor of XrML security tokens MUST conform to the required validation and processing rules defined in the XrML specification.

The following table illustrates how several different confirmation mechanisms are processed:

Mechanism	**RECOMMENDED Processing Rules**
`<xrml:keyHolder>`	The sender (the subject) includes an XML

	Signature that can be verified with the key information in the referenced security token.
<xrml:allPrincipals>	The sender (the subject) includes an XML Signature that can be verified. An implementation MAY choose to not require principals to "authenticate".

4.5. Error Codes

When using XrML tokens, it is RECOMMENDED to use the error codes defined in the WS-Security specification. However, implementations MAY use custom errors, defined in private namespaces if they desire. Care should be taken not to introduce security vulnerabilities in the errors returned.

4.6. Threat Model and Countermeasures

The use of XrML security tokens with WS-Security introduces no new threats beyond those identified for XrML or WS-Security with other types of security tokens.

Message alteration and eavesdropping can be addressed by using the integrity and confidentiality mechanisms described in WS-Security. Replay attacks can be addressed by using of message timestamps and caching, as well as other application-specific tracking mechanisms. For XrML tokens whose ownership is verified by use of keys, man-in-the-middle attacks are generally mitigated.

It is strongly RECOMMENDED that all relevant and immutable message data be signed.

It should be noted that transport-level security MAY be used to protect the message and the security token.

5. Security Considerations

In order to provide relying parties with the confidence that they can *trust* XML-based tokens, the issuers of those tokens SHOULD sign those tokens using the mechanisms outlined in this document. Signing XML tokens allows parties relying on them to be confident that the tokens haven't been forged or altered. It is strongly RECOMMENDED that <saml:Assertion> and <xrml:license> elements used in WS-Security header fields be signed (using either the token-signing mechanisms defined in the SAML or XrML specifications or the header-element signing mechanisms defined in the WS-Security specification, or both mechanisms)

It should be noted that references to unsigned or unsecured tokens represent potential security holes and make increase attack opportunities.

6. References

[KEYWORDS]

S. Bradner, "Key Words for Use in RFCs to Indicate Requirement Levels," RFC 2119, Harvard University, March 1997.

[SAML]

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

[SOAP]

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

Draft, SOAP 1.2, http://www.w3.org/TR/soap12-part0/

Draft, SOAP 1.2, http://www.w3.org/TR/soap12-part1/

Draft, SOAP 1.2, http://www.w3.org/TR/soap12-part2/

[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.

[XML-Encrypt]

W3C Working Draft, "XML Encryption Syntax and Processing," 04 March 2002.

[XML-ns]

W3C Recommendation, "Namespaces in XML," 14 January 1999.

[XML Signature]

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

[XrML]

"eXtensible rights Markup Language (XrML) 2.0 Specification", ContentGuard, 11/2001

[WS-Security]

"Web Services Security Language", IBM, Microsoft, VeriSign, April 2002.

"WS-Security Addendum", IBM, Microsoft, VeriSign, August 2002.