SAML & XML-Signature Syntax and Processing

This version:

File : draft-sstc-dsig-02.doc
Date : October 24, 2001

Authors

- Krishna Sankar [ksankar@Cisco.com]
-

Contributors

- Scott Cantor [cantor.2@osu.edu]
- Prateek Mishra [pmishra@netegrity.com]
- Stephen Farrell [stephen.farrell@baltimore.ie]
- Philip Hallam-Baker [pbaker@verisign.com]

Abstract

XML Signature is used in SAML for assertion integrity, assertion authentication and signer authentication as defined in [SIG]. The XML Signature specification [SIG] defines how this can be achieved and provides many options. This document details the use of XML Signature for SAML assertions and protocols.

Referenced Documents

[SIG] XML-Signature Syntax and Processing, W3C Proposed Recommendation.
http://www.w3.org/TR/2001/PR-xmldsig-core-20010820/

[RFC3126] RFC 3126 : Electronic Signature Formats for long term electronic signatures

[RFC3125] RFC 3125 : Electronic Signature Policies

Notational Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in Key Words for Use in RFC’s to Indicate Requirement Levels (RFC 2119).

Status of this Document

This document represents work in progress upon which no reliance should be made.

Document Version History

- Version 0.001:
- Version 0.002:

Related Files

The web site http://www.oasis-open/committees/security/xxxxx contains the current version of all the related files.

Table of Contents

1 Role of Digital Signatures in SAML

SAML Assertions, Request and Response messages may be signed, with the following benefits:

- An Assertion signed by the issuer (AP). This supports :
  (1) Message integrity
  (2) Authentication of the issuer to a relying party
  (3) If the signature is based on the issuer's public-private key pair, then it also provides for non-repudiation of origin.

- A SAML request or a SAML response message signed by the message originator. This supports :
  (1) Message integrity
  (2) Authentication of message origin to a destination
  (3) If the signature is based on the originator's public-private key pair, then it also provides for non-repudiation of origin.

Note :

- SAML documents may be the subject of signatures from in many different packaging contexts. [SIG] provides a framework for signing in XML and is the framework of choice. However, signing may also take place in the context of S/MIME or Java objects that contain SAML documents. One goal is to ensure compatibility with this type of "foreign" digital signing.
- It is useful to characterize situations when a digital signature is NOT required in SAML.

  (1) Assertions: asserting party has provided the assertion to the relying party and authenticated by means other than digital signature. In other words, the RP has obtained the assertion from the AP directly (no intermediaries) and the AP has authenticated to the RP.

  (2) Request/Response messages: the originator has authenticated to the destination and the destination has obtained the assertion directly from the originator (no intermediaries).

  Many different techniques are available for "direct" authentication between two parties. The list includes SSL, HMAC, password-based login etc. [QUESTION: Do we need to constrain this list further?]

- All other contexts require the use of digital signature for assertions and request and response messages. Specifically:

  (1) An assertion obtained by a relying party from an entity other than the asserting party MUST be signed by the issuer.

(2) SAML message obtained arriving at a destination from an entity other than the originating site MUST be signed by the origin site.

2 Signing Assertions

All SAML assertions MAY be signed using the XML Signature. This is reflected in the schema :

```
<element name = "Assertion" type = "saml:AssertionAbstractType"/>
    <complexType name = "AssertionAbstractType" abstract = "true">
        <sequence>
            <element ref = "saml:Conditions" minOccurs = "0"/>
            <element ref = "saml:Advice" minOccurs = "0"/>
            <element ref = “ds:Signature” minOccurs=”0” maxOccurs=”1”/>
        </sequence>
        <attribute name = "MajorVersion" use = "required" type = "integer"/>
        <attribute name = "MinorVersion" use = "required" type = "integer"/>
        <attribute name = "AssertionID" use = "required" type = "saml:IDType"/>
        <attribute name = "Issuer" use = "required" type = "string"/>
        <attribute name = "IssueInstant" use = "required" type = "timeInstant"/>
    </complexType>
```

3 Request/Response Signing

All SAML requests and responses MAY be signed using the XML Signature. This is reflected in the schema :

```
        <complexType name="RequestAbstractType" abstract="true">
            <attribute name="RequestID" type="saml:IDType" use="required"/>
            <attribute name="MajorVersion" type="integer" use="required"/>
            <attribute name="MinorVersion" type="integer" use="required"/>
            <element ref = “ds:Signature” minOccurs=”0” maxOccurs=”1”/>
        </complexType>

        <complexType name="ResponseAbstractType" abstract="true">
            <attribute name="ResponseID" type="saml:IDType" use="required"/>
            <attribute name="InResponseTo" type="saml:IDType" use="required"/>
            <attribute name="MajorVersion" type="integer" use="required"/>
            <attribute name="MinorVersion" type="integer" use="required"/>
            <element ref = “ds:Signature” minOccurs=”0” maxOccurs=”1”/>
        </complexType>
```

4 Signature Inheritance (a.k.a. super-signatures & sub-messages)

4.1 Context

SAML assertions may be embedded within request or response messages or other XML messages, which may be signed. Request or response messages may themselves be contained within other messages that are based on other XML messaging frameworks (e.g., SOAP) and the composite object may be the subject of a signature. Another possibility is that SAML assertions or request/response messages are embedded within a non-XML messaging object (e.g., MIME package) and signed.

In such a case, the SAML sub-message (Assertion, request, response) may be viewed as inheriting a signature from the "super-signature" over the enclosing object, provided certain constraints are met.

(1) An assertion may be viewed as inheriting a signature from a super signature, if the super signature applies all of the mandatory elements within the assertion.

(2) A SAML request or response may be viewed as inheriting a signature from a super signature, if the super signature applies to all of the mandatory elements within the response.

4.2 Proposal

Signatures MAY inherited in the SAML domain. i.e. if a SAML request/response has a signature, then if any of the assertions in the res/resp packages are not signed, they inherit the super-signature.

But if assertions need to be passed around by themselves, or embedded in other message they would need to be signed as per section 2.1

5 XML Signature Profile

The [SIG] specification calls out a general XML syntax for signing data with many flexibilities and choices. This section details the constraints on these facilities so that SAML processors do not have to deal with the full generality of [SIG] processing.

5.1 Signing formats

XML Signature has three ways of representing signature in a document viz: enveloping, enveloped and detached.

SAML assertions and protocols would use the enveloped signatures for signing assertions.

5.2 CanonicalizationMethod

[Sig] REQUIRES the Canonical XML (omits comments) (http://www.w3.org/TR/2001/REC-xml-c14n-20010315). SAML RECOMMENDS the Canonical XML with Comments (http://www.w3.org/TR/2001/REC-xml-c14n-20010315#WithComments)

5.3 Transforms

[Sig] REQUIRES the enveloped signature transform http://www.w3.org/2000/09/xmldsig#enveloped-signature

5.4 KeyInfo

Any valid key which is acceptable by the [SIG] is acceptable to SAML as well. SAML does not restrict or impose any additions in this area. Which means it is possible NOT to have the KeyInfo element and then arrive at the keyinfo by context.

5.5 Object

The Object element SHOULD NOT be present in the signature block

5.6 Binding between statements in a multi-statement assertion

5.6.1 MultipleAssertionType

This structure packs multiple related statements as one assertion. The relationships between these assertions are implied.

5.6.2 Multiple Assertions in a ResponseType

This structure packs multiple assertions in a response message. The relationship between the assertions in this case is arbitrary i.e. no inter-assertion relationship should be assumed just because all the assertions were packaged in the sane response message

5.7 Security considerations

5.7.1 Replay Attack

The mechanisms stated here-in does not offer any counter measures against a replay attack. Other mechanisms like sequence numbers, time stamps, expiration et al need to be explored to prevent a replay attack.

6 Issues, To Do

Issue	Status
√ Binding between different SAML fragments	N/A
√ Replay Attack ?	Added security considerations Para
√ Granularity	
Multiple signers	Next Version
Signing multiple assertions	Use MultipleAssertionType for related statements or have multiple assertions in response for unrelated assertions
Detached signature as attribute assertions to tie payload ?	?
Or a new assertion payload assertion ?	?
√ Trust assertion due to bearer or the stated issuer? [Kelvin Beeck]	Stated issuer
√ Encryption?	Next Version
√ Counter Signature	Next Version
√ Multiple Signature	Next Version
√ Manifest	Next Version
√ Bearer Assertion	Next Version