SOAP Security Extensions: Digital Signature

W3C NOTE 06 February 2001

This version:
http://www.w3.org/TR/2001/NOTE-SOAP-dsig-20010206/
Latest version:
http://www.w3.org/TR/SOAP-dsig/
Authors:
Allen Brown, Microsoft
Barbara Fox, Microsoft
Satoshi Hada, IBM
Brian LaMacchia, Microsoft
Hiroshi Maruyama, IBM

Abstract

This document specifies the syntax and processing rules of a SOAP header entry to carry digital signature information within a SOAP 1.1 Envelope.

Status

This document is a submission to the World Wide Web Consortium (see Submission Request, W3C Staff Comment). For a full list of all acknowledged Submissions, please see Acknowledged Submissions to W3C.

This document is a NOTE made available by the W3C for discussion only. Publication of this Note by the W3C indicates no endorsement by W3C or the W3C Team, or any W3C Members. The W3C has had no editorial control over the preparation of this Note. This document is a work in progress and may be updated, replaced, or rendered obsolete by other documents at any time.

A list of current W3C technical documents can be found at the Technical Reports page.

Table of Contents

1. Motivation
    1. Notational Conventions
2. Header Entry Syntax
    1. Namespace
    2. Signature HeaderEntry
    3. SOAP-SEC:id Attribute
    4. Example
3. Processing Rules
    1. Signature Header Entry Generation
    2. Signature Header Entry Validation
4. Security Considerations
5. References

1. Motivation

The motivation for this Note is to propose a standard way to use the XML Digital Signature syntax [XML-Signature] to sign SOAP 1.1 messages [SOAP]. We define a SOAP header entry <SOAP-SEC:Signature> for this purpose.

We also propose the definition of an extensible namespace for adding security features to the SOAP header. By extensible we mean that new elements can be added to the schema overtime but elements in the schema will not change. It is our intention that other security features, such as confidentiality of SOAP 1.1 messages, will be added within this namespace as appropriate standards, such as forthcoming XML Encryption, become available. What we specifically defer to another Note or working group is the definition of an authentication protocol for SOAP. By "protocol", we mean any expectation of processing by the recipient of a signed/encrypted message.

1.1 Notational Conventions

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

Namespace URIs of the general form "some-URI" represent some application-dependent or context-dependent URI as defined in RFC2396 [URI]. The namespace prefixes "SOAP-ENV" and "ds" used in this document are associated with the namespaces "http://schemas.xmlsoap.org/soap/envelope/" and "http://www.w3.org/2000/09/xmldsig#", respectively.

2 Header Entry Syntax

2.1 Namespace

The XML namespace [XML-ns] URI that MUST be used by implementations of this specification is:

```
http://schemas.xmlsoap.org/soap/security/2000-12
```

The namespace prefix "SOAP-SEC" used in this specification is associated with this URI.

2.2 Signature Header Entry

The header entry <SOAP-SEC:Signature> is defined by the following schema [XML-Schema1], [XML-Schema2]. The <SOAP-SEC:Signature> element contains a single digital signature conforming to the XML-Signature specification [XML-Signature].

```
<schema
  xmlns="http://www.w3.org/1999/XMLSchema"
  xmlns:SOAP-SEC="http://schemas.xmlsoap.org/soap/security/2000-12"
  targetNamespace="http://schemas.xmlsoap.org/soap/security/2000-12"
  xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">

  <import namespace="http://www.w3.org/2000/09/xmldsig#"/>
  <import namespace="http://schemas.xmlsoap.org/soap/envelope/"/>

  <element name="Signature" final="restriction">
    <complexType>
      <sequence>
        <element ref="ds:Signature" minOccurs="1" maxOccurs="1"/>
      </sequence>
      <attribute name="id" type="ID" use="optional"/>
      <attribute ref="SOAP-ENV:actor" use="optional"/>
      <attribute ref="SOAP-ENV:mustUnderstand" use="optional"/>
    </complexType>
  </element>

  <attribute name="id" type="ID"/>

</schema>
```

2.3 SOAP-SEC:id Attribute

The <ds:Reference> element needs to refer to the signed part of the SOAP Envelope. This can be achieved through the use of XML identifiers. Applications are responsible for determining which attributes are of the type ID. To help applications to identify attributes of the type ID, this specification defines the SOAP-SEC:id global attribute. This attribute MAY be used for referencing the signed part of the SOAP Envelope.

2.4 Example

Here is an example of a SOAP message with a signature header entry, where the SOAP Body is signed and the resulting signature <ds:Signature> is added to the <SOAP-SEC:Signature> header entry. Note that the "URI" attribute of the <ds:Reference> element refers to the <SOAP-ENV:SOAP-Body> element.

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header>
    <SOAP-SEC:Signature
      xmlns:SOAP-SEC="http://schemas.xmlsoap.org/soap/security/2000-12"
      SOAP-ENV:actor="some-URI"
      SOAP-ENV:mustUnderstand="1">
      <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <ds:SignedInfo>
          <ds:CanonicalizationMethod
            Algorithm="http://www.w3.org/TR/2000/CR-xml-c14n-20001026">
          </ds:CanonicalizationMethod>
          <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmlds
          <ds:Reference URI="#Body">
            <ds:Transforms>
              <ds:Transform Algorithm="http://www.w3.org/TR/2000/CR-xml-
            </ds:Transforms>
            <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsi
            <ds:DigestValue>j6lwx3rvEPO0vKtMup4NbeVu8nk=</ds:DigestValue
          </ds:Reference>
        </ds:SignedInfo>
        <ds:SignatureValue>MC0CFFrVLtRlk=...</ds:SignatureValue>
      </ds:Signature>
    </SOAP-SEC:Signature>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body
    xmlns:SOAP-SEC="http://schemas.xmlsoap.org/soap/security/2000-12"
    SOAP-SEC:id="Body">
    <m:GetLastTradePrice xmlns:m="some-URI">
      <m:symbol>IBM</m:symbol>
    </m:GetLastTradePrice>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

3. Processing Rules

The Signature header entry is used to carry a signature compliant with the XML-Signature specification [XML-Signature] within a SOAP Envelope for the purpose of signing one or more elements in the SOAP Envelope. Multiple signature header entries MAY be added into a single SOAP Envelope with either disjoint or overlapping signed elements. A future version of this specification may allow signature syntax other than the XML-Signature through extension [XML-Schema1] of the content model of <SOAP-SEC:Signature>.

SOAP applications conforming to this specification MUST satisfy the following conditions:

1. The application MUST be capable of processing XML Signature as defined in the XML-Signature specification [XML-Signature].
2. If a conforming SOAP application is to add a <SOAP-SEC:Signature> header entry in the SOAP Header, the header entry MUST have a <ds:Signature> element conforming to the XML-Signature specification [XML-Signature]. All the <ds:Reference> elements contained in the signature MUST refer to a resource within the enclosing SOAP Envelope, or

a resource in the enclosing SOAP message package [SOAP-attachment] if the envelope is the primary SOAP 1.1 message [SOAP-attachment] of the package.

3. When a conforming SOAP application receives a SOAP message containing one or more <SOAP-SEC:Signature> header entries intended for the application (either it is explicitly specified by the SOAP "actor" attribute or the application is the ultimate destination), for each such header entry, the application MUST perform the following steps:
    1. Decide whether or not to process the header entry (either forced by the `mustUnderstand="1"` attribute or voluntarily),
    2. If it is to be processed, the application MUST try to validate the signature using the processing model of XML Signature [XML-Signature].

Note that XML Canonicalization [XML-C14N] of <ds:SignedInfo> and other signed resources MUST each be done within its own context. This means, among other things, that the Canonical form [XML-C14N] of <ds:SigndInfo> always inherits the namespace declarations for SOAP-ENV and SOAP-SEC.

The rest of this section describes the operations to be performed for the signature header entry.

3.1 Signature Header Entry Generation

One way to create a <SOAP-SEC:Signature> header entry is as follows.

1. Prepare the target SOAP Envelope with the body and necessary headers.
2. Create a template of a <ds:Signature> element. The template is assumed to contain empty contents for <ds:DigestValue> or <ds:SignatureValue> elements, but contains appropriate values for the elements such as <ds:SignatureMethod> and <ds:Reference> required to calculate them.
3. Create a new header entry <SOAP-SEC:Signature> and add the template to this entry.
4. Add the header entry <SOAP-SEC:Signature> to the SOAP Header.
5. Add the SOAP "actor" and "mustUnderstand" attributes to the entry, if necessary.
6. Calculate the <ds:DigestValue> and <ds:SignatureValue> elements according to the core generation of the XML-Signature specification [XML-Signature].

XPath filtering can be used to specify objects to be signed, as described in the XML-Signature specification [XML-Signature]. However, since the SOAP message exchange model allows intermediate applications to modify the Envelope (add or delete a header entry, for example), XPath filtering does not always result in the same objects after message delivery. Care should be taken in using XPath filtering so that there is no subsequent validation failure due to such modifications.

The transform `http://www.w3.org/2000/09/xmldsig#enveloped-signature` defined in the XML-Signature specification [XML-Signature] may be useful when signing the entire Envelope including other header entries, if any.

3.2 Signature Header Entry Validation

The validation of a <SOAP-SEC:Signature> header entry fails if

1. The syntax of the content of the header entry does not conform to this specification, or
2. The validation of the signature contained in the header entry fails according to the core validation of the XML-Signature specification [XML-Signature], or
3. The receiving application program rejects the signature for some reason (e.g., the signature is created by an untrusted key).

If the validation of the signature header entry fails, applications MAY report the failure to the sender. It is out of the scope of this specification how to deal with it.

4. Security Considerations

This specification defines the use of XML Signature in SOAP 1.1 headers. As one of building blocks for securing SOAP messages, it is intended to be used in conjunction with other security techniques. Digital signatures need to be understood in the context of other security mechanisms and the threats to an entity. Digital signatures are, according to the IETF RFC 2828[DIGSIG],

> "A value computed with a cryptographic algorithm and appended to a data object in such a way that any recipient of the data can use the signature to verify the data's origin and integrity. (See:data origin authentication service, data integrity service,digitized signature, electronic signature, signer.)"
>
> "Data appended to, or a cryptographic transformation of, a data unit that allows a recipient of the data unit to prove the source and integrity of the data unit and protect against forgery, e.g. by the recipient." [I7498Part2]

For example, digital signatures alone do not provide message authentication. One can record a signed message and resend it (replay attack). To prevent this type of attack, digital signatures must be combined with an appropriate means to ensure the uniqueness of the message, such as nonces or time stamps. One way to add this information is to place an extra <ds:Object> element that is a child of the <ds:Signature>.

When digital signatures are used for verifying the identity of the sending party, the sender must prove the possession of the private key. One way to achieve this is to use a challenge-response type of protocol.

Implementers should also be aware of all the security implications resulting from the use of digital signatures in general and XML Signature in particular. In building trust into an application based on a digital signature there are other pieces of technology that must be identified in relation to the signature. There needs to be a

certificate trust model, whether hierarchical or peer-to-peer. There needs to be a way to generate and maintain trusted key pairs and certificates. And there must be a way to validate that the certificate has not been revoked.

5. References

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

[I7498Part2]
ISO/IEC 7498-2, "Information Processing Systems -- Open Systems Interconnection Reference Model Part 2: Security Architecture," 1989.

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

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

[SOAP-attachment]
W3C Note, "SOAP Messages with Attachments," 11 December 2000.

[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-C14N]
W3C Candidate Recommendation, "Canonical XML Version 1.0," 26 October 2000.

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

[XML-Schema1]
W3C Candidate Recommendation, "XML Schema Part 1: Structures," 24 October 2000.

[XML-Schema2]
W3C Candidate Recommendation, "XML Schema Part 2: Datatypes," 24 October 2000.

[XML-Signature]
W3C Candidate Recommendation, "XML-Signature Syntax and Processing," 31 October 2000.