XMLDSIG Working Group | Richard D. Brown
GlobeSet, Inc.
Expires December 1999 | June 1999

Digital Signatures for XML - Comments

Richard D. Brown
GlobeSet, Inc.

Document Status

This document, file name <draft-ietf-xmldsig-signature-comments-00.txt> is intended to become a Proposed Standard RFC. Distribution of this document is unlimited. Comments should be sent to the XMLDSIG mailing list or to the author. Additional information can be found on the web sites maintained by the working group.

This document is an Internet-Draft and is in full conformance with all provisions of Section 10 of RFC2026.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

The list of current Internet-Drafts can be accessed at http://www.ietf.org/ietf/1id-abstracts.txt

The list of Internet-Draft Shadow Directories can be accessed at http://www.ietf.org/shadow.html

Abstract

This specification consists of a series of comments and suggestions with regard to the Internet-Draft XMLDSIG main specification. It constitutes the primary source of information for forthcoming revisions of the main specification.

The main specification can be accessed at http://www.ietf.org/internet-drafts/draft-ietf-xmldsig-signature-00.txt

Contacts

Chair(s):
Donald Eastlake 3rd <dee3@torque.pothole.com>
Joseph Reagle Jr. <reagle@w3.org>

Author:
Richard D. Brown <rdbrown@globeset.com>

Mailing List:
Discussion: w3c-ietf-xmldsig@w3.org
Archive: http://lists.w3.org/Archives/Public/w3c-ietf-xmldsig
Subscription: w3c-ietf-xmldsig-request@w3.org
specify (un)subscribe in SUBJECT line with an empty body.

Web Sites:
IETF: http://www.ietf.org/html.charters/xmldsig-charter.html
W3C: http://www.w3.org/Signature

Revision History

18-June-99:
Create independent document,
Create an index per status,
Add entry #99061601,
Add entry #99061602,
Add entry #99061603.

04-April-99:
Initial draft in main specification.

Table of Contents

1. Comments and Suggestions

This chapter consists of a central repository for all the comments and suggestions that have been received with regard to the XMLDSIG main specification.

#98072901 - Syntax too verbose

```
Origin:   General
Status:   Superceded by 98091001, 98091002
```

Comments:

"The syntax is far too much verbose."

Author Notes:

There are multiple optimizations that could be envisioned:

- Promote syntax compactness instead of element reusability.
- Adopt syntax that enables shared algorithm definitions.
- Adopt default attribute and parameter values.

#98091001 - Enable shared algorithm definitions

Origin: Richard D. Brown
Status: Opened

Comments:

"Considering that in most circumstances the digest of authenticate resources will be computed by means of a common digest algorithm, it seems preferable to define a syntax that allows shared algorithm definitions.

Several approaches might be considered. The first one consists of adding a DigestAlgorithms element in every signature Manifest and using some linking mechanism (i.e. IDREF) to bind a digest value with a digest algorithm. Another approach is to allow definitions throughout the document and requiring the canonicalizer to inline the algorithm definition in the Digest elements."

Author Notes:

The first approach does not prevent replication of algorithm definitions in the header of the document and the different signature Manifests. On the other hand, the second approach requires particular behaviors in the canonicalizers and therefore cannot suffice with a surface string digest algorithm.

#98091002 - Promote compactness over reusability

Origin: Richard D. Brown
Status: Opened

Comments:

"The current specification makes use of reusable element types. This approach obviously increases the "verbosity" of the syntax. It might be preferable to promote compactness instead of reusability."

For example, the following optimizations might be considered:

- Collapsing Locator into Resource.
- Collapsing ContentInfo into Package.
- Replacing basic data types (i.e. Integer, Date, etc...) by a DCD attribute and a PCDATA contents on the parent element.

Author Notes:

#98111301 - Add a Resource criticality attribute

Origin: Milton M. Anderson
Status: Opened

Comments:

"The definition of <resource> does not contain a feature to allow the signer to declare that the resource is critical to the validity of the signature."

Author Notes:

I do not have all the elements for judging how relevant is this particular comment, but my first guess is that a signature applies to a particular content and semantics verification shall be left to the application layer.

#98111302 - Remove dsig:eval global attribute

Origin: Milton M. Anderson
Status: Closed

Comments:

"Having the dsig:eval attribute in the element that defines the authenticated block is probably a bad idea, since different signers can use different hash algorithm."

Author Notes:

Others have made a similar comment, but we have reached a different conclusion: dsig:eval shall be an IDREFS instead of an IDREF.

#98111303 - Add a logging attribute

Origin: Milton M. Anderson
Status: Closed

Comments:

"No ability to mark data for logging by a signing hardware is included."

Author Notes:

Also, very much application specific. I am not quite sure this shall be a concern for the XML DSIG Proposal.

#98111304 - Add a signature purpose attribute

Origin: Milton M. Anderson
Status: Closed

Comments:

"The signature doesn't have a "type of signature" element."

Author Notes:

This may be provided at the application level by means of a complementary attribute.

#98111305 - Add a Nonce in the Manifest

Origin: Milton M. Anderson
Status: Closed

Comments:

"No nonce to be used in hashing the resource is defined."

Author Notes:

Milton refers to a scheme that could be used for preventing birthday-attacks against the digest algorithm.

E-Check makes use of this artifice to make sure that a fraudulent signer or a Trojan Horse on the signer's computer does not have full control over the data being signed by the signing device. If the attacker were able to find two messages M1 and M2 such that H(M1) = H(M2), it could send M1 to the device and then substitute M2 later. The hardware log will record that M1 has been signed and not M2.

This problem may be addressed in the hardware log by itself. The device may sign H(M1), pick a nonce at random, and log the nonce value along with H(nonce||M1) for example.

#98112501 - Leverage DCD and other specifications

Origin: Hiroshi Maruyama
Status: Opened

Comments:

"For data types, I think we could refer to other activities such as DCD."

Author Notes:

Agreed as long as these specifications are adopted in time. Notice that DCD provides mostly for basic data types. This will not resolve the problem for the constructed data types such as IssuerAndSerialNumber, Package, etc...

#98121501 - IssuerAndSerialNumber is too restrictive

Origin: Richard D. Brown
Status: Opened

Comments:

"Identifying a certificate by means of the constructed data types IssuerAndSerialNumber might be too restrictive. It is not obvious that this construct is sufficient for certificates other than X509v3. The specification shall adopt syntax a bit more opened.

For example, a certificate might be uniquely identified by means of a Identifier element, whose syntax depends upon the type of the certificate."

Author Notes:

#98122601 - Allow multiple Resource in Manifest

Origin: Don Eastlake
Status: Adopted (990404)

Comments:

"I believe that multiple occurrences of Resource should be permitted in the Manifest"

Author Notes:

The Manifest now requires a Resources element.

#98122602 - Add a base locator for HREFs

Origin: Don Eastlake - IOTP WG
Status: Opened

Comments:

"Some of the IOTP concerns about many huge locator HREFs could be satisfied if a "base" attribute were permitted at the Resources level or, even more general, a Base element, which effected all following resource."

Author Notes:

#98122603 - Rename the attribute dsig:eval

Origin: Don Eastlake
Status: Opened

Comments:

"I do not like the choice of "eval" even if it is arbitrary. It makes me think of Lisp or the like. I would expect it to evaluate arithmetic expressions or the like. I think the previous "hash" was better and perhaps "dsig:digest" would be best..."

Author Notes:

#98122604 - Default encoding attribute to base64

Origin: Don Eastlake
Status: Opened

Comments:

"What's about defaulting to base64 instead of none for the encoding."

Author Notes:

#99010201 - Add a ID attribute to Algorithm

Origin: Mark Linehan
Status: Superceded (98091001)

Comments:

"I suggest adding an "id" attribute to the Algorithm element, and adding an algref attribute to any element that contains an Algorithm. Purpose is to avoid repeating the same Algorithm text in many places."

Author Notes:

One approach to address #98091001

#99010202 - Provide a default digest algorithm

Origin: Mark Linehan
Status: Closed

Comments:

"I suggest that it would be helpful to have a way to declare a default or standard digest algorithm. Reason: in most cases, the same algorithm will be used throughout a document."

Author Notes:

Adequate optimization should enable use of shared definitions. In such circumstances, the overhead on a Resource element will be limited to an IDREF. At first, removing all algorithm references from the Resource element does not seem a good idea

#99010203 - Add a type to RecipientInfo and OriginatorInfo

Origin: Mark Linehan
Status: Closed

Comments:

"I suggest that OriginatorInfo and RecipientInfo have a "type" attribute for the same reason as the Attribute element: to identify the format of the ANY content."

Author Notes:

RecipientInfo and OriginatorInfo are expected to be a collection of "attributes". Therefore, it does not make sense to define a "type" attribute for these elements.

#99020301 - Adopt URL instead of URI

Origin: Joseph Reagle
Status: Opened

Comments:

"What's about adopting URLs instead of URNs. This will prevent registration requirements. At the least, the specification should allow URIs."

Author Notes:

#99021201 - Allow multiple signatures on a Manifest

Origin: IOTP WG
Status: Opened

Comments:

"Ability to have multiple signatures on a single Manifest and ability to adhere a recipient to a Signature."

To address these concerns, IOTP DTD proposes the following construct:

```
  <!-- DTD Extract -->

  <!ELEMENT Signature (Manifest, Value+)>
  <!ELEMENT Manifest ((Manifest,
    Resources, Attributes?, OriginatorInfo, RecipientInfo+
  )>
  <!ELEMENT RecipientInfo ANY>
  <!ATTLIST RecipientInfo
    SignatureValueRef    IDREF    #REQUIRED
    SignatureCertRef     IDREF    #IMPLIED
  >

  <!-- Signature Example -->

  <Signature>
    <Manifest>
      <Resources>
        ...
      </Resources>
      <OriginatorInfo>
        ...
      </OriginatorInfo>
      <RecipientInfo
        SignatureValueRef='sigv1'
        SignatureCertRef='cert1'>
        ...
      </RecipientInfo>
      <RecipientInfo
        SignatureValueRef='sigv2'
        SignatureCertRef='cert2'>
        ...
      </RecipientInfo>
        ...
    </Manifest>
    <Value id='sigv1'>
      aBcdsejhtksagnbf==
    </Value>
    <Value id='sigv2'>
      ehlekjrekkjrk==
    </Value>
  </Signature>
```

```
<Certificates>
  <Certificate id='cert1' ...>
    ...
  </Certificate>
  <Certificate id='cert2' ...>
    ...
  </Certificate>
</Certificate>
```

Author Notes:

Assuming that the benefit expected from this construct is to prevent replication of a large Manifest in multiple Signature elements, I would remind that this specification allows for shared Resources element

I can see at least three potential drawbacks with this construct:

- Privacy: A signature value cannot be verified unless all the RecipientInfo elements are preserved into the Manifest. In some circumstances, it may not be desirable to disclose the identity of the other recipients. Notice however that this construct does not preclude the creation of independent signature elements.

- Complexity: This construct is obviously more complicated than the one currently proposed. Dual forward references are not always easy to handle.

- Inconsistency: Applying multiple signatures to a single document usually implies that the application has adopted some secret-key authentication scheme. In such circumstances, the originator may be known by the recipients under different names or accounts. But, this construct does not allow replication of the OriginatorInfo element (per recipient basis), which is supposed to carry such pieces of information.

#99021202 - Enable shared digest algorithm definitions

Origin: IOTP WG
Status: Superceded by 98091002

Comments:

"Ability to share digest algorithms for signatures, digest, and canonicalization"

To address this concern, the IOTP DTD proposes to insert a collection of algorithm definitions into the signature Manifest and to use a linking mechanism for binding Resource definitions and signature algorithms to these shared definitions.

```
<!-- DTD Extract -->

<!ELEMENT Manifest (
          Algorithm+,
          Resource+,
          Attributes?,
          OriginatorInfo,
          RecipientInfo+
)>

<!ELEMENT Digest (#PCDATA)>
<!ATTLIST Digest
          DigestAlgorithmRef    IDREF    #REQUIRED
>
```

Author Notes:

Potential solution to optimizing the syntax. However, I would suggest replacing the collection of Algorithm elements by a DigestAlgorithms element. Also, I would limit this functionality to the Digest elements.

#99031601 - Remove criticality attribute on Attribute

Origin: Dave Solo, Eric Riscola
Status: Opened

Comments:

... IETF Meeting - following a presentation of criticality flag...

Dave: "it's a bad idea: experience in CMS was to remove it"

Eric: "I reinforce Dave: this bogs down every new attribute with a debate over 'criticality' ... PKIX and S/MIME show signs of precisely this kind of bloat."

Author Notes:

Tend to agree that criticality shall be a matter of the relying party and, therefore, a criticality attribute provided by the signer is not necessary in the syntax.

#99040101 - Remove Attributes element

Origin: Yoshiaki Kawatsura
Status: Closed

Comments:

"I do not understand why the Attributes element is needed."

Author Notes:

Collection elements such Attributes, Certificates, and Signatures has been proposed to facilitate DOM manipulation. For example, one may call some trust verification engine by passing the Certificates sub-tree. This construct enables containment of similar and related elements.

#99040102 - Allow attributes on Resource

Origin: Richard D. Brown
Status: Opened

Comments:

"It seems that allowing per Resource attributes may have interesting applications. For example, a rating standard could define a rating:Public attribute that could be associated directly with the Resource element. In such circumstances, a rating standard could almost suffice with the current DTD definitions."

Author Notes:

#99040103 - Add an CanonicalizerAlgorithm element

Origin: Richard D. Brown
Status: Opened

Comments:

"All signature schemes require canonicalization and digest of the Manifest. Thence, all the signature algorithms require at least a digest-algorithm parameter and this has lead to some inconsistencies such as requiring a digest algorithm for rsa encryption or two hash functions for HMAC.

It may be preferable to add a CanonicalAlgorithm element in the signature Manifest. This element will identify the digest/canonical algorithm to be used for computation of the fingerprint of the Signature Manifest."

```
<!ELEMENT Manifest(
          DigestAlgorithms,
          Resources,
          Attributes?,
          OriginatorInfo,
          RecipientInfo,
          KeyAgreementAlgorithm?,
          CanonicalizerAlgorithm,
          SignatureAlgorithm
)>
```

Author Notes:

#99040104 - Allow attributes on Package

Origin: Richard D. Brown
Status: Opened

Comments:

"Similarly to adding attributes to the Resource element. For example, such added functionality could be used for attaching an origin attribute to a package. This may be used for binding indirectly a detached-signature with an internal Package element."

Author Notes:

#99040105 - Change Attributes contents to ANY

Origin: Richard D. Brown
Status: Opened

Comments:

"Currently, the Attributes element consists of a collection of Attribute elements, which specify a type and contain an inner value element

An alternative a bit more opened would consist of defining Attributes as an element of ANY contents and use constructed attribute element."

```
<Resource href='http://www.w3c.org/doc.xml'>
  <Attributes>
    <rating:Audience value='all'/>
  </Attributes>
  ...
</Resource>
```

Author Notes:

One could argue that origin, digest, and the forth are also attributes of the resource and, therefore, could be sealed in the Attributes element. In fact, we could remove the Attributes element and define Resource as an element of ANY contents. But, if we were to do so, it would be impossible to enforce the presence of mandatory attribute elements such as resource locator and digest by means of the DTD.

If this suggestion were to be adopted, we may consider converting the ContentInfo element into a standard attribute.

#99040106 - Change ContentInfo contents to PCDATA

Origin: Richard D. Brown
Status: Opened

Comments:

"The specification proposes a ContentInfo element with a type and subtype attribute. The type attribute is defined as an URN. Unfortunately, URN specification does not allow the character "/" in the NSS. Thence, it is not possible to map directly existing MIME types into an URN without adopting adequate NSS encoding. For example, "application/msword" shall be encoded into something like "urn:MIME:application%2fmsword."

An alternative could be to define the ContentInfo element as follows:

```
<!ELEMENT ContentInfo (#PCDATA)>
<!ATTLIST ContentInfo
          type    CDATA    #IMPLIED 'urn:MIME'
>

<!-- Examples -->

<ContentInfo>
  application/msword
</ContentInfo>

<ContentInfo type='urn:IOTP'>
  OrderDescription
</ContentInfo>
```

Author Notes:

#99040701 - Allow Resource by value in Manifest

Origin: John Boyer, Richard D. Brown
Status: Opened

Comments:

"It may be worth considering the possibility to define a Resource either by means of a locator and a digest or by value."

We may consider the following definition:

```
<!ELEMENT Resource ((Locator, Digest) | Value)>
```

Author Notes:

Adopting this approach will disallow collapsing the Locator element into the Resource element. If the value is provided it does not make a lot of sense to specify a resource location.

But, Xlink specification seems to require the href attribute (not clear in the specification).

#99040801 - Add a Certificate Appendix to specification

Origin: Richard D. Brown
Status: Opened

Comments:

"It may be worth considering adding a certificate appendix that documents specifics for the different certificate types or certificate locators. As a matter of fact, a LDAP URL might be used but it is not sufficient for locating a particular certificate"

Author Notes:

#99040802 - Add a Security Appendix to specification

Origin: Richard D. Brown
Status: Opened

Comments:

"It may be worth considering adding a security appendix such as the one mandated by IETF."

Author Notes:

#99040803 - Add a Compliance Appendix to specification

Origin: Richard D. Brown
Status: Opened

Comments:

"It may be worth considering adding an appendix that defines compliance requirements."

Author Notes:

#99040804 - Segregate basic data types

Origin: Richard D. Brown
Status: Opened

Comments:

"It may be worth considering segregation of the data types that do not directly relate to XML-DSIG. For example, elements such as Integer, Float, and value might be replaced by making use of some DCD attribute, and others such as IssuerAndSerialNumber or Package might be temporarily moved into some Data DTD. These element definitions will be later superceded by definitions adopted by specialized DTDs.

Author Notes:

#99061601 - Replace dsig:eval by a PI

Origin: Richard D. Brown
Status: Opened

Comments:

As mentioned earlier by Milton M. Anderson (98111302), having the dsig:eval attribute directly in the element being authenticated may be unpractical if different signers were to use different hash algorithms.

A different approach consists of the definition of a Processing Instruction that specifies the algorithm to be used. This processing instruction could be inserted just before the element to be hashed and is not part of the hash value. Change in the value of the processing instruction does not invalidate previous signatures. Moreover, because the digest algorithm definitions are replicated into the Manifest, an attacker cannot attack the authentication scheme by tampering with the processing instruction.

```
Document>

<dsig:DigestAlgorithms>
  <dsig:Algorithm id='sha1' type='urn:nist-gov:sha1'/>
  <dsig:Algorithm id='md5' type='urn:rsasdi-com:md5'/>
</dsig:DigestAlgorithms>
```

```
<Element>

<?xmldsig eval='sha1 md5'?>
<HashedElement>
...
</HashedElement>

</Element>
</Document>
```

Author Notes:

#99061602 - Adopt RDF Data Model

Origin: XMLDISG'99 Participants
Status: Opened

Comments:

"XML-Signature should use the RDF data model but need not use the RDF serialization syntax. " In other words, the XML Signature syntax should consist of a static representation of a RDF schema. In the short term this static representation will translate into a DTD that will be replaced by the actual RDF schema as RDF awareness will develop in the Industry.

Author Notes:

#99061603 - Add a index per Category

Origin: Don Eastlake
Status: Opened

Comments:

"Categorize the comments and suggestions and attach an Index per category."

Author Notes:

2. Index per Reference Number

#98072901 - Syntax too verbose

#98091001 - Enable shared algorithm definitions

#98091002 - Promote compactness over reusability

#98111301 - Add a Resource criticality attribute

#98111302 - Remove dsig:eval global attribute
#98111303 - Add a logging attribute
#98111304 - Add a signature purpose attribute

#98111305 - Add a Nonce in the Manifest
#98112501 - Leverage DCD and other specifications
#98121501 - IssuerAndSerialNumber is too restrictive
#98122601 - Allow multiple Resource in Manifest
#98122602 - Add a base locator for HREFs
#98122603 - Rename the attribute dsig:eval
#98122604 - Default encoding attribute to base64
#99010201 - Add a ID attribute to Algorithm
#99010202 - Provide a default digest algorithm
#99010203 - Add a type to RecipientInfo and OriginatorInfo
#99020301 - Adopt URL instead of URI
#99021201 - Allow multiple signatures on a Manifest
#99021202 - Enable shared digest algorithm definitions
#99031601 - Remove criticality attribute on Attribute
#99040101 - Remove Attributes element
#99040102 - Allow attributes on Resource
#99040103 - Add an CanonicalizerAlgorithm element
#99040104 - Allow attributes on Package
#99040105 - Change Attributes contents to ANY
#99040106 - Change ContentInfo contents to PCDATA
#99040701 - Allow Resource by value in Manifest
#99040801 - Add a Certificate Appendix to specification
#99040802 - Add a Security Appendix to specification
#99040803 - Add a Compliance Appendix to specification
#99040804 - Segregate basic data types
#99061601 - Replace dsig:eval by a PI
#99061602 - Adopt RDF Data Model

3. Index per Status

Opened

#98091001 - Enable shared algorithm definitions
#98091002 - Promote compactness over reusability
#98111301 - Add a Resource criticality attribute
#98112501 - Leverage DCD and other specifications
#98121501 - IssuerAndSerialNumber is too restrictive
#98122602 - Add a base locator for HREFs
#98122603 - Rename the attribute dsig:eval
#98122604 - Default encoding attribute to base64
#99020301 - Adopt URL instead of URI
#99021201 - Allow multiple signatures on a Manifest
#99031601 - Remove criticality attribute on Attribute
#99040102 - Allow attributes on Resource
#99040103 - Add an CanonicalizerAlgorithm element
#99040104 - Allow attributes on Package
#99040105 - Change Attributes contents to ANY
#99040106 - Change ContentInfo contents to PCDATA
#99040701 - Allow Resource by value in Manifest
#99040801 - Add a Certificate Appendix to specification
#99040802 - Add a Security Appendix to specification
#99040803 - Add a Compliance Appendix to specification
#99040804 - Segregate basic data types
#99061601 - Replace dsig:eval by a PI
#99061602 - Adopt RDF Data Model
#99061603 - Add a index per Category

Closed

#98111302 - Remove dsig:eval global attribute
#98111303 - Add a logging attribute
#98111304 - Add a signature purpose attribute
#98111305 - Add a Nonce in the Manifest
#99010202 - Provide a default digest algorithm
#99010203 - Add a type to RecipientInfo and OriginatorInfo
#99040101 - Remove Attributes element

Superceded

#98072901 - Syntax too verbose
#99010201 - Add a ID attribute to Algorithm
#99021202 - Enable shared digest algorithm definitions

Adopted

#98122601 - Allow multiple Resource in Manifest