# Web Services Security
# SOAP Messages with Attachments (SwA) Profile 1.0

## OASIS Draft 7, 30 July 2004

**Document identifier:**
wss-swa-profile-1.0-draft-07

**Location:**
http://www.oasis-open.org/committees/documents.php?wg_abbrev=wss

**Editors:**
Frederick Hirsch, Nokia

**Contributors:**
Thomas DeMartini, ContentGuard
Dale Moberg, Cyclone Commerce
Michael McIntosh, IBM
Frederick Hirsch, Nokia
Jerry Schwarz, Oracle
Blake Dournaee, Sarvega, Inc.
Pete Wenzel, SeeBeyond

**Abstract:**
This specification defines how to use the OASIS Web Services Security: SOAP Message Security standard [WSS-Sec] with SOAP Messages with Attachments [SwA].

**Status:**
This is a Draft proposal and has no standing.

Committee members should submit comments and potential errata to the wss@lists.oasis-open.org list. Others should submit them to the wss-comment@lists.oasis-open.org list (to post, you must subscribe; to subscribe, send a message to wss-comment-subscribe@lists.oasis-open.org with "subscribe" in the body) or use other OASIS-supported means of submitting comments.

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights web page for the WSS TC (http://www.oasis-open.org/committees/wss/ipr.php).

# Table of Contents

# 1 Introduction

This document describes how to use the OASIS Web Services Security: SOAP Message Security standard [WSS-Sec] with SOAP Messages with Attachments [SwA]. More specifically, it describes how a web service consumer can secure SOAP attachments using SOAP Message Security for attachment integrity, confidentiality and origin authentication, and how a receiver may process such a message.

A broad range of industries - automotive, insurance, financial, pharmaceutical, medical, retail, etc - require that their application data be secured from its originator to its ultimate consumer. While some of this data will be XML, quite a lot of it will not be. In order for these industries to deploy web service solutions, they need an interoperable standard for end-to-end security for both their XML data and their non-XML data.

Profiling SwA security may help interoperability between the firms and trading partners using attachments to convey non-XML data that is not necessarily linked to the XML payload. Many industries, such as the insurance industry require free-format document exchange in conjunction with web services messages. This profile of SwA should be of value in these cases.

In addition, some content that could be conveyed as part of the SOAP body may be conveyed as an attachment due to its large size to reduce the impact on message and XML processing, and may be secured as described in this profile.

This profile is applicable to using SOAP Message Security in conjunction with SOAP Messages with Attachments (SwA). This means the scope is limited to SOAP 1.1, the scope of SwA.

The existence of this profile does not preclude using other mechanisms to secure attachments conveyed in conjunction with SOAP messages, including the use of XML security technologies at the application layer or the use of security for the XML Infoset before a serialization that uses attachment technology [MTOM]. The requirements in this profile only apply when securing SwA attachments explicitly according to this profile.

**Note that in this document, lists of processing steps are descriptive in that an implementation may use a different procedure as long as the result is the same.**

## 1.1 Notations and Terminology

This section specifies the notations, namespaces, and terminology used in this specification.

### 1.1.1 Notational Conventions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this specification are to be interpreted as described in IETF RFC 2119 [RFC2119].

```
Listings of productions or other normative code appear like this.
```

```
Example code listings appear like this.
```

**Note: Non-normative notes and explanations appear like this.**

When describing abstract data models, this specification uses the notational convention used by the XML Infoset. Specifically, abstract property names always appear in square brackets (e.g., [some property]).

When describing concrete XML schemas [XML-Schema], this specification uses the notational convention of OASIS Web Services Security: SOAP Message Security. Specifically, each member of an element's [children] or [attributes] property is described using an XPath-like [XPath] notation (e.g., /x:MyHeader/x:SomeProperty/@value1). The use of {any} indicates the presence of an element wildcard (<xs:any/>). The use of @{any} indicates the presence of an attribute wildcard (<xs:anyAttribute/>).

Commonly used security terms are defined in the Internet Security Glossary [SECGLO]. Readers are presumed to be familiar with the terms in this glossary as well as the definitions in the SOAP Message Security specification [WSS-Sec] .

### 1.1.2 Namespaces

Namespace URIs (of the general form "some-URI") represent application-dependent or context-dependent URIs as defined in RFC 2396 [URI]. This specification is designed to work with the SOAP 1.1 [SOAP11] message structure and message processing model, the version of SOAP supported by SOAP Messages with Attachments. The current SOAP 1.1 namespace URI is used herein to provide detailed examples.

The namespaces used in this document are shown in the following table (note that for brevity, the examples use the prefixes listed below but do *not* include the URIs – those listed below are assumed).

| Prefix | Namespace |
|---|---|
| S11 | http://schemas.xmlsoap.org/soap/envelope/ |
| wsse | http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd |
| wsu | http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-utility-1.0.xsd |
| wsswa | http://docs.oasis-open.org/wss/2004/XX/oasis-2004XX-wss-swa-profile-1.0.xsd |

The URLs provided for the *wsse* and *wsu* namespaces can be used to obtain the schema files.

**Note: When this document is finalized the wsswa URL will be updated, replacing XX values and possibly making other changes.**

## 1.1.3 Acronyms and Abbreviations

The following (non-normative) table defines acronyms and abbreviations for this document, beyond those defined in the SOAP Message Security standard.

| Term | Definition |
|---|---|
| CID | Content ID scheme for URLs. Refers to Multipart MIME body part, that includes both MIME headers and content for that part. [RFC2392] |
| SwA | SOAP Messages with Attachments [SwA] |

# 2 MIME Processing

This profile is concerned with the securing SOAP messages with attachments, attachments that are conveyed as MIME parts in a multi-part MIME message as outlined in SOAP Messages with Attachments. In effect this combines two processing layers, a SOAP messaging layer and a MIME wrapping. A SOAP sender effectively transmits a SOAP message and corresponding attachments by passing them to a MIME layer that serializes them. A SOAP receiver receives a message and attachments after the MIME layer processes the MIME serialization. This is important since certain aspects of the MIME processing may be changed at different intermediary transport nodes, yet remain transparent to the SOAP layer. For example, a MIME processing node may change the transfer encoding of a MIME part, transparently to the SOAP nodes. The MIME layer may translate to and from a transfer encoding upon serialization and de-serialization.

The importance to this profile is two-fold. First, it means that certain aspects of MIME processing, such as transfer encoding processing, are out of scope of the profile and do not need to be addressed. Secondly, it means that many of the MIME headers are also out of scope of the profile and the profile does not support integrity protection of these headers, since they are expected to change. If more security protection is required then it must occur at a protocol layer below the MIME layer, for example transport security (with the understanding that such security may not always apply end-end).

SOAP message security is intended to provide security at the SOAP messaging layer, including support for SOAP intermediaries. Thus this profile supports securing the attachment content, possibly including MIME headers that are associated directly with the content (such as Content-Type, Content-Length and other Content related MIME headers) and not MIME headers associated with MIME serialization. This simplifies the profile and also delineates the layering.

# 3 XML Attachments

A SOAP Messages with Attachments multi-part MIME structure contains a primary SOAP envelope in the root part and one or more attachments in additional MIME parts. Some of these attachments may be have a content type corresponding to XML, but do not contain the primary SOAP envelope to be processed.

Attachments associated with the SOAP body are targeted at the SOAP Ultimate Receiver along with the SOAP body, and may be processed at the application layer along with the body. This means that XML processing may not be required for such XML media type MIME attachments until application layer processing is performed. For this reason the SOAP message layer may not need to perform XML canonicalization or parsing for such attachments and SOAP Message layer security may treat these attachments as text.

Attachments might also be associated with SOAP headers and targeted toward specific SOAP intermediaries, or actors. For SOAP headers specific to an application the attachment content is processed at the application layer, logically after SOAP message processing is complete.

This profile assumes that SOAP attachments (not including the root part containing the primary SOAP envelope) need not be processed as XML at the SOAP messaging layer, so do not require SOAP canonicalization or XML parsing and may be treated as opaque data by the SOAP Message Security layer security processing. MIME part canonicalization (as described below) is required to enable effective SOAP Message Security signatures that include SOAP with Attachments.

In cases where this is not true, XML canonicalization transforms may be used in addition to the transforms outlined in this document. This is noted within the profile.

# 4 Securing SOAP With Attachments

Attachments may be associated with SOAP messages, as outlined in SOAP Messages with Attachments (SwA). This profile defines how such attachments may be secured for integrity and confidentiality using the OASIS Web Services Security: SOAP Message Security standard. This does not preclude using other techniques. The requirements in this profile only apply when securing SwA attachments explicitly according to this profile.

## 4.1 Referencing Attachments

SwA attachments may be identified with one of two MIME mechanisms. The first mechanism uses a CID scheme URL to refer to the attachment that has a Content-Id MIME header value corresponding to the URL scheme, as defined in [RFC 2392]. For example, a content id of “foo” may be specified in the MIME part with the MIME header “Content-Id: <foo>” and be referenced using a the CID Schema URL “cid:foo”.

The second mechanism is to use a URL to refer to an attachment containing a Content-Location MIME header. In this case the URL may require resolution to determine the referenced attachment [RFC2557].

Support for both mechanisms is included in this profile to enable full support of SwA which outlines the use of both mechanisms.

## 4.2 MIME Part Reference Transforms

By definition of RFC 2392, a URI reference to a MIME attachment includes the MIME headers associated with that attachment as well as the MIME part content [RFC2392]. Since there may be some confusion as to what is referenced, it is useful to clearly indicate what is included in the referenced attachment. In addition, some applications may wish to only encrypt or include the attachment content in a signature reference hash, and others may wish to include MIME headers and content.

For these reasons, this profile defines two transforms, allowing a clear and explicit statement of what is included in a MIME reference. These transforms are called “MIME Part Reference Transforms”.

### 4.2.1 Attachment-Content-Only Reference Transform

The Attachment-Content-Only transform indicates that only the content of a MIME part is referenced. This transform MUST be identified using the URI value: http://docs.oasis-open.org/wss/2004/XX/oasis-2004XX-wss-swa-profile-1.0#Attachment-Content-Only-Transform.

> **Note: When this document is finalized this URL will be updated, replacing XX values and possibly making other changes.**

### 4.2.2 Attachment-Complete Reference Transform

The Attachment-Complete transform indicates that both the content and selected headers of the MIME part are referenced. This transform MUST be identified using the URI value: http://docs.oasis-open.org/wss/2004/XX/oasis-2004XX-wss-swa-profile-1.0#Attachment-Complete-Transform.

> **Note: When this document is finalized this URL will be updated, replacing XX values and possibly making other changes.**

This transform specifies that in addition to the content the following MIME headers are to be included (when present):

- MIME-Version

- Content-Id
- Content-Location
- Content-Type
- Content-Description
- Content-Length

Other MIME headers associated with the MIME part serialization are not referenced by the transform and are not to be included in signature or encryption calculations.

## 4.3 Integrity and Data Origin Authentication

Integrity and data origin authentication may be provided for SwA attachments using XML Digital Signatures, as outlined in the SOAP Message Security standard as profiled in this document. This is useful independent of the content of the MIME part – for example, it is possible to sign a MIME part that already contains a signed object created by an application. It may still be sensible to sign such an attachment as part of SOAP Message security so that the receiving SOAP node may verify that all attachments are intact before delivering them to an application. A SOAP intermediary may also choose to perform this verification, even if the attachments are not otherwise processed by the intermediary.

### 4.3.1 MIME header canonicalization

Each of the MIME headers listed for the Attachment-Complete transform must be canonicalized as part of that transform processing, as outlined in this section. This means the following:

1. Only MIME headers that are explicitly present in the attachment part and are listed in the Attachment-Complete transform section are to be included by the transform.
2. The MIME headers must precede the MIME content.
3. The MIME headers must be returned in the order listed in the Attachment-Complete section (i.e. MIME-Version would be first).
4. All parameter names and media type/subtype values must be converted to lowercase. The case of parameter values must be left as-is, unless the parameter's specification indicates that the value is case-insensitive, in which case it must be converted to lowercase.
5. Each header must be terminated by a single CRLF pair, without any trailing whitespace.
6. Whitespace must be canonicalized by replacing multiple adjacent whitespace characters, including folding whitespace tokens, with a single space character (ascii value 32). Refer to RFC2822 for definitions [RFC2822].
7. The last header must be followed by a single CRLF and then the content of the MIME part.
8. All comments, as defined by RFC2822, must be removed [RFC2822].

### 4.3.2 MIME Content Canonicalization

Before including attachment content in a signature reference hash calculation, that MIME attachment may need to be MIME canonicalized. The exact details of MIME part canonicalization depend on the Content-Type of the MIME part. To quote the S/MIME specification (section 3.1.1 “Canonicalization”) which deals with this issue [RFC2633]:

> The exact details of canonicalization depend on the actual MIME type and subtype of an entity, and are not described here. Instead, the standard for the particular MIME type should be consulted. For example, canonicalization of type text/plain is different from

canonicalization of audio/basic. Other than text types, most types have only one representation regardless of computing platform or environment which can be considered their canonical representation.

MIME types are registered. This registration includes a section on "Canonicalization and Format Requirements" [RFC2048] and requires each MIME type to have a canonical representation.

The MIME “text” type canonical form is defined in the MIME conformance specification (See “Canonical Encoding Model”) [RFC2049]. Important aspects of “text” media type canonicalization include line ending normalization to <CR><LF> and ensuring that the charset is a registered charset (see RFC 2633 section “Canonicalization”). [RFC2633, CHARSETS, RFC2045].

MIME attachment parts (other than the part containing the primary SOAP envelope) that contain XML do NOT require XML Canonicalization according to this profile, given the rationale in the previous section on XML attachments. These parts MUST be MIME canonicalized according the MIME “text” part requirements. MIME part canonicalization must be performed before signature hash generation or verification is performed. Signature validation requires an identical hash of content requiring consistent MIME part content.

Note that in cases where XML processing of an XML attachment is anticipated, perhaps by a SOAP intermediary, an XML canonicalization transform may also be specified as a <ds:Reference> transform, in addition to the MIME Part Reference Transform. Additional transforms MUST follow the MIME Part Reference transform.

### 4.3.3 Protecting against attachment insertion threat

Including an attachment in a signature calculation enables a receiver to detect modification of that attachment. Including all attachments in a signature calculation, by providing a <ds:Reference> for each, protects against the threat of attachment removal. This does not protect against insertion of a new attachment.

The simplest protection against attachment insertion is for the receiver to know that all attachments should be included in a signature calculation – unreferenced attachments are then an indication of an attachment insertion attack.

Such information may be communicated in or out of band. Definition of these approaches is out of the scope of this profile.

### 4.3.4 Processing Rules for Attachment Signing

The processing rule for signing is modified based on the SOAP Message Security rules.

After determining which attachments are to be included as references in a signature, create a <ds:Signature> element in a <wsse:Security> header block targeted at the recipient, including a <ds:Reference> for each attachment to be protected by the signature. Additional <ds:Reference> elements may refer to content in the SOAP envelope to be included in the signature.

For each attachment Reference, perform the following steps:

1. MIME Part Canonicalize the content of the attachment, as appropriate to the MIME type of the part.
2. If MIME headers are to be included in the signature, MIME part canonicalize the headers listed in this profile as outlined above.
3. Determine the URL to be used to reference the part. Use of a CID scheme reference is recommended when possible, since this avoids the need for reference resolution. The <ds:Reference> URL attribute value should be set to the URL determined in this step.
4. Include a <ds:Transforms> element in the <ds:Reference>. This <ds:Transforms> element MUST include a <ds:Transform> element with the Algorithm attribute having the URL value specified in this

profile - either Attachment-Complete or Attachment-Content-Only, depending on what is to be included in the hash calculation. This MUST be the first transform listed. Additional transforms, such as an XML canonicalization transform, MAY be included as required.

5. Extract the appropriate portion of the MIME part consistent with the selected transform.
6. Create the <ds:Reference> hash value as outlined in the W3C XML Digital Signature Recommendation.

### 4.3.5 Processing Rules for Attachment Signature Verification

Signature verification is performed as outlined in SOAP Message Security and the XML Digital Signature Recommendation, with the following considerations for SwA attachments.

To verify <ds:Reference> hashes for SwA attachments, the following steps must be performed for each reference to an attachment:

1. Find the attachment corresponding to the <ds:Reference> URL attribute value. This value should correspond to the Content-Id for the attachment or resolve to a URL corresponding to a Content-Location header [SwA].
2. MIME Part Canonicalize the content of the attachment, as appropriate to the MIME type of the part.
3. If MIME headers were included in the signature, canonicalize the headers listed in this profile as outlined above.
4. Extract the appropriate portion of the MIME part according to the MIME Part Signature Transform value.
5. Calculate the reference hash and verify the reference.

### 4.3.6 Example Signed Message

```
Content-Type: multipart/related; boundary=”arggh” type=text/xml
--arggh
Content-Type: text/xml
<S11:Envelope xmlns:S11="..." xmlns:wsse="..." xmlns:wsu="..."
xmlns:ds="..." xmlns:xenc="...">
  <S11:Header>
    <wsse:Security>
      <ds:Signature>
         <ds:Reference URI="cid:bar">
            <ds:Transforms>
              <ds:Transform Algorithm="http://docs.oasis-
open.org/wss/2004/XX/oasis-2004XX-wss-swa-profile-1.0#Attachment-Content-
Only-Transform"/>
            </ds:Transforms>
        <ds:DigestMethod
Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
          <ds:DigestValue>j6lwx3rvEPO0vKtMup4NbeVu8nk=</ds:DigestValue>
        </ds:Reference>
    </ds:Signature>
   </wsse:Security>
  </S11:Header>
  <S11:Body>
   some items
  </S11:Body>
</S11:Envelope>
--arggh
Content-Type: image/png
Content-Id: <bar>
Content-Transfer-Encoding: base64
the image
```

## 4.4 Encryption

A SwA attachment may be encrypted for confidentiality protection, protecting either the MIME part content including selected MIME headers, or only the MIME part content.

This is done using XML Encryption to encrypt the attachment, placing the resulting cipher text in the updated attachment body replacing the original content, and placing a new <xenc:EncryptedData> element in the <wsse:Security> header. An <xenc:CipherReference> is used to link the cipher data to the <xenc:EncryptedData> element.

No <xenc:ReferenceList> element is placed in the <wsse:Security> header, since the <xenc:EncryptedData> element is present in the header, eliminating the need for a reference. The SOAP Message Security standard recommends the use of <xenc:ReferenceList>, but this is only necessary when the <xenc:EncryptedData> element is not present in the <wsse:Security> header.

**Note: The same CID is used to refer to the attachment before encryption and after. This avoids the need to rewrite references to the attachment, avoiding issues related to generating unique CIDs and relating to preserving the correspondence to the original WSDL definition.**

### 4.4.1 MIME Part CipherReference

This profile requires that <xenc:EncryptedData> elements corresponding to encrypted SwA attachments use a <xenc:CipherReference> to refer to the cipher text, to be conveyed in the attachment. Upon encryption the MIME part attachment content is replaced with the encoded cipher text.

The <xenc:CipherReference> must have a <ds:Transforms> child element, and this element must have a <ds:Transform> child. The <ds:Transform> Algorithm attribute must have a URI value specifying the Content-Only MIME Part Reference Transform. This transform explicitly indicates that when dereferencing the CID referring to the MIME part, only the MIME part content is to be used as the cipher value

### 4.4.2 Encryption Processing Rules

The order of the following steps is not normative, although the result should be the same as if this order were followed.

1. Encrypt the attachment part using XML Encryption, according to the rules of XML Encryption. Encrypt either the attachment including content and selected MIME headers or only the attachment content.
2. Set the <xenc:EncryptedData> Type attribute value to a URI that specifies adherence to his profile and that specifies what was encrypted (MIME content or entire MIME part including headers). The following URIs MUST be used for this purpose:
   - Content Only: http://docs.oasis-open.org/wss/2004/XX/oasis-2004XX-wss-swa-profile-1.0#Attachment-Content-Only.
   - Content and headers: http://docs.oasis-open.org/wss/2004/XX/oasis-2004XX-wss-swa-profile-1.0#Attachment-Complete

   **Note: When this document is finalized these URLs will be updated, replacing XX values and possibly making other changes. Note that these URLs should match the related transforms apart from -Transform.**
3. Set the <xenc:EncryptedData> MimeType attribute to match the attachment MIME part Content-Type header before encryption.
4. Set the <xenc:EncryptedData> <xenc:CipherReference> to the same reference URL for the attachment that was used before encryption . This is either a CID scheme URL referring to the

attachment part Content-ID or a URL that resolves to the attachment part Content-Location header value. Ensure this MIME header is in the part conveying the cipher data after encryption.

5. Include the MIME Part CipherReference Transform in the <xenc:CipherReference> as outlined above.
6. Prepend the <xenc:EncryptedData> element to the <wsse:Security> SOAP header block. Do NOT add a <xenc:ReferenceList> element to the SOAP header block (even though recommended by SOAP Message Security).
7. Update the attachment MIME part, replacing the original content with the cipher text generated by the XML Encryption step.
8. Update the attachment MIME part header MIME Content-Type and Content-Length appropriate to the cipher data.

### 4.4.3 Decryption Processing Rules

The <xenc:CipherReference> URL MUST be a URL that refers to the MIME part containing the cipher text, and must also correspond to the reference value of the original attachment that was encrypted. This may either be a CID scheme URL or a URL that resolves to a Content_Location header for the MIME part.

Decryption may be initiated upon locating the <xenc:EncryptedData> element in the <wsse:Security> header.

The following decryption steps must be performed so that the result is as if they were performed in this order:

1. Extract the cipher text from the attachment referenced by the <xenc:CipherReference> URL attribute. The MIME Part CipherReference Transform defined in this profile indicates that the MIME part content is extracted as an octet stream and used as the cipher data.
2. Decrypt the cipher text using the information present in the appropriate <xenc:EncryptedData> element and possibly other out of band information, according to the XML Encryption Standard.
3. If the <xenc:EncryptedData>Type attribute indicates that selected MIME headers were encrypted, then those MIME headers must be replaced by the result of decryption, as well as the MIME part content.
4. If the <xenc:EncryptedData>Type attribute indicates that only the content of the MIME part was encrypted, then the cipher text content of the attachment part must be replaced by the result of decryption. In this case the MIME part Content-Type header value MUST be replaced by the <xenc:EncryptedData> MimeType attribute value.

### 4.4.4 Example

```
Content-Type: multipart/related; boundary=”arggh” type=text/xml
--arggh
Content-Type: text/xml
<S11:Envelope xmlns:S11="..." xmlns:wsse="..." xmlns:wsu="..."
xmlns:ds="..." xmlns:xenc="...">
  <S11:Header>
    <wsse:Security>
       <xenc:EncryptedData Id="foo_Part" Type=”http://docs.oasis-
open.org/wss/2004/XX/oasis-2004XX-wss-swa-profile-1.0#Attachment-
Complete” MimeType=”image/jpeg” Encoding=”base64”>
         <ds:KeyInfo>
           <ds:KeyName>someName</ds:KeyName>
         </ds:KeyInfo>
        <xenc:CipherData>
          <xenc:CipherReference URI=”cid:foo>
            <ds:Transforms>
```

```
                <ds:Transform Algorithm="http://docs.oasis-
open.org/wss/2004/XX/oasis-2004XX-wss-swa-profile-
1.0#ContentOnlyCipherText"/>
            </ds:Transforms>
          </xenc:CipherReference>
        </xenc:CipherData>
      </xenc:EncryptedData>
    </wsse:Security>
  </S11:Header>
  <S11:Body>
some information
  </S11:Body>
</S11:Envelope>
--arggh
Content-Type: something
Content-Id: <foo>
Content-Transfer-Encoding: base64
DEADBEEF
```

## 4.5 Signing and Encryption

When portions of content are both signed and encrypted, there is possible confusion as to whether encrypted content need first be decrypted before signature verification. This confusion can occur when the order of operations is not clear [DecryptT]. This problem may be avoided with SOAP Message Security for SwA attachments when attachments and corresponding signatures and encryptions are targeted for a single SOAP recipient (actor). The SOAP Message Security standard explicitly states that there may not be two <wsse:Security> headers targeted at the same actor, nor may there be two headers without a designated actor. In this case the SOAP Message Security and SwA profile processing rules may eliminate ambiguity since each signing or encryption produces an element in the <wsse:Security> header, and these elements are ordered. (Signing produces <ds:Signature> elements and encryption produces <xenc:EncryptedData> elements).

If an application produces different <wsse:Security> headers targeted at different recipients, these are processed independently by the recipients. Thus there is no need to correlate activities between distinct headers – the order is inherent in the SOAP node model represented by the distinct actors.

# 5 References


**[CHARSETS]** Character sets assigned by IANA. See ftp://ftp.isi.edu/in-notes/iana/assignments/character-sets.

**[DecryptT]** M. Hughes et al, “Decryption Transform for XML Signature”, W3C Recommendation 10 December 2002 http://www.w3.org/TR/xmlenc-decrypt/.

**[MTOM]** *Work in Progress – subject to change.* SOAP Message Transmission Optimization Mechanism, W3C Working Draft 8 June 2004, http://www.w3.org/TR/soap12-mtom/.

**[RFC2045]** Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies, http://www.ietf.org/rfc/rfc2045.txt.

**[RFC2048]** Multipurpose Internet Mail Extensions (MIME) Part Four: Registration Procedures, http://www.ietf.org/rfc/rfc2048.txt.

**[RFC2049]** Multipurpose Internet Mail Extensions(MIME) Part Five: Conformance Criteria and Examples, http://www.ietf.org/rfc/rfc2049.txt.

**RFC2119]** S. Bradner, Key words for use in RFCs to Indicate Requirement Levels, IETF RFC 2119, March 1997, http://www.ietf.org/rfc/rfc2119.txt.

**[RFC2392]** E. Levinson, *Content-ID and Message-ID Uniform Resource Locators,* IETF RFC 2392*,* http://www.ietf.org/rfc/rfc2392.txt

**[RFC2557]** MIME Encapsulation of Aggregate Documents, such as HTML (MHTML), IETF RFC 2557, March 1999, http://www.ietf.org/rfc/rfc2557.txt.

**[RFC2633]** Ramsdell B., “S/MIME Version 3 Message Specification”, Standards Track RFC 2633, June 1999. http://www.ietf.org/rfc/rfc2633.txt

**[RFC2822]** Internet Message Format, http://www.ietf.org/rfc/rfc2822.txt.

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

**[SOAP11]** W3C Note, "SOAP: Simple Object Access Protocol 1.1," 08 May 2000.

**[SwA]** W3C Note, “SOAP Messages with Attachments”, 11 December 2000, http://www.w3.org/TR/2000/NOTE-SOAP-attachments-20001211 .

**[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, http://www.ietf.org/rfc/rfc2396.txt.

**[WS-I-AP]** *Work in progress – subject to change.* Attachments Profile Version 1.0, Board Approval Draft, 2004-06-11, http://www.ws-i.org/Profiles/AttachmentsProfile-1.0.html

**[WSS-Sec]** A. Nadalin et al., Web Services Security: SOAP Message Security 1.0 (WS-Security 2004), OASIS Standard 200401, March 2004, http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-soap-message-security-1.0.pdf

**[XML-Schema]** W3C Recommendation, "XML Schema Part 1: Structures,"2 May 2001, http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/.
W3C Recommendation, "XML Schema Part 2: Datatypes," 2 May 2001, http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/.

**[XPath]** W3C Recommendation, "XML Path Language", 16 November 1999, http://www.w3.org/TR/xpath.

# A. Acknowledgments

The editors would like to acknowledge the contributions of the OASIS WSS Technical Committee, whose voting members at the time of publication were:

- TBD

# B. Revision History

| Rev | Date | By Whom | What |
|---|---|---|---|
| 1 | 05/25/04 | Frederick Hirsch | Initial version, put draft proposal into profile format. |
| 2 | 05/26/04 | Frederick Hirsch | Editorial and namespace suggestions from Michael McIntosh. Added rationale for SwA support to introduction. Completely rewrote processing rules for encryption and decryption. |
| 3 | 05/28/04 | Frederick Hirsch | Rewrote signature section, fixed cid references and Content-Ids, added examples. |
| 4 | 06/12/04 | Frederick Hirsch | Added Decrypt Transform section, added All-Attachments-Complete transform, changed MIME reference to v3, minor editorial changes. |
| 5 | 07/07/04 | Frederick Hirsch | Removed Decrypt transform material, since it is generally not needed and the approach had issues. Reorganized signatures section. Eliminated incorrect All-Attachments-Complete transform and replaced with discussion of attachment insertion threat. Clarified that only one wsse:Security header per actor/role minimizes signing,encryption confusion possibility. Added section for MIME Part CipherReference Transform. Editorial fixes. |
| 6 | 07/14/04 | Frederick Hirsch | ** Allow use of Content-Location, consistent with SwA.<br>** Proposed update to signature Content-Transfer-Encoding processing rules. Needs review.<br>Revised section on MIME canonicalization, added section on XML attachments. Only support SOAP 1.1. Clarified introduction. Added MTOM and additional MIME references.(Issue 297 should be closed – removed section on decryption transform and updated section on signing and encryption in version 5) Issue 303 – fixed, (see 3.2.4 example), Issue 306 – revised section on MIME canonicalization to close this issue. Issue 307 – revised to refer to SOAP 1.1 only, added section on XML attachments, defined MTOM and added reference. Editorial fixes. |

| Rev | Date | By Whom | What |
|---|---|---|---|
| 7 | 07/30/04 | Frederick Hirsch | Incorporate feedback from WS-I BSP. Limit MIME headers included in signature or encryption to those listed in profile. Clarify MIME layering approach. Remove processing rules associated with Content-Transfer-Encoding. Editorial correction throughout document to allow both CID and Content-Location references to attachments. Editorial revision to pull attachment referencing and reference transforms into section applicable to both signatures and encryption. Incorporated feedback from Pete Wenzel and Toshihiro Nishimura – separate URL for transform and encryption type, used Content-Only reference transform for Cipherdata as well. |

# C. 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.