SAML V2.0 Metadata Profile for Algorithm Support Version 1.0

Committee Draft 01
29 June 2010

Specification URIs:

This Version:

http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-metadata-algsupport-cd-01.html

http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-metadata-algsupport-cd-01.odt (Authoritative)

http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-metadata-algsupport-cd-01.pdf

Previous Version:

None

Latest Version:

http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-metadata-algsupport.html

http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-metadata-algsupport.odt (Authoritative)

http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-metadata-algsupport.pdf

Latest Approved Version:

http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-metadata-algsupport-cd-01.html

http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-metadata-algsupport-cd-01.odt (Authoritative)

http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-metadata-algsupport-cd-01.pdf

Technical Committee:

OASIS Security Services TC

Chair:

Thomas Hardjono, M.I.T.
Nate Kingenstein, Internet2

Editor:

Scott Cantor, Internet2

Related Work:

This specification defines an extension for use with SAML V2.0 Metadata [SAML2Meta].

Declared XML Namespace(s):

urn:oasis:names:tc:SAML:metadata:algsupport

Abstract:

The SAML V2.0 Metadata specification [SAML2Meta] includes an element allowing entities to describe the XML Encryption [XMLEnc] algorithms they support. This specification defines metadata extension elements to enable entities to describe the XML Signature [XMLSig] algorithms they support, and a profile for using both elements to enable better algorithm agility for profiles that rely on metadata.

Status

This document was last revised or approved by the SSTC on the above date. The level of approval is also listed above. Check the current location noted above for possible later revisions of this document. This document is updated periodically on no particular schedule.

TC members should send comments on this specification to the TC's email list. Others should send comments to the TC by using the "Send A Comment" button on the TC's web page at http://www.oasis-open.org/committees/security.

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 IPR section of the TC web page (http://www.oasis-open.org/committees/security/ipr.php).

The non-normative errata page for this specification is located at http://www.oasis-open.org/committees/security.

Notices

Table of Contents

1 Introduction 5
1.1 Notation 5
1.2 Normative References 6
1.3 Non-Normative References 6
2 SAML V2.0 Metadata Profile for Algorithm Support 7
2.1 Required Information 7
2.2 Profile Description 7
2.3 Expression of Encryption Capabilities 7
2.4 Expression of Signature Capabilities 8
2.4.1 Element <alg:DigestMethod> 8
2.4.2 Element <alg:SigningMethod> 8
2.5 Metadata Consumers 9
2.6 Security Considerations 9
2.7 Example 10
3 Conformance 11
3.1 SAML V2.0 Metadata Profile for Algorithm Support Version 1.0 11
Appendix A. Acknowledgements 12
Appendix B. Revision History 13

1 Introduction

The SAML V2.0 Metadata specification [SAML2Meta] includes an <md:EncryptionMethod> element intended to communicate the XML Encryption [XMLEnc] algorithms supported for use with the key described by a containing <md:KeyDescriptor>.element. The use of this element is not completely defined by the original specification, and there is no comparable support for communicating the XML Signature [XMLSig] algorithms supported by an entity. This profile addresses both considerations to improve algorithm agility and interoperability for deployments that make use of metadata.

There are more general standards for the description of security requirements of communicating endpoints, such as [WS-SecPol]. This specification is not intended as a replacement for such mechanisms, but is directed at systems with fewer requirements that are already designed around SAML V2.0 Metadata.

1.1 Notation

This specification uses normative text.

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

> …they MUST only be used where it is actually required for interoperation or to limit behavior which has potential for causing harm (e.g., limiting retransmissions)…

These keywords are thus capitalized when used to unambiguously specify requirements over protocol and application features and behavior that affect the interoperability and security of implementations. When these words are not capitalized, they are meant in their natural-language sense.

```
Listings of XML schemas appear like this.
```

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

Conventional XML namespace prefixes are used throughout the listings in this specification to stand for their respective namespaces as follows, whether or not a namespace declaration is present in the example:

Prefix	XML Namespace	Comments
saml:	urn:oasis:names:tc:SAML:2.0:assertion	This is the SAML V2.0 assertion namespace [SAML2Core].
samlp:	urn:oasis:names:tc:SAML:2.0:protocol	This is the SAML V2.0 protocol namespace [SAML2Core].
md:	urn:oasis:names:tc:SAML:2.0:metadata	This is the SAML V2.0 metadata namespace [SAML2Meta].
alg:	urn:oasis:names:tc:SAML:metadata:algsupport	This is the SAML V2.0 metadata extension namespace defined by this document and its accompanying schema [AlgSup-XSD].
xenc:	http://www.w3.org/2001/04/xmlenc#	This is the XML Encryption namespace [XMLEnc].
xsd:	http://www.w3.org/2001/XMLSchema	This namespace is defined in the W3C XML Schema specification [Schema1]. In schema listings, this is the default namespace and no prefix is shown.

This specification uses the following typographical conventions in text: `<SAMLElement>`, `<ns:ForeignElement>`, `Attribute`, **Datatype**, `OtherCode`.

1.2 Normative References

[AlgSup-XSD] OASIS Working Draft, *Metadata Extension Schema for SAML V2.0 Metadata Profile for Algorithm Support Version 1.0*, June 2010. http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-metadata-algsupport.xsd

[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

[SAML2Core] OASIS Standard, *Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0*, March 2005. http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf

[SAML2Errata] OASIS Approved Errata, *SAML V2.0 Errata*, October 2009. http://docs.oasis-open.org/security/saml/v2.0/sstc-saml-approved-errata-2.0.pdf

[SAML2Meta] OASIS Standard, *Metadata for the OASIS Security Assertion Markup Language (SAML) V2.0*, March 2005. http://docs.oasis-open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf

[XMLEnc] D. Eastlake et al. *XML Encryption Syntax and Processing*. World Wide Web Consortium Recommendation. http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/

[Schema1] H. S. Thompson et al. XML Schema Part 1: Structures. World Wide Web Consortium Recommendation, May 2001. http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/

[Schema2] Paul V. Biron, Ashok Malhotra. *XML Schema Part 2: Datatypes*. World Wide Web Consortium Recommendation, May 2001. http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/

[XMLSig] D. Eastlake et al. *XML-Signature Syntax and Processing, Second Edition*. World Wide Web Consortium Recommendation, June 2008. http://www.w3.org/TR/xmldsig-core/

1.3 Non-Normative References

[RFC4051] IETF RFC 4051, *Additional XML Security Uniform Resource Identifiers,* April 2005. http://www.ietf.org/rfc/rfc4051.txt

[WS-SecPol] OASIS Standard, *WS-SecurityPolicy 1.3*, February 2009. http://docs.oasis-open.org/ws-sx/ws-securitypolicy/v1.3/os/ws-securitypolicy-1.3-spec-os.pdf

2 SAML V2.0 Metadata Profile for Algorithm Support

2.1 Required Information

Identification: `urn:oasis:names:tc:SAML:metadata:algsupport`

Contact information: security-services-comment@lists.oasis-open.org

Description: Given below.

Updates: None.

2.2 Profile Description

One of the interoperability challenges in large-scale, and long-term, SAML deployments is the selection of XML Signature [XMLSig] and XML Encryption [XMLEnc] algorithms at runtime when communicating with peer entities. In particular, accounting for software limitations that prevent support of newer algorithms, while supporting those algorithms where possible to gradually strengthen systems, is difficult to manage without knowledge of a peer's capabilities. This profile makes use of SAML metadata to enable deployments to document their algorithm capabilities and preferences. It also allows for future expansion to address the interoperability requirements of more complex algorithms.

This profile provides guidance on the use of the `<md:EncryptionMethod>` element defined in the SAML V2.0 Metadata specification [SAML2Meta], and defines extension elements, `<alg:SigningMethod>` and `<alg:DigestMethod>`, to address comparable requirements related to XML Signature usage.

2.3 Expression of Encryption Capabilities

The SAML V2.0 Metadata specification [SAML2Meta] permits zero or more `<md:EncryptionMethod>` elements to appear inside a `<md:KeyDescriptor>` element. This profile provides guidance for the use of this element only in enclosing elements whose `use` attribute is omitted or set to `"encryption"`.

In the common case that a `<md:KeyDescriptor>` element contains an asymmetric encryption key, an `<md:EncryptionMethod>` element SHOULD be present for each of a Block or Stream Encryption, and a Key Transport or Key Agreement algorithm. The Key Transport or Key Agreement algorithm(s) listed MUST be compatible with the associated encryption key.

If the `<md:KeyDescriptor>` element contains or identifies by reference a symmetric key (e.g., a name referring to a shared master secret or password), then an `<md:EncryptionMethod>` element SHOULD be present for a Block or Stream Encryption algorithm, and MAY be present for other algorithm types such as Symmetric Key Wrap or Key Derivation.

Per [XMLEnc], the `<md:EncryptionMethod>` element MUST contain an `Algorithm` attribute containing the identifier for the algorithm defined for use with the specification. If the algorithm permits varying key sizes, the element MAY contain an `<xenc:KeySize>` element defining a key size for the algorithm that the entity will accept. If the algorithm definition includes the specification of additional public content that the party performing encryption needs, that content MAY also be present.

If multiple `<md:EncryptionMethod>` elements identifying algorithms of the same general type are present, they MUST be listed in order of preference by the entity.

2.4 Expression of Signature Capabilities

This profile defines a pair of extension elements for the expression of an entity's capability to verify digests and signatures with particular algorithms. While not strictly meant as an expression of policy, it is a natural assumption that a peer stating support for particular algorithms requires their use.

An entity SHOULD include one or more `<alg:DigestMethod>` and `<alg:SigningMethod>` elements in its metadata by means of the `<md:Extensions>` element in its `<md:EntityDescriptor>` element, and/or in its roles (elements whose type is based on **md:RoleDescriptorType**).

If a signature algorithm permits varying key sizes, the `<alg:SigningMethod>` element MAY contain `MinKeySize` and/or `MaxKeySize` attributes bounding the key size for the algorithm that the entity supports. If the algorithm definition includes the specification of additional public content that the party creating a signature or digest needs, that content MAY also be present.

If multiple elements of the same type are present, they MUST be listed in order of preference by the entity.

2.4.1 Element <alg:DigestMethod>

The `<alg:DigestMethod>` element describes a Message Digest algorithm. It contains the following attribute:

`Algorithm` [Required]

Identifies the algorithm by means of the URL defined for its use with the XML Signature specification [XMLSig].

This element also permits the use of arbitrary elements defined in any namespace.

The schema for the `<alg:DigestMethod>` element, and its corresponding **alg:DigestMethodType** complex type, is as follows:

```
<element name="DigestMethod" type="alg:DigestMethodType"/>
<complexType name="DigestMethodType">
   <sequence>
      <any namespace="##any" minOccurs="0" maxOccurs="unbounded"/>
   </sequence>
   <attribute name="Algorithm" type="anyURI" use="required"/>
</complexType>
```

2.4.2 Element <alg:SigningMethod>

The `<alg:SigningMethod>` element describes a Signature or Message Authentication Code algorithm. It contains the following attributes:

`Algorithm` [Required]

Identifies the algorithm by means of the URL defined for its use with the XML Signature specification [XMLSig].

`MinKeySize` [Optional]

The smallest key size, in bits, that the entity supports in conjunction with the algorithm. If omitted, no minimum is implied.

`MaxKeySize` [Optional]

The largest key size, in bits, that the entity supports in conjunction with the algorithm. If omitted, no maximum is implied.

This element also permits the use of arbitrary elements defined in any namespace.

The schema for the <alg:SigningMethod> element, and its corresponding **alg:SigningMethodType** complex type, is as follows:

```
<element name="SigningMethod" type="alg:SigningMethodType"/>
<complexType name="SigningMethodType">
   <sequence>
      <any namespace="##any" minOccurs="0" maxOccurs="unbounded"/>
   </sequence>
   <attribute name="Algorithm" type="anyURI" use="required"/>
   <attribute name="MinKeySize" type="positiveInteger"/>
   <attribute name="MaxKeySize" type="positiveInteger"/>
</complexType>
```

2.5 Metadata Consumers

A consumer of metadata that wishes to perform XML Signature or XML Encryption operations with knowledge of the peer entity (this is not always true of signatures) MUST consult the peer's metadata to determine the intersection of the algorithms, key sizes, and other parameters as defined by particular algorithms that it supports and that the peer entity supports.

The elements describing this support in metadata SHOULD be consulted in order, and the metadata consumer SHOULD select the first algorithm encountered that it supports for use with a particular entity (subject to local policy).

With respect to use of XML Signature, the presence of any <alg:DigestMethod> and <alg:SigningMethod> elements at the level of a role element MUST take precedence over any such elements at the level of of an <md:EntityDescriptor> element, and the two sets are not combined if both are present.

In the absence of an element describing support for a particular algorithm type (e.g., no <alg:DigestMethod> elements), the metadata consumer is free to select any algorithm that it supports. The absence of metadata therefore implies no information, rather than lack of support.

2.6 Security Considerations

The use of metadata as a means of "negotiating" the algorithms to use exposes both parties to attacks tradtionally associated with such mechanisms, such as step-down attacks in which the metadata is compromised to influence the selection of a weaker algorithm than the parties might otherwise support.

The exchange and verification of metadata should always be subject to appropriate security controls to mitigate this threat, and entities should always be prepared to reject the use of algorithms that they deem insufficiently secure.

2.7 Example

The example presented shows a partial metadata instance for a service provider that supports (as a relying party) a number of newer/stronger signature and digest algorithms defined in [RFC4051]. It also specifies support for encryption via two AES variants using an RSA key as a transport.

```
<EntityDescriptor xmlns="urn:oasis:names:tc:SAML:2.0:metadata"
      xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
      xmlns:alg="urn:oasis:names:tc:SAML:metadata:algsupport"
      entityID="https://serviceprovider.example.com/SAML">
   <Extensions>
      <alg:DigestMethod
         Algorithm="http://www.w3.org/2001/04/xmldsig-more#sha384"/>
      <alg:DigestMethod
         Algorithm="http://www.w3.org/2001/04/xmldsig-more#sha256"/>
      <alg:SignatureMethod MinKeySize="256" MaxKeySize="511"
         Algorithm="http://www.w3.org/2001/04/xmldsig-more#ecdsa-sha256"/>
      <alg:SignatureMethod MinKeySize="2048" MaxKeySize="4096"
         Algorithm="http://www.w3.org/2001/04/xmldsig-more#rsa-sha256"/>
   </Extensions>
   <SPSSODescriptor
         protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
      <KeyDescriptor>
         <ds:KeyInfo>...RSA key elided...</ds:KeyInfo>
         <EncryptionMethod
            Algorithm="http://www.w3.org/2001/04/xmlenc#aes128-cbc"/>
         <EncryptionMethod
            Algorithm="http://www.w3.org/2001/04/xmlenc#aes256-cbc"/>
         <EncryptionMethod
            Algorithm="http://www.w3.org/2001/04/xmlenc#rsa-oaep-mgf1p"/>
      </KeyDescriptor>
   ...
   </SPSSODescriptor>
...
</EntityDescriptor>
```

3 Conformance

3.1 SAML V2.0 Metadata Profile for Algorithm Support Version 1.0

A metadata producer conforms to this profile if it has the ability to produce metadata in accordance with sections 2.3 and 2.4.

A metadata consumer conforms to this profile if it can consume extended metadata produced in accordance with sections 2.3 and 2.4 and conforms to the normative statements in section 2.5.

Appendix A. Acknowledgements

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

- Rob Philpott, EMC Corporation
- John Bradley, Individual
- Scott Cantor, Internet2
- Nate Klingenstein, Internet2
- Thomas Hardjono, M.I.T.
- Anthony Nadalin, Microsoft Corporation
- Thinh Nguyenphu, Nokia Siemens Networks Gmb
- Phil Hunt, Oracle Corporation
- Ari Kermaier, Oracle Corporation
- Hal Lockhart, Oracle Corporation
- Emily Xu, Oracle Corporation
- Anil Saldhana, Red Hat
- David Staggs, Veterans Health Administration

Appendix B. Revision History

- Draft 01, first working draft.
- Committee Draft 01, CD edits.