



2 SAML V2.0 Metadata Interoperability Profile

3 Working Draft 01, 9 August 2008

4 **Specification URIs:**

5 TBD

6 **Technical Committee:**

7 OASIS Security Services TC

8 **Chair(s):**

9 Hal Lockhart, BEA Systems, Inc.

10 Brian Campbell, Ping Identity Corporation

11 **Editors:**

12 Scott Cantor, Internet2

13 **Abstract:**

14 This profile describes a set of rules for SAML metadata producers and consumers to follow such
15 that federated relationships can be interoperably provisioned, and controlled at runtime in a
16 secure, understandable, and self-contained fashion.

17 **Status**

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

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

24 For information on whether any patents have been disclosed that may be essential to
25 implementing this specification, and any offers of patent licensing terms, please refer to the IPR
26 section of the TC web page (<http://www.oasis-open.org/committees/security/ipr.php>).

27 The non-normative errata page for this specification is located at [http://www.oasis-](http://www.oasis-open.org/committees/security)
28 [open.org/committees/security](http://www.oasis-open.org/committees/security).

29 Notices

30 Copyright © OASIS Open 2008. All Rights Reserved.

31 All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual
32 Property Rights Policy (the "OASIS IPR Policy"). The full Policy may be found at the OASIS website.

33 This document and translations of it may be copied and furnished to others, and derivative works that
34 comment on or otherwise explain it or assist in its implementation may be prepared, copied, published,
35 and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice
36 and this section are included on all such copies and derivative works. However, this document itself may
37 not be modified in any way, including by removing the copyright notice or references to OASIS, except as
38 needed for the purpose of developing any document or deliverable produced by an OASIS Technical
39 Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be
40 followed) or as required to translate it into languages other than English.

41 The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors
42 or assigns.

43 This document and the information contained herein is provided on an "AS IS" basis and OASIS
44 DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY
45 WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY
46 OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A
47 PARTICULAR PURPOSE.

48 OASIS requests that any OASIS Party or any other party that believes it has patent claims that would
49 necessarily be infringed by implementations of this OASIS Committee Specification or OASIS Standard, to
50 notify OASIS TC Administrator and provide an indication of its willingness to grant patent licenses to such
51 patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced
52 this specification.

53 OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of any
54 patent claims that would necessarily be infringed by implementations of this specification by a patent
55 holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR
56 Mode of the OASIS Technical Committee that produced this specification. OASIS may include such
57 claims on its website, but disclaims any obligation to do so.

58 OASIS takes no position regarding the validity or scope of any intellectual property or other rights that
59 might be claimed to pertain to the implementation or use of the technology described in this document or
60 the extent to which any license under such rights might or might not be available; neither does it represent
61 that it has made any effort to identify any such rights. Information on OASIS' procedures with respect to
62 rights in any document or deliverable produced by an OASIS Technical Committee can be found on the
63 OASIS website. Copies of claims of rights made available for publication and any assurances of licenses
64 to be made available, or the result of an attempt made to obtain a general license or permission for the
65 use of such proprietary rights by implementers or users of this OASIS Committee Specification or OASIS
66 Standard, can be obtained from the OASIS TC Administrator. OASIS makes no representation that any
67 information or list of intellectual property rights will at any time be complete, or that any claims in such list
68 are, in fact, Essential Claims.

69 The name "OASIS" is a trademark of OASIS, the owner and developer of this specification, and should be
70 used only to refer to the organization and its official outputs. OASIS welcomes reference to, and
71 implementation and use of, specifications, while reserving the right to enforce its marks against
72 misleading uses. Please see <http://www.oasis-open.org/who/trademark.php> for above guidance.

73 **Table of Contents**

74	1 Introduction.....	4
75	1.1 Notation.....	4
76	1.2 Normative References.....	5
77	1.3 Non-Normative References.....	5
78	1.4 Conformance.....	5
79	1.4.1 SAML V2.0 Metadata Interoperability Profile.....	5
80	2 SAML V2.0 Metadata Interoperability Profile.....	6
81	2.1 Required Information.....	6
82	2.2 Profile Overview.....	6
83	2.3 Metadata Acceptance.....	6
84	2.4 Other Assumptions.....	6
85	2.5 Metadata Producer Requirements.....	6
86	2.5.1 Key Representation.....	7
87	2.6 Metadata Consumer Requirements.....	7
88	2.6.1 Key Processing.....	8
89	2.7 Security Considerations.....	8
90	Appendix A. Acknowledgements.....	10
91	Appendix B. Revision History.....	11
92		

93 1 Introduction

94 The SAML V2.0 metadata specification [SAML2Meta] defines an XML schema and a set of basic
95 processing rules intended to facilitate the use of SAML profiles, and generally any profile or specification
96 involving SAML. Practical experience has shown that the most complex aspects of implementing most
97 SAML profiles, and obtaining interoperability between such implementations, are in the areas of
98 provisioning federated relationships between deployments, and establishing the validity of cryptographic
99 signatures and handshakes. Because the metadata specification was largely intended to solve those
100 exact problems, additional profiling is needed to improve and clarify the use of metadata in addressing
101 those aspects of deployment.

102 This profile is the product of the implementation experience of several SAML solution providers and has
103 been widely deployed and successfully used in furtherance of the goal of scaling deployment beyond small
104 numbers into the hundreds and thousands of sites, without sacrificing security.

105 Experience has shown that the most frustrating part of using SAML (and many similar technologies) is that
106 products approach the use of cryptography and trust in wildly inconsistent ways, and often the libraries
107 that such products depend on do the same in their own domains. Key management is hard, and often
108 relies on complicated tools with cryptic output. Standards only help insofar as they can be understood and
109 widely implemented; this has generally not occurred above a basic level of cryptographic correctness. A
110 formal PKI is a tremendously complex, and some would say intractable, goal; it could be argued that
111 SAML itself is a reaction to this. Often, the security of deployments is based on a presumption that
112 required practices like revocation checking are being performed, when in fact they are not.

113 The purpose of this profile is to guarantee that in a correct implementation, all security considerations not
114 deriving from the particular cryptography used (i.e. algorithm strength, key sizes) can be isolated to
115 metadata exchange and acceptance, and not affect the runtime processing of messages. If a deployment
116 can be shown to rely solely on metadata to derive trust, it can be reasoned about in a much simpler way,
117 and the security exposures can be well understood.

118 Furthermore, this profile accomplishes a number of practical goals:

- 119 • simplifying ordinary implementations and deployments
- 120 • reducing the technical foundation required to understand and use implementations
- 121 • scaling the provisioning of federated relationships (via processing of metadata batches)
- 122 • radically simplifying interactions between existing federated deployments (i.e. interfederation)

123 Most importantly, these goals can be accomplished without sacrificing security. Too often, the reaction to
124 security complexity is to produce competing approaches that start by rejecting the notion that a substantial
125 degree of security is achievable in the general case.

126 Another benefit of this profile is to produce a greater awareness of the importance of securing the
127 exchange of metadata. Deployers have sometimes tended to ignore this issue by falling back on the
128 assumption that the underlying PKI would provide the real security of the system, resulting in other
129 exposures due to insecure provisioning of other important information.

130 1.1 Notation

131 This specification uses normative text.

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

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

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

140 Listings of XML schemas appear like this.

141
142 Example code listings appear like this.

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

Prefix	XML Namespace	Comments
saml:	urn:oasis:names:tc:SAML:2.0:assertion	This is the SAML V2.0 assertion namespace defined in the SAML V2.0 core specification [SAML2Core].
md:	urn:oasis:names:tc:SAML:2.0:metadata	This is the SAML V2.0 metadata namespace defined in the SAML V2.0 metadata specification [SAML2Meta].
ds:	http://www.w3.org/2000/09/xmldsig#	This is the XML Signature namespace [XMLSig].
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.
xsi:	http://www.w3.org/2001/XMLSchema-instance	This is the XML Schema namespace for schema-related markup that appears in XML instances [Schema1].

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

148 1.2 Normative References

- 149 **[RFC2119]** S. Bradner. *Key words for use in RFCs to Indicate Requirement Levels*. IETF
150 RFC 2119, March 1997. <http://www.ietf.org/rfc/rfc2119.txt>.
- 151 **[RFC3280]** R. Housley, et al. *Internet X.509 Public Key Infrastructure Certificate and
152 Certificate Revocation List (CRL) Profile*. IETF RFC 3280, April 2002.
153 <http://www.ietf.org/rfc/rfc3280.txt>.
- 154 **[SAML2Bind]** S. Cantor et al. *Bindings for the OASIS Security Assertion Markup Language
155 (SAML) V2.0*. OASIS Standard, March 2005. Document ID saml-bindings-2.0-os.
156 See <http://docs.oasis-open.org/security/saml/v2.0/saml-bindings-2.0-os.pdf>.
- 157 **[SAML2Core]** S. Cantor et al. *Assertions and Protocols for the OASIS Security Assertion
158 Markup Language (SAML) V2.0*. OASIS Standard, March 2005. Document ID
159 saml-core-2.0-os. See [http://docs.oasis-open.org/security/saml/v2.0/saml-
160 core-2.0-os.pdf](http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf).
- 161 **[SAML2Meta]** S. Cantor et al. *Metadata for the OASIS Security Assertion Markup Language
162 (SAML) V2.0*. OASIS Standard, March 2005. Document ID saml-metadata-2.0-
163 os. See <http://docs.oasis-open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf>.
- 164 **[SAML2Prof]** S. Cantor et al. *Profiles for the OASIS Security Assertion Markup Language
165 (SAML) V2.0*. OASIS Standard, March 2005. Document ID saml-profiles-2.0-os.
166 See <http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf>.

- 167 **[Schema1]** H. S. Thompson et al. *XML Schema Part 1: Structures*. World Wide Web
168 Consortium Recommendation, May 2001. See [http://www.w3.org/TR/2001/REC-](http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/)
169 [xmlschema-1-20010502/](http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/). Note that this specification normatively references
170 [Schema2], listed below.
- 171 **[Schema2]** Paul V. Biron, Ashok Malhotra. *XML Schema Part 2: Datatypes*. World Wide Web
172 Consortium Recommendation, May 2001. See [http://www.w3.org/TR/2001/REC-](http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/)
173 [xmlschema-2-20010502/](http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/).
- 174 **[XMLSig]** D. Eastlake et al. *XML-Signature Syntax and Processing*. World Wide Web
175 Consortium Recommendation, February 2002. See
176 <http://www.w3.org/TR/xmlsig-core/>.

177 **1.3 Non-Normative References**

- 178 **[RFC4346]** T. Dierks, E. Rescorla. *The Transport Layer Security (TLS) Protocol Version 1.1*.
179 IETF RFC 4346, April 2006. <http://www.ietf.org/rfc/rfc4346.txt>.

180 **1.4 Conformance**

181 **1.4.1 SAML V2.0 Metadata Interoperability Profile**

182 A metadata producer conforms to this profile if it can produce metadata consistent with the normative text
183 in section 2.5.

184 A metadata consumer conforms to this profile if it can "accept" metadata in accordance with section 2.3
185 and process it consistent with the normative text in section 2.6.

2 SAML V2.0 Metadata Interoperability Profile

2.1 Required Information

Identification: urn:oasis:names:tc:SAML:2.0:profiles:metadata-iop

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

Description: Given below.

Updates: None.

2.2 Profile Overview

The SAML V2.0 profiles [SAML2Prof] and metadata [SAML2Meta] specifications, and subsequent profiles within OASIS and in other communities, describe the use of SAML metadata as a means of describing deployment capabilities and providing partners with information about endpoints, keys, profile support, processing requirements, etc.

This profile extends these practices by guaranteeing that a given metadata document will be consistently interpreted by any conforming implementation of higher level profiles. To this end, it requires that metadata be usable as a self-contained vehicle for communicating trust such that a user of a conforming implementation can be guaranteed that any and all rules for processing digital signatures, encrypted XML, and transport layer cryptography (e.g. TLS/SSL [RFC4346]) can be derived from the metadata alone, with no additional trust requirements imposed.

This profile requires that all runtime decisions are made solely on the basis of key comparisons, and not on any traditionally certificate-influenced basis. A signed metadata file following this specification is semantically equivalent to a PKI, hence there is little value in the additional layer of complexity provided by certificate validation as in [RFC3280]. Operational experience also shows that managing signed metadata is easier than managing a PKI of the corresponding size and scale.

2.3 Metadata Acceptance

This profile does not seek to constrain the method by which metadata is published or acquired, but only its content and interpretation. It is assumed that, subject to the security and deployment requirements of the participants, some means of exchanging metadata exists that results in the "acceptance" of metadata by a consumer. Acceptance in this profile is defined as an explicit treatment of everything in the metadata as "true", for the purposes of the metadata consumer's operational behavior.

In other words, this profile does not define how metadata is exchanged or how and why it is trusted, but rather assumes that it is exchanged and trusted, and proceeds from that starting point. Dynamic exchange (as described in [SAML2Meta]), manual exchange, the aggregation and signing of metadata by third parties, or any other mechanism, can be used in conjunction with this profile.

The rest of this profile deals with the requirements for producing metadata that will be accepted, and a consumer's obligations having accepted it.

2.4 Other Assumptions

An additional assumption is necessitated by the inability of SAML metadata to express authentication requirements of back-channel communications between SAML entities, such as the SAML SOAP binding [SAML2Bind]. In lieu of extending metadata to capture such requirements, this profile assumes that such communications are secured by means of some combination of TLS/SSL and digital signing. If this assumption does not hold, this profile might need supplementing in some scenarios.

226 2.5 Metadata Producer Requirements

227 A producer of metadata that adheres to this profile may be an actual participant in a SAML (or other)
228 profile, or an aggregator of metadata describing many such participants. In either case, the content of the
229 metadata itself is independent of its source and MUST stand alone as a description of the cryptographic
230 requirements for securely communicating with the entity (or entities) described therein, to the extent that
231 the constructs of the SAML V2.0 metadata specification [SAML2Meta] can express these requirements.

232 Subject to the constraints of the exchange mechanisms in use, a conforming metadata instance MAY be
233 rooted by either an `<md:EntityDescriptor>` or `<md:EntitiesDescriptor>` element. This profile
234 further applies to any `<md:RoleDescriptor>` element (or any derived elements and types) that may be
235 included.

236 Within the context of a particular role (and the protocols it supports, as expressed in its
237 `protocolSupportEnumeration` attribute), any and all cryptographic keys that are known to be valid at
238 the time of metadata production MUST appear, each in its own `<md:KeyDescriptor>` element, with the
239 appropriate `use` attribute (see section 2.4.1.1 of [SAML2Meta]). This includes not only signing and
240 encryption keys, but also any keys used to establish mutual authentication with technologies such as TLS/
241 SSL.

242 Signing or transport authentication keys intended for future use MAY be included as a means of preparing
243 for migration from an older to a newer key (i.e. key rollover). Once an allowable period of time has elapsed
244 (with this period dependent on deployment-specific policies), the older key can be removed, completing
245 the change.

246 Expired keys (those not in use anymore by an entity, for reasons other than compromise) SHOULD be
247 removed once the rollover process to a new key (or keys) has been completed.

248 Compromised keys MUST be removed from an entity's metadata. The metadata producer MUST NOT
249 rely on the metadata consumer utilizing online or offline mechanisms for verifying the validity of a key (e.g.
250 X.509 revocation lists, OCSP, etc.). The exact time by which a compromise is reflected in metadata is left
251 to the requirements of the parties involved, the metadata's validity period (as defined by a `validUntil` or
252 `cacheDuration` attribute), and the exchange mechanism in use.

253 2.5.1 Key Representation

254 Each key included in a metadata role MUST be placed within its own `<md:KeyDescriptor>` element
255 and expressed using the `<ds:KeyInfo>` element within. One or more of the following representations
256 within a `<ds:KeyInfo>` element MUST be present:

- 257 • `<ds:KeyValue>`
- 258 • `<ds:X509Certificate>` (child element of `<ds:X509Data>`)

259 In the case of the latter, only a single certificate is permitted. If both forms are used, then they MUST
260 represent the same key.

261 Any other representation in the form of a `<ds:KeyInfo>` child element (such as `<ds:KeyName>`,
262 `<ds:X509SubjectName>`, `<ds:X509IssuerSerial>`, etc.) MAY appear, but MUST NOT be required
263 in order to identify the key (they are hints only).

264 In the case of an X.509 certificate, there are no requirements as to the content of the certificate apart from
265 the requirement that it contain the appropriate public key. Specifically, the certificate MAY be expired, not
266 yet valid, carry critical or non-critical extensions or usage flags, and contain any subject or issuer.

267 **2.6 Metadata Consumer Requirements**

268 A metadata consumer MUST have the ability to fully provision and configure itself based on the content of
269 a metadata instance that it has accepted (see section 2.3), within the constraints of the information
270 represented by the SAML V2.0 metadata specification [SAML2Meta] and any profiles that make use of it.
271 A consumer need not provision policy that is outside the scope of metadata, but MUST have the ability to
272 interoperate with the entities described by a metadata instance that it accepts, constrained by whatever
273 default policies it applies.

274 Subject to the constraints of the exchange mechanism(s) in use, a metadata consumer MUST be able to
275 process instances rooted with either an `<md:EntityDescriptor>` or `<md:EntitiesDescriptor>`
276 element. When processing an `<md:EntitiesDescriptor>` element, each `<md:EntityDescriptor>`
277 element contained within it MUST be processed in accordance with this profile (subject to their validity).

278 **2.6.1 Key Processing**

279 Each key expressed by a `<md:KeyDescriptor>` element within a particular role MUST be accepted
280 when processing messages or assertions in the context of that role. Specifically, any signatures or
281 transport communications (e.g. TLS/SSL sessions) verifiable with a signing key MUST be accepted, and
282 any encryption keys found may be used to encrypt messages or assertions to the containing entity.

283 Subsequent to accepting a metadata instance, a consumer MUST NOT apply additional criteria of any
284 kind on the acceptance, or validity, of the keys found within it or their use at runtime. Specifically,
285 consumers SHALL NOT apply any online or offline techniques including, but not limited to, X.509 path
286 validation or revocation lists, OCSP responders, etc.

287 The following key representations within a `<ds:KeyInfo>` element MUST be supported:

- 288 • `<ds:KeyValue>`
- 289 • `<ds:X509Certificate>` (child element of `<ds:X509Data>`)

290 In the case of the latter, a metadata consumer MUST extract the public key found in the certificate and
291 MUST NOT honor, interpret, or make use of any of the information found in the certificate other than as an
292 aid in identifying the appropriate key to try (based, for example, on information found at runtime in an XML
293 digital signature's `<ds:KeyInfo>` element or the certificate presented by a transport peer).

294 A metadata consumer, when implementing authentication of a transport peer via TLS/SSL, MAY retain the
295 checking of server certificate names (e.g. subject cn or subjectAltName) in accordance with [RFC3280],
296 but even then it MUST accept a properly named certificate that contains a public key that corresponds to a
297 valid key found in that peer's metadata, even if the exact certificate presented is not found in that
298 metadata.

299 **2.7 Security Considerations**

300 A number of important exposures arise from the reliance on metadata alone to control runtime trust
301 decisions.

302 Metadata becomes a critical tool for the revocation of compromised sites and keys, and all of the standard
303 practices in the use of tools like CRLs become relevant to the consumption of metadata. The specification
304 has the mechanisms to address these issues, but they have to be used. Specifically, metadata obtained
305 via an insecure transport should be both signed, and should expire, so that consumers are forced to
306 refresh it often enough to limit the damage from compromised information.

307 In addition, distributing signed metadata without an expiration over an untrusted channel (e.g. posting it on
308 a public web site) creates an exposure. An attacker can corrupt the channel and substitute an old
309 metadata file containing a compromised key and proceed to use that key together with other attacks to

310 impersonate a site. Repeatedly expiring and reissuing the metadata limits the window of exposure, just as
311 a CRL does.

312 A broad set of concerns arises in the dynamic exchange of metadata self-published by a site. In such
313 cases, it may seem untenable to trust someone to properly identify their own key, and of course it may be.
314 Rather than constraining the acceptance of that key, this profiles relies on securing the exchange and
315 acceptance of the metadata. Traditional PKI protections can be applied to that document and/or its
316 exchange, subsequently leveraging that protection to establish trust in the key within the metadata.

317 For example, when using the Well Known Location resolution profile [SAML2Meta], a producer may use
318 an X.509 certificate to sign the metadata. This certificate can be bound to the metadata through its subject
319 or subjectAltName (which might contain a SAML entityID). This ensures the appropriate key/name binding
320 for the signature.

321 **Appendix A. Acknowledgements**

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

- 324 • TBD

325 The editor would also like to acknowledge the following contributors:

- 326 • Walter Hoehn, University of Memphis
- 327 • Chad LaJoie, SWITCH
- 328 • Ian Young, EDINA, University of Edinburgh

329 **Appendix B. Revision History**

- 330 ● Draft 01.