



SAML V2.0 Metadata Interoperability Profile Version 1.0

Committee Draft 01

10 March 2009

Specification URIs:

This Version:

<http://docs.oasis-open.org/security/saml/Post2.0/sstc-metadata-iop-cd-01.html>
<http://docs.oasis-open.org/security/saml/Post2.0/sstc-metadata-iop-cd-01.odt> (Authoritative)
<http://docs.oasis-open.org/security/saml/Post2.0/sstc-metadata-iop-cd-01.pdf>

Previous Version:

None

Latest Version:

<http://docs.oasis-open.org/security/saml/Post2.0/sstc-metadata-iop.html>
<http://docs.oasis-open.org/security/saml/Post2.0/sstc-metadata-iop.odt>
<http://docs.oasis-open.org/security/saml/Post2.0/sstc-metadata-iop.pdf>

Technical Committee:

OASIS Security Services TC

Chair(s):

Hal Lockhart, BEA Systems, Inc.
Brian Campbell, Ping Identity Corporation

Editors:

Scott Cantor, Internet2

Abstract:

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

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>.

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

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

40 Notices

41 Copyright © OASIS Open 2009. All Rights Reserved.

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

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

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

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

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

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

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

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

84 Table of Contents

85	1 Introduction.....	5
86	1.1 Notation.....	6
87	1.2 Normative References.....	7
88	1.3 Non-Normative References.....	7
89	2 SAML V2.0 Metadata Interoperability Profile.....	8
90	2.1 Required Information.....	8
91	2.2 Profile Overview.....	8
92	2.3 Metadata Exchange and Acceptance.....	8
93	2.4 Implementation Constraints.....	9
94	2.4.1 Peer Authentication.....	9
95	2.5 Metadata Producer Requirements.....	9
96	2.5.1 Key Representation.....	9
97	2.6 Metadata Consumer Requirements.....	10
98	2.6.1 Key Processing.....	10
99	2.7 Security Considerations.....	11
100	3 Conformance.....	12
101	3.0.1 SAML V2.0 Metadata Interoperability Profile.....	12
102	Appendix A. Acknowledgements.....	13
103	Appendix B. Revision History.....	14
104		

1 Introduction

105

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

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

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

126 Of course, it is the case that some deployments, at least to date, have overcome some or all of these
127 problems. They may have a mature PKI, possibly one that existed long before their use of SAML, or they
128 may require such a PKI for other purposes. In such cases, it is obviously less beneficial to deploy a
129 second trust infrastructure based on SAML metadata. Another factor in this profile's usefulness is the
130 relationship between SAML and the other security technologies involved in a deployment; if the use of
131 SAML is subordinated to a secondary role, this profile may be less applicable.

132 The purpose of this profile is to guarantee that in a correct implementation, all security considerations not
133 deriving from the particular cryptography used (i.e., algorithm strength, key sizes) can be isolated to
134 metadata exchange and acceptance, and not affect the runtime processing of messages. In other words,
135 given a metadata instance and presuming that it is successfully processed and has not been updated or
136 superseded, it must be possible with no other information supplied to determine whether a given
137 credential (e.g., a key or certificate) will be accepted by an implementation when used to secure a SAML
138 protocol or assertion.

139 If an implementation can be shown to rely solely on the acceptance of metadata to derive trust, it can be
140 reasoned about in a much simpler way, and the security exposures can be well understood. Furthermore,
141 this profile accomplishes a number of additional practical goals:

- 142 • simplifying ordinary implementations and deployments
- 143 • reducing the technical foundation required to understand and use implementations
- 144 • scaling the provisioning of federated relationships (via processing of metadata batches)
- 145 • facilitating the use of XML encryption without dependency on weaker methods for establishing
146 knowledge of public keys (e.g., guessing based on TLS server certificates)
- 147 • radically simplifying interactions between existing federated deployments (i.e. interfederation)

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

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

155 Finally, note that, in addition to SAML V2.0 itself, this profile is applicable to any set of use cases
156 supported by SAML metadata, including SAML V1.x profiles (as in [SAML1Meta]) and any other
157 specifications that may profile SAML metadata.

158 1.1 Notation

159 This specification uses normative text.

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

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

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

168 Listings of XML schemas appear like this.

169 Example code listings appear like this.

171 Conventional XML namespace prefixes are used throughout the listings in this specification to stand for
172 their respective namespaces as follows, whether or not a namespace declaration is present in the
173 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].

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

176 1.2 Normative References

- 177 [RFC2119] S. Bradner. *Key words for use in RFCs to Indicate Requirement Levels*. IETF
178 RFC 2119, March 1997. <http://www.ietf.org/rfc/rfc2119.txt>.
- 179 [RFC2818] E. Rescorla. *HTTP Over TLS*. IETF RFC 2818, May 2000.
180 <http://www.ietf.org/rfc/rfc2818.txt>.
- 181 [SAML2Bind] OASIS Standard, *Bindings for the OASIS Security Assertion Markup Language*
182 *(SAML) V2.0*. March 2005. [http://docs.oasis-open.org/security/saml/v2.0/saml-](http://docs.oasis-open.org/security/saml/v2.0/saml-bindings-2.0-os.pdf)
183 [bindings-2.0-os.pdf](http://docs.oasis-open.org/security/saml/v2.0/saml-bindings-2.0-os.pdf).
- 184 [SAML2Core] OASIS Standard, *Assertions and Protocols for the OASIS Security Assertion*
185 *Markup Language (SAML) V2.0*. March 2005. [http://docs.oasis-open.org/security/](http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf)
186 [saml/v2.0/saml-core-2.0-os.pdf](http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf).
- 187 [SAML2Errata] OASIS Standard Errata, *SAML V2.0 Errata*. August 2007. [http://docs.oasis-](http://docs.oasis-open.org/security/saml/v2.0/sstc-saml-approved-errata-2.0.pdf)
188 [open.org/security/saml/v2.0/sstc-saml-approved-errata-2.0.pdf](http://docs.oasis-open.org/security/saml/v2.0/sstc-saml-approved-errata-2.0.pdf).
- 189 [SAML2Meta] OASIS Standard, *Metadata for the OASIS Security Assertion Markup Language*
190 *(SAML) V2.0*. March 2005. [http://docs.oasis-open.org/security/saml/v2.0/saml-](http://docs.oasis-open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf)
191 [metadata-2.0-os.pdf](http://docs.oasis-open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf).
- 192 [SAML2Prof] OASIS Standard, *Profiles for the OASIS Security Assertion Markup Language*
193 *(SAML) V2.0*. March 2005. [http://docs.oasis-open.org/security/saml/v2.0/saml-](http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf)
194 [profiles-2.0-os.pdf](http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf).
- 195 [Schema1] H. S. Thompson et al. *XML Schema Part 1: Structures*. World Wide Web
196 Consortium Recommendation, May 2001. See [http://www.w3.org/TR/2001/REC-](http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/)
197 [xmlschema-1-20010502/](http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/). Note that this specification normatively references
198 [Schema2], listed below.
- 199 [Schema2] Paul V. Biron, Ashok Malhotra. *XML Schema Part 2: Datatypes*. World Wide Web
200 Consortium Recommendation, May 2001. See [http://www.w3.org/TR/2001/REC-](http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/)
201 [xmlschema-2-20010502/](http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/).
- 202 [XMLSig] D. Eastlake et al. *XML-Signature Syntax and Processing*. World Wide Web
203 Consortium Recommendation, February 2002. See
204 <http://www.w3.org/TR/xmlsig-core/>.

205 1.3 Non-Normative References

- 206 [RFC4346] T. Dierks, E. Rescorla. *The Transport Layer Security (TLS) Protocol Version 1.1*.
207 IETF RFC 4346, April 2006. <http://www.ietf.org/rfc/rfc4346.txt>.
- 208 [RFC5280] D. Cooper, et al. *Internet X.509 Public Key Infrastructure Certificate and*
209 *Certificate Revocation List (CRL) Profile*. IETF RFC 5280, May 2008.
210 <http://www.ietf.org/rfc/rfc5280.txt>.
- 211 [SAML1Meta] OASIS Standard, *Metadata Profile for the OASIS Security Assertion Markup*
212 *Language (SAML) V1.x*. November 2007. [http://docs.oasis-](http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml1x-metadata-os.pdf)
213 [open.org/security/saml/Post2.0/sstc-saml1x-metadata-os.pdf](http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml1x-metadata-os.pdf)

214 **2 SAML V2.0 Metadata Interoperability Profile**

215 **2.1 Required Information**

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

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

218 **Description:** Given below.

219 **Updates:** None.

220 **2.2 Profile Overview**

221 The SAML V2.0 profiles [SAML2Prof] and metadata [SAML2Meta] specifications, and subsequent profiles
222 within OASIS and in other communities (e.g., [SAML1Meta]), describe the use of SAML metadata as a
223 means of describing deployment capabilities and providing partners with information about endpoints,
224 keys, profile support, processing requirements, etc.

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

231 This profile requires that all runtime decisions are made solely on the basis of key comparisons, and not
232 on any traditionally certificate-influenced basis. A signed metadata file conforming to this specification is
233 semantically equivalent to an X.509-based public key infrastructure (PKI), hence there is little value in the
234 additional layer of complexity provided by certificate validation as specified in [RFC5280]. Operational
235 experience also shows that managing signed metadata is easier than managing a PKI of the
236 corresponding size and scale.

237 **2.3 Metadata Exchange and Acceptance**

238 This profile does not constrain the method(s) by which metadata is published or acquired, but only its
239 content and interpretation. It is assumed that, subject to the security and deployment requirements of the
240 participants, some means of exchanging metadata exists that results in the "acceptance" of metadata by
241 a consumer. Acceptance in this profile is defined as an explicit treatment of everything in the metadata as
242 "true", for the purposes of the metadata consumer's operational behavior. The truth of a given set of
243 metadata is of course contingent upon the metadata not being superseded by newer metadata, which
244 may conflict with, and therefore render obsolete, the earlier information.

245 In other words, this profile does not define how (or how often) metadata is exchanged or how and why it
246 is trusted, but rather assumes that it is exchanged and trusted, and proceeds from that starting point.
247 Dynamic exchange (as described in [SAML2Meta]), manual exchange, the aggregation and signing of
248 metadata by third parties, or any other mechanism, can be used in conjunction with this profile. Note that
249 verification of metadata signatures, if applicable, is considered to be part of this prerequisite step.

250 The rest of this profile deals with the requirements for producing metadata, and a conformant consumer's
251 obligations having accepted it.

252 Finally, note that accepting metadata does not imply that a relying party will interoperate with a specific
253 asserting party; it implies only that if it does so, it does so in a predictable fashion based on the metadata
254 it accepts about that party.

255 2.4 Implementation Constraints

256 2.4.1 Peer Authentication

257 An additional constraint is necessitated by the inability of SAML metadata to express the authentication
258 requirements of back-channel communications between SAML-using entities, such as via the SAML
259 SOAP binding [SAML2Bind]. In lieu of extending metadata to capture such requirements, this profile
260 assumes that such communications are secured by means of some combination of TLS/SSL and digital
261 signing. If this assumption cannot be made, this profile might need to be supplemented in such scenarios.

262 2.5 Metadata Producer Requirements

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

268 Subject to any constraints of the exchange mechanisms in use, a conforming metadata instance may be
269 rooted by either an `<md:EntityDescriptor>` or `<md:EntitiesDescriptor>` element. Any
270 `<md:RoleDescriptor>` element (or any derived element or type) appearing in the metadata instance
271 MUST conform to this profile's requirements.

272 Within the context of a particular role (and the protocols it supports, as expressed in its
273 `protocolSupportEnumeration` attribute), any and all cryptographic keys that are known by the
274 producer to be valid at the time of metadata production MUST appear within that role's element, in the
275 manner described below in section 2.5.1. This includes not only signing and encryption keys, but also any
276 keys used to establish mutual authentication with technologies such as TLS/SSL.

277 Signing or transport authentication keys intended for future use MAY be included as a means of preparing
278 for migration from an older to a newer key (i.e., key rollover). Once an allowable period of time has
279 elapsed (with this period dependent on deployment-specific policies), the older key can be removed,
280 completing the change. Expired keys (those not in use anymore by an entity, for reasons other than
281 compromise) SHOULD be removed once the rollover process to a new key (or keys) has been
282 completed.

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

288 2.5.1 Key Representation

289 Each key included in a metadata role MUST be placed within its own `<md:KeyDescriptor>` element,
290 with the appropriate `use` attribute (see section 2.4.1.1 of [SAML2Meta], as revised by E62 in
291 [SAML2Errata]), and expressed using the `<ds:KeyInfo>` element.

292 One or more of the following representations within a `<ds:KeyInfo>` element MUST be present:

- 293 • `<ds:KeyValue>`
- 294 • `<ds:X509Certificate>` (child element of `<ds:X509Data>`)

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

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

300 In the case of an X.509 certificate, there are no requirements as to the content of the certificate apart from
301 the requirement that it contain the appropriate public key. Specifically, the certificate may be expired, not
302 yet valid, carry critical or non-critical extensions or usage flags, and contain any subject or issuer. The use
303 of the certificate structure is merely a matter of notational convenience to communicate a key and has no
304 semantics in this profile apart from that. However, it is RECOMMENDED that certificates be unexpired.

305 2.6 Metadata Consumer Requirements

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

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

316 2.6.1 Key Processing

317 Each key expressed by a `<md:KeyDescriptor>` element within a particular role MUST be treated as
318 valid when processing messages or assertions in the context of that role. Specifically, any signatures or
319 transport communications (e.g., TLS/SSL sessions) verifiable with a signing key MUST be treated as
320 valid, and any encryption keys found MAY be used to encrypt messages or assertions (or encryption
321 keys) intended for the containing entity.

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

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

- 327 • `<ds:KeyValue>`
- 328 • `<ds:X509Certificate>` (child element of `<ds:X509Data>`)

329 In the case of the former, the key itself is explicitly identified. In the case of the latter, a metadata
330 consumer MUST extract the public key found in the certificate and MUST NOT honor, interpret, or make
331 use of any of the information found in the certificate other than as an aid in identifying the key used
332 (based, for example, on information found at runtime in an XML digital signature's `<ds:KeyInfo>`
333 element or the certificate presented by a transport peer).

334 Upon identifying a candidate key, a signature can be directly evaluated based on whether it is verifiable
335 with the key. Authentication of a transport peer can be evaluated by extracting the key presented by the
336 peer (often in the form of an X.509 certificate) and comparing it by value to the candidate key. This
337 process has the effect of decoupling the certificates that may be present in metadata from those
338 presented at runtime, provided that the public keys are in fact the same.

339 A metadata consumer, when implementing authentication of a transport peer via TLS/SSL, MAY retain the
340 checking of server certificate names (e.g., `subjectAltName` or `Common Name`) in accordance with
341 [RFC2818]. Note that this constrains the certificates that may be used at runtime for TLS/SSL server
342 authentication, but does not affect certificates that might appear in metadata, because the eventual
343 comparison is based solely on the key.

344 **2.7 Security Considerations**

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

347 Metadata becomes a critical tool for the revocation of compromised sites and keys, and all of the
348 standard practices in the use of tools like CRLs become relevant to the consumption of metadata. The
349 specification has the mechanisms to address these issues, but they have to be used. Specifically,
350 metadata obtained via an insecure transport should be both signed, and should expire, so that consumers
351 are forced to refresh it often enough to limit the damage from compromised information. Either the
352 `validUntil` or `cacheDuration` attribute may be appropriate to mitigate this threat, depending on the
353 exchange mechanism.

354 In addition, distributing signed metadata without an expiration over an untrusted channel (e.g., posting it
355 on a public web site) creates an exposure. An attacker can corrupt the channel and substitute an old
356 metadata file containing a compromised key and proceed to use that key together with other attacks to
357 impersonate a site. Repeatedly expiring (using a `validUntil` attribute) and reissuing the metadata
358 limits the window of exposure, just as a CRL does. Note that the `cacheDuration` attribute does not
359 prevent this attack.

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

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

369 **3 Conformance**

370 **3.0.1 SAML V2.0 Metadata Interoperability Profile**

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

373 A metadata consumer conforms to this profile if it can process metadata consistent with the normative text
374 in section 2.6.

375 **Appendix A. Acknowledgements**

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

- 378 • Rob Philpott, EMC Corporation
- 379 • John Bradley, Individual
- 380 • Jeff Hodges, Individual
- 381 • Scott Cantor, Internet2
- 382 • Nate Klingenstein, Internet2
- 383 • Bob Morgan, Internet2
- 384 • Joni Brennan, Liberty Alliance Project
- 385 • Tom Scavo, National Center for Supercomputing Applications (NCSA)
- 386 • Frederick Hirsch, Nokia Corporation
- 387 • Ari Kermaier, Oracle Corporation
- 388 • Hal Lockhart, Oracle Corporation
- 389 • Brian Campbell, Ping Identity Corporation
- 390 • Anil Saldhana, Red Hat
- 391 • Kent Spaulding, Skyworth TTG Holdings Limited
- 392 • Emily Xu, Sun Microsystems
- 393 • Duane DeCouteau, Veterans Health Administration
- 394 • David Staggs, Veterans Health Administration

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

- 396 • Walter Hoehn, University of Memphis
- 397 • Chad LaJoie, SWITCH
- 398 • Ian Young, EDINA, University of Edinburgh

399 **Appendix B. Revision History**

- 400 ● Draft 01
- 401 ● Draft 02, feedback and discussion ([http://lists.oasis-open.org/archives/security-services/200808/
402 msg00038.html](http://lists.oasis-open.org/archives/security-services/200808/msg00038.html))
- 403 ● Draft 03, feedback and discussion ([http://lists.oasis-open.org/archives/security-services/200902/
404 msg00013.html](http://lists.oasis-open.org/archives/security-services/200902/msg00013.html))
- 405 ● Draft 04, improvements to introductory material
- 406 ● Committee Draft 01, CD edits