

1



2

3 **Kantara Initiative eGovernment** 4 **Implementation Profile of SAML V2.0**

5 **Version:** 2.0

6 **Date:** June 11, 2010

7 **Editor:** Scott Cantor, Internet2

8 **Contributors:**

- 9 • <http://kantarainitiative.org/confluence/x/igCDAg>

10 **Status:** This document is a **Kantara Initiative Final Report**, created by the
11 eGovernment WG (see section 3.9 and 4 of the Kantara Initiative Operating Procedures)

12 **Abstract:**

13 This document contains an implementation profile for eGovernment use of SAML
14 V2.0, suitable for the purposes of testing conformance of implementations of
15 SAML V2.0. It is not a deployment profile, and does not provide for or reflect
16 specific behavior expected of implementations when used within a particular
17 deployment context.

18 **Filename:** kantara-report-egov-saml2-profile-2.0

19

Notice:

20

This work is licensed under the Creative Commons Attribution-Share Alike 3.0 Unported License.

21

22

You are free:

23

- to Share -- to copy, distribute and transmit the work

24

- to Remix -- to adapt the work

25

Under the Following Conditions:

26

- Attribution — You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).

27

28

29

- Share Alike — If you alter, transform, or build upon this work, you may distribute the resulting work only under the same, similar or a compatible license.

30

31

32

With the understanding that:

33

- Waiver — Any of the above conditions can be waived if you get permission from the copyright holder.

34

35

- Public Domain — Where the work or any of its elements is in the public domain under applicable law, that status is in no way affected by the license.

36

37

38

- Other Rights — In no way are any of the following rights affected by the license:

39

40

- Your fair dealing or fair use rights, or other applicable copyright exceptions and limitations;

41

42

- The author's moral rights;

43

- Rights other persons may have either in the work itself or in how the work is used, such as publicity or privacy rights.

44

45

- Notice — For any reuse or distribution, you must make clear to others the license terms of this work. The best way to do this is with a link to this web page.

46

47

48

Copyright © 2010 Kantara Initiative

49

50	Contents	
51	1 Introduction	4
52	1.1 Notation	4
53	2 SAML V2.0 Implementation Profile	6
54	2.1 Required Information	6
55	2.2 Metadata and Trust Management	6
56	2.2.1 Metadata Profiles	6
57	2.2.2 Metadata Exchange	7
58	2.3 Name Identifiers	8
59	2.4 Attributes	8
60	2.5 Browser Single Sign-On	9
61	2.5.1 Identity Provider Discovery	9
62	2.5.2 Authentication Requests	9
63	2.5.3 Responses	10
64	2.5.4 Artifact Resolution	11
65	2.6 Browser Holder of Key Single Sign-On	12
66	2.7 SAML 2.0 Proxying	12
67	2.7.1 Authentication Requests	12
68	2.7.2 Responses	13
69	2.8 Single Logout	13
70	2.8.1 Logout Requests	13
71	2.8.2 Logout Responses	14
72	3 Conformance Classes	15
73	3.1 Standard	15
74	3.1.1 Signature and Encryption Algorithms	15
75	3.2 Standard with Logout	16
76	3.3 Full	16
77	4 References	17
78	4.1 Normative References	17
79	5 Appendix A. Revision History	19
80		

81 **1 INTRODUCTION**

82 SAML V2.0 is a rich and extensible standard that must be profiled to be used
83 interoperably, and the profiles that typically emerge from the broader standardization
84 process usually remain fairly broad and include a number of options and features that
85 increase the burden for implementers and make deployment-time decisions more
86 difficult.

87 The Kantara Initiative eGovernment Implementation Profile provides a SAML V2.0
88 conformance specification for Identity Provider and Service Provider implementations
89 operating in eGovernment federations and deployments. The profile is based on the
90 SAML V2.0 specifications created by the Security Services Technical Committee
91 (SSTC) of OASIS, and related specifications approved by that body. It constrains and
92 supplements the base SAML V2.0 features, elements, and attributes required for
93 eGovernment federations and deployments.

94 Implementation profiles define the features that software implementations must support
95 such that deployers can be assured of the ability to meet their own (possibly varied)
96 deployment requirements. Deployment profiles define specific options and constraints to
97 which deployments are required to conform; they guide product configuration and
98 federation operations, and provide criteria against which actual deployments may be
99 tested. This document does not include a deployment profile, but reflects the features
100 deemed necessary or desirable from software implementations in support of a variety of
101 deployment profiles planned and in use. This includes requirements deemed useful to
102 further the eventual goal of interfederation between deployments.

103 **1.1 Notation**

104 This specification uses normative text to describe the use of SAML capabilities.

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

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

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

115 Listings of XML schemas appear like this.

116 Example code listings appear like this.

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

- 120 • The prefix `saml2:` stands for the SAML 2.0 assertion namespace,
121 `urn:oasis:names:tc:SAML:2.0:assertion`
- 122 • The prefix `saml2p:` stands for the SAML 2.0 protocol namespace,
123 `urn:oasis:names:tc:SAML:2.0:protocol`
- 124 • The prefix `md:` stands for the SAML 2.0 metadata namespace,
125 `urn:oasis:names:tc:SAML:2.0:metadata`
- 126 • The prefix `idpdisc:` stands for the Identity Provider Discovery Service
127 Protocol and Profile [IdPDisco] namespace,
128 `urn:oasis:names:tc:SAML:profiles:SSO:idp-discovery-`
129 `protocol`
- 130 • The prefix `mdattr:` stands for the Metadata Extension for Entity Attributes
131 Version 1.0 [MetaAttr] namespace,
132 `urn:oasis:names:tc:SAML:metadata:attribute`

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

135 **2 SAML V2.0 IMPLEMENTATION PROFILE**

136 This profile specifies behavior and options that implementations of a selected set of
137 SAML V2.0 profiles [SAML2Prof] are required to support. The requirements specified
138 are *in addition to* all normative requirements of the original profiles, as modified by the
139 Approved Errata [SAML2Err], and readers should be familiar with all relevant reference
140 documents. Any such requirements are not repeated here except where deemed necessary
141 to highlight a point of discussion or draw attention to an issue addressed in errata, but
142 remain implied.

143 SAML leaves substantial latitude to implementations with regard to how software is
144 architected and combined with authentication and application infrastructure. Where the
145 terms "Identity Provider" and "Service Provider" are used, they should be understood to
146 include the total software footprint intended to provide the desired functionality; no
147 specific assumptions are made as to how the required features are exposed to deployers,
148 only that there is some method for doing so.

149 **2.1 Required Information**

150 **Identification:** <http://kantarainitiative.org/eGov/profiles/SAML2.0/v2.0>

151 **Contact information:** <http://kantarainitiative.org/confluence/display/eGov/Home>

152 **Description:** Given below

153 **Updates:** Liberty Alliance eGov Profile for SAML 2.0 [eGov15]

154 **2.2 Metadata and Trust Management**

155 Identity Provider, Service Provider, and Discovery Service implementations **MUST**
156 support the use of SAML V2.0 Metadata [SAML2Meta] in conjunction with their support
157 of the SAML V2.0 profiles referenced by subsequent sections. Additional expectations
158 around the use of particular metadata elements related to profile behavior may be
159 encountered in those sections.

160 **2.2.1 Metadata Profiles**

161 Implementations **MUST** support the SAML V2.0 Metadata Interoperability Profile
162 Version 1.0 [MetaIOP].

163 In addition, implementations **MUST** support the use of the `<md:KeyDescriptor>`
164 element as follows:

- 165 • Implementations **MUST** support the <ds:X509Certificate> element as
166 input to subsequent requirements. Support for other key representations, and for
167 other mechanisms for credential distribution, is **OPTIONAL**.
- 168 • Implementations **MUST** support some form of path validation of signing, TLS,
169 and encryption credentials used to secure SAML exchanges against one or more
170 trusted certificate authorities. Support for PKIX [RFC5280] is
171 **RECOMMENDED**; implementations **SHOULD** document the behavior of the
172 validation mechanisms they employ, particular with respect to limitations or
173 divergence from PKIX [RFC5280].
- 174 • Implementations **MUST** support the use of OCSP [RFC2560] and Certificate
175 Revocation Lists (CRLs) obtained via the "CRL Distribution Point" X.509
176 extension [RFC5280] for revocation checking of those credentials.
- 177 • Implementations **MAY** support additional constraints on the contents of
178 certificates used by particular entities, such as "subjectAltName" or "DN", key
179 usage constraints, or policy extensions, but **SHOULD** document such features and
180 make them optional to enable where possible.

181 Note that these metadata profiles are intended to be mutually exclusive within a given
182 deployment context; they are alternatives, rather than complimentary or compatible uses
183 of the same metadata information.

184 Implementations **SHOULD** support the SAML V2.0 Metadata Extension for Entity
185 Attributes Version 1.0 [MetaAttr] and provide policy controls on the basis of SAML
186 attributes supplied via this extension mechanism.

187 **2.2.2 Metadata Exchange**

188 It is **OPTIONAL** for implementations to support the generation or exportation of
189 metadata, but implementations **MUST** support the publication of metadata using the
190 Well-Known-Location method defined in section 4.1 of [SAML2Meta] (under the
191 assumption that entityID values used are suitable for such support).

192 Implementations **MUST** support the following mechanisms for the importation of
193 metadata:

- 194 • local file
- 195 • remote resource at fixed location accessible via HTTP 1.1 [RFC2616] or HTTP
196 1.1 over TLS/SSL [RFC2818]

197 In the case of HTTP resolution, implementations **MUST** support use of the "ETag" and
198 "Last-Modified" headers for cache management. Implementations **SHOULD** support the

199 use of more than one fixed location for the importation of metadata, but MAY leave their
200 behavior unspecified if a single entity's metadata is present in more than one source.

201 Importation of multiple entities' metadata contained within an
202 `<md:EntitiesDescriptor>` element MUST be supported.

203 Finally, implementations SHOULD allow for the automated updating/reimportation of
204 metadata without service degradation or interruption.

205 **2.2.2.1 Metadata Verification**

206 Verification of metadata, if supported, MUST include XML signature verification at least
207 at the root element level, and SHOULD support the following mechanisms for signature
208 key trust establishment:

- 209 • Direct comparison against known keys.
- 210 • Some form of path-based certificate validation against one or more trusted
211 certificate authorities, along with certificate revocation lists and/or OCSP
212 [RFC2560]. Support for PKIX [RFC5280] is RECOMMENDED;
213 implementations SHOULD document the behavior of the validation mechanisms
214 they employ, particular with respect to limitations or divergence from PKIX
215 [RFC5280].

216 **2.3 Name Identifiers**

217 In conjunction with their support of the SAML V2.0 profiles referenced by subsequent
218 sections, Identity Provider and Service Provider implementations MUST support the
219 following SAML V2.0 name identifier formats, in accordance with the normative
220 obligations associated with them by [SAML2Core]:

- 221 • `urn:oasis:names:tc:SAML:2.0:nameid-format:persistent`
- 222 • `urn:oasis:names:tc:SAML:2.0:nameid-format:transient`

223 Support for other formats is OPTIONAL.

224 **2.4 Attributes**

225 In conjunction with their support of the SAML V2.0 profiles referenced by subsequent
226 sections, Identity Provider and Service Provider implementations MUST support the
227 generation and consumption of `<saml2:Attribute>` elements that conform to the
228 SAML V2.0 X.500/LDAP Attribute Profile [SAML-X500].

229 The ability to support `<saml2:AttributeValue>` elements whose values are not
230 simple strings (e.g., `<saml2:NameID>`, or other XML values) is OPTIONAL. Such
231 content could be base64-encoded as an alternative.

232 **2.5 Browser Single Sign-On**

233 This section defines an implementation profile of the SAML V2.0 Web Browser SSO
234 Profile [SAML2Prof].

235 **2.5.1 Identity Provider Discovery**

236 Service Provider and Discovery Service implementations MUST support the Identity
237 Provider Discovery Service Protocol Profile in conformance with section 2.4.1 of
238 [IdPDisco].

239 **2.5.2 Authentication Requests**

240 **2.5.2.1 Binding and Security Requirements**

241 Identity Provider and Service Provider implementations MUST support the use of the
242 HTTP-Redirect binding [SAML2Bind] for the transmission of
243 `<saml2p:AuthnRequest>` messages, including the generation or verification of
244 signatures in conjunction with this binding.

245 Support for other bindings is OPTIONAL.

246 **2.5.2.2 Message Content**

247 In addition to standard core- and profile-driven requirements, Service Provider
248 implementations MUST support the inclusion of at least the following
249 `<saml2p:AuthnRequest>` child elements and attributes (when appropriate):

- 250 • `AssertionConsumerServiceURL`
- 251 • `ProtocolBinding`
- 252 • `ForceAuthn`
- 253 • `IsPassive`
- 254 • `AttributeConsumingServiceIndex`
- 255 • `<saml2p:RequestedAuthnContext>`
- 256 • `<saml2p:NameIDPolicy>`

257 Identity Provider implementations **MUST** support all `<saml2p:AuthnRequest>`
258 child elements and attributes defined by [SAML2Core], but **MAY** provide that support in
259 the form of returning appropriate errors when confronted by particular request options.
260 However, implementations **MUST** fully support the options enumerated above, and be
261 configurable to utilize those options in a useful manner as defined by [SAML2Core].

262 Implementations **MAY** limit their support of the
263 `<saml2p:RequestedAuthnContext>` element to the value "exact" for the
264 Comparison attribute, but **MUST** otherwise support any allowable content of the
265 element.

266 Identity Provider implementations **MUST** support verification of requested
267 AssertionConsumerServiceURL locations via comparison to
268 `<md:AssertionConsumerService>` elements supplied via metadata using
269 case-sensitive string comparison. It is **OPTIONAL** to support other means of
270 comparison (e.g., canonicalization or other manipulation of URL values) or
271 alternative verification mechanisms.

272 **2.5.3 Responses**

273 **2.5.3.1 Binding and Security Requirements**

274 Identity Provider and Service Provider implementations **MUST** support the use of the
275 HTTP-POST and HTTP-Artifact bindings [SAML2Bind] for the transmission of
276 `<saml2p:Response>` messages.

277 Support for other bindings, and for artifact types other than
278 `urn:oasis:names:tc:SAML:2.0:artifact-04`, is **OPTIONAL**.

279 Identity Provider and Service Provider implementations **MUST** support the generation
280 and consumption of unsolicited `<saml2p:Response>` messages (i.e., responses that are
281 not the result of a `<saml2p:AuthnRequest>` message).

282 Identity Provider implementations **MUST** support the issuance of
283 `<saml2p:Response>` messages (with appropriate status codes) in the event of an
284 error condition, provided that the user agent remains available and an acceptable location
285 to which to deliver the response is available. The criteria for "acceptability" of a response
286 location are not formally specified, but are subject to Identity Provider policy and reflect
287 its
288 responsibility to protect users from being sent to untrusted or possibly malicious parties.
289 Note that this is a stronger requirement than the comparable language in [SAML2Prof].

290 Identity Provider and Service Provider implementations MUST support the signing of
291 <saml2:Assertion> elements in responses; support for signing of the
292 <saml2p:Response> element is OPTIONAL.

293 Identity Provider and Service Provider implementations MUST support the use of XML
294 Encryption via the <saml2:EncryptedAssertion> element when using the
295 HTTP-POST binding; support for the <saml2:EncryptedID> and
296 <saml2:EncryptedAttribute> elements is OPTIONAL.

297 **2.5.3.2 Message Content**

298 The Web Browser SSO Profile allows responses to contain any number of assertions and
299 statements. Identity Provider implementations MUST allow the number of
300 <saml2:Assertion>, <saml2:AuthnStatement>, and
301 <saml2:AttributeStatement> elements in the <saml2p:Response> message
302 to be limited to one. In turn, Service Provider implementations MAY limit support to a
303 single instance of those elements when processing <saml2p:Response> messages.

304 Identity Provider implementations MUST support the inclusion of a Consent attribute
305 in <saml2p:Response> messages, and a SessionIndex attribute in
306 <saml2:AuthnStatement> elements.

307 Service Provider implementations that provide some form of session semantics MUST
308 support the <saml2:AuthnStatement> element's SessionNotOnOrAfter
309 attribute.

310 Service Provider implementations MUST support the acceptance/rejection of assertions
311 based on the content of the <saml2:AuthnStatement> element's
312 <saml2:AuthnContext> element. Implementations also MUST support the
313 acceptance/rejection of particular <saml2:AuthnContext> content based on the
314 identity of the Identity Provider. [IAP] provides one such mechanism via SAML
315 V2.0 metadata and is RECOMMENDED; though this specification is in draft form,
316 the technical details are not expected to change prior to eventual approval.

317 **2.5.4 Artifact Resolution**

318 Pursuant to the requirement in section 2.5.3.1 for support of the HTTP-Artifact binding
319 [SAML2Bind] for the transmission of <saml2p:Response> messages,
320 implementations MUST support the SAML V2.0 Artifact Resolution profile
321 [SAML2Prof] as constrained by the following subsections.

322 **2.5.4.1 Artifact Resolution Requests**

323 Identity Provider and Service Provider implementations **MUST** support the use of the
324 SAML SOAP (using HTTP as a transport) binding [SAML2Bind] for the transmission of
325 <saml2p:ArtifactResolve> messages.

326 Implementations **MUST** support the use of SAML message signatures and TLS server
327 authentication to authenticate requests; support for TLS client authentication, or other
328 forms of authentication in conjunction with the SAML SOAP binding, is **OPTIONAL**.

329 **2.5.4.2 Artifact Resolution Responses**

330 Identity Provider and Service Provider implementations **MUST** support the use of the
331 SAML SOAP (using HTTP as a transport) binding [SAML2Bind] for the transmission of
332 <saml2p:ArtifactResponse> messages.

333 Implementations **MUST** support the use of SAML message signatures and TLS server
334 authentication to authenticate responses; support for TLS client authentication, or other
335 forms of authentication in conjunction with the SAML SOAP binding, is **OPTIONAL**.

336 **2.6 Browser Holder of Key Single Sign-On**

337 This section defines an implementation profile of the SAML V2.0 Holder-of-Key Web
338 Browser SSO Profile Version 1.0 [HoKSSO].

339 The implementation requirements defined in section 2.5 for the non-holder-of-key profile
340 apply to implementations of this profile.

341 **2.7 SAML 2.0 Proxying**

342 Section 3.4.1.5 of [SAML2Core] defines a formalized approach to proxying the SAML
343 2.0 Authentication Request protocol between multiple Identity Providers. This section
344 defines an implementation profile for this behavior suitable for composition with the
345 Single Sign-On profiles defined in sections 2.5 and 2.6.

346 The requirements of the profile are imposed on Identity Provider implementations acting
347 as a proxy. These requirements are in addition to the technical requirements outlined in
348 section 3.4.1.5.1 of [SAML2Core], which also **MUST** be supported.

349 **2.7.1 Authentication Requests**

350 Proxying Identity Provider implementations **MUST** support the mapping of incoming to
351 outgoing <saml2p:RequestedAuthnContext> and
352 <saml2p:NameIDPolicy> elements, such that deployers may choose to pass through
353 values or map between different vocabularies as required.

354 Proxying Identity Provider implementations MUST support the suppression/eliding of
355 <saml2p:RequesterID> elements from outgoing <saml2p:AuthnRequest>
356 messages to allow for hiding the identity of the Service Provider from proxied Identity
357 Providers.

358 **2.7.2 Responses**

359 Proxying Identity Provider implementations MUST support the mapping of incoming to
360 outgoing <saml2:AuthnContext> elements, such that deployers may choose to pass
361 through values or map between different vocabularies as required.

362 Proxying Identity Provider implementations MUST support the suppression of
363 <saml2:AuthenticatingAuthority> elements from outgoing
364 <saml2:AuthnContext> elements to allow for hiding the identity of the proxied
365 Identity Provider from Service Providers.

366 **2.8 Single Logout**

367 This section defines an implementation profile of the SAML V2.0 Single Logout Profile
368 [SAML2Prof].

369 For clarification, the technical requirements for each message type below reflect the
370 intent to normatively require initiation of logout by a Service Provider using either the
371 front- or back-channel, and initiation/propagation of logout by an Identity Provider using
372 the back-channel.

373 **2.8.1 Logout Requests**

374 **2.8.1.1 Binding and Security Requirements**

375 Identity Provider implementations MUST support the SAML SOAP (using HTTP as a
376 transport) binding [SAML2Bind] for the issuance of <saml2p:LogoutRequest>
377 messages, and MUST support the SAML SOAP (using HTTP as a transport) and HTTP-
378 Redirect bindings [SAML2Bind] for the reception of <saml2p:LogoutRequest>
379 messages.

380 Service Provider implementations MUST support the SAML SOAP (using HTTP as a
381 transport) binding [SAML2Bind] for both issuance and reception of
382 <saml2p:LogoutRequest> messages.

383 Support for other bindings is OPTIONAL.

384 Implementations MUST support the use of SAML message signatures and TLS server
385 authentication to authenticate <saml2p:LogoutRequest> messages; support for

386 TLS client authentication, or other forms of authentication in conjunction with the SAML
387 SOAP binding, is OPTIONAL.

388 Identity Provider and Service Provider implementations MUST support the use of XML
389 Encryption via the `<saml2:EncryptedID>` element when using the HTTP-Redirect
390 binding.

391 **2.8.1.2 User Interface Behavior**

392 Identity Provider implementations MUST support both user-initiated termination of the
393 local session only and user-initiated Single Logout. Upon receipt of a
394 `<saml2p:LogoutRequest>` message via a front-channel binding, Identity Provider
395 implementations MUST support user intervention governing the choice of propagating
396 logout to other Service Providers, or limiting the operation to the Identity Provider. Of
397 course, implementations MUST return status information to the requesting entity (e.g.
398 partial logout indication) as appropriate.

399 Service Provider implementations MUST support both user-initiated termination of the
400 local session only and user-initiated Single Logout.

401 Identity Provider implementations MUST also support the administrative initiation of
402 Single Logout for any active session, subject to appropriate policy.

403 **2.8.2 Logout Responses**

404 **2.8.2.1 Binding and Security Requirements**

405 Identity Provider implementations MUST support the SAML SOAP (using HTTP as a
406 transport) and HTTP-Redirect bindings [SAML2Bind] for the issuance of
407 `<saml2p:LogoutResponse>` messages, and MUST support the SAML SOAP
408 (using HTTP as a transport) binding [SAML2Bind] for the reception of
409 `<saml2p:LogoutResponse>` messages.

410 Service Provider implementations MUST support the SAML SOAP (using HTTP as a
411 transport) binding [SAML2Bind] for both issuance and reception of
412 `<saml2p:LogoutResponse>` messages.

413 Support for other bindings is OPTIONAL.

414 Implementations MUST support the use of SAML message signatures and TLS server
415 authentication to authenticate `<saml2p:LogoutResponse>` messages; support for
416 TLS client authentication, or other forms of authentication in conjunction with the SAML
417 SOAP binding, is OPTIONAL.

418 **3 CONFORMANCE CLASSES**

419 **3.1 Standard**

420 Conforming Identity Provider and/or Service Provider implementations **MUST** support
421 the normative requirements in sections 2.2, 2.3, 2.4, and 2.5.

422 **3.1.1 Signature and Encryption Algorithms**

423 Implementations **MUST** support the signature and digest algorithms identified by the
424 following URIs in conjunction with the creation and verification of XML Signatures
425 [XMLSig]:

- 426 • <http://www.w3.org/2001/04/xmldsig-more#rsa-sha256> (defined in
427 [RFC4051])
- 428 • <http://www.w3.org/2001/04/xmlenc#sha256> (defined in [XMLEnc])

429 Implementations **SHOULD** support the signature and digest algorithms identified by the
430 following URIs in conjunction with the creation and verification of XML Signatures
431 [XMLSig]:

- 432 • <http://www.w3.org/2001/04/xmldsig-more#ecdsa-sha256> (defined in
433 [RFC4051])

434 Implementations **MUST** support the block encryption algorithms identified by the following URIs in
435 conjunction with the use of XML Encryption [XMLEnc]:

- 436 • <http://www.w3.org/2001/04/xmlenc#tripledes-cbc>
- 437 • <http://www.w3.org/2001/04/xmlenc#aes128-cbc>
- 438 • <http://www.w3.org/2001/04/xmlenc#aes256-cbc>

439 Implementations **MUST** support the key transport algorithms identified by the following URIs in
440 conjunction with the use of XML Encryption [XMLEnc]:

- 441 • http://www.w3.org/2001/04/xmlenc#rsa-1_5
- 442 • <http://www.w3.org/2001/04/xmlenc#rsa-oaep-mgf1p>

443 Implementations **SHOULD** support the key agreement algorithms identified by the following URIs
444 in conjunction with the use of XML Encryption [XMLEnc]:

- 445 • <http://www.w3.org/2009/xmlenc11#ECDH-ES> (defined in [XMLEnc11])

446

447 (This is a Last Call Working Draft of XML Encryption 1.1, and this normative requirement
448 is contingent on W3C ratification of this specification without normative changes to this
449 algorithm's definition.)

450 Support for other algorithms is OPTIONAL.

451 **3.2 Standard with Logout**

452 Conforming Identity Provider and/or Service Provider implementations **MUST** meet the
453 conformance requirements in section 3.1, and **MUST** in addition support the normative
454 requirements in section 2.8.

455 **3.3 Full**

456 Conforming Identity Provider and/or Service Provider implementations **MUST** meet the
457 conformance requirements in section 3.1, and **MUST** in addition support the normative
458 requirements in sections 2.6, 2.7, and 2.8.

459 4 REFERENCES

460 4.1 Normative References

- 461 [RFC2119] IETF RFC 2119, *Key words for use in RFCs to Indicate Requirement*
462 *Levels*, March 1997. <http://www.ietf.org/rfc/rfc2119.txt>
- 463 [RFC2560] IETF RFC 2560, *X.509 Internet Public Key Infrastructure Online*
464 *Certificate Status Protocol*, June 1999.
465 <http://www.ietf.org/rfc/rfc2560.txt>
- 466 [RFC2616] IETF RFC 2616, *Hypertext Transfer Protocol – HTTP/1.1*, June 1999.
467 <http://www.ietf.org/rfc/rfc2616.txt>
- 468 [RFC2818] IETF RFC 2818, *HTTP Over TLS*, May 2000.
469 <http://www.ietf.org/rfc/rfc2818.txt>
- 470 [RFC4051] IETF RFC 4051, *Additional XML Security Uniform Resource Identifiers*,
471 April 2005. <http://www.ietf.org/rfc/rfc4051.txt>
- 472 [RFC5280] IETF RFC 5280, *Internet X.509 Public Key Infrastructure Certificate*
473 *and Certificate Revocation List (CRL) Profile*, May 2008.
474 <http://www.ietf.org/rfc/rfc5280.txt>
- 475 [HoKSSO]_OASIS Committee Specification, *SAML V2.0 Holder-of-Key Web*
476 *Browser SSO Profile Version 1.0*, July 2009. [http://docs.oasis-](http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-holder-of-key-browser-sso-cs-01.pdf)
477 [open.org/security/saml/Post2.0/sstc-saml-holder-of-key-browser-](http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-holder-of-key-browser-sso-cs-01.pdf)
478 [sso-cs-01.pdf](http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-holder-of-key-browser-sso-cs-01.pdf)
- 479 [IAP]_OASIS Committee Draft, *Identity Assurance Profiles, Version 1.0*, September
480 2009. [http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-](http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-assurance-profile-cd-01.pdf)
481 [assurance-profile-cd-01.pdf](http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-assurance-profile-cd-01.pdf)
- 482 [IdPDisco]_OASIS Committee Specification, *Identity Provider Discovery Service*
483 *Protocol and Profile*, March 2008. [http://docs.oasis-](http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-idp-discovery.pdf)
484 [open.org/security/saml/Post2.0/sstc-saml-idp-discovery.pdf](http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-idp-discovery.pdf)
- 485 [MetaAttr]_OASIS Committee Specification, *SAML V2.0 Metadata Extension for*
486 *Entity Attributes Version 1.0*, August 2009. [http://docs.oasis-](http://docs.oasis-open.org/security/saml/Post2.0/sstc-metadata-attr.pdf)
487 [open.org/security/saml/Post2.0/sstc-metadata-attr.pdf](http://docs.oasis-open.org/security/saml/Post2.0/sstc-metadata-attr.pdf)
- 488 [MetaIOP]_OASIS Committee Specification, *SAML V2.0 Metadata Interoperability*
489 *Profile Version 1.0*, August 2009. [http://docs.oasis-](http://docs.oasis-open.org/security/saml/Post2.0/sstc-metadata-iop.pdf)
490 [open.org/security/saml/Post2.0/sstc-metadata-iop.pdf](http://docs.oasis-open.org/security/saml/Post2.0/sstc-metadata-iop.pdf)
- 491 [SAML2Core] OASIS Standard, *Assertions and Protocols for the OASIS Security*
492 *Assertion Markup Language (SAML) V2.0*, March 2005.
493 <http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf>

- 494 [SAML2Meta] OASIS Standard, *Metadata for the OASIS Security Assertion Markup*
495 *Language (SAML) V2.0*, March 2005. [http://docs.oasis-
open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf](http://docs.oasis-
496 open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf)
- 497 [SAML2Bind] OASIS Standard, *Bindings for the OASIS Security Assertion Markup*
498 *Language (SAML) V2.0*, March 2005. [http://docs.oasis-
open.org/security/saml/v2.0/saml-bindings-2.0-os.pdf](http://docs.oasis-
499 open.org/security/saml/v2.0/saml-bindings-2.0-os.pdf)
- 500 [SAML2Prof] OASIS Standard, *Profiles for the OASIS Security Assertion Markup*
501 *Language (SAML) V2.0*, March 2005. [http://docs.oasis-
open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf](http://docs.oasis-
502 open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf)
- 503 [SAML2Err] OASIS Approved Errata, *SAML V2.0 Errata*, Dec 2009.
504 [http://www.oasis-open.org/committees/download.php/37166/ssstc-
saml-approved-errata-2.0-02.pdf](http://www.oasis-open.org/committees/download.php/37166/ssstc-
505 saml-approved-errata-2.0-02.pdf)
- 506 [SAML-X500] OASIS Committee Specification, *SAML V2.0 X.500/LDAP Attribute*
507 *Profile*, March 2008. [http://docs.oasis-
open.org/security/saml/Post2.0/ssstc-saml-attribute-x500.pdf](http://docs.oasis-
508 open.org/security/saml/Post2.0/ssstc-saml-attribute-x500.pdf)
- 509 [XMLEnc] D. Eastlake et al. *XML Encryption Syntax and Processing*. World Wide
510 Web Consortium Recommendation. [http://www.w3.org/TR/2002/REC-
xmlenc-core-20021210/](http://www.w3.org/TR/2002/REC-
511 xmlenc-core-20021210/)
- 512 [XMLEnc11] D. Eastlake et al. *XML Encryption Syntax and Processing Version 1.1*.
513 World Wide Web Consortium Last Call Working Draft.
514 <http://www.w3.org/TR/2010/WD-xmlenc-core1-20100513/>
- 515 [XMLSig] D. Eastlake et al. *XML-Signature Syntax and Processing, Second Edition*.
516 World Wide Web Consortium Recommendation, June 2008.
517 <http://www.w3.org/TR/xmlsig-core/>

518 4.2 Non-Normative References

- 519 [eGov15] Kyle Meadors, *Liberty Alliance eGov Profile for SAML 2.0 Version*
520 *1.5*.
521 [http://www.projectliberty.org/liberty/content/download/4711/3
2210/file/Liberty Alliance eGov Profile 1.5 Final.pdf](http://www.projectliberty.org/liberty/content/download/4711/3
522 2210/file/Liberty Alliance eGov Profile 1.5 Final.pdf)

523 5 APPENDIX A. REVISION HISTORY

- 524 • Draft 01: first working draft based on similar document created by InCommon
525 Federation
- 526 • Draft 02: first round of feedback incorporated, deployment section dropped, new
527 section on Artifact Resolution added, artifact added for SSO responses, SOAP
528 added for logout, discovery moved under SSO, language on non-string attributes
529 added, changed SHOULD to MUST for IdP support of selected AuthnRequest
530 features
- 531 • Draft 03: moved Artifact Resolution into a SSO profile subsection, new language
532 on SOAP security and SLO bindings, added metadata publication via WKL,
533 added language on IdP error handling, added Holder of Key SSO profile, added
534 Conformance Classes
- 535 • Draft 04: added UI language around SLO, layered conformance language and
536 added MTI algorithms, added section for Proxying
- 537 • Draft 05: revised language for IdP error handling, added text on ACS checking,
538 added proxying privacy language, heavily revised metadata section and added a
539 "pseudo-profile" for combining certificates in metadata with PKI as an IOP
540 alternative
- 541 • Draft 06: added normative reference to RFC5280 in path validation text,
542 expanded algorithm requirements, added sentence on administrative logout
- 543 • Draft 07, clarifications on AuthnContext support and reference to IAP, additional
544 algorithm reference, change to boilerplate sections to match Kantara template