

---

# Kantara Initiative eGovernment Implementation Profile of SAML V2.0

~~May 14, 2010~~

~~Working Draft 0~~



~~Version 2.0~~

**Document identifierVersion:**

~~draft-kantara-egov-saml2-profile-2.0Working Draft 07~~

**Date:**

~~May 14, 2010~~

**Location:**

~~TBD~~

**Editors:**

~~Scott Cantor, Internet2~~

**Contributors:**

~~Kantara eGovernment Working Group~~

~~Andreas Åkre Solberg, UNINETT~~

**Abstract:**

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

**Filename:**

~~draft-kantara-egov-saml2-profile-2.0-07~~

**Notice:**

~~This document has been prepared by Participants of Kantara Initiative. Permission is hereby granted to use the document solely for the purpose of implementing the Specification. No rights are granted to prepare derivative works of this Specification. Entities seeking permission to reproduce portions of this document for other uses must contact Kantara Initiative to determine whether an appropriate license for such use is available.~~

~~-  
Implementation or use of certain elements of this document may require licenses under third party~~

32 intellectual property rights, including without limitation, patent rights. The Participants of and any  
33 other contributors to the Specification are not and shall not be held responsible in any manner for  
34 identifying or failing to identify any or all such third party intellectual property rights. This  
35 Specification is provided "AS IS," and no Participant in Kantara Initiative makes any warranty of  
36 any kind, expressed or implied, including any implied warranties of merchantability, non-  
37 infringement of third party intellectual property rights, and fitness for a particular purpose.  
38 Implementers of this Specification are advised to review Kantara Initiative's website  
39 (<http://www.kantarainitiative.org/>) for information concerning any Necessary Claims Disclosure  
40 Notices that have been received by the Kantara Initiative Board of Trustees. This work is licensed  
41 under the Creative Commons Attribution-Share Alike 3.0 Unported License.

42 You are free:

- 43 • to Share -- to copy, distribute and transmit the work
- 44 • **— to Remix -- to adapt the work**

45 Under the Following Conditions:

- 46 • Attribution — You must attribute the work in the manner specified by the author or  
47 licensor (but not in any way that suggests that they endorse you or your use of the work).
- 48 • Share Alike — If you alter, transform, or build upon this work, you may distribute the  
49 resulting work only under the same, similar or a compatible license.

50 •

51 With the understanding that:

- 52 •
- 53 • Waiver — Any of the above conditions can be waived if you get permission from the  
54 copyright holder.
- 55 •
- 56 • Public Domain — Where the work or any of its elements is in the public domain under  
57 applicable law, that status is in no way affected by the license.
- 58 •
- 59 • Other Rights — In no way are any of the following rights affected by the license:
  - 60 • \_\_\_\_\_ Your fair dealing or fair use rights, or other applicable copyright  
61 exceptions and limitations;
  - 62 • \_\_\_\_\_ The author's moral rights;
  - 63 • \* \_\_\_\_\_ Rights other persons may have either in the work itself or in how  
64 the work is used, such as publicity or privacy rights.
- 65 • Notice — For any reuse or distribution, you must make clear to others the license terms of  
66 this work. The best way to do this is with a link to this web page.

67 Copyright © 2010 Kantara Initiative

68 -

69 Copyright: The content of this document is copyright of Kantara Initiative. © 2010 Kantara  
70 Initiative.

# 71 Table of Contents

72	1 Introduction.....	4
73	1.1 Notation.....	4
74	2 SAML V2.0 Implementation Profile.....	6
75	2.1 Required Information.....	6
76	2.2 Metadata and Trust Management.....	6
77	2.2.1 Metadata Profiles.....	6
78	2.2.2 Metadata Exchange.....	7
79	2.2.2.1 Metadata Verification.....	7
80	2.3 Name Identifiers.....	7
81	2.4 Attributes.....	8
82	2.5 Browser Single Sign-On.....	8
83	2.5.1 Identity Provider Discovery.....	8
84	2.5.2 Authentication Requests.....	8
85	2.5.2.1 Binding and Security Requirements.....	8
86	2.5.2.2 Message Content.....	8
87	2.5.3 Responses.....	9
88	2.5.3.1 Binding and Security Requirements.....	9
89	2.5.3.2 Message Content.....	9
90	2.5.4 Artifact Resolution.....	10
91	2.5.4.1 Artifact Resolution Requests.....	10
92	2.5.4.2 Artifact Resolution Responses.....	10
93	2.6 Browser Holder of Key Single Sign-On.....	10
94	2.7 SAML 2.0 Proxying.....	10
95	2.7.1 Authentication Requests.....	10
96	2.7.2 Responses.....	11
97	2.8 Single Logout.....	11
98	2.8.1 Logout Requests.....	11
99	2.8.1.1 Binding and Security Requirements.....	11
100	2.8.1.2 User Interface Behavior.....	11
101	2.8.2 Logout Responses.....	12
102	2.8.2.1 Binding and Security Requirements.....	12
103	3 Conformance Classes.....	13
104	3.1 Standard.....	13
105	3.1.1 Signature and Encryption Algorithms.....	13
106	3.2 Standard with Logout.....	13
107	3.3 Full.....	14
108	4 References.....	15
109	4.1 Normative References.....	15
110	Appendix A. Revision History.....	17
111		

---

# 1 Introduction

112

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

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

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

## 1.1 Notation

131

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

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

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

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

141       Listings of XML schemas appear like this.

142       Example code listings appear like this.

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

- 147 • The prefix `saml2:` stands for the SAML 2.0 assertion namespace,  
148     `urn:oasis:names:tc:SAML:2.0:assertion`
- 149 • The prefix `saml2p:` stands for the SAML 2.0 protocol namespace,  
150     `urn:oasis:names:tc:SAML:2.0:protocol`
- 151 • The prefix `md:` stands for the SAML 2.0 metadata namespace,  
152     `urn:oasis:names:tc:SAML:2.0:metadata`
- 153 • The prefix `idpdisc:` stands for the Identity Provider Discovery Service Protocol and Profile  
154     [IdPDisco] namespace, `urn:oasis:names:tc:SAML:profiles:SSO:idp-discovery-  
155     protocol`

156 • The prefix `mdattr:` stands for the Metadata Extension for Entity Attributes Version 1.0 [MetaAttr]  
157 namespace, `urn:oasis:names:tc:SAML:metadata:attribute`

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

---

## 2 SAML V2.0 Implementation Profile

160

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

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

### 2.1 Required Information

172

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

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

175 **Description:** Given below

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

### 2.2 Metadata and Trust Management

177

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

#### 2.2.1 Metadata Profiles

182

183 Implementations MUST support the SAML V2.0 Metadata Interoperability Profile Version 1.0 [MetalOP].

184 In addition, implementations MUST also support an alternative to that profile's language on the use of the  
185 `<md:KeyDescriptor>` element as follows:

- 186 • Implementations MUST support the `<ds:X509Certificate>` element as input to subsequent  
187 requirements. Support for other key representations, and for other mechanisms for credential  
188 distribution, is OPTIONAL.
- 189 • Implementations MUST support some form of path validation of signing, TLS, and encryption  
190 credentials used to secure SAML exchanges against one or more trusted certificate authorities.  
191 Support for PKIX [RFC5280] is RECOMMENDED; implementations SHOULD document the  
192 behavior of the validation mechanisms they employ, particular with respect to limitations or  
193 divergence from PKIX [RFC5280].
- 194 • Implementations MUST support the use of OCSP [RFC2560] and eCertificate rRevocation lLists  
195 (CRLs) obtained via the "CRL Distribution Point" X.509 extension [RFC5280] for revocation  
196 checking of those credentials.
- 197 • Implementations MAY support additional constraints on the contents of certificates used by  
198 particular entities, such as "subjectAltName" or "DN", key usage constraints, or policy extensions,  
199 but SHOULD document such features and make them optional to enable where possible.

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

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

## 205 **2.2.2 Metadata Exchange**

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

210 Implementations MUST support the following mechanisms for the importation of metadata:

- 211 • local file
- 212 • remote resource at fixed location accessible via HTTP 1.1 [RFC2616] or HTTP 1.1 over TLS/SSL  
213 [RFC2818]

214 In the case of HTTP resolution, implementations MUST support use of the "ETag" and "Last-Modified"  
215 headers for cache management. Implementations SHOULD support the use of more than one fixed  
216 location for the importation of metadata, but MAY leave their behavior unspecified if a single entity's  
217 metadata is present in more than one source.

218 Importation of multiple entities' metadata contained within an <md:EntitiesDescriptor> element  
219 MUST be supported.

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

### 222 **2.2.2.1 Metadata Verification**

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

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

## 231 **2.3 Name Identifiers**

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

- 235 • urn:oasis:names:tc:SAML:2.0:nameid-format:persistent
- 236 • urn:oasis:names:tc:SAML:2.0:nameid-format:transient

237 Support for other formats is OPTIONAL.

## 238 2.4 Attributes

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

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

## 246 2.5 Browser Single Sign-On

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

### 248 2.5.1 Identity Provider Discovery

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

### 251 2.5.2 Authentication Requests

#### 252 2.5.2.1 Binding and Security Requirements

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

256 Support for other bindings is OPTIONAL.

#### 257 2.5.2.2 Message Content

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

- 261 • `AssertionConsumerServiceURL`
- 262 • `ProtocolBinding`
- 263 • `ForceAuthn`
- 264 • `IsPassive`
- 265 • `AttributeConsumingServiceIndex`
- 266 • `<saml2p:RequestedAuthnContext>`
- 267 • `<saml2p:NameIDPolicy>`
- 268 •

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



274 Implementations MAY limit their support of the `<saml2p:RequestedAuthnContext>` element to the  
275 value "exact" for the `Comparison` attribute, but MUST otherwise support any allowable content of the  
276 element.

277 Identity Provider implementations MUST support verification of requested  
278 `AssertionConsumerServiceURL` locations via comparison to `<md:AssertionConsumerService>`  
279 elements supplied via metadata using case-sensitive string comparison. It is OPTIONAL to support other  
280 means of comparison (e.g., canonicalization or other manipulation of URL values) or alternative verification  
281 mechanisms.

## 282 **2.5.3 Responses**

### 283 **2.5.3.1 Binding and Security Requirements**

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

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

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

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

297 Identity Provider and Service Provider implementations MUST support the signing of  
298 `<saml2:Assertion>` elements in responses; support for signing of the `<saml2p:Response>` element  
299 is OPTIONAL.

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

### 303 **2.5.3.2 Message Content**

304 The Web Browser SSO Profile allows responses to contain any number of assertions and statements.  
305 Identity Provider implementations MUST allow the number of `<saml2:Assertion>`,  
306 `<saml2:AuthnStatement>`, and `<saml2:AttributeStatement>` elements in the  
307 `<saml2p:Response>` message to be limited to one.

308 In turn, Service Provider implementations MAY limit support to a single instance of those elements when  
309 processing `<saml2p:Response>` messages.

310 Identity Provider implementations MUST support the inclusion of a `Consent` attribute in  
311 `<saml2p:Response>` messages, and a `SessionIndex` attribute in `<saml2:AuthnStatement>`  
312 elements.

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

315 Service Provider implementations MUST support the acceptance/rejection of assertions based on the  
316 content of the `<saml2:AuthnStatement>` element's `<saml2:AuthnContext>` element.

317 Implementations also MUST support the acceptance/rejection of particular <saml2:AuthnContext>  
318 content based on the identity of the Identity Provider. [IAP] provides one such mechanism via SAML V2.0  
319 metadata and is RECOMMENDED; though this specification is in draft form, the technical details are not  
320 expected to change prior to eventual approval.

## 321 **2.5.4 Artifact Resolution**

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

### 325 **2.5.4.1 Artifact Resolution Requests**

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

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

### 332 **2.5.4.2 Artifact Resolution Responses**

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

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

## 339 **2.6 Browser Holder of Key Single Sign-On**

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

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

## 344 **2.7 SAML 2.0 Proxying**

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

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

### 351 **2.7.1 Authentication Requests**

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

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

## 358 **2.7.2 Responses**

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

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

## 365 **2.8 Single Logout**

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

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

### 370 **2.8.1 Logout Requests**

#### 371 **2.8.1.1 Binding and Security Requirements**

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

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

378 Support for other bindings is OPTIONAL.

379 Implementations MUST support the use of SAML message signatures and TLS server authentication to  
380 authenticate <saml2p:LogoutRequest> messages; support for TLS client authentication, or other  
381 forms of authentication in conjunction with the SAML SOAP binding, is OPTIONAL.

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

#### 384 **2.8.1.2 User Interface Behavior**

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

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

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

## 395 **2.8.2 Logout Responses**

### 396 **2.8.2.1 Binding and Security Requirements**

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

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

403 Support for other bindings is OPTIONAL.

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

---

## 407 3 Conformance Classes

### 408 3.1 Standard

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

#### 411 3.1.1 Signature and Encryption Algorithms

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

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

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

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

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

- 421 • <http://www.w3.org/2001/04/xmlenc#tripleDES-cbc>
- 422 • <http://www.w3.org/2001/04/xmlenc#aes128-cbc>
- 423 • <http://www.w3.org/2001/04/xmlenc#aes256-cbc>

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

- 426 • [http://www.w3.org/2001/04/xmlenc#rsa-1\\_5](http://www.w3.org/2001/04/xmlenc#rsa-1_5)
- 427 • <http://www.w3.org/2001/04/xmlenc#rsa-oaep-mgf1p>

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

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

431 (This is a Last Call Working Draft of XML Encryption 1.1, and this normative requirement is  
432 contingent on W3C ratification of this specification without normative changes to this algorithm's  
433 definition.)  
434

435 Support for other algorithms is OPTIONAL.

### 436 3.2 Standard with Logout

437 Conforming Identity Provider and/or Service Provider implementations MUST meet the conformance  
438 requirements in section 3.1, and MUST in addition support the normative requirements in section 2.8.

439 **3.3 Full**

440 Conforming Identity Provider and/or Service Provider implementations MUST meet the conformance  
441 requirements in section 3.1, and MUST in addition support the normative requirements in sections 2.6,  
442 2.7, and 2.8.

## 4 References

### 4.1 Normative References

- 445     **[RFC2119]**     IETF RFC 2119, *Key words for use in RFCs to Indicate Requirement Levels*,  
446                   March 1997. <http://www.ietf.org/rfc/rfc2119.txt>
- 447     **[RFC2560]**     IETF RFC 2560, *X.509 Internet Public Key Infrastructure Online Certificate Status*  
448                   *Protocol*, June 1999. <http://www.ietf.org/rfc/rfc2560.txt>
- 449     **[RFC2616]**     IETF RFC 2616, *Hypertext Transfer Protocol – HTTP/1.1*, June 1999.  
450                   <http://www.ietf.org/rfc/rfc2616.txt>
- 451     **[RFC2818]**     IETF RFC 2818, *HTTP Over TLS*, May 2000. <http://www.ietf.org/rfc/rfc2818.txt>
- 452     **[RFC4051]**     IETF RFC 4051, *Additional XML Security Uniform Resource Identifiers*, April  
453                   2005. <http://www.ietf.org/rfc/rfc4051.txt>
- 454     **[RFC5280]**     IETF RFC 5280, *Internet X.509 Public Key Infrastructure Certificate and*  
455                   *Certificate Revocation List (CRL) Profile*, May 2008.  
456                   <http://www.ietf.org/rfc/rfc5280.txt>
- 457     **[HoKSSO]**     OASIS Committee Specification, *SAML V2.0 Holder-of-Key Web Browser SSO*  
458                   *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)  
459                   [open.org/security/saml/Post2.0/sstc-saml-holder-of-key-browser-sso-cs-01.pdf](http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-holder-of-key-browser-sso-cs-01.pdf)
- 460     **[IAP]**         OASIS Committee Draft, *Identity Assurance Profiles, Version 1.0, September*  
461                   *2009*. [http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-assurance-](http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-assurance-profile-cd-01.pdf)  
462                   [profile-cd-01.pdf](http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-assurance-profile-cd-01.pdf)
- 463     **[IdPDisco]**    OASIS Committee Specification, *Identity Provider Discovery Service*  
464                   *Protocol and Profile*, March 2008. [http://docs.oasis-](http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-idp-discovery.pdf)  
465                   [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)
- 466     **[MetaAttr]**    OASIS Committee Specification, *SAML V2.0 Metadata Extension for Entity*  
467                   *Attributes Version 1.0*, August 2009. [http://docs.oasis-](http://docs.oasis-open.org/security/saml/Post2.0/sstc-metadata-attr.pdf)  
468                   [open.org/security/saml/Post2.0/sstc-metadata-attr.pdf](http://docs.oasis-open.org/security/saml/Post2.0/sstc-metadata-attr.pdf)
- 469     **[MetalOP]**    OASIS Committee Specification, *SAML V2.0 Metadata Interoperability Profile*  
470                   *Version 1.0*, August 2009. [http://docs.oasis-open.org/security/saml/Post2.0/sstc-](http://docs.oasis-open.org/security/saml/Post2.0/sstc-metadata-iop.pdf)  
471                   [metadata-iop.pdf](http://docs.oasis-open.org/security/saml/Post2.0/sstc-metadata-iop.pdf)
- 472     **[SAML2Core]**   OASIS Standard, *Assertions and Protocols for the OASIS Security Assertion*  
473                   *Markup Language (SAML) V2.0*, March 2005. [http://docs.oasis-](http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf)  
474                   [open.org/security/saml/v2.0/saml-core-2.0-os.pdf](http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf)
- 475     **[SAML2Meta]**   OASIS Standard, *Metadata for the OASIS Security Assertion Markup Language*  
476                   *(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)  
477                   [metadata-2.0-os.pdf](http://docs.oasis-open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf)
- 478     **[SAML2Bind]**   OASIS Standard, *Bindings for the OASIS Security Assertion Markup Language*  
479                   *(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)  
480                   [bindings-2.0-os.pdf](http://docs.oasis-open.org/security/saml/v2.0/saml-bindings-2.0-os.pdf)
- 481     **[SAML2Prof]**   OASIS Standard, *Profiles for the OASIS Security Assertion Markup Language*  
482                   *(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)  
483                   [profiles-2.0-os.pdf](http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf)
- 484     **[SAML2Err]**    OASIS Approved Errata, *SAML V2.0 Errata*, Dec 2009. [http://www.oasis-](http://www.oasis-open.org/committees/download.php/37166/sstc-saml-approved-errata-2.0-02.pdf)  
485                   [open.org/committees/download.php/37166/sstc-saml-approved-errata-2.0-02.pdf](http://www.oasis-open.org/committees/download.php/37166/sstc-saml-approved-errata-2.0-02.pdf)
- 486     **[SAML-X500]**   OASIS Committee Specification, *SAML V2.0 X.500/LDAP Attribute Profile*, March  
487                   2008. [http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-attribute-](http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-attribute-x500.pdf)  
488                   [x500.pdf](http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-attribute-x500.pdf)

489       **[XMLEnc]**       D. Eastlake et al. *XML Encryption Syntax and Processing*. World Wide Web  
490       Consortium Recommendation. [http://www.w3.org/TR/2002/REC-xmlenc-core-](http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/)  
491       [20021210/](http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/)

492       **[XMLEnc11]**       D. Eastlake et al. *XML Encryption Syntax and Processing Version 1.1*. World  
493       Wide Web Consortium Last Call Working Draft. [http://www.w3.org/TR/2010/WD-](http://www.w3.org/TR/2010/WD-xmlenc-core1-20100513/)  
494       [xmlenc-core1-20100513/](http://www.w3.org/TR/2010/WD-xmlenc-core1-20100513/)

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

498       **Non-Normative References**

499       **[eGov15]**       Kyle Meadors, *Liberty Alliance eGov Profile for SAML 2.0 Version 1.5*.  
500       [http://www.projectliberty.org/liberty/content/download/4711/32210/file/Liberty\\_Allia-](http://www.projectliberty.org/liberty/content/download/4711/32210/file/Liberty_Alliance_eGov_Profile_1.5_Final.pdf)  
501       [nce\\_eGov\\_Profile\\_1.5\\_Final.pdf](http://www.projectliberty.org/liberty/content/download/4711/32210/file/Liberty_Alliance_eGov_Profile_1.5_Final.pdf)



502

## Appendix A. ~~Change Log~~Revision History

503

- Draft 01: first working draft based on similar document created by InCommon Federation

504

505

506

507

- Draft 02: first round of feedback incorporated, deployment section dropped, new section on Artifact Resolution added, artifact added for SSO responses, SOAP added for logout, discovery moved under SSO, language on non-string attributes added, changed SHOULD to MUST for IdP support of selected AuthnRequest features

508

509

510

- Draft 03: moved Artifact Resolution into a SSO profile subsection, new language on SOAP security and SLO bindings, added metadata publication via WKL, added language on IdP error handling, added Holder of Key SSO profile, added Conformance Classes

511

512

- Draft 04: added UI language around SLO, layered conformance language and added MTI algorithms, added section for Proxying

513

514

515

- Draft 05: revised language for IdP error handling, added text on ACS checking, added proxying privacy language, heavily revised metadata section and added a "pseudo-profile" for combining certificates in metadata with PKI as an IOP alternative

516

517

- Draft 06: added normative reference to RFC5280 in path validation text, expanded algorithm requirements, added sentence on administrative logout

518

519

- [Draft 07, clarifications on AuthnContext support and reference to IAP, additional algorithm reference, change to boilerplate sections to match Kantara template](#)