

---

# Kantara Initiative eGovernment Implementation Profile of SAML V2.0

## Version 2.0

### Working Draft 05 May 3, 2010

#### Document identifier:

draft-kantara-egov-saml2-profile-2.0

#### Location:

TBD

#### Editors:

Scott Cantor, Internet2

#### Contributors:

Kantara eGovernment WG

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.

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

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

# 40 Table of Contents

41	1 Introduction.....	3
42	1.1 Notation.....	3
43	1.2 Normative References.....	4
44	2 SAML V2.0 Implementation Profile.....	6
45	2.1 Required Information.....	6
46	2.2 Metadata and Trust Management.....	6
47	2.2.1 Metadata Profiles.....	6
48	2.2.2 Metadata Exchange.....	7
49	2.2.2.1 Metadata Verification.....	7
50	2.3 Name Identifiers.....	7
51	2.4 Attributes.....	7
52	2.5 Browser Single Sign-On.....	8
53	2.5.1 Identity Provider Discovery.....	8
54	2.5.2 Authentication Requests.....	8
55	2.5.2.1 Binding and Security Requirements.....	8
56	2.5.2.2 Message Content.....	8
57	2.5.3 Responses.....	9
58	2.5.3.1 Binding and Security Requirements.....	9
59	2.5.3.2 Message Content.....	9
60	2.5.4 Artifact Resolution.....	9
61	2.5.4.1 Artifact Resolution Requests.....	10
62	2.5.4.2 Artifact Resolution Responses.....	10
63	2.6 Browser Holder of Key Single Sign-On.....	10
64	2.7 SAML 2.0 Proxying.....	10
65	2.7.1 Authentication Requests.....	10
66	2.7.2 Responses.....	10
67	2.8 Single Logout.....	11
68	2.8.1 Logout Requests.....	11
69	2.8.1.1 Binding and Security Requirements.....	11
70	2.8.1.2 User Interface Behavior.....	11
71	2.8.2 Logout Responses.....	11
72	2.8.2.1 Binding and Security Requirements.....	11
73	3 Conformance Classes.....	13
74	3.1 Standard.....	13
75	3.2 Standard with Logout.....	13
76	3.3 Full.....	13
77	Appendix A. Open Issues.....	14
78	Appendix B. Change Log.....	15
79		

---

# 1 Introduction

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

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

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

## 1.1 Notation

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

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

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

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

Listings of XML schemas appear like this.

Example code listings appear like this.

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

- The prefix `saml2:` stands for the SAML 2.0 assertion namespace, `urn:oasis:names:tc:SAML:2.0:assertion`
- The prefix `saml2p:` stands for the SAML 2.0 protocol namespace, `urn:oasis:names:tc:SAML:2.0:protocol`
- The prefix `md:` stands for the SAML 2.0 metadata namespace, `urn:oasis:names:tc:SAML:2.0:metadata`
- The prefix `idpdisc:` stands for the Identity Provider Discovery Service Protocol and Profile [IdPDisco] namespace, `urn:oasis:names:tc:SAML:profiles:SSO:idp-discovery-protocol`

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

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

## 128 1.2 Normative References

- 129 **[RFC2119]** IETF RFC 2119, *Key words for use in RFCs to Indicate Requirement Levels*,  
130 March 1997. <http://www.ietf.org/rfc/rfc2119.txt>
- 131 **[RFC2560]** IETF RFC 2560, *X.509 Internet Public Key Infrastructure Online Certificate Status*  
132 *Protocol*, June 1999. <http://www.ietf.org/rfc/rfc2560.txt>
- 133 **[RFC2616]** IETF RFC 2616, *Hypertext Transfer Protocol – HTTP/1.1*, June 1999.  
134 <http://www.ietf.org/rfc/rfc2616.txt>
- 135 **[RFC2818]** IETF RFC 2818, *HTTP Over TLS*, May 2000. <http://www.ietf.org/rfc/rfc2818.txt>
- 136 **[RFC4051]** IETF RFC 4051, *Additional XML Security Uniform Resource Identifiers*, April  
137 2005. <http://www.ietf.org/rfc/rfc4051.txt>
- 138 **[RFC5280]** IETF RFC 5280, *Internet X.509 Public Key Infrastructure Certificate and*  
139 *Certificate Revocation List (CRL) Profile*, May 2008.  
140 <http://www.ietf.org/rfc/rfc5280.txt>
- 141 **[HoKSSO]** OASIS Committee Specification, *SAML V2.0 Holder-of-Key Web Browser SSO*  
142 *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)  
143 [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)
- 144 **[IdPDisco]** OASIS Committee Specification, *Identity Provider Discovery Service Protocol*  
145 *and Profile*, March 2008. [http://docs.oasis-open.org/security/saml/Post2.0/sstc-](http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-idp-discovery.pdf)  
146 [saml-idp-discovery.pdf](http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-idp-discovery.pdf)
- 147 **[MetaAttr]** OASIS Committee Specification, *SAML V2.0 Metadata Extension for Entity*  
148 *Attributes Version 1.0*, August 2009. [http://docs.oasis-](http://docs.oasis-open.org/security/saml/Post2.0/sstc-metadata-attr.pdf)  
149 [open.org/security/saml/Post2.0/sstc-metadata-attr.pdf](http://docs.oasis-open.org/security/saml/Post2.0/sstc-metadata-attr.pdf)
- 150 **[MetaIOP]** OASIS Committee Specification, *SAML V2.0 Metadata Interoperability Profile*  
151 *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)  
152 [metadata-iop.pdf](http://docs.oasis-open.org/security/saml/Post2.0/sstc-metadata-iop.pdf)
- 153 **[SAML2Core]** OASIS Standard, *Assertions and Protocols for the OASIS Security Assertion*  
154 *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)  
155 [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)
- 156 **[SAML2Meta]** OASIS Standard, *Metadata for the OASIS Security Assertion Markup Language*  
157 *(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)  
158 [metadata-2.0-os.pdf](http://docs.oasis-open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf)
- 159 **[SAML2Bind]** OASIS Standard, *Bindings for the OASIS Security Assertion Markup Language*  
160 *(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)  
161 [bindings-2.0-os.pdf](http://docs.oasis-open.org/security/saml/v2.0/saml-bindings-2.0-os.pdf)
- 162 **[SAML2Prof]** OASIS Standard, *Profiles for the OASIS Security Assertion Markup Language*  
163 *(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)  
164 [profiles-2.0-os.pdf](http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf)
- 165 **[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)  
166 [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)
- 167 **[SAML-X500]** OASIS Committee Specification, *SAML V2.0 X.500/LDAP Attribute Profile*, March  
168 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)  
169 [x500.pdf](http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-attribute-x500.pdf)

170 **[XMLEnc]** D. Eastlake et al. *XML Encryption Syntax and Processing*. World Wide Web  
171 Consortium Recommendation. [http://www.w3.org/TR/2002/REC-xmlenc-core-](http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/)  
172 [20021210/](http://www.w3.org/TR/2002/REC-xmlenc-core-20021210/)

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

## 176 **Non-Normative References**

177 **[eGov15]** Kyle Meadors, *Liberty Alliance eGov Profile for SAML 2.0 Version 1.5*.  
178 [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)  
179 [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)

---

## 2 SAML V2.0 Implementation Profile

180

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

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

### 2.1 Required Information

192

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

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

195 **Description:** Given below

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

### 2.2 Metadata and Trust Management

197

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

#### 2.2.1 Metadata Profiles

202

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

204 Implementations MUST also support an alternative to that profile's language on use of the  
205 `<md:KeyDescriptor>` element as follows:

- 206 • Implementations MUST support the `<ds:X509Certificate>` element as input to subsequent  
207 requirements. Support for other representations, and for other mechanisms for credential  
208 distribution, is OPTIONAL.
- 209 • Implementations MUST support some form of path validation of signing, TLS, and encryption  
210 credentials used to secure SAML exchanges against one or more trusted root certificates.  
211 Implementations SHOULD document the behavior of the validation mechanisms they employ.
- 212 • Implementations MUST support the use of OCSP [RFC2560] and certificate revocation lists  
213 (CRLs) obtained via the "CRL Distribution Point" X.509 extension [RFC5280] for revocation  
214 checking of those credentials.
- 215 • Implementations MAY support additional constraints on the contents of certificates used by  
216 particular entities, such as "subjectAltName" or "DN", key usage constraints, or policy extensions,  
217 but SHOULD document such features and make them optional to enable where possible.

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

## 221 **2.2.2 Metadata Exchange**

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

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

- 227 • local file
- 228 • remote resource at fixed location accessible via HTTP 1.1 [RFC2616] or HTTP 1.1 over TLS/SSL  
229 [RFC2818]

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

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

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

### 238 **2.2.2.1 Metadata Verification**

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

- 241 • direct comparison against known keys
- 242 • some form of path-based certificate validation against one or more trusted root certificates and  
243 certificate revocation lists

244 The latter mechanism does not impose a particular profile for certificate validation, but should be  
245 understood as being consistent with the "usual" practices encountered in the implementation of certificate  
246 validation. Where possible, implementations SHOULD document known limitations of the mechanisms  
247 they employ.

## 248 **2.3 Name Identifiers**

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

- 252 • `urn:oasis:names:tc:SAML:2.0:nameid-format:persistent`
- 253 • `urn:oasis:names:tc:SAML:2.0:nameid-format:transient`

254 Support for other formats is OPTIONAL.

## 255 **2.4 Attributes**

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

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

## 263 **2.5 Browser Single Sign-On**

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

### 265 **2.5.1 Identity Provider Discovery**

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

### 268 **2.5.2 Authentication Requests**

#### 269 **2.5.2.1 Binding and Security Requirements**

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

273 Support for other bindings is OPTIONAL.

#### 274 **2.5.2.2 Message Content**

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

- 278 • `AssertionConsumerServiceURL`
- 279 • `ProtocolBinding`
- 280 • `ForceAuthn`
- 281 • `IsPassive`
- 282 • `AttributeConsumingServiceIndex`
- 283 • `<saml2p:RequestedAuthnContext>`
- 284 • `<saml2p:NameIDPolicy>`

285 Identity Provider implementations MUST support all `<saml2p:AuthnRequest>` child elements and  
286 attributes defined by [SAML2Core], but MAY provide that support in the form of returning appropriate  
287 errors when confronted by particular request options. However, implementations MUST fully support the  
288 options enumerated above.

289 Implementations MAY limit their support of the `<saml2p:RequestedAuthnContext>` element to the  
290 value "exact" for the `Comparison` attribute.



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

## 296 **2.5.3 Responses**

### 297 **2.5.3.1 Binding and Security Requirements**

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

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

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

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

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

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

### 317 **2.5.3.2 Message Content**

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

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

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

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

## 329 **2.5.4 Artifact Resolution**

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

### 333 **2.5.4.1 Artifact Resolution Requests**

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

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

### 340 **2.5.4.2 Artifact Resolution Responses**

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

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

## 347 **2.6 Browser Holder of Key Single Sign-On**

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

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

## 352 **2.7 SAML 2.0 Proxying**

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

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

### 359 **2.7.1 Authentication Requests**

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

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

## 366 **2.7.2 Responses**

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

370 Proxying Identity Provider implementations MUST support the suppression of  
371 `<saml2:AuthenticatingAuthority>` elements from outgoing `<saml2:AuthnContext>` elements  
372 to allow for hiding the identity of the proxied Identity Provider from Service Providers.

## 373 **2.8 Single Logout**

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

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

### 378 **2.8.1 Logout Requests**

#### 379 **2.8.1.1 Binding and Security Requirements**

380 Identity Provider implementations MUST support the SAML SOAP (using HTTP as a transport) binding  
381 [SAML2Bind] for the issuance of `<saml2p:LogoutRequest>` messages, and MUST support the SAML  
382 SOAP (using HTTP as a transport) and HTTP-Redirect bindings [SAML2Bind] for the reception of  
383 `<saml2p:LogoutRequest>` messages.

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

386 Support for other bindings is OPTIONAL.

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

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

#### 392 **2.8.1.2 User Interface Behavior**

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

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

400 TBD: Requirements on administrative logout (i.e., not the user)?

401 **2.8.2 Logout Responses**

402 **2.8.2.1 Binding and Security Requirements**

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

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

409 Support for other bindings is OPTIONAL.

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

---

413 **3 Conformance Classes**

414 **3.1 Standard**

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

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

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

421 **3.2 Standard with Logout**

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

424 **3.3 Full**

425 Conforming Identity Provider and/or Service Provider implementations MUST meet the conformance  
426 requirements in section 3.1, and MUST in addition support the normative requirements in sections 2.6,  
427 2.7, and 2.8.

---

428

## Appendix A. Open Issues

429

430

- Single logout language around UI and consent needs review, and need text on administrative logout.

431

- Need to bump HoK reference to new profile version if it reaches CS-02

---

432

## Appendix B. Change Log

433

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

434

435

436

437

- 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

438

439

440

- 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

441

442

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

443

444

445

- 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