



Holder-of-Key Web Browser SSO Profile

Working Draft 01

23 February 2008

Specification URIs:

This Version:

<http://docs.oasis-open.org/security/saml/Post2.0/ssstc-saml-holder-of-key-browser-ssso-wd-01.html>

<http://docs.oasis-open.org/security/saml/Post2.0/ssstc-saml-holder-of-key-browser-ssso-wd-01.odt>

<http://docs.oasis-open.org/security/saml/Post2.0/ssstc-saml-holder-of-key-browser-ssso-wd-01.pdf>

Latest Version:

<http://docs.oasis-open.org/security/saml/Post2.0/ssstc-saml-holder-of-key-browser-ssso.html>

<http://docs.oasis-open.org/security/saml/Post2.0/ssstc-saml-holder-of-key-browser-ssso.odt>

<http://docs.oasis-open.org/security/saml/Post2.0/ssstc-saml-holder-of-key-browser-ssso.pdf>

Technical Committee:

OASIS Security Services TC

Chair(s):

Hal Lockhart, BEA Systems, Inc.

Brian Campbell, Ping Identity Corporation

Editor(s):

Nate Klingenstein, Internet2

Related Work:

This specification is an alternative to the SAML V2.0 Web Browser SSO Profile in the SAML V2.0 Profiles specification [SAML2Prof].

Declared XML Namespace(s):

urn:oasis:names:tc:SAML:2.0:profiles:SSO:browser:holder-of-key

Abstract:

This profile allows for transport and validation of holder-of-key assertions by standard HTTP user agents with no modification of client software and maximum compatibility with existing deployments. Most of the flows are as in standard Web Browser SSO, but an x.509 certificate presented by the user agent supplies a valid keypair through client TLS authentication for HTTP transactions. The keypair is used as the vehicle for holder-of-key validation of a SAML assertion. This strengthens the assurance of the resulting authentication context and protects against credential theft, giving the service provider fresh authentication and attribute information without requiring it to perform successful PKIX validation of the certificate.

35 **Status:**

36 This document was last revised or approved by the SSTC on the above date. The level of
37 approval is also listed above. Check the "Latest Version" or "Latest Approved Version" location
38 noted above for possible later revisions of this document.

39 Technical Committee members should send comments on this specification to the Technical
40 Committee's email list. Others should send comments to the Technical Committee by using the
41 "Send A Comment" button on the Technical Committee's web page at [http://www.oasis-
open.org/committees/security](http://www.oasis-
42 open.org/committees/security).

43 For information on whether any patents have been disclosed that may be essential to
44 implementing this specification, and any offers of patent licensing terms, please refer to the
45 Intellectual Property Rights section of the Technical Committee web page ([http://www.oasis-
open.org/committees/security/ipr.php](http://www.oasis-
46 open.org/committees/security/ipr.php)).

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

49 Notices

50 Copyright © OASIS® 2008. All Rights Reserved.

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

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

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

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

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

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

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

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

93 Table of Contents

94	1 Introduction.....	5
95	1.1 Terminology.....	5
96	1.2 Normative References.....	6
97	1.3 Conformance.....	6
98	1.3.1 Holder-of-Key Web Browser SSO Profile.....	6
99	2 Holder-of-Key Web Browser SSO Profile.....	7
100	2.1 Required Information.....	7
101	2.2 Background.....	7
102	2.3 Profile Overview.....	8
103	2.4 Profile Description.....	9
104	2.4.1 HTTP Request to Service Provider.....	9
105	2.4.2 Service Provider Determines Identity Provider.....	10
106	2.4.3 <samlp:AuthnRequest> Issued by Service Provider to Identity Provider.....	10
107	2.4.4 Identity Provider Identifies Principal and Verifies Key Possession.....	10
108	2.4.5 Identity Provider Issues <samlp:Response> to Service Provider.....	11
109	2.4.6 Service Provider Grants or Denies Access to Principal.....	11
110	2.5 Use of Authentication Request Protocol.....	11
111	2.5.1 <samlp:AuthnRequest> Usage.....	12
112	2.5.2 <samlp:AuthnRequest> Message Processing Rules.....	12
113	2.5.3 <samlp:Response> Usage.....	12
114	2.5.4 <samlp:Response> Message Processing Rules.....	13
115	2.5.4.1 Artifact-Specific <samlp:Response> Message Processing Rules.....	14
116	2.5.4.2 POST-Specific <samlp:Response> Message Processing Rules.....	14
117	2.6 Unsolicited Responses.....	14
118	2.7 Use of Metadata.....	14
119	2.8 Compatibility.....	15
120		

121

1 Introduction

122 In the scenario addressed by this profile, which is an extended version of the Web Browser SSO Profile
 123 in 4.1 of [SAML2Prof], a principal uses an HTTP user agent to either access a web-based resource at a
 124 service provider or access an identity provider such that the service provider and desired resource are
 125 understood or implicit. In either case, the user agent needs to acquire a SAML assertion from the identity
 126 provider. The user agent makes a request to the identity provider using client TLS authentication. The
 127 X.509 certificate supplied in this transaction is used primarily to supply a public key that is associated with
 128 the principal. The identity provider authenticates the principal by way of this TLS authentication or any
 129 other method of its choice. The identity provider then produces a response containing at least an
 130 assertion with holder-of-key subject confirmation and an authentication statement for the user agent to
 131 transport to the service provider. This assertion is presented by the user agent to the service provider
 132 over client TLS authentication to prove possession of the private key matching the holder-of-key
 133 confirmation in the assertion. The service provider should rely on no information from the certificate
 134 beyond the key; instead, it consumes the assertion to create a security context. The TLS key may then
 135 be used to persist the security context rather than a cookie or other application-layer session.

136 To implement this scenario, a profile of the SAML Authentication Request protocol is used in conjunction
 137 with the HTTP Redirect, HTTP POST and HTTP Artifact bindings. It is assumed that the user is using an
 138 HTTP user agent capable of presenting client certificates during TLS session establishment, such as a
 139 standard web browser.

1.1 Terminology

141 The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD
 142 NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this specification are to be interpreted as
 143 described in [RFC 2119].

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

147 Conventional XML namespace prefixes are used throughout this specification to stand for their respective
 148 namespaces as follows:

Prefix	XML Namespace	Comments
ds:	http://www.w3.org/2000/09/xmldsig#.	This is the XML-Dsig Schema defined in [DSig].
md:	urn:oasis:names:tc:SAML:2.0:metadata	This is the SAML V2.0 metadata namespace defined in the SAML V2.0 metadata specification [SAML2Meta].
saml:	urn:oasis:names:tc:SAML:2.0:assertion	This is the SAML V2.0 assertion namespace defined in the SAML V2.0 core specification [SAML2Core].
samlp:	urn:oasis:names:tc:SAML:2.0:protocol	This is the SAML V2.0 protocol namespace defined in the SAML V2.0 core specification [SAML2Core].

149

150 This specification uses the following typographical conventions in text: <namespace:Element>,
 151 Attribute, **Datatype**, OtherKeyword.

152 1.2 Normative References

- 153 **[DSig]** D. Eastlake, J. Reagle, D. Solo. *XML-Signature Syntax and Processing*. World
154 Wide Web Consortium Recommendation, 12 February 2002. See
155 <http://www.w3.org/TR/xmldsig-core/>.
- 156 **[IDPDisco]** R. Widdowson, S. Cantor. Identity Provider Discovery Service Protocol and
157 Profile, OASIS SSTC October 2007. Document ID sstc-saml-idp-discovery. See
158 <http://www.oasis-open.org/committees/security/>.
- 159 **[RFC 2119]** S. Bradner. *Key words for use in RFCs to Indicate Requirement Levels*. IETF
160 RFC 2119, March 1997. <http://www.ietf.org/rfc/rfc2119.txt>.
- 161 **[RFC 4346]** T. Dierks, E. Rescorla. *The Transport Layer Security (TLS) Protocol*. IETF RFC
162 4346, April 2006.
163 <http://www.ietf.org/rfc/rfc4346.txt>.
- 164 **[SAML2Bind]** S. Cantor et al. *Assertions and Protocols for the OASIS Security Assertion
165 Markup Language (SAML) V2.0*. OASIS Standard, March 2005. Document ID
166 saml-core-2.0-os. See [http://docs.oasis-open.org/security/saml/v2.0/saml-
167 bindings-2.0-os.pdf](http://docs.oasis-open.org/security/saml/v2.0/saml-bindings-2.0-os.pdf).
- 168 **[SAML2Core]** S. Cantor et al. *Assertions and Protocols for the OASIS Security Assertion
169 Markup Language (SAML) V2.0*. OASIS Standard, March 2005. Document ID
170 saml-core-2.0-os. See [http://docs.oasis-open.org/security/saml/v2.0/saml-
171 core-2.0-os.pdf](http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf).
- 172 **[SAML2Meta]** S. Cantor et al. *Metadata for the OASIS Security Assertion Markup Language
173 (SAML) V2.0*. OASIS Standard, March 2005. Document ID saml-metadata-2.0-
174 os. See <http://docs.oasis-open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf>.
- 175 **[SAML2Prof]** S. Cantor et al. *Profiles for the OASIS Security Assertion Markup Language
176 (SAML) V2.0*. OASIS Standard, March 2005. Document ID saml-profiles-2.0-os.
177 See <http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf>.
- 178 **[SAML2Secure]** F. Hirsch et al. *Security and Privacy Considerations for the OASIS Security
179 Assertion Markup Language (SAML) v2.0*. OASIS SSTC, March 2005.
180 Document ID saml-sec-consider-2.0-os. See [http://docs.oasis-
open.org/security/saml/v2.0/saml-sec-consider-2.0-os.pdf](http://docs.oasis-
181 open.org/security/saml/v2.0/saml-sec-consider-2.0-os.pdf).

182 1.3 Conformance

183 1.3.1 Holder-of-Key Web Browser SSO Profile

184 A conformant implementation of a service provider and an identity provider MUST support holder-of-key
185 assertions and the acquisition of client keys from TLS connections, for validation and issuance of these
186 assertions, respectively.

2 Holder-of-Key Web Browser SSO Profile

2.1 Required Information

Identification: urn:oasis:names:tc:SAML:2.0:profiles:SSO:browser:holder-of-key

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

SAML Confirmation Method Identifiers: The SAML V2.0 “holder-of-key” confirmation method identifier, urn:oasis:names:tc:SAML:2.0:cm:holder-of-key, is included in all assertions issued under this profile. Imbedded in the <saml:SubjectConfirmation> element is an x.509 certificate using XML-Signature <ds:KeyInfo> with identifier <http://www.w3.org/2000/09/xmldsig#>.

Description: Given below.

Updates: Provides an alternative to the SAML V2.0 Web Browser SSO Profile given in 4.1 of [SAML2Prof].

2.2 Background

This profile is designed to enhance the security of SAML assertion and message exchange without requiring modifications to client software while improving the user experience. The amount of benefit depends on the alignment of the certificate with the discovery service and identity provider and the extent to which a service provider has been enabled. Deployments should minimize user interaction and avoid mutually conflicting CA requirements by coordinating certificate issuance and TLS configuration.

If both the identity provider and service provider use this profile, but assume no knowledge of the certificate's contents, enhanced security is the primary benefit. There is a small chance that a bearer token will be stolen in transit, as described in [SAML2Secure]. Confirming that the presenter of the token is the intended holder through public key cryptography virtually eliminates this chance, improving the viability of SAML-based HTTP SSO for highly sensitive applications. The session created by the service provider in the security context resulting from the Holder-of-Key Web Browser SSO Profile can be keyed by the TLS public key or session key. Application-layer sessions, such as maintained by cookies, are often poorly protected by user agents, allowing for theft of this session and impersonation of the user.

If a certificate can be used by the identity provider for principal authentication, there is no need for the user to further confirm its identity, and potentially no user interaction is needed. Phishing is eliminated, as there are greater challenges and no benefits to tricking the user into authenticating with legitimate credentials to a fraudulent party.

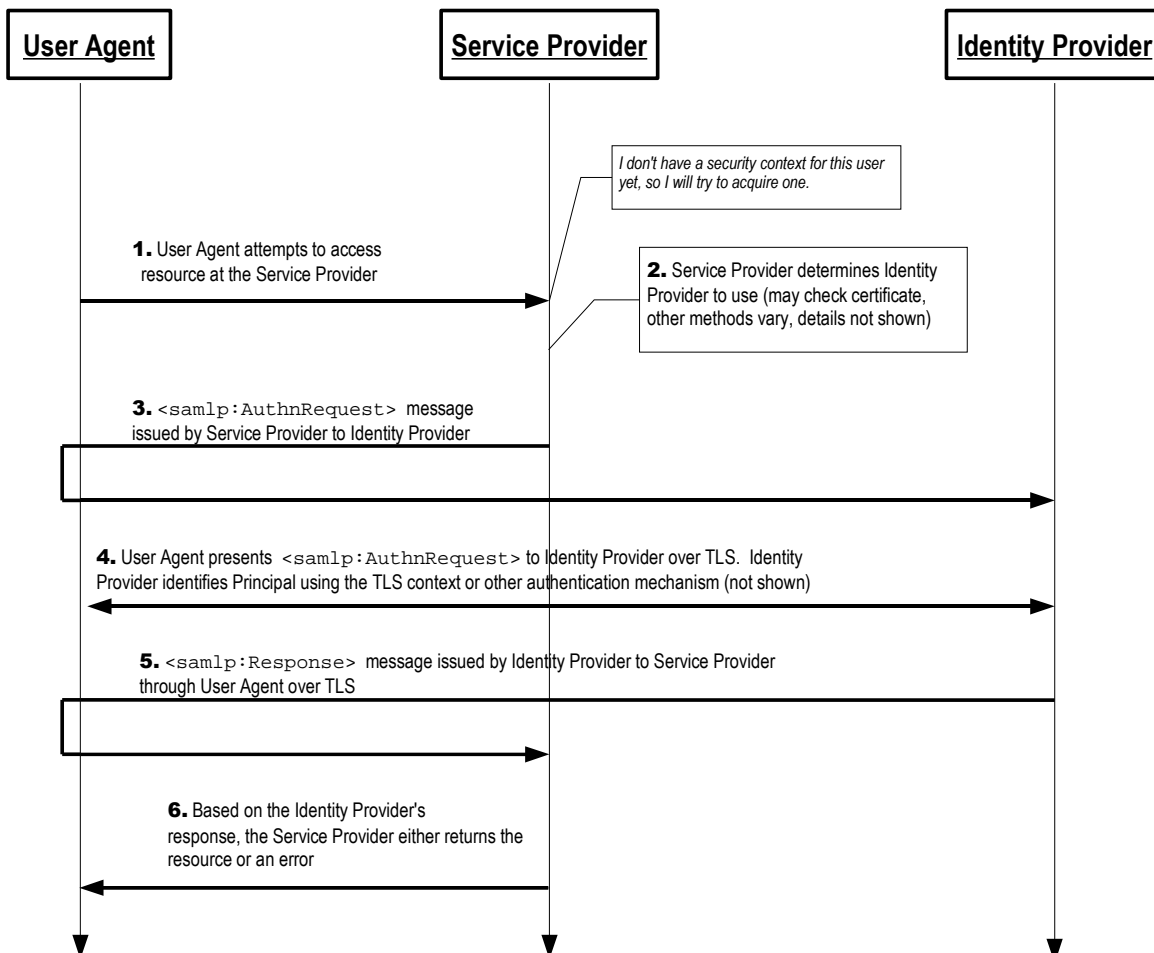
Further, if the user accesses the service provider first, discovery of the user's identity provider may be performed by matching fields within the certificate presented; however, that is beyond the scope of this specification.

This profile offers meaningful advantages over traditional PKI, as well. There is no requirement for a mutually or universally trusted root, distributed OCSP or CRL-based revocation, a globally unique namespace, PKIX validation (particularly by the SP), or for all participants in SSO to utilize X.509. The authentication token can be customized for every transaction, including fresh attributes and appropriate revelation of identity.

There are limitations on the degree to which users can remain private under this profile, particularly as most end-user X.509 certificates have a unique distinguished name for the subject regularly containing personally identifying information. Additional information about the subject may be implicitly revealed through the issuer. The ideal certificate for use with this profile contains a pseudonym for the user as subject that the identity provider can map to a principal, the domain of the identity provider included in the subject, and optionally the unique SAML entityID of the identity provider included in the certificate as an X.509 subjectAltName. However, even in this case it's not generally feasible for the

231 user to remain truly anonymous, as transient identifiers and short-lived assertions permit, unless a new
 232 keypair is issued for every transaction. The public key is a de-facto persistent ID, as discussed in
 233 [SAML2Secure].

234 **2.3 Profile Overview**



235 Figure 1 illustrates the basic template for achieving SSO. The following steps are described by the
 236 profile. Within an individual step, there may be one or more actual message exchanges depending on
 237 the binding used for that step and other implementation-dependent behavior.

238 **1. HTTP Request to Service Provider**

239 The principal, via an HTTP user agent, makes an HTTP request for a secured resource at the service
 240 provider. The service provider determines that no security context exists, and attempts to create
 241 one.

242 **2. Service Provider Determines Identity Provider**

243 The service provider determines the proper identity provider to which to direct the user agent. This
 244 may be done through use of a discovery service as described in [IDPDisco], by examining fields in a
 245 certificate presented through client TLS authentication, such the X.509 `subject` or
 246 `subjectAltName`, or by any other means appropriate.

247 3. <samlp:AuthnRequest> issued by Service Provider to Identity Provider

248 The service provider issues a <samlp:AuthnRequest> message to be delivered by the user agent
249 to the identity provider. If the initial HTTP Request for a resource protected by the service provider
250 was made over client TLS authentication and the <samlp:AuthnRequest> will be signed, the
251 service provider MAY include the certificate presented by the client for holder-of-key
252 <saml:SubjectConfirmation>. The HTTP Redirect, HTTP POST, or HTTP Artifact binding can
253 be used to transport the message to the identity provider through the user agent, unless holder-of-
254 key <saml:SubjectConfirmation> is included, in which case HTTP Redirect MAY NOT be used.

255 4. Identity Provider identifies Principal

256 The principal is identified by the identity provider. The identity provider MUST identify the principal
257 using any authentication method at its discretion honoring any requirements imposed by the service
258 provider in the <samlp:AuthnRequest>, including validation of the certificate presented in client
259 TLS authentication. However, the identity provider MUST establish that the private key
260 corresponding to the public key that will be included for holder-of-key proofing is held by this user
261 agent, typically through a successful TLS handshake.

262 5. Identity Provider issues <samlp:Response> to Service Provider

263 The identity provider issues a <samlp:Response> message to be delivered by the user agent to the
264 service provider. Either the HTTP POST or HTTP Artifact binding can be used to transfer the
265 message to the service provider through the user agent. The message may indicate an error or will
266 include at least an authentication statement in an assertion with holder-of-key
267 <saml:SubjectConfirmation> containing a <ds:KeyInfo> element containing the public key of
268 principal. The HTTP Redirect binding MUST NOT be used, as the response will typically exceed the
269 URL length permitted by most user agents.

270 6. Service Provider grants or denies access to Principal

271 The response is received by the service provider, which can respond to the principal's user agent
272 with its own error, an error passed by the identity provider, or establish a security context for the
273 principal and return the requested resource.

274 Note that an identity provider can initiate this profile at step 5 by issuing a <samlp:Response> message
275 to a service provider without the preceding steps.

276 2.4 Profile Description

277 If the profile is initiated by the service provider, start with Section 2.4.1. If initiated by the identity
278 provider, start with Section 2.4.5. The descriptions refer to a Single Sign-On Service and Assertion
279 Consumer Service in accordance with their use in section 4.1.3 of [SAML2Prof].

280 2.4.1 HTTP Request to Service Provider

281 The profile may be initiated by an arbitrary request to the service provider. The service provider is free to
282 use any means it wishes to associate the subsequent interactions with the original request. Each of the
283 bindings provides a RelayState mechanism that the service provider MAY use to associate the profile
284 exchange with the original request. In particular, the TLS session itself MAY be used.

285 2.4.2 Service Provider Determines Identity Provider

286 The service provider determines the primary identity provider with which the principal is associated
287 through a variety of mechanisms as selected by the service provider implementation or deployment. The
288 service provider MAY check the certificate presented by the user agent, to attempt to use the `x.509`
289 `subject`, `subjectAltName`, or other field or extension in the certificate to determine the principal's
290 identity provider or single sign-on service endpoint. The common domain cookie approach described in
291 4.3 of [SAML2Prof], a discovery service as described in [IDPDisco], or other mechanism MAY be used if
292 the correct identity provider cannot be determined through inspection of the certificate.

293 2.4.3 <samlp:AuthnRequest> Issued by Service Provider to Identity 294 Provider

295 Once an identity provider is selected, the location of a single sign-on service to which to send an
296 <samlp:AuthnRequest> is determined based on the SAML binding chosen by the service provider.
297 Metadata as described in [SAML2Meta] MAY be used for this purpose. Following an HTTP request by
298 the user agent, an HTTP response is returned containing an <samlp:AuthnRequest> message or an
299 artifact, depending on the SAML binding used, to be delivered to the identity provider's single sign-on
300 service.

301 Profile-specific rules for the contents of the <samlp:AuthnRequest> are defined in Section 2.5.1. If
302 the HTTP Redirect or POST binding is used, the <samlp:AuthnRequest> message is delivered
303 directly to the identity provider in this step. If the HTTP Artifact binding is used, the Artifact Resolution
304 profile defined in Section 5 of [SAML2Prof] is used by the identity provider, which makes a callback to the
305 service provider to retrieve the <samlp:AuthnRequest> message using, for example, the SOAP
306 binding.

307 The <samlp:AuthnRequest> message MAY be signed if authentication of the request issuer is
308 required. If a certificate is included in the request, the HTTP Redirect binding MUST NOT be used to
309 transport the <samlp:AuthnRequest> due to size limitations.

310 It is REQUIRED that the <samlp:AuthnRequest> be presented to the identity provider over mutually
311 authenticated TLS to supply the identity provider with a public key associated with the user agent and
312 establish the user agent's possession of the corresponding private key.

313 2.4.4 Identity Provider Identifies Principal and Verifies Key Possession

314 The identity provider must perform two functions in this step: identification of the principal presenting the
315 <samlp:AuthnRequest>, and verification that the principal possesses the private key associated with
316 the public key that will be included in the <saml:SubjectConfirmation>.

317 The identity provider MUST establish the identity of the principal (unless it will return an error) prior to the
318 issuance of the <samlp:Response>. If the <samlp:AuthnRequest> attribute `ForceAuthn` is
319 present and true, the identity provider MUST freshly establish this identity rather than relying on any
320 existing session it may have with the principal. Otherwise, and in all other respects, the identity provider
321 may use any means to authenticate the user agent, subject to any requirements included in the
322 <samlp:AuthnRequest>.

323 The identity provider MUST also establish that the public key that will be included as a holder-of-key
324 <saml:SubjectConfirmation> in the subsequent <samlp:Response> is the one presented by the
325 user agent in step 2.4.3. The user agent MUST have demonstrated possession of this key through
326 successful TLS authentication.

327 Preferably, both of these requirements will be simultaneously addressed by validation of an x.509
328 certificate presented by the user agent in TLS authentication from an issuer trusted by the identity
329 provider, but this is not mandatory unless such an authentication context is requested by the service
330 provider.

331 **2.4.5 Identity Provider Issues <samlp:Response> to Service Provider**

332 Regardless of the success or failure of the <samlp:AuthnRequest>, the identity provider SHOULD
333 produce an HTTP response to the user agent containing a <samlp:Response> message or an artifact,
334 depending on the SAML binding used, to be delivered to the service provider's assertion consumer
335 service.

336 The exact format of this HTTP response and the subsequent HTTP request to the assertion consumer
337 service is defined by [SAML2Bind]. Profile-specific rules on the contents of the <samlp:Response> are
338 included in section 2.5.2. If the HTTP POST binding is used, the <samlp:Response> message is
339 delivered directly to the service provider in this step. If the HTTP Artifact binding is used, the Artifact
340 Resolution profile defined in Section 5 is used by the service provider, which makes a callback to the
341 identity provider to retrieve the <samlp:Response> message, using for example the SOAP binding.

342 The location of the assertion consumer service MAY be determined using metadata defined in
343 [SAML2Meta]. The identity provider MUST have some means to establish that this location is in fact
344 controlled by the service provider. A service provider MAY indicate the SAML binding and the specific
345 assertion consumer service to use in its <samlp:AuthnRequest> and the identity provider MUST honor
346 them if it can.

347 It is REQUIRED that the HTTP requests in this step be made over mutually authenticated TLS to
348 demonstrate possession of the private key corresponding to the public key included in the assertion's
349 <saml:SubjectConfirmation> as well as maintain confidentiality and message integrity. The
350 <saml:Assertion> element(s) in the <samlp:Response> MUST be signed, if the HTTP POST
351 binding is used, and MAY be signed if the HTTP Artifact binding is used.

352 The service provider MUST process the <samlp:Response> message and any enclosed
353 <saml:Assertion> elements as described in [SAML2Core].

354 **2.4.6 Service Provider Grants or Denies Access to Principal**

355 To complete the profile, the service provider processes the <samlp:Response> and
356 <saml:Assertion>(s) and grants or denies access to the resource. The service provider MAY
357 establish a security context with the user agent using any session mechanism it chooses. Any
358 subsequent use of the <saml:Assertion>(s) provided is at the discretion of the service provider and
359 other relying parties, subject to any restrictions on use contained within them.

360 **2.5 Use of Authentication Request Protocol**

361 This profile is based upon the Web Browser SSO Profile defined in [SAML2Prof] and the Authentication
362 Request protocol defined in [SAML2Core]. In the nomenclature of actors enumerated in Section 3.4 of
363 that document, the service provider is the request issuer and the relying party, the user agent is the
364 attesting entity and presenter, and the principal is the requested subject. There may be additional relying
365 parties at the discretion of the identity provider.

366 2.5.1 <samlp:AuthnRequest> Usage

367 A service provider MAY include any message content described in [SAML2Core], Section 3.4.1. All
368 processing rules are as defined in [SAML2Core]. The request MUST conform to the following:

- 369 ● The <saml:Issuer> element MUST be present and MUST contain the unique identifier of the
370 requesting service provider. The Format attribute MUST be omitted or have a value of
371 urn:oasis:names:tc:SAML:2.0:nameid-format:entity.
- 372 ● If the initial request was made over TLS and this message is signed, a<saml:Subject>
373 element MAY be included in the request that includes the certificate presented by the user agent
374 for which the service provider wishes to receive an assertion in a holder-of-key
375 <saml:SubjectConfirmation> element. A <saml:NameID> SHOULD NOT be included, as
376 the names used by the certificate authority may differ from those used by the identity provider. If
377 the user agent fails this confirmation, then the identity provider MUST respond with a
378 <samlp:Response> message containing an error status and no assertions.
- 379 ● If the service provider wishes to permit the identity provider to establish a new identifier for the
380 principal if none exists, it MUST include a <saml:NameIDPolicy> element with the
381 AllowCreate attribute set to true. Otherwise, only a principal for whom the identity provider
382 has previously established an identifier usable by the service provider can be authenticated
383 successfully.
- 384 ● The <samlp:AuthnRequest> message MAY be signed (as directed by the SAML binding
385 used). If the HTTP Artifact binding is used, authentication of the parties is OPTIONAL and any
386 mechanism permitted by the binding MAY be used.

387 2.5.2 <samlp:AuthnRequest> Message Processing Rules

388 If the identity provider cannot or will not satisfy the request, it MUST respond with a message containing
389 an appropriate error status code or codes.

390 If the <samlp:AuthnRequest> is not authenticated and/or integrity protected, the information in it
391 MUST NOT be trusted except as advisory. The <samlp:AuthnRequest> must be processed as
392 follows:

- 393 ● It is RECOMMENDED that any AssertionConsumerServiceURL or
394 AssertionConsumerServiceIndex attributes in the <samlp:AuthnRequest> are verified
395 as belonging to the entityID to whom the response will be sent. However, holder-of-key
396 confirmation eliminates the potential for assertion theft and encryption prevents privacy loss.
397 Encrypted assertions issued under this profile do NOT require this verification.
- 398 ● It is NOT obligated to honor the requested set of <saml:Conditions> in the
399 <samlp:AuthnRequest>, if any.

400 2.5.3 <samlp:Response> Usage

401 If the identity provider wishes to return an error for this request, it MUST NOT include any assertions in
402 the <samlp:Response> message. Otherwise, if the request is successful or the response is not
403 associated with a request, the <samlp:Response> element MUST conform to the following:

- 404 ● The <saml:Issuer> element of the <samlp:Response> MAY be omitted, but if present it
405 MUST contain the unique identifier of the issuing identity provider; the Format attribute MUST be
406 omitted or have a value of urn:oasis:names:tc:SAML:2.0:nameid-format:entity.

- 407 ● It MUST contain at least one `<saml:Assertion>`. Each assertion's `<saml:Issuer>` element
408 MUST contain the unique identifier of the issuing identity provider, and the `Format` attribute
409 MUST be omitted or have a value of `urn:oasis:names:tc:SAML:2.0:nameid-`
410 `format:entity`.
- 411 ● The set of one or more assertions MUST collectively contain one `<saml:AuthnStatement>`
412 that reflects the authentication of the principal to the identity provider.
- 413 ● The assertion containing an `<saml:AuthnStatement>` MUST also contain a
414 `<saml:Subject>` element with exactly one `<saml:SubjectConfirmation>` element with a
415 Method of `urn:oasis:names:tc:SAML:2.0:cm:holder-of-key`. Its
416 `<saml:SubjectConfirmationData>` MUST contain the public key of the principal. This will
417 typically take the form of a `<ds:KeyInfo>` element containing a `<ds:X509Data>` element with
418 the principal's certificate encoded inside.
- 419 ● If the identity provider supports the Single Logout profile, defined in Section 4.4 of [SAML2Prof],
420 the `<saml:AuthnStatement>` MUST include a `SessionIndex` attribute to enable per-session
421 logout requests by the service provider.
- 422 ● Additional statements MAY be included in the assertion(s) at the discretion of the identity
423 provider. The `<samlp:AuthnRequest>` MAY contain an
424 `AttributeConsumingServiceIndex` XML attribute referencing information about desired or
425 required attributes in [SAML2Meta]. The identity provider MAY ignore this, or send other
426 attributes at its discretion.
- 427 ● The assertion containing the `<saml:AuthnStatement>` MUST contain a
428 `<saml:AudienceRestriction>` including the service provider's unique identifier as a
429 `<saml:Audience>`.
- 430 ● Other conditions (and other `<saml:Audience>` elements) MAY be included as requested by the
431 service provider or at the discretion of the identity provider. All such conditions MUST be
432 understood by and accepted by the service provider in order for the assertion to be considered
433 valid.

434 **2.5.4 `<samlp:Response>` Message Processing Rules**

435 Regardless of the SAML binding used, the service provider MUST do the following:

- 436 ● Verify any signatures present on the assertion(s) or the response.
- 437 ● Verify that the key in the certificate presented by the user agent in mutual TLS authentication to
438 the service provider matches the public key in the holder-of-key
439 `<saml:SubjectConfirmationData>`. The service provider SHOULD NOT rely on any other
440 data in the certificate to process the assertion.
- 441 ● Verify that any assertions relied upon are valid in other respects.

442 Any assertion which is not valid, or whose subject confirmation requirements cannot be met, SHOULD be
443 discarded and SHOULD NOT be used to establish a security context for the principal.

444 **2.5.4.1 Artifact-Specific <samlp:Response> Message Processing** 445 **Rules**

446 If the HTTP Artifact binding is used to deliver the <samlp:Response>, the dereferencing of the artifact
447 using the Artifact Resolution profile MUST be mutually authenticated, integrity protected, and confidential.

448 If the assertion is not encrypted, it is RECOMMENDED that the identity provider ensure that only the
449 service provider to whom the <samlp:Response> message has been issued is given the message as
450 the result of an <samlp:ArtifactResolve> request.

451 Either the SAML binding used to dereference the artifact or message signatures can be used to
452 authenticate the parties and protect the messages.

453 **2.5.4.2 POST-Specific <samlp:Response> Message Processing Rules**

454 If the HTTP POST binding is used to deliver the <samlp:Response>, the enclosed assertion(s) MUST
455 be signed.

456 **2.6 Unsolicited Responses**

457 An identity provider MAY initiate this profile by delivering an unsolicited <samlp:Response> message to
458 a service provider.

459 An unsolicited <samlp:Response> MUST NOT contain an InResponseTo attribute. If metadata as
460 specified in [SAML2Meta] is used, the <samlp:Response> or artifact SHOULD be delivered to the
461 <md:AssertionConsumerService> endpoint of the service provider designated as the default.

462 Of special mention is that the identity provider MAY include a binding-specific "RelayState" parameter
463 that indicates, based on mutual agreement with the service provider, how to handle subsequent
464 interactions with the user agent. This MAY be the URL of a resource at the service provider. The service
465 provider SHOULD be prepared to handle unsolicited responses by designating a default location to send
466 the user agent subsequent to processing a response successfully.

467 **2.7 Use of Metadata**

468 [SAML2Meta] defines an endpoint element, <md:SingleSignOnService>, to describe supported
469 bindings and location(s) to which a service provider may send requests to an identity provider using this
470 profile.

471 The <md:IDPSSODescriptor> element's WantAuthnRequestsSigned attribute MAY be used by an
472 identity provider to indicate a requirement that requests be signed. The <md:SPSSODescriptor>
473 element's AuthnRequestsSigned attribute MAY be used by a service provider to indicate the intention
474 to sign all of its requests. If one of these attributes is present, the requirement MUST be met by
475 counterparties.

476 The providers MAY document the key(s) used to sign requests, responses, and assertions with
477 <md:KeyDescriptor> elements with a use attribute of sign. When encrypting SAML elements,
478 <md:KeyDescriptor> elements with a use attribute of encrypt MAY be used to document supported
479 encryption algorithms and settings, and public keys used to receive bulk encryption keys. If no use
480 attribute is included, then the key MAY be used for both signing and encryption.

481 The indexed endpoint element <md:AssertionConsumerService> is used to describe supported
482 bindings and location(s) to which an identity provider may send responses to a service provider using this

483 profile. The `index` attribute is used to distinguish the possible endpoints that may be specified by
484 reference in the `<samlp:AuthnRequest>` message. The `isDefault` attribute is used to specify the
485 endpoint to use if not specified in a request.

486 **2.8 Compatibility**

487 This profile is based on the Web Browser SSO Profile in [SAML2Prof]. The primary difference is the
488 mandatory holder-of-key `<saml:SubjectConfirmation>`, prohibition of alternative
489 `<saml:SubjectConfirmation>`, and the resulting mandate of client TLS authentication for user agent
490 interactions. The confirmation of the subject by key allows several other requirements within that profile
491 to be relaxed or removed. Furthermore, because of its satisfy-any nature, inclusion of additional (in
492 particular, bearer) `<saml:SubjectConfirmation>` is prohibited in order to ensure that relying parties
493 perform key validation to accept the assertion.

494 The `urn:oasis:names:tc:SAML:2.0:profiles:SSO:browser:holder-of-key` profile is
495 therefore mutually incompatible with the `urn:oasis:names:tc:SAML:`
496 `2.0:profiles:SSO:browser` profile. However, one provider could easily register endpoints for each
497 profile in parallel in metadata to allow a single entity to transact with counterparties using both profiles.

498 **Appendix A. Acknowledgments**

499 The following individuals have participated in the creation of this specification and are gratefully
500 acknowledged. In addition, the editor would like to thank the National Institute of Informatics and the
501 UPKI initiative for their support of this work.

502 **Participants:**

503 Scott Cantor, Internet2
504 Patrick Harding, Ping Identity Corporation
505 Toshiyuki Kataoka, NII
506 Chad La Joie, SWITCH
507 Diego Lopez, RedIRIS
508 Tom Scavo, NCSA