



# Holder-of-Key Web Browser SSO Profile

## Working Draft 04

22 June 2008

### Specification URIs:

#### This Version:

<http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-holder-of-key-browser-ss0-draft-04.html>

<http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-holder-of-key-browser-ss0-draft-04.odt>

<http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-holder-of-key-browser-ss0-draft-04.pdf>

#### Previous Version:

<http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-holder-of-key-browser-ss0-draft-03.html>

<http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-holder-of-key-browser-ss0-draft-03.odt>

<http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-holder-of-key-browser-ss0-draft-03.pdf>

#### Latest Version:

<http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-holder-of-key-browser-ss0.html>

<http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-holder-of-key-browser-ss0.odt>

<http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-holder-of-key-browser-ss0.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

35 transactions. Cryptographic data resulting from TLS authentication is used for holder-of-key  
36 validation of a SAML assertion. This strengthens the assurance of the resulting authentication  
37 context and protects against credential theft, giving the service provider fresh authentication and  
38 attribute information without requiring it to perform successful validation of the certificate.

39 **Status:**

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

43 Technical Committee members should send comments on this specification to the Technical  
44 Committee's email list. Others should send comments to the Technical Committee by using the  
45 "Send A Comment" button on the Technical Committee's web page at [http://www.oasis-  
open.org/committees/security](http://www.oasis-<br/>46 open.org/committees/security).

47 For information on whether any patents have been disclosed that may be essential to  
48 implementing this specification, and any offers of patent licensing terms, please refer to the  
49 Intellectual Property Rights section of the Technical Committee web page ([http://www.oasis-  
open.org/committees/security/ipr.php](http://www.oasis-<br/>50 open.org/committees/security/ipr.php)).

51 The non-normative errata page for this specification is located at [http://www.oasis-  
open.org/committees/security](http://www.oasis-<br/>52 open.org/committees/security).

---

# Notices

53

54 Copyright © OASIS® 2008. All Rights Reserved.

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

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

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

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

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

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

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

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

## Table of Contents

98	1 Introduction.....	5
99	1.1 Terminology.....	5
100	1.2 Normative References.....	6
101	1.3 Conformance.....	6
102	1.3.1 Identity Provider.....	6
103	1.3.2 Service Provider.....	7
104	2 Holder-of-Key Web Browser SSO Profile.....	8
105	2.1 Required Information.....	8
106	2.2 Background.....	8
107	2.3 Profile Overview.....	8
108	2.4 Profile Description.....	10
109	2.4.1 HTTP Request to Service Provider.....	10
110	2.4.2 Service Provider Determines Identity Provider.....	10
111	2.4.3 <samlp:AuthnRequest> Issued by Service Provider to Identity Provider.....	10
112	2.4.4 Identity Provider Identifies Principal and Verifies Key Possession.....	11
113	2.4.5 Identity Provider Issues <samlp:Response>.....	11
114	2.4.6 Service Provider Grants or Denies Access to Principal.....	12
115	2.5 Use of Authentication Request Protocol.....	12
116	2.5.1 <samlp:AuthnRequest> Usage.....	12
117	2.5.2 <samlp:AuthnRequest> Message Processing Rules.....	13
118	2.5.3 <samlp:Response> Usage.....	13
119	2.5.4 <samlp:Response> Message Processing Rules.....	14
120	2.5.4.1 Artifact-Specific <samlp:Response> Message Processing Rules.....	14
121	2.5.4.2 POST-Specific <samlp:Response> Message Processing Rules.....	14
122	3 Compatibility.....	15
123	4 Security and Privacy Considerations.....	16
124		

# 1 Introduction

125

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

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

## 1.1 Terminology

144

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

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

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

Prefix	XML Namespace	Comments
md:	urn:oasis:names:tc:SAML:2.0:metadata	This is the SAML V2.0 metadata namespace defined in the SAML V2.0 metadata specification [SAML2Meta].
ds:	http://www.w3.org/2000/09/xmldsig#	This is the XML digital signature namespace defined in the XML Signature Syntax and Processing specification [DSig].
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].

153 For purposes of this document, Transport Layer Security (TLS) includes Secure Sockets Layer V3.0.

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

## 156 1.2 Normative References

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

## 186 1.3 Conformance

### 187 1.3.1 Identity Provider

188 A conforming implementation of an identity provider MUST support the following normative text of this  
189 profile: Sections 2.4.3, 2.4.4, 2.4.5, 2.5.2, and 2.5.3.

190 In addition to the typical requirements for a SAML implementation, a conforming identity provider MUST  
191 support the following functionality to support interoperability:

- 192 ● Retrieving a certificate presented by a user agent and validating that the user agent possesses  
193 the corresponding private key through TLS; and
- 194 ● Encoding this certificate using Base64 as `<ds:KeyInfo>` in a  
195 `<saml:SubjectConfirmationData>` element.

196 Implementations MAY additionally support the substitution of certificate fingerprints, public keys, and  
197 public key fingerprints for this certificate. Matches against `<ds:KeyName>` are discouraged because of  
198 the associated need for coordinated namespaces and certificate authorities. These alternatives SHOULD  
199 NOT be used unless the identity provider is specifically aware of the service provider's support for them.

## 200 1.3.2 Service Provider

201 A conforming implementation of a service provider MUST support the following normative text of this  
202 profile: Sections 2.4.1, 2.4.2, 2.4.3, 2.4.6, 2.5.1, and 2.5.4.

203 In addition to the typical requirements for a SAML implementation, a conforming service provider MUST  
204 support the following functionality to support interoperability:

- 205 ● Retrieving a certificate presented by a user agent and validating that the user agent possesses  
206 the corresponding private key through TLS; and
- 207 ● Decoding a Base64-encoded certificate within a `<ds:KeyInfo>` in a  
208 `<saml:SubjectConfirmationData>` element and matching it against a presented certificate.

209 Implementations MAY additionally support the substitution of certificate fingerprints, public keys, and  
210 public key fingerprints for this certificate. Matches against `<ds:KeyName>` are discouraged because of  
211 the associated need for coordinated namespaces and certificate authorities. A service provider SHOULD  
212 NOT trust assertions received through this profile unless it can confidently interpret and match the  
213 `<saml:SubjectConfirmationData>`.

---

## 214 2 Holder-of-Key Web Browser SSO Profile

### 215 2.1 Required Information

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

217 **Contact information:** [security-services-comment@lists.oasis-open.org](mailto:security-services-comment@lists.oasis-open.org)

218 **SAML Confirmation Method Identifiers:** The SAML V2.0 “holder-of-key” confirmation method identifier,  
219 urn:oasis:names:tc:SAML:2.0:cm:holder-of-key, is included in all assertions issued under this  
220 profile.

221 **Description:** Given below.

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

### 224 2.2 Background

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

230 If both the identity provider and service provider use this profile, but assume no knowledge of the  
231 certificate's contents, enhanced security is the primary benefit. There is a small chance that a bearer  
232 token will be stolen in transit, as described in [SAML2Secure]. Confirming that the presenter of the token  
233 is the intended holder through public key cryptography virtually eliminates this chance, improving the  
234 viability of SAML-based HTTP SSO for sensitive applications.

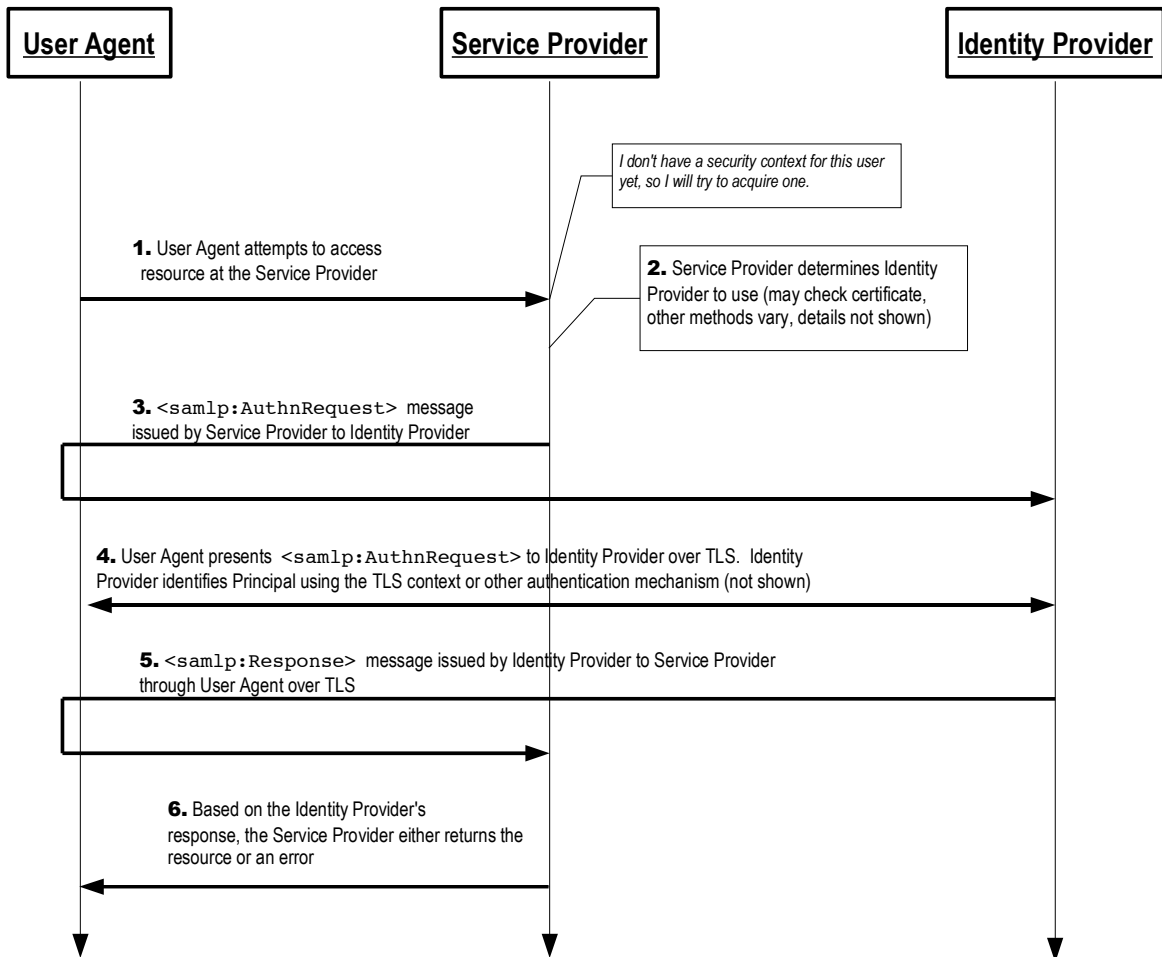
235 If a certificate can be used by the identity provider for principal authentication, there is no need for the  
236 user to further confirm its identity, and potentially no user interaction is needed.

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

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

### 245 2.3 Profile Overview

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



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

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

252 **2. Service Provider Determines Identity Provider**

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

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

258 The service provider issues a <samlp:AuthnRequest> message to be delivered by the user agent  
 259 to the identity provider. The HTTP Redirect, HTTP POST, or HTTP Artifact binding can be used to  
 260 transport the message to the identity provider through the user agent.

261 **4. Identity Provider identifies Principal**

262 The user agent makes a request to the identity provider using TLS. The principal is identified by the  
 263 identity provider. The identity provider identifies the principal using any authentication method at its  
 264 discretion honoring any requirements imposed by the service provider in the  
 265 <samlp:AuthnRequest>, including validation of the certificate presented in client TLS

266 authentication. However, the identity provider must establish that the private key corresponding to  
267 the keying material that will be included for holder-of-key proofing is held by this user agent.

## 268 **5. Identity Provider issues <samlp:Response> to Service Provider**

269 The identity provider issues a <samlp:Response> message to be delivered by the user agent to the  
270 service provider. The user agent presents this response to the service provider using TLS. Either the  
271 HTTP POST or HTTP Artifact binding can be used to transfer the message to the service provider  
272 through the user agent. The message may indicate an error or will include at least an authentication  
273 statement in an assertion with holder-of-key <saml:SubjectConfirmation> containing keying  
274 information associated with the principal.

## 275 **6. Service Provider grants or denies access to Principal**

276 The response is received by the service provider, which can respond to the principal's user agent by  
277 either establishing a security context for the principal and returning the requested resource or  
278 returning its own error or an error passed by the identity provider.

279 Note that an identity provider can initiate this profile at step 5 by issuing a <samlp:Response> message  
280 to a service provider without the preceding steps. The user agent or a third party may also initiate this  
281 profile by spoofing the authentication request if there is no requirement it be signed.

## 282 **2.4 Profile Description**

283 If the profile is initiated by the service provider, start with Section 2.4.1. If the request is unsigned and  
284 spoofed by the user agent or a third party, start with Section 2.4.4. If initiated by the identity provider,  
285 start with Section 2.4.5. The descriptions refer to a Single Sign-On Service and Assertion Consumer  
286 Service in accordance with their use in section 4.1.3 of [SAML2Prof]. Processing rules for all messages  
287 are specified in Section 2.5.

### 288 **2.4.1 HTTP Request to Service Provider**

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

### 293 **2.4.2 Service Provider Determines Identity Provider**

294 The service provider determines the primary identity provider with which the principal is associated  
295 through a variety of mechanisms as selected by the service provider implementation or deployment. The  
296 service provider MAY check the certificate presented by the user agent, to attempt to use the `x.509`  
297 `subject`, `subjectAltName`, or other field or extension in the certificate to determine the principal's  
298 identity provider or single sign-on service endpoint.

### 299 **2.4.3 <samlp:AuthnRequest> Issued by Service Provider to Identity 300 Provider**

301 Once an identity provider is selected, the location of a single sign-on service to which to send a  
302 <samlp:AuthnRequest> is determined based on the SAML binding chosen by the service provider.  
303 Metadata as described in [SAML2Meta] MAY be used for this purpose. Following an HTTP request by  
304 the user agent, an HTTP response is returned containing a <samlp:AuthnRequest> message or an

305 artifact, depending on the SAML binding used, to be delivered to the identity provider's single sign-on  
306 service.

307 Profile-specific rules for the contents of the `<samlp:AuthnRequest>` are defined in Section 2.5.1.

308 The `<samlp:AuthnRequest>` message MUST be signed if the identity provider requires the request  
309 issuer to be verified and the HTTP Redirect or HTTP POST bindings are used; if the HTTP Artifact  
310 binding is used, then the request issuer MAY be verified through other means.

311 If a certificate or public key is used as holder-of-key keying material in the request, the HTTP Redirect  
312 binding MUST NOT be used to transport the `<samlp:AuthnRequest>` due to size limitations.

#### 313 **2.4.4 Identity Provider Identifies Principal and Verifies Key Possession**

314 If the HTTP Redirect or POST binding is used, a `<samlp:AuthnRequest>` message is delivered directly  
315 to the identity provider to begin this step. If the HTTP Artifact binding is used, the Artifact Resolution  
316 profile defined in Section 5 of [SAML2Prof] is used by the identity provider, which makes a callback to the  
317 service provider to retrieve the `<samlp:AuthnRequest>` message using, for example, the SOAP  
318 binding.

319 The identity provider must perform two functions in this step: identification of the principal presenting the  
320 `<samlp:AuthnRequest>`, and verification that the principal possesses the private key associated with  
321 the keying information included in `<saml:SubjectConfirmation>`.

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

328 It is REQUIRED that the `<samlp:AuthnRequest>` be presented to the identity provider over mutually  
329 authenticated TLS to supply the identity provider with keying information and establish the user agent's  
330 possession of the corresponding private key. Keying information resulting from this process MUST match  
331 information included as holder-of-key `<saml:SubjectConfirmation>` in the subsequent  
332 `<samlp:Response>`.

333 These requirements can be simultaneously addressed by validation of an x.509 certificate presented by  
334 the user agent in TLS authentication from an issuer trusted by the identity provider, but this is not  
335 mandatory unless such an authentication context is requested by the service provider.

#### 336 **2.4.5 Identity Provider Issues `<samlp:Response>`**

337 The identity provider presents an HTTP response to the user agent containing a `<samlp:Response>`  
338 message or an artifact, depending on the SAML binding used, to be delivered to the service provider's  
339 assertion consumer service. The exact format of this HTTP response and the subsequent HTTP request  
340 to the assertion consumer service is defined by [SAML2Bind].

- 341 ● If the HTTP POST binding is used, the `<samlp:Response>` message is delivered directly to the  
342 service provider in this step.
- 343 ● The HTTP Redirect binding MUST NOT be used, as the response will typically exceed the URL  
344 length permitted by most user agents.

345 Profile-specific rules on the contents of the `<samlp:Response>` are included in section 2.5.3.

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

## 351 2.4.6 Service Provider Grants or Denies Access to Principal

352 The HTTP request presenting the message resulting from Section 2.4.5 to the service provider MUST be  
353 made over mutually authenticated TLS to demonstrate possession of the private key corresponding to the  
354 keying information included in the assertion's <saml:SubjectConfirmation> as well as maintain  
355 confidentiality and message integrity. The <saml:Assertion> element(s) in the <samlp:Response>  
356 MUST be signed if the HTTP POST binding is used, and MAY be signed if the HTTP Artifact binding is  
357 used.

358 If the HTTP Artifact binding is used, the Artifact Resolution profile defined in Section 5 of [SAML2Prof] is  
359 used by the service provider, which makes a callback to the identity provider to retrieve the  
360 <samlp:Response> message, using for example the SOAP binding. The TLS session could be used to  
361 persist client state during artifact resolution, or establish state afterwards by claiming a resolved  
362 assertion.

363 To complete the profile, the service provider processes the <samlp:Response> and  
364 <saml:Assertion>(s) and grants or denies access to the resource. The service provider MUST  
365 process the <samlp:Response> message and any enclosed <saml:Assertion> elements as  
366 described in [SAML2Core].

367 The service provider MAY establish a security context with the user agent using any session mechanism  
368 it chooses. Any subsequent use of the <saml:Assertion>(s) provided is at the discretion of the  
369 service provider and other relying parties, subject to any restrictions on use contained within them.

## 370 2.5 Use of Authentication Request Protocol

371 This profile uses the Authentication Request protocol defined in [SAML2Core]. In the nomenclature of  
372 actors enumerated in Section 3.4 of that document, the service provider is the request issuer and the  
373 relying party, the user agent is the attesting entity and presenter, and the principal is the requested  
374 subject. There may be additional relying parties at the discretion of the identity provider.

### 375 2.5.1 <samlp:AuthnRequest> Usage

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

- 378 ● The <saml:Issuer> element MUST be present and MUST contain the unique identifier of the  
379 requesting service provider. The `Format` attribute MUST be omitted or have a value of  
380 `urn:oasis:names:tc:SAML:2.0:nameid-format:entity`.
- 381 ● If the initial request was made over TLS and the <samlp:AuthnRequest> is to be signed, a  
382 <saml:Subject> element MAY be included in the request that includes keying information  
383 presented by the user agent for which the service provider wishes to receive an assertion in a  
384 holder-of-key <saml:SubjectConfirmation> element. If a <saml:NameID> is included to  
385 reference an existing user, subject information from the x.509 certificate SHOULD NOT be used  
386 for this purpose, as names used by the certificate authority may differ from those used by the  
387 identity provider.

- 388 ● If the service provider wishes to permit the identity provider to establish a new identifier for the  
389 principal if none exists, it MUST include a `<saml:NameIDPolicy>` element with the  
390 `AllowCreate` attribute set to `true`.
- 391 ● The `<samlp:AuthnRequest>` message MAY be signed (as directed by the SAML binding  
392 used). If the HTTP Artifact binding is used, authentication of the parties is OPTIONAL and any  
393 mechanism permitted by the binding MAY be used.

## 394 2.5.2 `<samlp:AuthnRequest>` Message Processing Rules

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

397 If the `<samlp:AuthnRequest>` is not authenticated and/or integrity protected, the information in it  
398 MUST NOT be trusted except as advisory. The `<samlp:AuthnRequest>` MUST be processed as  
399 follows:

- 400 ● It is RECOMMENDED that any `AssertionConsumerServiceURL` or  
401 `AssertionConsumerServiceIndex` attributes in the `<samlp:AuthnRequest>` are verified  
402 as belonging to the `entityID` to whom the response will be sent.
- 403 ● If the user agent cannot satisfy the `<saml:SubjectConfirmation>` present in the  
404 `<samlp:AuthnRequest>`, the identity provider MUST respond with a `<samlp:Response>`  
405 message containing an error status and no assertions.
- 406 ● The identity provider is NOT obligated to honor the requested set of `<saml:Conditions>` in the  
407 `<samlp:AuthnRequest>`, if any.

## 408 2.5.3 `<samlp:Response>` Usage

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

- 412 ● The `<saml:Issuer>` element of the `<samlp:Response>` MAY be omitted, but if present it  
413 MUST contain the unique identifier of the issuing identity provider; the `Format` attribute MUST be  
414 omitted or have a value of `urn:oasis:names:tc:SAML:2.0:nameid-format:entity`.
- 415 ● It MUST contain at least one `<saml:Assertion>`. Each assertion's `<saml:Issuer>` element  
416 MUST contain the unique identifier of the issuing identity provider, and the `Format` attribute  
417 MUST be omitted or have a value of `urn:oasis:names:tc:SAML:2.0:nameid-  
418 format:entity`.
- 419 ● The set of one or more assertions MUST collectively contain one and only one  
420 `<saml:AuthnStatement>` that reflects the authentication of the principal to the identity  
421 provider.
- 422 ● The assertion containing the `<saml:AuthnStatement>` MUST also contain a  
423 `<saml:Subject>` element with at least one `<saml:SubjectConfirmation>` element with a  
424 `Method` of `urn:oasis:names:tc:SAML:2.0:cm:holder-of-key`. Its  
425 `<saml:SubjectConfirmationData>` MUST contain cryptographically secure keying material  
426 associated with the user's private key that will be available to the service provider as a result of  
427 TLS authentication, such as an X.509 certificate, a public key, or a collision resistant hash of the  
428 public key. See Section 1.3.1 for minimal conformance requirements. Additional

- 429 <saml:SubjectConfirmation> elements MAY be included, though deployers should be  
430 aware of the implications of allowing weaker confirmation, as the processing is satisfy-any.
- 431 ● If the identity provider supports the Single Logout profile, defined in Section 4.4 of [SAML2Prof],  
432 the <saml:AuthnStatement> MUST include a `SessionIndex` attribute or a uniquely  
433 identifying <saml:NameID> to enable per-session logout requests by the service provider.
  - 434 ● Additional statements MAY be included in the assertion(s) at the discretion of the identity  
435 provider.
  - 436 ● The assertion containing the <saml:AuthnStatement> MUST contain a  
437 <saml:AudienceRestriction> including the service provider's unique identifier as a  
438 <saml:Audience>.
  - 439 ● Other conditions (and other <saml:Audience> elements) MAY be included as requested by the  
440 service provider or at the discretion of the identity provider. All such conditions MUST be  
441 understood by and accepted by the service provider in order for the assertion to be considered  
442 valid.

## 443 **2.5.4 <samlp:Response> Message Processing Rules**

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

- 445 ● Verify any signatures present on the assertion(s) or the response.
- 446 ● Verify that cryptographic data resulting from the mutual TLS authentication to the service provider  
447 matches the keying information in the holder-of-key <saml:SubjectConfirmationData>.  
448 The service provider SHOULD NOT rely on any other data in the certificate to process the  
449 assertion.
- 450 ● Verify that any assertions relied upon are valid according to processing rules in [SAML2Core].

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

### 453 **2.5.4.1 Artifact-Specific <samlp:Response> Message Processing 454 Rules**

455 If the HTTP Artifact binding is used to deliver the <samlp:Response>, the dereferencing of the artifact  
456 using the Artifact Resolution profile MUST be mutually authenticated, integrity protected, and confidential.  
457 Either the SAML binding used to dereference the artifact or message signatures can be used to  
458 authenticate the parties and protect the messages.

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

### 462 **2.5.4.2 POST-Specific <samlp:Response> Message Processing Rules**

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

---

465

## 3 Compatibility

466  
467  
468  
469  
470

This profile is based on the Web Browser SSO Profile in [SAML2Prof]. The primary difference is the mandatory holder-of-key `<saml:SubjectConfirmation>` and the resulting mandate of client TLS authentication for user agent interactions. Because of its satisfy-any nature, inclusion of additional (in particular, bearer) `<saml:SubjectConfirmation>` must be done cautiously in order to preserve the security benefits.

471  
472  
473

The `urn:oasis:names:tc:SAML:2.0:profiles:SSO:browser:holder-of-key` profile is technically compatible with the `urn:oasis:names:tc:SAML:2.0:profiles:SSO:browser` profile, but it is RECOMMENDED that separate endpoints be used to remove any potential ambiguity.

474

## 4 Security and Privacy Considerations

475  
476  
477

The holder-of-key assertions and protocols supporting their issuance and verification in this profile have some different security and privacy characteristics from the bearer assertions used in the Web Browser SSO Profile.

478  
479  
480

- If a certificate is used by the identity provider for principal authentication, phishing is eliminated, as there are greater challenges and no benefits to tricking the user into authenticating with legitimate credentials to a fraudulent party.

481  
482  
483  
484  
485  
486

- There are limitations on the degree to which users can remain private under this profile, since the X.509 certificate is presented to the service provider. 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 other fields. Furthermore, unless a new keypair is issued for every transaction, the public key is a de-facto persistent ID, as discussed in [SAML2Secure].

487  
488  
489

- Holder-of-key confirmation of the assertion issued eliminates the potential for assertion theft and encryption prevents privacy loss, eliminating attacks that would have required a check of the request issuer in Section 2.5.2.

490  
491  
492  
493

- The use of holder-of-key verification and encryption eliminate man-in-the-middle attacks. However, without the inclusion of `<saml:AudienceRestriction>` in Section 2.5.3, there would remain the possibility of collusion between the principal and the intended recipient to re-encrypt and replay the assertion to another service provider.

494  
495  
496  
497  
498  
499  
500

- The `<md:IDPSSODescriptor>` element's `WantAuthnRequestsSigned` attribute MAY be used by an identity provider to indicate a requirement that requests be signed. The `<md:SPSSODescriptor>` element's `AuthnRequestsSigned` attribute MAY be used by a service provider to indicate the intention to sign all of its requests. If one of these attributes is present, the requirement SHOULD be met by counterparties. Deployers should consider the limited vulnerabilities associated with spoofed authentication requests and significant complexity resulting from authentication request signing.

501  
502  
503  
504

- 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 the theft of this session and impersonation of the user.

---

505 **Appendix A. Acknowledgments**

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

509 **Participants:**

510 Scott Cantor, Internet2  
511 Patrick Harding, Ping Identity Corporation  
512 Enrique de la Hoz, University of Alcala de Henares  
513 Toshiyuki Kataoka, National Institute of Informatics  
514 Chad La Joie, SWITCH  
515 Diego Lopez, RedIRIS  
516 Tom Scavo, NCSA  
517 David Waite, Ping Identity Corporation