



# **SAML V2.0 Holder-of-Key Web Browser SSO Profile**

**Working Draft 067**

**~~26 August~~ 22 September 2008**

## **Specification URIs:**

### **This Version:**

<http://docs.oasis-open.org/security/saml/Post2.0/sssc-saml-holder-of-key-browser-ss0-draft-07.html>

<http://docs.oasis-open.org/security/saml/Post2.0/sssc-saml-holder-of-key-browser-ss0-draft-07.odt>

<http://docs.oasis-open.org/security/saml/Post2.0/sssc-saml-holder-of-key-browser-ss0-draft-07.pdf>

### **Previous Version:**

<http://docs.oasis-open.org/security/saml/Post2.0/sssc-saml-holder-of-key-browser-ss0-draft-06.html>

<http://docs.oasis-open.org/security/saml/Post2.0/sssc-saml-holder-of-key-browser-ss0-draft-06.odt>

<http://docs.oasis-open.org/security/saml/Post2.0/sssc-saml-holder-of-key-browser-ss0-draft-06.pdf>

### **Latest Version:**

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

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

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

34 deployments. Most of the flows are as in standard Web Browser SSO, but an X.509 certificate  
35 presented by the user agent supplies a valid keypair through client TLS authentication for HTTP  
36 transactions. Proof of key possession resulting from TLS authentication is used for holder-of-key  
37 validation of a SAML assertion. This strengthens the assurance of the resulting authentication  
38 context and protects against credential theft, giving the service provider fresh authentication and  
39 attribute information without requiring it to perform successful validation of the certificate.

40 **Status:**

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

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

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

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

---

# Notices

54

55 Copyright © OASIS® 2008. All Rights Reserved.

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

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

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

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

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

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

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

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

## Table of Contents

99	1 Introduction.....	5
100	1.1 Terminology.....	5
101	1.2 Normative References.....	6
102	1.3 Non-normative References.....	6
103	1.4 Conformance.....	7
104	1.4.1 Identity Provider.....	7
105	1.4.2 Service Provider.....	7
106	2 Holder-of-Key Web Browser SSO Profile.....	8
107	2.1 Required Information.....	8
108	2.2 Background.....	8
109	2.3 Profile Overview.....	8
110	2.4 Profile Description.....	10
111	2.4.1 HTTP Request to Service Provider.....	10
112	2.4.2 Service Provider Determines Identity Provider.....	10
113	2.4.3 <samlp:AuthnRequest> Issued by Service Provider to Identity Provider.....	10
114	2.4.4 Identity Provider Identifies Principal and Verifies Key Possession.....	11
115	2.4.5 Identity Provider Issues <samlp:Response>.....	11
116	2.4.6 Service Provider Grants or Denies Access to Principal.....	12
117	2.5 Use of Authentication Request Protocol.....	12
118	2.5.1 <samlp:AuthnRequest> Usage.....	12
119	2.5.2 <samlp:AuthnRequest> Message Processing Rules.....	13
120	2.5.3 <samlp:Response> Usage.....	13
121	2.5.4 <samlp:Response> Message Processing Rules.....	14
122	2.5.4.1 Artifact-Specific <samlp:Response> Message Processing Rules.....	14
123	2.5.4.2 POST-Specific <samlp:Response> Message Processing Rules.....	14
124	2.6 Use of Metadata.....	14
125	3 Compatibility.....	16
126	4 Security and Privacy Considerations.....	17
127		

128

# 1 Introduction

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

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

## 1.1 Terminology

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

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

155 Conventional XML namespace prefixes are used throughout this specification to stand for their respective  
 156 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].
hok:	<a href="#">urn:oasis:names:tc:SAML:2.0:profiles:holder-of-key:SSO:browser:holder-of-key</a>	This is the web browser holder-of-key namespace is defined by this document and its accompanying schema [HoK-XSD].
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].

157

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

160 For purposes of this document, Transport Layer Security (TLS) as defined in [RFC 4346] includes Secure  
161 Sockets Layer V3.0.

## 162 1.2 Normative References

- 163 **[DSig]** D. Eastlake, J. Reagle, D. Solo, F. Hirsch, T. Roessler. *XML Signature Syntax*  
164 *and Processing (Second Edition)*. World Wide Web Consortium  
165 Recommendation, 10 June 2008. See <http://www.w3.org/TR/xmlsig-core/>.
- 166 **[HoK-XSD]** N. Klingenstein. Schema for SAML V2.0 Holder-of-Key Web Browser SSO  
167 Profile. OASIS SSTC, 4 August 2008. See [http://www.oasis-](http://www.oasis-open.org/committees/security)  
168 [open.org/committees/security](http://www.oasis-open.org/committees/security).
- 169 **[RFC 2119]** S. Bradner. *Key words for use in RFCs to Indicate Requirement Levels*. IETF  
170 RFC 2119, March 1997. <http://www.ietf.org/rfc/rfc2119.txt>.
- 171 **[RFC 4346]** T. Dierks, E. Rescorla. *The Transport Layer Security (TLS) Protocol*. IETF RFC  
172 4346, April 2006. <http://www.ietf.org/rfc/rfc4346.txt>.
- 173 **[SAML2Bind]** S. Cantor et al. *Bindings for the OASIS Security Assertion Markup Language*  
174 *(SAML) V2.0*. OASIS Standard, March 2005. Document ID saml-bindings-2.0-os.  
175 See <http://docs.oasis-open.org/security/saml/v2.0/saml-bindings-2.0-os.pdf>.
- 176 **[SAML2Core]** S. Cantor et al. *Assertions and Protocols for the OASIS Security Assertion*  
177 *Markup Language (SAML) V2.0*. OASIS Standard, March 2005. Document ID  
178 saml-core-2.0-os. See [http://docs.oasis-open.org/security/saml/v2.0/saml-core-](http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf)  
179 [2.0-os.pdf](http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf).
- 180 **[SAML2HoKAP]** T. Scavo. *SAML V2.0 Holder-of-Key Assertion Profile. OASIS Working Draft 03,*  
181 *August 2008. Document ID sstc-saml2-holder-of-key. See* [http://www.oasis-](http://www.oasis-open.org/committees/security)  
182 [open.org/committees/security](http://www.oasis-open.org/committees/security).
- 183 **[SAML2Meta]** S. Cantor et al. *Metadata for the OASIS Security Assertion Markup Language*  
184 *(SAML) V2.0*. OASIS Standard, March 2005. Document ID saml-metadata-2.0-  
185 os. See <http://docs.oasis-open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf>.
- 186 **[SAML2Prof]** ~~J. Hughes~~~~S. Cantor~~ et al. *Profiles for the OASIS Security Assertion Markup*  
187 *Language (SAML) V2.0*. OASIS Standard, March 2005. Document ID saml-  
188 profiles-2.0-os. See [http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-](http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf)  
189 [2.0-os.pdf](http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf).

## 190 1.3 Non-normative References

- 191
- 192 **[IDPDisco]** R. Widdowson, S. Cantor. Identity Provider Discovery Service Protocol and  
193 Profile. OASIS SSTC, October 2007. Document ID sstc-saml-idp-discovery. See  
194 <http://www.oasis-open.org/committees/security>.
- 195
- 196 **[SAML2Secure]** F. Hirsch et al. *Security and Privacy Considerations for the OASIS Security*  
197 *Assertion Markup Language (SAML) v2.0*. OASIS ~~SSTC~~**Standard**, March 2005.  
198 Document ID saml-sec-consider-2.0-os. See [http://docs.oasis-](http://docs.oasis-open.org/security/saml/v2.0/saml-sec-consider-2.0-os.pdf)  
199 [open.org/security/saml/v2.0/saml-sec-consider-2.0-os.pdf](http://docs.oasis-open.org/security/saml/v2.0/saml-sec-consider-2.0-os.pdf).

## 200 1.4 Conformance

### 201 1.4.1 Identity Provider

202 A conforming implementation of an identity provider MUST support the following normative text of this  
203 profile: sections ~~2.4.3~~, 2.4.4, 2.4.5, 2.5.2, and 2.5.3. If the identity provider uses metadata, it MUST also  
204 support section 2.6.

205 In addition to the typical requirements for a SAML implementation, a conforming identity provider MUST  
206 ~~meet the conformance requirements listed in [SAML2HoKAP]. support the following functionality to~~  
207 ~~support interoperability:~~

- 208 ● ~~Retrieving a certificate presented by a user agent and validating that the user agent possesses~~  
209 ~~the corresponding private key through TLS; and~~
- 210 ● ~~Encoding this certificate using Base64 as <ds:KeyInfo> in a~~  
211 ~~<saml:SubjectConfirmationData> element.~~

212

### 213 1.4.2 Service Provider

214 A conforming implementation of a service provider MUST support the following normative text of this  
215 profile: sections 2.4.1, 2.4.2, 2.4.3, 2.4.6, 2.5.1, and 2.5.4. If the service provider uses metadata, it MUST  
216 also support section 2.6.

217 In addition to the typical requirements for a SAML implementation, a conforming service provider MUST  
218 ~~meet the conformance requirements listed in [SAML2HoKAP]. support the following functionality to~~  
219 ~~support interoperability:~~

- 220 ● ~~Retrieving a certificate presented by a user agent and validating that the user agent possesses~~  
221 ~~the corresponding private key through TLS; and~~
- 222 ● ~~Decoding a Base64-encoded certificate within a <ds:KeyInfo> in a~~  
223 ~~<saml:SubjectConfirmationData> element and matching it against a presented certificate.~~

---

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

### 2.1 Required Information

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

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

**Description:** Given below.

**Updates:** Provides an alternative to the SAML V2.0 Web Browser SSO Profile given in [section 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. The SAML assertion is delivered to the service provider over mutually authenticated TLS using keying information vetted by the identity provider, resulting in strong association of the resulting security context with the intended user and elimination of many attacks. Additional usability benefits can be realized by leveraging the end user’s certificate for discovery and authentication.

Deployments should minimize user interaction, avoid mutually conflicting CA requirements, and must ensure presentation of the same certificate to all services by coordinating certificate issuance and TLS configuration.

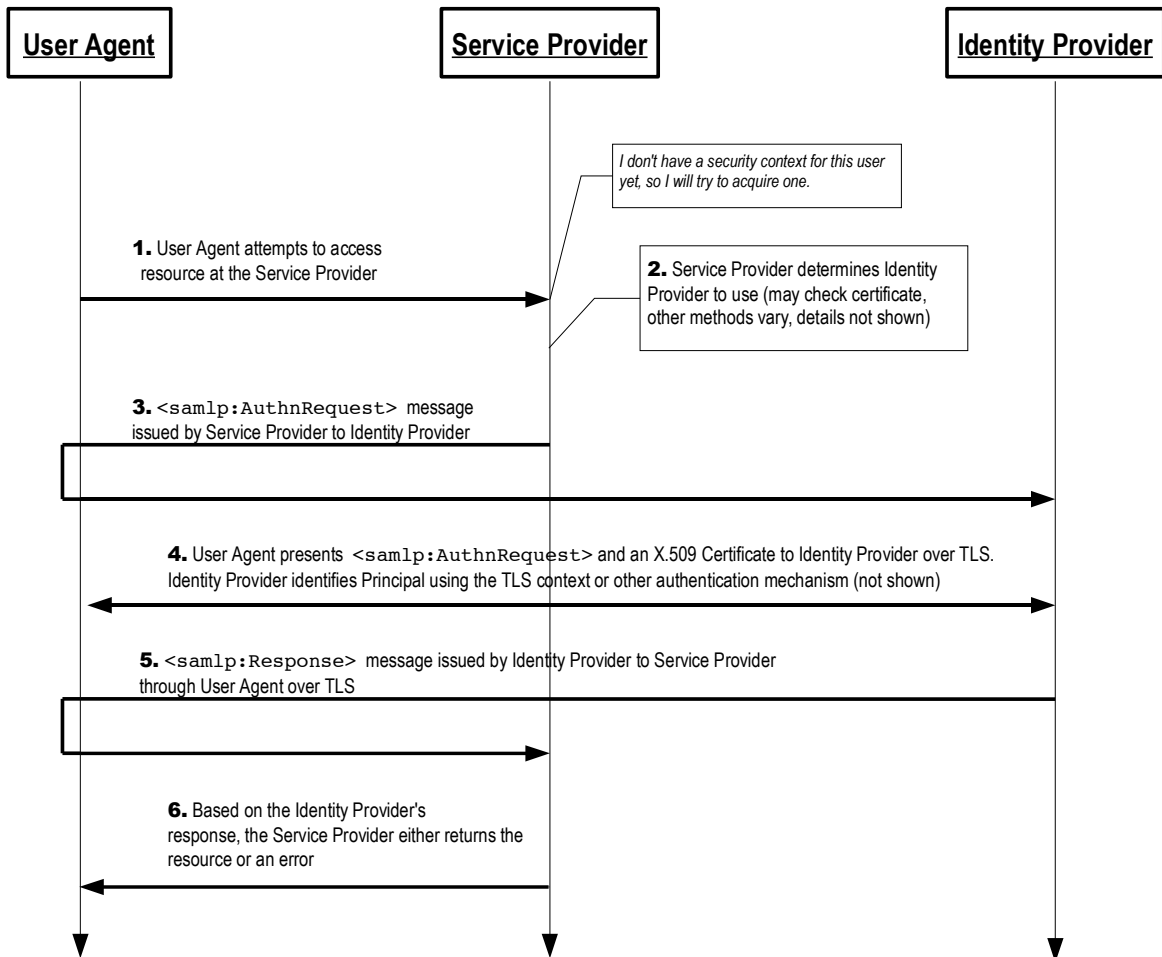
If both the identity provider and service provider use this profile, but assume no knowledge of the certificate's contents, enhanced security is the primary benefit. There is a small chance that a bearer token will be stolen in transit, as described in [SAML2Secure]. Confirming that the presenter of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

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. 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~~[public key infrastructure \(PKI\)](#), as well. There is no requirement for a mutually or universally trusted root, distributed OCSP or CRL-based revocation, a globally unique namespace, PKI 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.

### 2.3 Profile Overview

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



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

264 The principal, via an HTTP user agent, makes an HTTP request for a secured resource at the service  
 265 provider. This request may or may not be made over TLS. The service provider determines that no  
 266 security context exists, and attempts to create one.

267 **2. Service Provider Determines Identity Provider**

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

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

273 The service provider issues a <samlp:AuthnRequest> message to be delivered by the user agent  
 274 to the identity provider. The HTTP Redirect, HTTP POST, or HTTP Artifact binding can be used to  
 275 transport the message to the identity provider through the user agent. The user agent presents this  
 276 message in a request to the identity provider using TLS.

277 **4. Identity Provider identifies Principal**

278 ~~The user agent makes a request to the identity provider using TLS.~~The principal is identified by the  
 279 identity provider. The identity provider identifies the principal using any authentication method at its  
 280 discretion honoring any requirements imposed by the service provider in the

281 <samlp:AuthnRequest>, including validation of the certificate presented in client TLS  
282 authentication. However, the identity provider must establish that the private key corresponding to  
283 the X.509 certificate that will be included for holder-of-key proofing is held by this user agent.

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

285 The identity provider issues a <samlp:Response> message to be delivered by the user agent to the  
286 service provider. The user agent presents this response to the service provider using TLS. Either the  
287 HTTP POST or HTTP Artifact binding can be used to transfer the message to the service provider  
288 through the user agent. The message may indicate an error or will include at least an authentication  
289 statement in an assertion with holder-of-key <saml:SubjectConfirmation> containing an X.509  
290 certificate associated with the principal.

## 291 6. Service Provider grants or denies access to Principal

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

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

## 298 2.4 Profile Description

299 If the profile is initiated by the service provider, start with section 2.4.1. If the request is unsigned and  
300 spoofed by the user agent or a third party, start with section 2.4.4. If initiated by the identity provider, start  
301 with section 2.4.5. The descriptions refer to a [single sign-on service](#)~~Single Sign-On Service~~ and [Assertion-](#)  
302 [Consumer Service](#)~~assertion consumer service~~ in accordance with their use in section 4.1.3 of  
303 [SAML2Prof]. Processing rules for all messages are specified in section 2.5.

### 304 2.4.1 HTTP Request to Service Provider

305 The profile may be initiated by an arbitrary [HTTP](#) request to the service provider. The service provider is  
306 free to use any means it wishes to associate the subsequent interactions with the original request. Each  
307 of the bindings provides a `RelayState` mechanism that the service provider MAY use to associate the  
308 profile exchange with the original request. If a TLS session is established for this initial request, it MAY be  
309 used for discovery in section 2.4.2.

### 310 2.4.2 Service Provider Determines Identity Provider

311 The service provider determines the identity provider with which the principal is associated through a  
312 variety of mechanisms as selected by the service provider implementation or deployment. If the initial  
313 request in section 2.4.1 was made over mutually authenticated TLS, the service provider MAY check the  
314 certificate presented by the user agent and use the X.509 subject, subjectAltName, or other field or  
315 extension in the certificate to determine the principal's identity provider or single sign-on service endpoint.

### 316 2.4.3 <samlp:AuthnRequest> Issued by Service Provider to Identity 317 Provider

318 Once an identity provider is selected, the location of a single sign-on service to which to send a  
319 <samlp:AuthnRequest> is determined based on the SAML binding chosen by the service provider.  
320 Metadata as described in section 2.6 MAY be used for this purpose. Following an HTTP request by the

321 user agent, an HTTP response is returned containing a `<samlp:AuthnRequest>` message or an  
322 artifact, depending on the SAML binding used, to be delivered to the identity provider's single sign-on  
323 service.

324 Profile-specific rules for the contents of the `<samlp:AuthnRequest>` are defined in section 2.5.1. ~~The  
325 `<samlp:AuthnRequest>` message should be signed if the identity provider requires the request issuer  
326 to be verified and the HTTP Redirect or HTTP POST bindings are used; if the HTTP Artifact binding is  
327 used, then the request issuer can be verified through other means.~~

## 328 2.4.4 Identity Provider Identifies Principal and Verifies Key Possession

329 If the HTTP Redirect or POST binding is used, a `<samlp:AuthnRequest>` message is delivered directly  
330 to the identity provider to begin this step. If the HTTP Artifact binding is used, the Artifact Resolution  
331 profile defined in section 5 of [SAML2Prof] is used by the identity provider, which makes a callback to the  
332 service provider to retrieve the `<samlp:AuthnRequest>` message using, for example, the SOAP  
333 binding.

334 The identity provider must perform two functions in this step: identification of the principal presenting the  
335 `<samlp:AuthnRequest>`, and verification that the principal possesses the private key associated with  
336 the X.509 certificate that will be included as `<saml:SubjectConfirmation>`.

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

343 It is REQUIRED that the `<samlp:AuthnRequest>` be presented to the identity provider over mutually  
344 authenticated TLS to supply the identity provider with the X.509 certificate and establish the user agent's  
345 possession of the corresponding private key. The certificate resulting from this process MUST match the  
346 one included as holder-of-key `<saml:SubjectConfirmation>` in the subsequent  
347 `<samlp:Response>`.

## 348 2.4.5 Identity Provider Issues `<samlp:Response>`

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

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

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

358 The location of the assertion consumer service MAY be determined using metadata as described in  
359 section 2.6. The identity provider MUST have some means to establish that this location is in fact  
360 controlled by the service provider. A service provider MAY indicate the SAML binding and the specific  
361 assertion consumer service to use in its `<samlp:AuthnRequest>` and the identity provider MUST honor  
362 them ~~if it can~~.

## 363 2.4.6 Service Provider Grants or Denies Access to Principal

364 The HTTP request presenting the message resulting from section 2.4.5 to the service provider MUST be  
365 made over mutually authenticated TLS to demonstrate possession of the private key corresponding to the  
366 certificate included in the assertion's `<saml:SubjectConfirmation>` as well as maintain  
367 confidentiality and message integrity. ~~The `<saml:Assertion>` element(s) in the `<samlp:Response>`  
368 MUST be signed if the HTTP POST binding is used, and MAY be signed if the HTTP Artifact binding is  
369 used.~~

370 If the HTTP Artifact binding is used, the Artifact Resolution profile defined in section 5 of [SAML2Prof] is  
371 used by the service provider, which makes a callback to the identity provider to retrieve the  
372 `<samlp:Response>` message, using for example the SOAP binding. The [front-channel](#) TLS session  
373 could be used to persist client state during artifact resolution, or establish state afterwards by claiming a  
374 resolved assertion.

375 To complete the profile, the service provider processes the `<samlp:Response>` and  
376 `<saml:Assertion>`(s) and creates a security context for the user. The service provider MUST process  
377 the `<samlp:Response>` message and any enclosed `<saml:Assertion>` elements as described in  
378 [SAML2Core].

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

## 382 2.5 Use of Authentication Request Protocol

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

### 387 2.5.1 `<samlp:AuthnRequest>` Usage

388 A service provider MAY include any `<samlp:AuthnRequest>` message content described in  
389 [SAML2Core], section 3.4.1. All processing rules are as defined in [SAML2Core]. The request MUST  
390 conform to the following:

- 391 ● The `<saml:Issuer>` element MUST be present and MUST contain the unique identifier of the  
392 requesting service provider. The `Format` attribute MUST be omitted or have a value of  
393 `urn:oasis:names:tc:SAML:2.0:nameid-format:entity`.
- 394 ● If the initial request was made over TLS and the `<samlp:AuthnRequest>` is to be signed, a  
395 `<saml:Subject>` element MAY be included in the request. If a `<saml:NameID>` is included to  
396 reference an existing user, subject information from the X.509 certificate SHOULD NOT be used  
397 for this purpose, as names used by the certificate authority may differ from those used by the  
398 providers.
- 399 ● If the service provider wishes to permit the identity provider to establish a new identifier for the  
400 principal if none exists, it MUST include a `<saml:NameIDPolicy>` element with the  
401 `AllowCreate` attribute set to `true`.
- 402 ● The `<samlp:AuthnRequest>` message MAY be signed (as directed by the SAML binding  
403 used).

## 404 2.5.2 <samlp:AuthnRequest> Message Processing Rules

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

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

- 410 ● ~~AnyIt is RECOMMENDED that any~~ AssertionConsumerServiceURL or  
411 AssertionConsumerServiceIndex attributes in the <samlp:AuthnRequest> are MUST be  
412 verified as belonging to the entityID to whom the response will be sent.
- 413 ● If the user agent cannot satisfy the <saml:SubjectConfirmation> present in the  
414 <samlp:AuthnRequest>, or it fails to obtain a key from the user agent, the identity provider  
415 MUST respond with a <samlp:Response> message containing an error status and no  
416 assertions.
- 417 ● The identity provider is NOT obligated to honor the requested set of <saml:Conditions> in the  
418 <samlp:AuthnRequest>, if any.

## 419 2.5.3 <samlp:Response> Usage

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

- 423 ● The <saml:Issuer> element of the <samlp:Response> MAY be omitted, but if present it  
424 MUST contain the unique identifier of the issuing identity provider; the Format attribute MUST be  
425 omitted or have a value of urn:oasis:names:tc:SAML:2.0:nameid-format:entity.
- 426 ● ~~#The response~~ MUST contain at least one <saml:Assertion>. Each assertion's  
427 <saml:Issuer> element MUST contain the unique identifier of the issuing identity provider, and  
428 the Format attribute MUST be omitted or have a value of  
429 urn:oasis:names:tc:SAML:2.0:nameid-format:entity.
- 430 ● The set of one or more assertions MUST collectively contain one and only one  
431 <saml:AuthnStatement> that reflects the authentication of the principal to the identity  
432 provider.
- 433 ● The assertion containing the <saml:AuthnStatement> is considered a holder-of-key assertion  
434 and MUST conform to [SAML2HoKAP]. Unless otherwise indicated by the service provider, the  
435 keying material SHOULD be <ds:X509Certificate>MUST also contain a <saml:Subject>  
436 element with at least one <saml:SubjectConfirmation> element with a Method of  
437 urn:oasis:names:tc:SAML:2.0:cm:holder-of-key. Its  
438 <saml:SubjectConfirmationData> MUST embed the X.509 certificate directly in the  
439 <saml:SubjectConfirmationData> element by placing its base64 encoded representation  
440 in a <ds:KeyInfo> element.
- 441 ● Additional <saml:SubjectConfirmation> elements MAY be included, though deployers  
442 should be aware of the implications of allowing weaker confirmation, as the processing is satisfy-  
443 any. See section 3 for compatibility considerations.

- 444 ● If the identity provider supports the Single Logout profile, defined in section 4.4 of [SAML2Prof],  
445 the <saml:AuthnStatement> MUST include a `SessionIndex` attribute or a uniquely  
446 identifying <saml:NameID> to enable per-session logout requests by the service provider.
- 447 ● Additional statements MAY be included in the assertion(s) at the discretion of the identity  
448 provider.
- 449 ● The assertion containing the <saml:AuthnStatement> MUST contain a  
450 <saml:AudienceRestriction> including the service provider's unique identifier as a  
451 <saml:Audience>.
- 452 ● Other conditions (and other <saml:Audience> elements) MAY be included as requested by the  
453 service provider or at the discretion of the identity provider. All such conditions MUST be  
454 understood by and accepted by the service provider in order for the assertion to be considered  
455 valid.

## 456 2.5.4 <samlp:Response> Message Processing Rules

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

- 458 ● Verify any signatures present on the assertion(s) or the response.
- 459 ● ~~Verify that cryptographic data~~ The <saml:SubjectConfirmation> MUST be confirmed in  
460 accordance with the requirements in [SAML2HoKAP] using the certificate resulting from  
461 successful the mutual TLS authentication to the service provider ~~by the user agent~~ strongly  
462 matches the X.509 certificate in the holder-of-key <saml:SubjectConfirmationData>. The  
463 service provider SHOULD NOT rely on any other data in the certificate to process the assertion  
464 but MAY utilize it more generally as additional information about the user.
- 465 ● Verify that any assertions relied upon are valid according to processing rules in [SAML2Core].

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

### 468 2.5.4.1 Artifact-Specific <samlp:Response> Message Processing 469 Rules

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

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

### 477 2.5.4.2 POST-Specific <samlp:Response> Message Processing Rules

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

## 480 2.6 Use of Metadata

481 [SAML2Meta] defines endpoint elements to describe supported bindings and locations for providers.  
482 However, the metadata specification offers no way to distinguish the profile used by an endpoint. A  
483 boolean flag extension is not sufficient to signal use of this profile: because SAML implementations that  
484 don't implement this profile would ignore this optional attribute, they could send users to an inappropriate  
485 endpoint, potentially impacting interoperability and user experience. Rather than define new endpoint  
486 elements, it's sufficient to use the `binding` attribute to disambiguate between  
487 `urn:oasis:names:tc:SAML:2.0:profiles:SSO:browser:holder-of-key` profile use and  
488 `urn:oasis:names:tc:SAML:2.0:profiles:SSO:browser` profile use. The actual binding to be  
489 used in this profile, as specified in [SAML2Bind], is instead placed into an extension attribute on the same  
490 endpoint. The combined information is sufficient to distinguish and utilize the correct profile and binding  
491 when making a request to an endpoint.

492 All `<md:SingleSignOnService>` endpoints and all `<md:AssertionConsumerService>` endpoints  
493 to be used exclusively with this profile MUST have a `binding` attribute of:

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

495 If an endpoint has the `binding` attribute

496 `urn:oasis:names:tc:SAML:2.0:profiles:SSO:browser:holder-of-key`, it MUST also  
497 include a separate extension `hok:Protocol` attribute. The `hok:Protocol` attribute contains the  
498 identifier of the original protocol binding.

499 The following schema fragment defines the `hok:Protocol` attribute:

```
500 <attribute name="Protocol" type="anyURI"/>
```

501 This is an example `<md:SingleSignOnService>` endpoint to be used **exclusively** with this profile:

```
502 <SingleSignOnService  
503   xmlns:hok="urn:oasis:names:tc:SAML:2.0:profiles:SSO:browser:holder-of-  
504   key:SSO:browser"  
505   Binding="urn:oasis:names:tc:SAML:2.0:profiles:SSO:browser:holder-of-  
506   key:SSO:browser" hok:Protocol="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-  
507   Redirect" Location="http://your-idp.example.org/some/path"/>
```

508 This is an example `<md:AssertionConsumerService>` endpoint to be used **exclusively** with this  
509 profile:

```
510 <AssertionConsumerService index="1" isDefault="true"  
511   xmlns:hok="urn:oasis:names:tc:SAML:2.0:profiles:SSO:browser:holder-of-  
512   key:SSO:browser"  
513   Binding="urn:oasis:names:tc:SAML:2.0:profiles:SSO:browser:holder-of-  
514   key:SSO:browser" hok:Protocol="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-  
515   POST" Location="http://your-sp.example.org/some/path"/>
```

516 The `<md:IDPSSODescriptor>` element's `WantAuthnRequestsSigned` attribute MAY be used by an  
517 identity provider to indicate a requirement that requests be signed. The `<md:SPSSODescriptor>`  
518 element's `AuthnRequestsSigned` attribute MAY be used by a service provider to indicate the intention  
519 to sign all of its requests. If one of these attributes is present, the requirement SHOULD be met by  
520 counterparties.

521 The `<md:SPSSODescriptor>` element's `WantAssertionsSigned` attribute MAY be used by a service  
522 provider to indicate a requirement that assertions delivered with this profile be signed. If present, the  
523 requirement SHOULD be met by counterparties.

---

524

## 3 Compatibility

525 This profile is derived from the Web Browser SSO Profile in [SAML2Prof]. The primary difference is the  
526 mandatory holder-of-key `<saml:SubjectConfirmation>` and the resulting mandate of client TLS  
527 authentication for user agent interactions. Because of its satisfy-any nature, inclusion of additional (in  
528 particular, bearer) `<saml:SubjectConfirmation>` must be done cautiously. An assertion including  
529 both holder-of-key and bearer subject confirmation could be issued in accordance with this profile and  
530 accepted as valid with no proof of possession of key, reintroducing attacks such as man-in-the-middle and  
531 replay.

532 The `urn:oasis:names:tc:SAML:2.0:profiles:SSO:browser:holder-of-key` profile is  
533 technically compatible with the `urn:oasis:names:tc:SAML:2.0:profiles:SSO:browser` profile,  
534 but it is RECOMMENDED that separate endpoints be used to ensure all processing is performed in  
535 accordance with each profile's requirements and avoid any negative impact on user experience.

---

## 4 Security and Privacy Considerations

536

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

- 540 ● ~~The identity provider's requirements for user authentication and keying material as described in~~  
541 ~~section 2.4.4 can be simultaneously addressed by validation of an x.509 certificate presented by~~  
542 ~~the user agent in TLS authentication from an issuer trusted by the identity provider, but this is not~~  
543 ~~mandatory unless such an authentication context is requested by the service provider. If a~~  
544 ~~certificate is used by the identity provider for principal authentication, pPhishing is also~~ eliminated,  
545 as there are greater challenges and no benefits to tricking the user into authenticating with  
546 legitimate credentials to a fraudulent party.
- 547 ● There ~~are~~ **may be** limitations on the degree to which users can remain private under this profile,  
548 since the X.509 certificate is presented to the service provider. Most end-user X.509 certificates  
549 have a unique distinguished name for the subject regularly containing personally identifying  
550 information. Additional information about the subject may be implicitly revealed through other  
551 fields. Furthermore, unless a new keypair is issued for every transaction, the public key is a de-  
552 facto persistent ID, as discussed in [SAML2Secure].
- 553 ● Holder-of-key confirmation of the assertion issued eliminates the potential for assertion theft and  
554 encryption prevents privacy loss, eliminating man-in-the-middle attacks.
- 555 ● ~~The identity provider's requirements for user authentication and keying material as described in~~  
556 ~~section 2.4.4 can be simultaneously addressed by validation of an x.509 certificate presented by~~  
557 ~~the user agent in TLS authentication from an issuer trusted by the identity provider, but this is not~~  
558 ~~mandatory unless such an authentication context is requested by the service provider.~~
- 559 ● Replay attacks that would have required tracking and checking assertion ID's are prevented by  
560 validation of key possession.
- 561 ● ~~The <md:IDPSSODescriptor> element's WantAuthnRequestsSigned attribute MAY be~~  
562 ~~used by an identity provider to indicate a requirement that requests be signed. The~~  
563 ~~<md:SPSSODescriptor> element's AuthnRequestsSigned attribute MAY be used by a~~  
564 ~~service provider to indicate the intention to sign all of its requests. If one of these attributes is~~  
565 ~~present, the requirement SHOULD be met by counterparties.~~ Deployers should consider the  
566 limited vulnerabilities associated with spoofed authentication requests and significant complexity  
567 resulting from authentication request signing.
- 568 ● The session created by the service provider in the security context resulting from the Holder-of-  
569 Key Web Browser SSO Profile can be keyed by the TLS public key or session key. Application-  
570 layer sessions, such as maintained by cookies, are often poorly protected by user agents,  
571 allowing for theft of this session and impersonation of the user.

---

572 **Appendix A. Acknowledgments**

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

576 **Participants:**

577 Scott Cantor, Internet2  
578 Paul Friedrichs, Defense Information Services Agency  
579 Patrick Harding, Ping Identity Corporation  
580 Enrique de la Hoz, University of Alcala de Henares  
581 Toshiyuki Kataoka, National Institute of Informatics  
582 Chad La Joie, SWITCH  
583 Diego Lopez, RedIRIS  
584 Tom Scavo, NCSA  
585 David Waite, Ping Identity Corporation