Holder-of-Key Web Browser SSO Profile

Committee Draft 01

14 February 2008

Specification URIs:

This Version:

http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-holder-of-key-browser-sso-cd-01.html

http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-holder-of-key-browser-sso-cd-01.odt

http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-holder-of-key-browser-sso-cd-01.pdf

Latest Version:

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

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

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

Latest Approved Version:

http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-holder-of-key-browser-sso-cd-01.html

http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-holder-of-key-browser-sso-cd-01.odt

http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-holder-of-key-browser-sso-cd-01.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 offers a potentially compatible addition 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 web browser 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 public/private keypair through client TLS authentication in one or more web browser transactions. The certificate itself is used as the token for holder-of-key validation of the resulting 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.

Status:

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

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

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

The non-normative errata page for this specification is located at http://www.oasis-open.org/committees/security.

Notices

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

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

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

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

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

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

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

The names "OASIS", [insert specific trademarked names, abbreviations, etc. here] are trademarks of OASIS, the owner and developer of this specification, and should be used only to refer to the organization and its official outputs. OASIS welcomes reference to, and implementation and use of, specifications, while reserving the right to enforce its marks against misleading uses. Please see http://www.oasis-open.org/who/trademark.php for above guidance.

Table of Contents

1 Introduction

In the scenario addressed by this profile, which is an extended version of the Web Browser SSO Profile in 4.1 of [SAML2Prof], a web user either accesses a web-based resource at a service provider or accesses an identity provider such that the service provider and desired resource are understood or implicit. In either case, the principal needs to acquire a SAML assertion from the identity provider. The user agent makes a request to the identity provider using client TLS authentication. The X.509 V3 certificate supplied in this transaction is used primarily to supply a public key that is associated with the principal, and may be used by the identity provider for authentication of the user, which may be done using any method of its choice. The identity provider then produces a response containing at least an assertion with holder-of-key subject confirmation and an authentication statement for the web browser to transport to the service provider. This assertion is presented by the user agent to the service provider over client TLS authentication to validate possession of the private key matching the holder-of-key confirmation in the assertion. The service provider relies on no information from the certificate beyond the key; instead, it consumes the assertion to create a security context. The TLS key may then be used to persist the security context rather than a cookie.

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

1.1 Terminology

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

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

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

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 Dsig Schema defined in [DSig].

This specification uses the following typographical conventions in text: `<SAMLElement>`, `<ns:ForeignElement>`, `Attribute`, **Datatype**, `OtherKeyword`.

1.2 Normative References

[DSig] D. Eastlake, J. Reagle, D. Solo. *XML-Signature Syntax and Processing*. World Wide Web Consortium Recommendation, 12 February 2002. See http://www.w3.org/TR/xmldsig-core/.

[IDPDisco] R. Widdowson, S. Cantor. Identity Provider Discovery Service Protocol and Profile, OASIS SSTC October 2007. Document ID sstc-saml-idp-discovery. See http://www.oasis-open.org/committees/security/.

[RFC 2119] S. Bradner. *Key words for use in RFCs to Indicate Requirement Levels*. IETF RFC 2119, March 1997. http://www.ietf.org/rfc/rfc2119.txt.

[RFC 4346] T. Dierks, E. Rescorla. *The Transport Layer Security (TLS) Protocol*. IETF RFC 4346, April 2006. http://www.ietf.org/rfc/rfc4346.txt.

[SAML2Bind] S. Cantor et al. Assertions *and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0*. OASIS Standard, March 2005. Document ID saml-core-2.0-os. See http://docs.oasis-open.org/security/saml/v2.0/saml-bindings-2.0-os.pdf.

[SAML2Core] S. Cantor et al. *Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0*. OASIS Standard, March 2005. Document ID saml-core-2.0-os. See http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf.

[SAML2Meta] S. Cantor et al. *Metadata for the OASIS Security Assertion Markup Language (SAML) V2.0*. OASIS Standard, March 2005. Document ID saml-metadata-2.0-os. See http://docs.oasis-open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf.

[SAML2Prof] S. Cantor et al. *Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0*. OASIS Standard, March 2005. Document ID saml-profiles-2.0-os. See **http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf**.

[SAML2Secure] F. Hirsch et al. *Security and Privacy Considerations for the OASIS Security Assertion Markup Language (SAML) v2.0*. OASIS SSTC, March 2005. 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**.

1.3 Conformance

1.3.1 Keyed Web Browser SSO Profile

A conformant implementation of a provider will support everything required in [SAML2Prof] for the Web Browser SSO profile as described in section 4.1. Both conformant service providers and identity providers MUST also support holder-of-ksey assertions and the acquisition of client keys from TLS connections, for validation and issuance of these 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 this confirmation method is an x.509 certificate using XML-Signature `<ds:KeyInfo>` with identifier `http://www.w3.org/2000/09/xmldsig#`. A SAML V2.0 bearer confirmation method with identifier `urn:oasis:names:tc:SAML:2.0:cm:bearer` MAY be included in assertions for consumption by endpoints that don't support holder-of-key confirmation.

Description: Given below.

Updates: Provides a compatible 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.

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 further lessens this chance, improving the viability of SAML-based browser 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, rather than the typical cookie. Cookies are often poorly protected by web browsers, allowing for theft of this session and impersonation of the user.

If a certificate can be used 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 is are greater challenges and no benefits to tricking the user into authenticating with legitimate credentials to a fraudulent party.

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

The combination of these two technologies 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.

The primary shortfall, beyond issuing x.509 end-entity certificates, is that such certificates contain a unique distinguished name for the issuer and the subject regularly containing personally identifying information. 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 `entityID` of the identity provider included in the certificate as `subjectAltName`. However, even in this case it's not generally feasible for the user to remain truly

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

2.3 Profile Overview

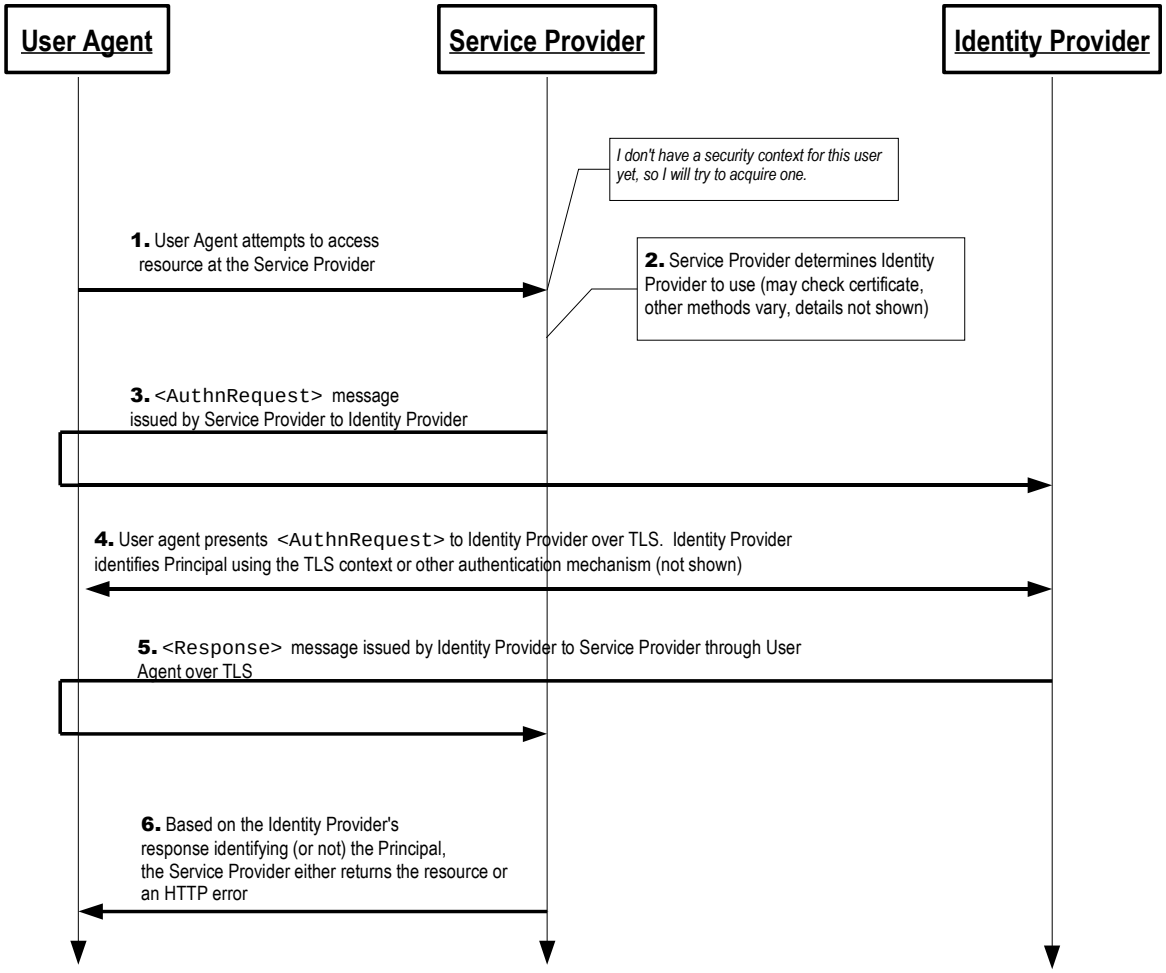

Figure 1 illustrates the basic template for achieving SSO. The following steps are described by the profile. Within an individual step, there may be one or more actual message exchanges depending on the binding used for that step and other implementation-dependent behavior.

1. HTTP Request to Service Providers

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

2. Service Provider Determines Identity Provider

The service provider determines the proper identity provider to which to direct the user agent. This may be done through use of a discovery service as described in [IDPDisco], by examining fields in a

certificate presented through client TLS authentication, such the `subject` or `subjectAltName`, or by any other means appropriate.

3. <AuthnRequest> issued by Service Provider to Identity Provider

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

4. Identity Provider identifies Principal

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

5. Identity Provider issues <Response> to Service Provider

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

6. Service Provider grants or denies access to Principal

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

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

2.4 Profile Description

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

2.4.1 HTTP Request to Service Provider

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

2.4.2 Service Provider Determines Identity Provider

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

2.4.3 <AuthnRequest> Issued by Service Provider to Identity Provider

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

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

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

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

2.4.4 Identity Provider Identifies Principal and Verifies Key Possession

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

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

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

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

2.4.5 Identity Provider Issues <Response> to Service Provider

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

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

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

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

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

2.4.6 Service Provider Grants or Denies Access to Principal

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

2.5 Use of Authentication Request Protocol

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

2.5.1 <AuthnRequest> Usage

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

- The <Issuer> element MUST be present and MUST contain the unique identifier of the requesting service provider. The Format attribute MUST be omitted or have a value of urn:oasis:names:tc:SAML:2.0:nameid-format:entity.

- If the initial request was made over TLS and this message is signed, a<Subject> element MAY be included in the request that includes the certificate presented by the user agent for which the service provider wishes to receive an assertion in a holder-of-key <SubjectConfirmation> element. A <NameID> SHOULD NOT be included, as the names used by the certificate authority may differ from those used by the identity provider. If the user agent fails this confirmation, then the identity provider MUST respond with a <Response> message containing an error status and no assertions.
- If the service provider wishes to permit the identity provider to establish a new identifier for the principal if none exists, it MUST include a <NameIDPolicy> element with the AllowCreate attribute set to true. Otherwise, only a principal for whom the identity provider has previously established an identifier usable by the service provider can be authenticated successfully.
- The <AuthnRequest> message MAY be signed (as directed by the SAML binding used). If the HTTP Artifact binding is used, authentication of the parties is OPTIONAL and any mechanism permitted by the binding MAY be used.

2.5.2 <AuthnRequest> Message Processing Rules

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

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

- It is RECOMMENDED that any <AssertionConsumerServiceURL> or <AssertionConsumerServiceIndex> elements in the request are verified as belonging to the entityID to whom the response will be sent. Use of holder-of-key confirmation eliminates the potential for assertion theft and encryption can prevent privacy loss; however, fulfilling this requirement is mandatory for compatibility with the standard Web Browser SSO profile.
- It is NOT obligated to honor the requested set of <Conditions> in the <AuthnRequest>, if any.

2.5.3 <Response> Usage

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

- The <Issuer> element of the <Response> MAY be omitted, but if present it MUST contain the unique identifier of the issuing identity provider; the Format attribute MUST be omitted or have a value of urn:oasis:names:tc:SAML:2.0:nameid-format:entity.
- It MUST contain at least one <Assertion>. Each assertion's <Issuer> element MUST contain the unique identifier of the issuing identity provider, and the Format attribute MUST be omitted or have a value of urn:oasis:names:tc:SAML:2.0:nameid-format:entity.
- The set of one or more assertions MUST collectively contain one <AuthnStatement> that reflects the authentication of the principal to the identity provider.
- The assertion containing an <AuthnStatement> MUST also contain a <Subject> element with at least one <SubjectConfirmation> element with a Method of urn:oasis:names:tc:SAML:2.0:cm:holder-of-key. Its

<SubjectConfirmationData> MUST contain the public key of the principal. This will typically take the form of a <ds:KeyInfo> element containing a <ds:X509Data> element with the principal's certificate encoded inside.

- If a bearer <SubjectConfirmation> element is also included, it MUST contain a <SubjectConfirmationData> element that contains a Recipient attribute containing the service provider's assertion consumer service URL and a NotOnOrAfter attribute that limits the window during which the assertion can be delivered. It MAY contain an Address attribute limiting the client address from which the assertion can be delivered. It MUST NOT contain a NotBefore attribute. If the containing message is in response to an <AuthnRequest>, then the InResponseTo attribute MUST match the request's ID.
- If the identity provider supports the Single Logout profile, defined in Section 4.4 of [SAML2Prof], the <AuthnStatement> MUST include a SessionIndex attribute to enable per-session logout requests by the service provider.
- <AttributeStatement> elements MAY be included in the assertion(s) at the discretion of the identity provider. The <AuthnRequest> MAY contain an AttributeConsumingServiceIndex XML attribute referencing information about desired or required attributes in [SAML2Meta]. The identity provider MAY ignore this, or send other attributes at its discretion.
- If the assertion containing the <AuthnStatement> is issued with an additional <SubjectConfirmation> of type bearer, it MUST contain an <AudienceRestriction> including the service provider's unique identifier as an <Audience>. Otherwise, it is RECOMMENDED for compatibility with the Web Browser SSO Profile.
- Other conditions (and other <Audience> elements) MAY be included as requested by the service provider or at the discretion of the identity provider. All such conditions MUST be understood by and accepted by the service provider in order for the assertion to be considered valid.

2.5.4 <Response> Message Processing Rules

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

- Verify any signatures present on the assertion(s) or the response.
- Verify that the key in the certificate presented by the user agent in mutual TLS authentication to the service provider matches the public key in the holder-of-key <SubjectConfirmationData>. The service provider SHOULD NOT rely on any other data in the certificate to process the assertion.
- Verify that the Recipient attribute in any bearer <SubjectConfirmationData> matches the assertion consumer service URL to which the <Response> or artifact was delivered.
- Verify that the NotOnOrAfter attribute in any bearer <SubjectConfirmationData> has not passed, subject to allowable clock skew between the providers.
- Verify that the InResponseTo attribute in any bearer <SubjectConfirmationData> equals the ID of its original <AuthnRequest> message, unless the response is unsolicited, in which case the attribute MUST NOT be present.
- Verify that any assertions relied upon are valid in other respects.

- If any bearer <SubjectConfirmationData> includes an Address attribute, the service provider MAY check the user agent's client address against it.

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

2.5.4.1 Artifact-Specific <Response> Message Processing Rules

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

It is RECOMMENDED that the identity provider ensure that only the service provider to whom the <Response> message has been issued is given the message as the result of an <ArtifactResolve> request.

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

2.5.4.2 POST-Specific <Response> Message Processing Rules

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

The service provider MUST ensure that bearer assertions are not replayed by maintaining the set of used ID values for the length of time for which the assertion would be considered valid based on the NotOnOrAfter attribute in the <SubjectConfirmationData>.

2.6 Unsolicited Responses

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

An unsolicited <Response> MUST NOT contain an InResponseTo attribute, nor should any bearer <SubjectConfirmationData> elements contain one. If metadata as specified in [SAML2Meta] is used, the <Response> or artifact SHOULD be delivered to the <md:AssertionConsumerService> endpoint of the service provider designated as the default.

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

2.7 Use of Metadata

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

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 MUST be met by counterparties.

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

The indexed endpoint element `<md:AssertionConsumerService>` is used to describe supported bindings and location(s) to which an identity provider may send responses to a service provider using this profile. The `index` attribute is used to distinguish the possible endpoints that may be specified by reference in the `<AuthnRequest>` message. The `isDefault` attribute is used to specify the endpoint to use if not specified in a request.

2.8 Compatibility

This profile is based on the Web Browser SSO Profile in [SAML2Prof]. The primary difference is the required holder-of-key `<SubjectConfirmation>`, no requirement for bearer `<SubjectConfirmation>`, and the resulting mandate of client TLS authentication for user agent interactions. The confirmation of the subject by key allows several of the requirements within that profile to be relaxed or removed, but there is nothing prohibiting meeting such requirements. All such requirements have been retained as RECOMMENDED or MAY. Additionally, inclusion of a secondary bearer `<SubjectConfirmation>` is possible in assertions and requests due to the satisfy-one nature of subject attestation.

As such, a request or assertion can be made sufficiently general to satisfy the requirements of both profiles and sent to an endpoint that only supports `urn:oasis:names:tc:SAML:2.0:profiles:SSO:browser` without special processing by that handler. This may be desirable to maximize interoperability with minimal implementation and deployment. However, deployers must be aware that in transacting with endpoints for `urn:oasis:names:tc:SAML:2.0:profiles:SSO:browser` they may be susceptible again to the some of the attacks mentioned in the introduction and as described in [SAML2Secure].

Appendix A. Acknowledgments

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

Participants:

Scott Cantor, Internet2

Patrick Harding, Ping Identity Corporation

Toshiyuki Kataoka, NII