SAML V2.0 Enhanced Client or Proxy Profile Version 2.0

Working Draft 01
14 October 2010

Specification URIs:

This Version:

http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-ecp-2.0-cd-01.html

http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-ecp-2.0-cd-01.odt (Authoritative)

http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-ecp-2.0-cd-01.pdf

Previous Version:

None

Latest Version:

http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-ecp-2.0.html

http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-ecp-2.0.odt (Authoritative)

http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-ecp-2.0.pdf

Technical Committee:

OASIS Security Services TC

Chair(s):

Thomas Hardjono, M.I.T.
Nate Kingenstein, Internet2

Editor(s):

Scott Cantor, Internet2

Related Work:

This specification updates the original ECP profile in [SAML2Prof] with backward-compatible additions of channel bindings and "Holder of Key" support.

Abstract:

The SAML V2.0 Enhanced Client or Proxy profile is a SSO profile for use with HTTP, and clients with the capability to directly contact a principal's identity provider(s) without requiring discovery and redirection by the service provider, as in the case of a browser. This specification updates the original profile by adding support for "Holder of Key" subject confirmation [SAML2HOK] and channel bindings [ChanBind].

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 current location noted above for possible later revisions of this document. This document is updated periodically on no particular schedule.

TC members should send comments on this specification to the TC's email list. Others should send comments to the TC by using the "Send A Comment" button on the TC'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 IPR section of the TC 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 name "OASIS" is a trademark 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 5
1.1 Notation 5
1.2 Terminology 6
1.3 Normative References 6
1.4 Non-Normative References 7
2 Enhanced Client or Proxy (ECP) Profile Version 2.0 8
2.1 Required Information 8
2.2 Profile Overview 8
2.3 Profile Description 8
2.3.1 ECP issues HTTP Request to Service Provider 8
2.3.1.1 Example 9
2.3.2 Service Provider Issues <samlp:AuthnRequest> to ECP 9
2.3.2.1 Example 10
2.3.3 ECP Determines Identity Provider 10
2.3.4 ECP issues <samlp:AuthnRequest> to Identity Provider 11
2.3.4.1 Example 11
2.3.5 Identity Provider Identifies Principal 11
2.3.6 Identity Provider issues <samlp:Response> to ECP 12
2.3.6.1 Verification of Channel Bindings 12
2.3.6.2 Example 12
2.3.7 ECP Conveys <samlp:Response> Message to Service Provider 13
2.3.7.1 Example 13
2.3.8 Service Provider Grants or Denies Access to Principal 14
2.3.9 Security Considerations 14
2.3.10 Use of Metadata 15
3 Conformance 16
3.1 SAML V2.0 Enhanced Client or Proxy Profile Version 2.0 16

1 Introduction

The SAML V2.0 Enhanced Client or Proxy (ECP) profile is a SSO profile for use with HTTP, and clients with the capability to directly contact a principal's identity provider(s) without requiring discovery and redirection by the service provider, as in the case of a browser. It is particularly useful for desktop or server-side HTTP clients.

This specification updates the original profile by adding support for "Holder of Key" subject confirmation [SAML2HOK] and channel bindings [ChanBind]. These additions are optional from a deployment perspective, and are incorporated in a backward-compatible fashion for use with existing implementations when the new features are not used. Both features can be used independently or together, to strengthen the security of the profile.

The addition of "Holder of Key" support has been well-motivated by previous work (e.g., [HOKSSO]), and is equally useful here to strenghten the security and widen the applicability of the original ECP Profile. Incorporation of this addition is accomplished in an analagous manner to [HOKSSO], but additional non-TLS options are permitted to allow for proof of key possession based on XML Signatures [XMLSig].

The addition of channel bindings takes advantage of the enhanced client's capability to intelligently add information to its exchange with the identity provider, in this case channel bindings between itself and the service provider. Combining this with channel bindings transmitted by the service provider in its (signed) `<samlp:AuthnRequest>` message allows the identity provider to perform channel bindings verification on behalf of both parties without introducing a requirement for key management into the enhanced client. This in turn allows the identity provider's typically strong and flexible authentication of the service provider to supplement (or substitute for) the typically ineffectual authentication that commercial TLS certificates allow the client to perform.

1.1 Notation

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this specification are to be interpreted as described in [RFC2119]. 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
`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].
`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].
`cb:`	urn:oasis:names:tc:SAML:protocol:ext:channel-binding	This is the SAML V2.0 channel binding extension namespace defined by this document and its accompanying schema.

`paos:`	urn:liberty:paos:2003-08	This is the PAOS V1.1 namespace defined in the PAOS V1.1 specification [PAOS].
`ecp:`	urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp	The is the Enhanced Client or Proxy Profile namespace defined in [SAML2Prof].
`S:`	http://schemas.xmlsoap.org/soap/envelope/	This is the SOAP 1.1 envelope namespace defined in [SOAP1.1].
`xsd:`	http://www.w3.org/2001/XMLSchema	This namespace is defined in the W3C XML Schema specification [Schema1]. In schema listings, this is the default namespace and no prefix is shown.
`xsi:`	http://www.w3.org/2001/XMLSchema-instance	This is the XML Schema namespace for schema-related markup that appears in XML instances [Schema1].

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

This specification uses the following typographical conventions in XML listings:

```
Listings of XML schemas appear like this.
```

```
Listings of XML examples appear like this.  These listings are non-normative.
```

1.2 Terminology

The term *TLS* as used in this specification refers to either the Secure Sockets Layer (SSL) Protocol 3.0 [SSL3] or any version of the Transport Layer Security (TLS) Protocol [RFC2246][RFC4346][RFC5246]. As used in this specification, the term *TLS* specifically does **not** refer to the SSL Protocol 2.0 [SSL2].

Unless otherwise noted, the term *X.509 certificate* refers to an X.509 client certificate as specified in the relevant version of the TLS protocol.

1.3 Normative References

[CBReg] Channel Binding Types Registry, IANA. http://www.iana.org/assignments/channel-binding-types/

[ChanBind] OASIS Working Draft, *SAML V2.0 Channel Binding Extensions Version 1.0*, September 2010. http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-channel-binding-ext.pdf

[ChanBind-XSD] OASIS Working Draft, *Extension Schema for SAML V2.0 Channel Binding Extensions Version 1.0*, September 2010. http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-channel-binding-ext.xsd

[PAOS] R. Aarts. *Liberty Reverse HTTP Binding for SOAP Specification* Version 1.1. Liberty Alliance Project, 2003. http://www.projectliberty.org/liberty/content/download/1219/7957/file/liberty-paos-v1.1.pdf

[RFC2045] N. Freed et al. *Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies*. IETF RFC 2045, November 1996. http://www.ietf.org/rfc/rfc2045.txt

[RFC2119] S. Bradner. *Key words for use in RFCs to Indicate Requirement Levels*. IETF RFC 2119, March 1997. http://www.ietf.org/rfc/rfc2119.txt

[RFC2246] T. Dierks, C. Allen. *The Transport Layer Security Protocol Version 1.0*. IETF RFC 2246, January 1999. http://www.ietf.org/rfc/rfc2246.txt

[RFC4346] T. Dierks, E. Rescorla. *The Transport Layer Security Protocol Version 1.1*. IETF RFC 4346, April 2006. http://www.ietf.org/rfc/rfc4346.txt

[RFC5056] N. Williams. *On the Use of Channel Bindings to Secure Channels*. IETF RFC 5056, November 2007. http://www.ietf.org/rfc/rfc5056.txt

[RFC5246] T. Dierks, E. Rescorla. *The Transport Layer Security Protocol Version 1.2*. IETF RFC 5246, August 2008. http://www.ietf.org/rfc/rfc5246.txt

[SAML2Bind] OASIS Standard, *Bindings for the OASIS Security Assertion Markup Language (SAML) V2.0*, March 2005. http://docs.oasis-open.org/security/saml/v2.0/saml-bindings-2.0-os.pdf

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

[SAML2Errata] OASIS Approved Errata, *SAML V2.0 Errata*, October 2009. http://docs.oasis-open.org/security/saml/v2.0/sstc-saml-approved-errata-2.0.pdf

[SAML2HOK] OASIS Committee Specification**,** *SAML V2.0 Holder-of-Key Assertion Profile Version 1.0*, January 2010. http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml2-holder-of-key-cs-02.pdf

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

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

[Schema1] H. S. Thompson et al. XML Schema Part 1: Structures. World Wide Web Consortium Recommendation, May 2001. http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/

[Schema2] Paul V. Biron, Ashok Malhotra. XML Schema Part 2: Datatypes. World Wide Web Consortium Recommendation, May 2001. http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/

[SOAP1.1] D. Box et al. *Simple Object Access Protocol (SOAP) 1.1.* World Wide Web Consortium Note, May 2000. http://www.w3.org/TR/SOAP

[SSL3] A. Freier, P. Karlton, P. Kocher. *The SSL Protocol Version 3.0*. Netscape Communications Corp., November 18, 1996. http://www.mozilla.org/projects/security/pki/nss/ssl/draft302.txt

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

1.4 Non-Normative References

[HOKSSO] OASIS Committee Specification**,** *SAML V2.0 Holder-of-Key Web Browser SSO Profile Version 1.0*, August 2010. http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-holder-of-key-browser-sso-cs-02.pdf

[RFC5929] J. Altman, et al. *Channel Bindings for TLS*. IETF RFC 5929, July 2010. http://www.ietf.org/rfc/rfc5929.txt

[SSL2] K. Hickman. *The SSL Protocol*. Netscape Communications Corp., February 9, 1995. http://www.mozilla.org/projects/security/pki/nss/ssl/draft02.html

2 Enhanced Client or Proxy (ECP) Profile Version 2.0

2.1 Required Information

Identification: `urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp:2.0`

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

Description: Given below.

Updates: The Enhanced Client or Proxy profile in Section 4.2 of [SAML2Prof].

2.2 Profile Overview

The original Enhanced Client or Proxy Profile [SAML2Prof] is a SAML "SSO" profile based on the Authentication Request protocol in [SAML2Core]. This profile builds on the original in a backwards-compatible fashion by adding two additional options:

- Channel Bindings
- "Holder of Key Subject" Confirmation

Both features are optional additions to the base profile, and use of this profile without either feature is by design wholly compatible with (and indistinguishable from) the original profile. The two additional options are independent and can be deployed together or separately.

The reader should be familiar with the original profile, and some of the normative content of this profile makes reference to the original. The steps outlined in the profile overview, Section 4.2.2, in [SAML2Prof] apply equally here.

2.3 Profile Description

The following sections describe each step in the profile. Some of the normative requirements of the original profile are repeated here for completeness, and to improve the technical presentation of the original material, which has proven somewhat confusing to follow. The normative definitions of the various header blocks, and their schemas, can be found in [PAOS] and [SAML2Prof].

In the steps that follow, all SOAP header blocks described by the profile MUST contain `actor` and `mustUnderstand` attributes set to `"http://schemas.xmlsoap.org/soap/actor/next"` and `"1"` respectively.

2.3.1 ECP issues HTTP Request to Service Provider

The client makes an arbitrary HTTP request to a service provider for a resource.

To indicate support for this profile, and the PAOS binding, the request MUST include the following HTTP header fields:

1. An `Accept` header indicating acceptance of the MIME type `"application/vnd.paos+xml"`
2. A `PAOS` header specifying the PAOS version with a value, at minimum, of `"urn:liberty:paos:2003-08"` and a supported service value of `"urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp"`. The service value MAY contain option values as follows:

- Support for channel bindings indicated by the option value "urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp:2.0:cb"
- Support for Holder-of-Key indicated by the option value "urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp:2.0:hok"

As defined by [PAOS], service values are delimited by semicolons, and options are comma-delimited from the service value and each other.

A client that supports the Holder-of-Key option MAY utilize TLS client authentication using an X.509 certificate (particularly assuming it plans to do so in subsequent communication with the service provider), but proof of key possession is not formally required during this step.

2.3.1.1 Example

The example demonstrates a client that supports both new options requesting a page. The `PAOS` header is one continuous line.

```
GET /secure/ HTTP/1.1
Host: sp.example.org
Accept: text/html; application/vnd.paos+xml
PAOS: ver="urn:liberty:paos:2003-08";
  "urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp",
  "urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp:2.0:cb",
  "urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp:2.0:hok"
```

2.3.2 Service Provider Issues <samlp:AuthnRequest> to ECP

If the service provider requires a security context for the principal before allowing access to the specified resource, it responds to the HTTP request in the previous step using the PAOS binding, including a `<samlp:AuthnRequest>` message in its HTTP response.

The HTTP response contains a `Status` code of 200, and the body consists of a SOAP 1.1 Envelope, which MUST contain the following:

1. A `<samlp:AuthnRequest>` element in the SOAP body. The rules for the request specified in the Browser SSO profile in Section 4.1.4.1 of [SAML2Prof] MUST be followed.
2. A `<paos:Request>` SOAP header block element (see Section 10 of [PAOS]). Its content MUST be as follows:
   - `service` MUST be set to "urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp"
   - `responseConsumerURL` MUST contain an absolute URL that specifies where error responses generated by the client should be sent; it MUST match the value of the `AssertionServiceConsumerURL` attribute in the `<samlp:AuthnRequest>` (or in its absence the location to which the identity provider is expected to target its response, such as a location derived from SAML metadata).
   - `messageID` MAY be set but is not required
3. An `<ecp:Request>` SOAP header block. This header contains information related to the authentication request that the client may need, such as a list of identity providers acceptable to the service provider, whether the client may interact with the principal through the user interface, and the service provider's (self-asserted) human-readable name. See Section 4.2.4.2 of [SAML2Prof].

The SOAP envelope MAY contain an `<ecp:RelayState>` SOAP header block (see Section 4.2.4.3 of [SAML2Prof]).

If the client includes the "urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp:2.0:cb" option value in its `PAOS` header, the service provider MAY include any number of `<cb:ChannelBindings>`

[ChanBind] SOAP header blocks in the SOAP envelope. Each element MUST contain no content and have a distinct `Type` attribute identifying a type of channel bindings supported by the service provider.

If the service provider requires channel bindings, but the client does not support the option, it MUST instead fail the original request directly. A client MAY require the use of channel bindings by requiring at least one `<cb:ChannelBindings>` SOAP header block be returned to it.

2.3.2.1 Example

```
<S:Envelope
   xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
   xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
   xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <paos:Request xmlns:paos="urn:liberty:paos:2003-08"
       service="urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp"
       responseConsumerURL="https://sp.example.org/PAOSConsumer"
       S:actor="http://schemas.xmlsoap.org/soap/actor/next"
       S:mustUnderstand="1"/>
    <ecp:Request xmlns:ecp="urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp"
       ProviderName="Example Service Provider" IsPassive="0"
       S:actor="http://schemas.xmlsoap.org/soap/actor/next"
       S:mustUnderstand="1">
      <saml:Issuer>https://sp.example.org/entity</saml:Issuer>
      <samlp:IDPList>
        <samlp:IDPEntry ProviderID="https://idp.example.org/entity"
          Name="Example Identity Provider"
          Loc="https://idp.example.org/saml2/sso"/>
      </samlp:IDPList>
    </ecp:Request>
    <ecp:RelayState xmlns:ecp="urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp"
        S:actor="http://schemas.xmlsoap.org/soap/actor/next"
        S:mustUnderstand="1">
      AGDY854379dskssda
    </ecp:RelayState>
    <cb:ChannelBindings xmlns:cb="urn:oasis:names:tc:SAML:ext:channel-binding"
        Type="tls-server-end-point"
        S:actor="http://schemas.xmlsoap.org/soap/actor/next"
        S:mustUnderstand="1"/>
  </S:Header>
  <S:Body>
    <samlp:AuthnRequest>
    ....
      <samlp:Extensions>
        <cb:ChannelBindings
          xmlns:cb="urn:oasis:names:tc:SAML:ext:channel-binding"
          Type="tls-server-end-point">
          ...base64-encoded hash of server's SSL cert...
        </cb:ChannelBindings>
      </samlp:Extensions>
    ....
    </samlp:AuthnRequest>
  </S:Body>
</S:Envelope>
```

2.3.3 ECP Determines Identity Provider

The client determines which identity provider is appropriate, possibly influenced by information found in the `<ecp:Request>` header block received in the previous step. It is out of scope how the client is provisioned with identity provider information, but SAML V2.0 metadata [SAML2Meta], or a derivative, MAY be used.

2.3.4 ECP issues <samlp:AuthnRequest> to Identity Provider

The client routes the SOAP envelope containing the `<samlp:AuthnRequest>` message on to the selected identity provider, using a modified form of the SAML SOAP binding [SAML2Bind]. Any header blocks received from the service provider MUST be removed.

The SAML request is submitted via the SAML SOAP binding in the usual fashion, but the identity provider MAY respond to the client's HTTP request with an HTTP response containing, for example, an HTML login form or some other presentation-oriented response. A sequence of HTTP exchanges MAY take place, but ultimately the identity provider MUST complete the SAML SOAP binding exchange and return a SAML response.

However, the use of HTML and a presentation-oriented interface for authentication is NOT RECOMMENDED. Identity providers and clients SHOULD support the use of SOAP- or HTTP-based authentication mechanisms that can be implemented without (or with minimal) user interface support.

If the client supports the use of channel bindings and the service provider requested their use, the client MUST include at least one `<cb:ChannelBindings>` SOAP header block in the SOAP message to the identity provider containing channel bindings of a type supported by the service provider. (The channel bindings are those of the channel between the client and the service provider.)

2.3.4.1 Example

Typically this request would be accompanied by some form of HTTP or TLS client authentication.

```
<S:Envelope
   xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
   xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <cb:ChannelBindings xmlns:cb="urn:oasis:names:tc:SAML:ext:channel-binding"
        Type="tls-server-end-point"
        S:actor="http://schemas.xmlsoap.org/soap/actor/next"
        S:mustUnderstand="1">
          ...base64-encoded hash of SP's SSL cert...
    </cb:ChannelBindings>
  </S:Header>
  <S:Body>
    <samlp:AuthnRequest>
    ....
      <samlp:Extensions>
        <cb:ChannelBindings
          xmlns:cb="urn:oasis:names:tc:SAML:ext:channel-binding"
          Type="tls-server-end-point">
          ...base64-encoded hash of server's SSL cert...
        </cb:ChannelBindings>
      </samlp:Extensions>
    ....
    </samlp:AuthnRequest>
  </S:Body>
</S:Envelope>
```

2.3.5 Identity Provider Identifies Principal

At any time during or subsequent to the previous step, the identity provider MUST establish the identity of the principal (unless it returns an error to the service provider). The `ForceAuthn` `<samlp:AuthnRequest>` attribute, if present with a value of `true`, obligates the identity provider to freshly establish this identity, rather than relying on an 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 `<samlp:AuthnRequest>` message in the form of the `<samlp:RequestedAuthnContext>` element.

2.3.6 Identity Provider issues <samlp:Response> to ECP

Regardless of the success or failure of authentication of the principal and of processing the <samlp:AuthnRequest> message, the identity provider MUST return a <samlp:Response> message or SOAP fault. The response is conveyed using the SAML SOAP binding [SAML2Bind], with the <samlp:Response> message in the body (unless a SOAP fault is signaled).

The rules for the response specified in the Browser SSO profile in Section 4.1.4.2 of [SAML2Prof] MUST be followed.

If a response is included, the SOAP envelope MUST contain an <ecp:Response> SOAP header block whose AssertionConsumerServiceURL attribute is set to the location to which the <samlp:Response> message is to be delivered by the client. The location is derived from the <samlp:AuthnRequest> message. See Section 4.2.4.4 of [SAML2Prof].

The SOAP envelope MAY contain an <ecp:RelayState> SOAP header block (typically in the case of an unsolicited response).

2.3.6.1 Verification of Channel Bindings

The identity provider is also responsible for verifying channel bindings supplied by the client and service provider (by comparing them).

The service provider's channel bindings (if any) are located within <cb:ChannelBindings> elements in the <samlp:Extensions> element of the <samlp:AuthnRequest> message. If such extensions exist but the <samlp:AuthnRequest> message is unsigned, or if the client did not supply at least one matching <cb:ChannelBindings> SOAP header block, the identity provider MUST respond with a <samlp:Response> message containing an error status.

Additionally, if the service provider does not include any <cb:ChannelBindings> elements in its <samlp:AuthnRequest> message, and the client includes a <cb:ChannelBindings> SOAP header block in its message, then the identity provider MUST respond with a <samlp:Response> message containing an error status.

Assuming channel bindings are supplied by both parties, and they match, then the identity provider MUST include at least one <cb:ChannelBindings> element in the <saml:Advice> element of any <saml:Assertion> elements that it returns to the client for delivery to the service provider. It also MUST include the same <cb:ChannelBindings> element(s) as SOAP header blocks in its message to the client. All such <cb:ChannelBindings> elements MAY contain no element content (indicating only the type of channel bindings verified).

2.3.6.2 Example

```
<S:Envelope
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
    xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
    xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
  <S:Header>
    <ecp:Response xmlns:ecp="urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp"
       AssertionConsumerServiceURL="https://sp.example.org/PAOSConsumer"
       S:actor="http://schemas.xmlsoap.org/soap/actor/next"
       S:mustUnderstand="1"/>
    <cb:ChannelBindings xmlns:cb="urn:oasis:names:tc:SAML:ext:channel-binding"
        Type="tls-server-end-point"
        S:actor="http://schemas.xmlsoap.org/soap/actor/next"
        S:mustUnderstand="1"/>
  </S:Header>
  <S:Body>
    <samlp:Response>
      ....
      <saml:Assertion>
```

```
        ....
        <saml:Advice>
          <cb:ChannelBindings
            xmlns:cb="urn:oasis:names:tc:SAML:ext:channel-binding"
            Type="tls-server-end-point"/>
        <saml:Advice>
        ....
      </saml:Assertion>
      ....
    </samlp:Response>
  </S:Body>
</S:Envelope>
```

2.3.7 ECP Conveys <samlp:Response> Message to Service Provider

The client MUST compare the `AssertionConsumerServiceURL` attribute from the identity provider's `<ecp:Response>` SOAP header block to the `responseConsumerURL` attribute found in the `<paos:Request>` SOAP header block sent to the client by the service provider (see Section 2.3.2). This comparison is used for security purposes to confirm the correct response destination. If the values do not match, then the client MUST generate a SOAP fault response to the service provider and MUST NOT return the SAML response it received from the identity provider.

If the client included one or more `<cb:ChannelBindings>` SOAP header blocks in its request to the identity provider, but no `<cb:ChannelBindings>` SOAP header blocks are in the response from the identity provider, the client MUST generate a SOAP fault response to the service provider. While a conformant identity provider would generate a SAML error response anyway, the absence of such information could instead indicate that the identity provider did not support the channel bindings extension at all.

Otherwise, the client routes the SOAP envelope containing the `<samlp:Response>` message (or SOAP fault) back to the service provider at the location designated by the identity provider's `<ecp:Response>` SOAP header block using the PAOS binding. Any header blocks received from the identity provider MUST be removed first.

The client may need to add `<paos:Response>` and `<ecp:RelayState>` SOAP header blocks to the SOAP Envelope as follows:

The `<paos:Response>` SOAP header block in the response to the service provider is generally used to correlate the response to an earlier request from the service provider. In this profile, the header is not strictly required since the `<samlp:Response>` element's `InResponseTo` attribute can be used for this purpose, but if the `<paos:Request>` SOAP header block contained a `messageID`, then a `<paos:Response>` SOAP header block MUST be added, with its `refToMessageID` attribute set to that value. See Section 10 of [PAOS].

The `<ecp:RelayState>` header block value is typically provided by the service provider to the client with its request, but if the identity provider is producing an unsolicited response (without having received a corresponding SAML request), then it MAY include a header block in its response to the client that indicates, based on mutual agreement with the service provider, how to handle subsequent interactions with the client. This MAY be the URL of a resource at the service provider.

If the service provider included an `<ecp:RelayState>` SOAP header block in its request, or if the identity provider included an `<ecp:RelayState>` SOAP header block in its response, then the client MUST include an identical header block with the response sent to the service provider. The service provider's value for this header block (if any) MUST take precedence.

2.3.7.1 Example

```
<S:Envelope
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
    xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
    xmlns:S="http://schemas.xmlsoap.org/soap/envelope/">
```

```
  <S:Header>
    <ecp:RelayState xmlns:ecp="urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp"
        S:actor="http://schemas.xmlsoap.org/soap/actor/next"
        S:mustUnderstand="1">
      AGDY854379dskssda
    </ecp:RelayState>
  </S:Header>
  <S:Body>
    <samlp:Response>
      ....
      <saml:Assertion>
        ....
        <saml:Advice>
          <cb:ChannelBindings
            xmlns:cb="urn:oasis:names:tc:SAML:ext:channel-binding"
            Type="tls-server-end-point"/>
        <saml:Advice>
        ....
      </saml:Assertion>
      ....
    </samlp:Response>
  </S:Body>
</S:Envelope>
```

2.3.8 Service Provider Grants or Denies Access to Principal

Once the service provider has received the SAML response in an HTTP request (in a SOAP Envelope using PAOS), it MUST process the response in accordance with the rules specified by the Browser SSO profile in Sections 4.1.4.3 and 4.1.4.5 of [SAML2Prof]. That is, the same processing rules used when receiving the `<samlp:Response>` with the HTTP POST binding generally apply to the use of PAOS.

In addition, if the service provider included at least one `<cb:ChannelBindings>` extension in its `<samlp:AuthnRequest>`, any `<saml:Assertion>` received SHOULD be rejected if it does not contain a corresponding `<cb:ChannelBindings>` extension in its `<saml:Advice>` element.

In the case of an error in processing the response, the service provider MUST return a an HTTP error status. Otherwise, it may respond with the service data or other information, or with a redirection to the original request location, or any other valid HTTP response.

2.3.9 Security Considerations

The `<samlp:AuthnRequest>` message MUST be signed if the channel bindings extension option is used.

Per the rules specified by the Browser SSO profile, the assertions enclosed in the `<samlp:Response>` MUST be integrity protected at either the individual assertion or response level.

The delivery of the response in the SOAP envelope via PAOS is essentially analogous to the use of the HTTP POST binding and security countermeasures appropriate to that binding are assumed.

All SOAP headers SHOULD be integrity protected, such as with the use of TLS over every HTTP exchange with the client, though alternative mechanisms MAY be employed.

The service provider SHOULD be authenticated to the client. Server-side TLS authentication may be used, but channel bindings are RECOMMENDED for this purpose, as they address many of the exposures common to commercial TLS infrastructure.

The client MUST authenticate the identity provider during the transmission of the `<samlp:AuthnRequest>` message and prior to the submission of credentials vulnerable to theft. The client SHOULD be authenticated to the identity provider, such as by maintaining an authenticated session. Any HTTP exchanges subsequent to the delivery of the `<samlp:AuthnRequest>` message and before

the identity provider returns a <samlp:Response> MUST be securely associated with the original request.

The assertions issued by the identity provider SHOULD be encrypted with a key that can be securely associated with the service provider. The key used SHOULD NOT be derived from a TLS certificate believed to belong to the service provider by means of probing endpoints unless that key is otherwise authenticatable and known to be usable for encryption.

2.3.10 Use of Metadata

The rules specified in the Browser SSO profile in Section 4.1.6 of [SAML2Prof] apply to this profile as well. Specifically, <md:AssertionConsumerService> element(s) with a Binding attribute of "urn:oasis:names:tc:SAML:2.0:bindings:PAOS" SHOULD be used to describe the supported location(s) to which an identity provider may send responses to a service provider using this profile.

In addition, <md:SingleSignOnService> elements(s) with a Binding attribute of "urn:oasis:names:tc:SAML:2.0:bindings:SOAP" SHOULD be used to describe the supported location(s) to which a client may relay requests to an identity provider using this profile.

The cb:supportsChannelBindings attribute defined in [ChanBind] SHOULD be added to both types of endpoints to indicate support for channel bindings in conjunction with this profile.

3 Conformance

3.1 SAML V2.0 Enhanced Client or Proxy Profile Version 2.0

Appendix A. Acknowledgments

The editors would like to acknowledge the contributions of the OASIS Security Services Technical Committee, whose voting members at the time of publication were:

- TBD

The editor would also like to acknowledge the following contributors:

- Nicolas Williams, Oracle Corporation

Appendix B. Revision History

- Working Draft 01 - Initial draft. Channel bindings material added, but not (yet) holder of key.