SAML V2.0 Enhanced Client or Proxy Profile Version 2.0

Working Draft 034

223 August 2011

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 is a Working Draft and as such as no official standing with regard to the OASIS Technical Committee Process.

Related Work:

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

Declared XML Namespace(s):

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

Table of Contents

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 (and non-public key) options are permitted to allow for proof of key possession based on XML Signatures [XMLSig] or HTTP-compatible mechanisms that may emerge in the future.

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 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 [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 [ChanBind].

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] and updated by this specification.
S:	http://schemas.xmlsoap.org/soap/envelope/	This is the SOAP 1.1 envelope namespace defined in [SOAP1.1].
ds:	http://www.w3.org/2000/09/xmldsig#	This is the XML digital signature namespace defined in the XML Signature Syntax and Processing specification [XMLSig].
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*, August 2011. 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*, August 2011. http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-channel-binding-ext.xsd

[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

[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

[RFC2616] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee. *Hypertext Transfer Protocol – HTTP 1.1*. IETF RFC 2616, June 1999. http://www.ietf.org/rfc/rfc2616.txt

[RFC2617] J. Franks, P. Hallam-Baker, J. Hostetler, S. Lawrence, P. Leach, A Luotonen, L.Stewart. *HTTP Authentication: Basic and Digest Authentication*. IETF RFC 2617, June 1999. http://www.ietf.org/rfc/rfc2617.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

[RFC5280] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, W. Polk. *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*. IETF RFC 5280, May 2008. http://www.ietf.org/rfc/rfc5280.txt

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

[RFC6265] A. Barth. *HTTP State Management Mechanism*. IETF RFC 6265, April 2011. http://www.ietf.org/rfc/rfc6265.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

[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 authentication profile based on the Authentication Request protocol in [SAML2Core]. This profile builds on the original in a backwardly-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 may wish 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 unless otherwise indicated.

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:protocol:ext:channel-binding"
- Support for Holder-of-Key subject confirmation indicated by the option value "urn:oasis:names:tc:SAML:2.0:cm:holder-of-key"

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:protocol:ext:channel-binding",
  "urn:oasis:names:tc:SAML:2.0:cm:holder-of-key"
```

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:protocol:ext:channel-binding" option value in its PAOS header, then the service provider MAY include any number of <cb:ChannelBindings> [ChanBind] SOAP header blocks in the SOAP envelope. Each element MUST contain no content (i.e., be an empty element) and have a distinct Type attribute identifying a type of channel bindings supported by the service provider. If the service provider supports channel bindings via an application layer API that limits its knowledge as to the types supported, then it MUST instead include a single, empty <cb:ChannelBindings> SOAP header block with no Type attribute.

In parallel, the service provider MUST include a corresponding <cb:ChannelBindings> element in the <samlp:Extensions> element of its <samlp:AuthnRequest> message for each SOAP header block it attaches, containing channel bindings of ~~the~~a ~~associated~~particular type. Within each extension element, the Type attribute M~~UST~~AY be set to the channel binding type (if known), and the raw channel binding data MUST be base64-encoded and the result used as the content of the element. When channel bindings are included, the <samlp:AuthnRequest> message MUST be signed via [XMLSig].

If the service provider requires channel bindings, but the client does not support the option, then it MUST instead fail the original request directly. A client MAY require the use of channel bindings by requiring that at least one <cb:ChannelBindings> SOAP header block be returned to it. If the Type is not specified, then it is assumed that the appropriate type to use is known out of band.

If the client includes the "urn:oasis:names:tc:SAML:2.0:cm:holder-of-key" option value in its PAOS header, then the service provider MAY include one or more <ecp:SubjectConfirmation> SOAP header blocks in the SOAP envelope. Each element MUST contain no content and have a distinct Method attribute identifying a type of subject confirmation supported by the service provider. See below for a formal description of this header block.

In the absence of any <ecp:SubjectConfirmation> SOAP header blocks, the client MUST rely on out-of-band knowledge, or assume the use of the "urn:oasis:names:tc:SAML:2.0:cm:bearer" confirmation type (as in the original profile). There is no precendence implied if more than one method is included.

Use of Method values other than "urn:oasis:names:tc:SAML:2.0:cm:bearer" or "urn:oasis:names:tc:SAML:2.0:cm:holder-of-key" are undefined by this profile.

2.3.2.1 <ecp:SubjectConfirmation> Header Block

The <ecp:SubjectConfirmation> element is a SOAP header block that identifies a method of subject confirmation supported by a service provider, or how an identity provider expects subject confirmation to be performed by the client. It contains the following attributes and elements:

S:mustUnderstand [Required]

The value MUST be "1" (true).

S:actor [Required]

The value MUST be "http://schemas.xmlsoap.org/soap/actor/next".

Method [Required]

A URI reference that identifies a protocol or mechanism to be used to confirm the subject.

<saml:SubjectConfirmationData> [Optional]

Identifies the subject confirmation data bound into the issued assertion(s) by an identity provider.

The following schema fragment defines the <ecp:SubjectConfirmation> element and its **ecp:SubjectConfirmationType** complex type:

```
<element name="SubjectConfirmation" type="ecp:SubjectConfirmationType"/>
<complexType name="SubjectConfirmationType">
  <sequence>
    <element ref="saml:SubjectConfirmationData" minOccurs="0"/>
  </sequence>
  <attribute ref="S:mustUnderstand" use="required"/>
  <attribute ref="S:actor" use="required"/>
  <attribute name="Method" type="anyURI" use="required"/>
</complexType>
```

2.3.2.2 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>
    <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 SP'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 Routes <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~~ derived from the channel between the client and the service provider.~~)~~ Within each header block, the Type attribute ~~MUST~~MAY be set to the channel binding type (if known), and the raw channel binding data MUST be base64-encoded and the result used as the content of the element.

2.3.4.1 "Holder of Key" Subject Confirmation

If the client, service provider, and identity provider all support the use of the "Holder of Key" subject confirmation method (and if it is to be used), then the client MUST demonstrate proof of possession of a key in communicating with the identity provider. This specification does not prescribe the means by which this is done, but for interoperability the following mechanisms are enumerated:

- TLS Client Authentication
- An XML Signature over the entire SOAP message via a single <ds:Reference> with a whole document URI reference ("") and no transforms.

Other forms of authentication MAY be used in conjunction with this; see Section 2.3.5.1 for further discussion.

In the case that an XML Signature or related mechanism is used (in other words, if proof of possession is independent of the transport), the client MAY attach an additional set of <cb:ChannelBindings> SOAP header blocks to the message that carry channel bindings between the client and the identity provider, using the same encoding rules. ~~These~~Such header blocks are distinguished from those representing the client/service provider channel by the **absence** of the S:actor XML attribute.

2.3.4.2 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 SP'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.

2.3.5.1 "Holder of Key" Authentication

If "Holder of Key" subject confirmation is to be used, then the identity provider MAY use TLS client authentication to identify the principal. The identity provider MAY validate the presented X.509 certificate as described in [RFC5280], but this is by no means a requirement.

The key obtained as a result of the TLS handshake, XML Signature, or other mechanism MUST be known to be associated with the principal (see Section 2.4 of [SAML2HOK] in the case that an X.509 certificate is obtained). Precisely how the identity provider satisfies this requirement is out of scope, but of course direct authentication of the principal via an X.509 certificate may offer significant benefits for some deployments.

Failure to demonstrate proof of possession of a key known to be associated with the principal MUST result in an authentication failure.

In the case that TLS Client Authentication is not used but the SOAP message is integrity protected in some other fashion, the identity provider MAY rely on the assistance of any included `<cb:ChannelBindings>` SOAP header blocks without an `S:actor` attribute to verify the channel between the client and itself ~~is intact~~. An identity provider SHOULD insist on verification of channel bindings between itself and the client before accepting a signed message as proof of key possession. If channel bindings are supplied and cannot be verified, then the identity provider MUST fail the authentication.

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).

In the case of "Bearer" subject confirmation, the rules for the response specified in the Browser SSO profile in Section 4.1.4.2 of [SAML2Prof] MUST be followed.

In the case of "Holder of Key" subject confirmation with an X.509 certificate, the rules for the response specified in the Holder of Key Web Browser profile in Section 2.7.3 of [HOKSSO] MUST be followed. If an X.509 certificate is not used, then the same rules MUST be followed, except that the `<ds:KeyInfo>` element in the included `<saml:SubjectConfirmationData>` element is not constrained by [SAML2HOK] and is left to the discretion of the identity provider. Typically a bare key representation is suggested.

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 ~~of a given type~~ are supplied by both parties, and ~~they~~a match exists, 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 (optionally indicating ~~only~~ the type of channel bindings that it verified, if known, or simply acting as an empty signalling element).

Note that the identity provider need not understand or “support” the various types of channel bindings it may encounter in these comparisons. It need only match the `Type` attributes (if set) and element content via a binary comparison.

2.3.6.2 "Holder of Key" Subject Confirmation

If "Holder of Key" subject confirmation is used, and the response from the identity provider is not an error or fault, then the identity provider MUST include a `<ecp:SubjectConfirmation>` SOAP header block with a `Method` of `"urn:oasis:names:tc:SAML:2.0:cm:holder-of-key"`. The header block MUST contain a `<saml:SubjectConfirmationData>` element identical to that from the SAML assertion(s) included in the response for the "Holder of Key" confirmation method. That is, it must identify the proof key to be used by the client.

2.3.6.3 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 Routes <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 "Holder of Key" Subject Confirmation

If "Holder of Key" subject confirmation is used, the client MUST demonstrate proof of possession of the key identified by the <ecp:SubjectConfirmation> header block described by Section 2.3.6.2. This specification does not prescribe the means by which this is done, but for interoperability the following mechanisms are enumerated:

- TLS Client Authentication
- An XML Signature over the entire SOAP message via a single <ds:Reference> with a whole document URI reference ("") and no transforms.

2.3.7.2 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: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.

If "Holder of Key" subject confirmation is used in conjunction with an X.509 certificate, then any such assertion(s) contained in the response MUST be confirmed in accordance with the SAML V2.0 Holder-of-Key Assertion Profile [SAML2HOK], with the confirmation key obtained via the verification of a supported proof mechanism as described by Section 2.3.7.1. If an X.509 certificate is not used, then the service provider MUST verify that the key identified by the `<saml:SubjectConfirmationData>` element matches the key used by the client, but the exact means are outside the scope of this specification.

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 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. It MAY rely on cookies [RFC6265] to maintain a session with the client.

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 and Holder of Key Browser profiles, 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 (even in the case of "Bearer" subject confirmation), 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 can help to address many of the exposures common to commercial TLS infrastructure (assuming the identity provider is trustworthy).

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.

If "Holder of Key" subject confirmation is used in conjunction with a message-level proof of possession to the identity provider or service provider such as an XML Signature [XMLSig] instead of a transport-level mechanism like TLS client authentication, then the use of channel bindings is RECOMMENDED. Absent such a mechanism, it is possible for a MITM to replay a signed message obtained from the legitimate client. Replay and freshness checking partially mitigate this threat.

Implementers are also encouraged to review the applicable security and privacy considerations outlined in [HOKSSO] and [SAML2HOK] (presuming that X.509 certificates are used).

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.

If "Holder of Key" subject confirmation is supported, the metadata usage described in Section 2.8 of [HOKSSO] SHOULD be used in combination with appropriate hoksso:ProtocolBinding values.

An example of a conforming <md:SingleSignOnService> element with "Holder of Key" support is as follows:

```
<md:SingleSignOnService
 xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
 xmlns:hoksso="urn:oasis:names:tc:SAML:2.0:profiles:holder-of-key:SSO:browser"
 hoksso:ProtocolBinding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"
 Binding="urn:oasis:names:tc:SAML:2.0:profiles:holder-of-key:SSO:browser"
 Location="https://your-idp.example.org/some/path" />
```

Similarly, an example of a conforming <md:AssertionConsumerService> element with "Holder of Key" support is as follows:

```
<md:AssertionConsumerService index="1" isDefault="true"
 xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
 xmlns:hoksso="urn:oasis:names:tc:SAML:2.0:profiles:holder-of-key:SSO:browser"
 hoksso:ProtocolBinding="urn:oasis:names:tc:SAML:2.0:bindings:PAOS"
 Binding="urn:oasis:names:tc:SAML:2.0:profiles:holder-of-key:SSO:browser"
 Location="https://your-sp.example.org/some/path" />
```

3 Conformance

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

3.1.1 Identity Provider Conformance

An identity provider that conforms to this profile MUST adhere to the relevant normative text in Section 2.3, including the verification of channel bindings and the use of "Holder of Key" subject confirmation. The use of X.509 certificates as a proof mechanism MUST be supported. Other key forms are OPTIONAL.

It MUST support the use of HTTP Basic Authentication, TLS Client Authentication, and the XML Signature mechanism described in section 2.3.4.1.

It MUST also support verification of channel bindings of type "tls-server-end-point" [RFC5929] between itself and the client during authentication via signed message.

3.1.2 Service Provider Conformance

A service provider that conforms to this profile MUST adhere to the relevant normative text in Section 2.3, and MUST support the use of channel bindings of type "tls-server-end-point" [RFC5929].

Support for "Holder of Key" subject confirmation is OPTIONAL, but if supported then both TLS Client Authentication and the XML Signature mechanism described in Section 2.3.7.1 MUST be supported as proof of possession mechanisms. The use of X.509 certificates with these mechanisms MUST be supported. Other key forms are OPTIONAL.

3.1.3 Enhanced Client or Proxy Conformance

An enhanced client or proxy that conforms to this profile MUST adhere to the relevant normative text in Section 2.3, and MUST support HTTP 1.1 [RFC2616] and the use of cookies [RFC6265].

It MUST support the use of channel bindings of type "tls-server-end-point" [RFC5929], both with respect to the service provider and identity provider channels (the latter only if “Holder of Key” via a signature-based authentication mechanism is supported).

It MUST support the use of HTTP Basic Authentication [RFC2617] and TLS Client Authentication to an identity provider.

Support for "Holder of Key" subject confirmation is OPTIONAL.

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
- Simon Josefsson, SJD AB

Appendix B. Revision History

- WD 01 – Initial draft. Channel bindings material added, but not (yet) holder of key.
- WD 02 – Added Holder of Key material with normative call outs to earlier HoK profiles.
- WD 03 – Nailed down encoding of channel bindings, added HTTP-related conformance requirements for clients, client/IdP channel bindings, more extensive support for XML Signature, and optional use of non-X.509 key proofs.
- WD 04 – Allow for type-less channel bindings so that GSS-API and SASL scenarios can coexist compatibly with pure SAML usage.