SAML V2.0 Holder-of-Key Assertion Request Profiles

Working Draft 021, 2 June 2009 – December 2008

Specification URLs:
TBD

Technical Committee:
OASIS Security Services TC

Chair(s):
Hal Lockhart, BEA Systems, Inc.
Brian Campbell, Ping Identity Corporation

Editors:
Tom Scavo, National Center for Supercomputing Applications (NCSA)

Abstract:
This series of specifications describe how a subject self-issues a SAML request and obtains a holder-of-key SAML assertion, which the subject subsequently uses as a strong security token at its discretion. Specific profile requirements for a <samlp:AuthnRequest> and a <samlp:AttributeQuery> are given, but any request type derived from the abstract samlp:RequestAbstractType complex type is applicable.

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

Copyright © OASIS Open 2008–2009. All Rights Reserved.

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.

OASIS distinguishes itself from licensing agencies. It does not, and cannot, negotiate licensing terms. 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 .................................................................................................................. 4
   1.1 Notation .................................................................................................................. 4
   1.2 Terminology ............................................................................................................ 5
   1.3 Normative References ............................................................................................ 5
   1.4 Non-normative References ..................................................................................... 6
   1.5 Conformance ........................................................................................................... 6
       1.5.1 SAML V2.0 Holder-of-Key Self-Request Profile ............................................. 6
       1.5.2 SAML V2.0 Self-AuthnRequest Profile ......................................................... 6
       1.5.3 SAML V2.0 Attribute Self-Query Profile ....................................................... 6
2 SAML V2.0 Holder-of-Key Self-Request Profile ......................................................... 7
   2.1 Required Information ............................................................................................... 7
   2.2 Profile Description .................................................................................................. 7
   2.3 X.509 Certificate Usage .......................................................................................... 8
   2.4 Issuing the SAML Request ...................................................................................... 8
   2.5 Negotiating the SAML Exchange .......................................................................... 9
   2.6 Issuing the SAML Response ................................................................................... 9
   2.7 Processing the SAML Response ............................................................................ 10
   2.8 Use of Metadata .................................................................................................... 10
   2.9 Security and Privacy Considerations .................................................................... 11
       2.9.1 Comparison with Web Browser SSO ............................................................. 11
       2.9.2 Policy issues .................................................................................................. 11
3 SAML V2.0 Self-AuthnRequest Profile ....................................................................... 13
   3.1 Required Information ............................................................................................. 13
   3.2 Profile Description .................................................................................................. 13
   3.3 Issuing the Authentication Request ....................................................................... 13
   3.4 Issuing the Authentication Response .................................................................... 13
   3.5 Example .................................................................................................................. 14
4 SAML V2.0 Attribute Self-Query Profile ..................................................................... 15
   4.1 Required Information ............................................................................................. 15
   4.2 Profile Description .................................................................................................. 15
   4.3 Issuing the Attribute Query ................................................................................... 15
   4.4 Issuing the Attribute Response .............................................................................. 15
   4.5 Example .................................................................................................................. 15

Appendix A. Acknowledgments ....................................................................................... 17
Appendix B. Revision History .......................................................................................... 18
1 Introduction

The SAML V2.0 Holder-of-Key Assertion Request Profiles are a series of profile specifications that describe how a subject self-issues a SAML request and obtains a holder-of-key SAML assertion, which the subject subsequently uses as a strong security token at its discretion. Specific profile requirements for a `<samlp:AuthnRequest>` and a `<samlp:AttributeQuery>` are given, but any request type derived from the abstract `samlp:RequestAbstractType` complex type is applicable.

The following specifications are included in this document:

- SAML V2.0 Holder-of-Key Self-Request Profile
- SAML V2.0 Self-AuthnRequest Profile
- SAML V2.0 Attribute Self-Query Profile

The SAML V2.0 Holder-of-Key Self-Request Profile is a refinement of the SAML V2.0 Request-Response Protocol [SAML2Core] where the subject is the requester. The subject self-issues a SAML request (both `<samlp:AuthnRequest>` and a `<samlp:AttributeQuery>` are profiled), which is presented to an identity provider along with an X.509 certificate. The subject proves possession of the private key corresponding to the public key bound to the X.509 certificate. The identity provider authenticates the subject and binds selected X.509 data from the presented certificate to a holder-of-key assertion. The subject consumes the response and subsequently uses the resulting holder-of-key assertion at its discretion.

The SAML V2.0 Self-AuthnRequest Profile and the SAML V2.0 Attribute Self-Query Profile are profiles of the `<samlp:AuthnRequest>` and `<samlp:AttributeQuery>` request elements, respectively. Each is a concrete profile of the abstract SAML V2.0 Holder-of-Key Self-Request Profile.

1.1 Notation

This specification uses normative text.

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]:

…they MUST only be used where it is actually required for interoperation or to limit behavior which has potential for causing harm (e.g., limiting retransmissions)…

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.

Listings of XML schemas appear like this.

Example code listings appear like this.

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:

<table>
<thead>
<tr>
<th>Prefix</th>
<th>XML Namespace</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>saml:</td>
<td>urn:oasis:names:tc:SAML:2.0:assertion</td>
<td>This is the SAML V2.0 assertion namespace defined in the SAML V2.0 core specification [SAML2Core].</td>
</tr>
<tr>
<td>samlp:</td>
<td>urn:oasis:names:tc:SAML:2.0:protocol</td>
<td>This is the SAML V2.0 protocol namespace defined in the SAML V2.0 core</td>
</tr>
</tbody>
</table>
This specification uses the following typographical conventions in text: `<SAMLElement>`, `<ns:ForeignElement>`, `Attribute`, `Datatype`, `OtherCode`.

### 1.2 Terminology

The term *X.509 certificate* as used in this specification refers to an X.509 end entity certificate [RFC5280] or a certificate based on an X.509 end entity certificate (such as an X.509 proxy certificate [RFC3820]).

The term *SSL/TLS* as used in this specification refers to SSL 3.0 [SSL3] or TLS 1.0 [RFC4346].

### 1.3 Normative References

<table>
<thead>
<tr>
<th>Prefix</th>
<th>XML Namespace</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>ds:</td>
<td><a href="http://www.w3.org/2000/09/xmldsig#">http://www.w3.org/2000/09/xmldsig#</a></td>
<td>This is the XML Signature namespace [XMLSig].</td>
</tr>
<tr>
<td>xs:</td>
<td><a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a></td>
<td>This is the XML Schema namespace [Schema1].</td>
</tr>
<tr>
<td>xsi:</td>
<td><a href="http://www.w3.org/2001/XMLSchema-instance">http://www.w3.org/2001/XMLSchema-instance</a></td>
<td>This is the XML Schema namespace for schema-related markup that appears in XML instances [Schema1].</td>
</tr>
</tbody>
</table>


1.4 Non-normative References


1.5 Conformance

1.5.1 SAML V2.0 Holder-of-Key Self-Request Profile

TBD

1.5.2 SAML V2.0 Self-AuthnRequest Profile

TBD

1.5.3 SAML V2.0 Attribute Self-Query Profile

TBD
2 SAML V2.0 Holder-of-Key Self-Request Profile

2.1 Required Information

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 associated with every <saml:Assertion> element issued under this profile.

Description: Given below.

Updates: Supplements the abstract request-response protocol specified in section 3.2 of the SAML V2.0 Core specification [SAML2Core].

2.2 Profile Description

This SAML V2.0 Holder-of-Key Self-Request Profile is an abstract profile of the abstract request-response protocol specified in section 3.2 of the SAML V2.0 Core specification [SAML2Core]. In particular, a request for a holder-of-key assertion must satisfy section 3.2.1 of Core. Likewise the response must satisfy section 3.2.2 of Core (and all of its subsections). Where this profile conflicts with Core, this profile takes precedence.

An important characteristic of this abstract profile is that the SAML requester is the subject, that is, the subject self-issues a SAML request. The subject presents this request and an X.509 certificate to a SAML identity provider. The subject proves possession of the private key corresponding to the public key of the presented certificate and authenticates to the identity provider by unspecified means.

The identity provider consumes the request and issues a response. The identity provider binds data from the X.509 certificate to an assertion in the response. The requester validates and consumes the response and outputs the holder-of-key assertion.

The following sequence of steps is profiled:

1. The subject self-issues a SAML request (section 2.4). The subject presents the request and an X.509 certificate to the identity provider.

2. The subject proves possession of the private key corresponding to the public key bound to the certificate and authenticates to the identity provider by unspecified means. See section 2.5.

3. The identity provider consumes the request and issues a SAML response (section 2.6). The identity provider binds data from the X.509 certificate to an assertion in the response.

4. The requester validates and consumes the response and outputs the holder-of-key assertion. See section 2.7.
SAML assertions with bound X.509 data are called holder-of-key assertions [SAML2HoK]. The contents of the <saml:SubjectConfirmation> element instruct a relying party how to confirm the subject. The subject must be confirmed before the assertion is accepted by the relying party.

2.3 X.509 Certificate Usage

As required by [SAML2HoK], the certificate presented to the identity provider MUST be an X.509 v3 certificate [RFC5280]. As discussed in section 2.5, the requester presents the certificate via an SSL/TLS exchange. For its part, the identity provider MUST be able to retrieve the presented certificate, decode it (if necessary), and bind the resulting X.509 data to the assertion. Otherwise the identity provider MUST return an error.

The X.509 certificate supplied to the identity provider via the SSL/TLS transaction is used primarily to supply a public key that is associated with the principal. The identity provider authenticates the principal by way of the presented X.509 certificate or any other method of its choosing. In any case, the identity provider MUST establish the identity of the principal before issuing a response. Thereafter the identity provider produces a SAML response containing a holder-of-key authentication assertion for the principal to use at its discretion.

Since the subject presents the X.509 certificate via an SSL/TLS exchange, the SSL/TLS key is a potential session key. However, the identity provider MUST NOT establish a session for any given request. In other words, the identity provider MUST freshly establish the identity of the principal for each request. Thus the requester MUST provide authentication credentials with each request.

2.4 Issuing the SAML Request

The requester MUST use a synchronous binding such as the SAML SOAP Binding. Any synchronous binding may be used. In particular, an HTTP binding MAY be used. The choice of binding is out of scope with respect to this profile.

The SAML request type of the SAML request message is unspecified by this profile. However, the SAML request type MUST be a type derived from the abstract samlp:RequestAbstractType complex type. Note
that the SAML request may be implicit, that is, the identity provider may issue an unsolicited SAML response without having received an explicit SAML request.

Since the SAML request type is derived from the abstract samlp:RequestAbstractType complex type, the SAML request is guaranteed to have a <saml:Issuer> element. To signal that the SAML requester is the subject, the value of the <saml:Issuer> element in the request MUST be the subject distinguished name (DN) of the presented certificate. (This is how the identity provider distinguishes this use case from other similar use cases such as the SAML V2.0 Holder-of-Key Web Browser SSO Profile [SAML2HoKSSO].) The Format attribute on the <saml:Issuer> element MUST be urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName. Following the XML Signature specification [XMLSig], it is RECOMMENDED that the DN conform to [RFC4514].

To signal that the SAML requester is the subject, the Consent attribute on the request MUST be present and its value MUST be urn:oasis:names:tc:SAML:2.0:consent:self. This is how the identity provider distinguishes this use case from other similar use cases such as the SAML V2.0 Holder-of-Key Web Browser SSO Profile [SAML2HoKSSO].

The requester SHOULD NOT sign the request.

### 2.5 Negotiating the SAML Exchange

The requester MUST prove possession of the private key corresponding to the public key bound to the presented X.509 certificate. Additionally, the requester MUST authenticate to the identity provider, but the method by which the requester does so is unspecified.

Note that the proof of possession step and the authentication step are completely independent of one another. Although X.509 authentication is not ruled out, it is by no means required. In fact, it is expected that client authentication will be by some other means, perhaps involving a username/password, since that is the credential most prevalent in today’s SAML deployments.

The SAML exchange MUST be mutually authenticated and integrity protected. Mutually authenticated SSL/TLS [RFC4346] MUST be used for this purpose. Note that if the identity provider can validate the presented certificate [RFC5280], the requester simultaneously satisfies both requirements of the SAML exchange.

If the presented certificate is an untrusted X.509 certificate, the requester MUST present a meaningless X.509 certificate [AIXCM] to the identity provider (via SSL/TLS). In that case, the SSL/TLS exchange proves possession of the private key only. Since a meaningless certificate is an inherently untrusted security token, the requester MUST authenticate to the identity provider by some other means.

If the identity provider can not (or will not) validate the presented certificate, and the presented certificate is not a meaningless X.509 certificate [AIXCM], then the identity provider MUST return an error. If the presented certificate is a meaningless X.509 certificate, then the identity provider MUST authenticate the requester by some other means, otherwise the identity provider MUST return an error.

The identity provider MUST check that the value of the <saml:Issuer> element in the request matches the subject distinguished name (DN) of the presented certificate.

### 2.6 Issuing the SAML Response

If the presenter can prove possession of the private key and successfully authenticate by whatever means, the identity provider issues a <samlp:Response> element containing a holder-of-key assertion. If the presenter is unable to prove possession of the private key, or the identity provider wishes to return an error for any other reason, the identity provider MUST NOT include any assertions in the <samlp:Response> message. Otherwise the <samlp:Response> element MUST conform to the following rules:
• The `<samlp:Response>` element MAY be signed by the identity provider.

• The `<samlp:Response>` element MAY contain a `<saml:Issuer>` element. If the `<saml:Issuer>` element is present in the response, it MUST contain the unique identifier of the issuing identity provider. The Format attribute on the `<saml:Issuer>` element MUST be omitted or have a value of urn:oasis:names:tc:SAML:2.0:nameid-format:entity.

• The `<samlp:Response>` element MUST contain exactly one `<saml:Assertion>` element and that `<saml:Assertion>` element MUST be signed. The assertion's `<saml:Issuer>` element 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.

• The assertion MUST contain a holder-of-key `<saml:SubjectConfirmation>` element and that element MUST conform to the SAML V2.0 Holder-of-Key Assertion Profile [SAML2HoK] as outlined in section 1.4.1 of that specification.

• The `<saml:SubjectConfirmationData>` element MAY include NotBefore and NotOnOrAfter attributes as specified in [SAML2HoK]. Beyond that, the `<saml:SubjectConfirmationData>` element MAY include other attributes as specified in [SAML2Core] except for the InResponseTo and Address attributes, which MUST be omitted.

• This specification does not preclude other `<saml:SubjectConfirmation>` elements from being included alongside the holder-of-key `<saml:SubjectConfirmation>` element (i.e., contained within the same `<saml:Subject>` element), but deployers should be aware of the implications of allowing weaker alternate confirmation methods (such as bearer), as the processing is implicitly satisfy-any.

• The assertion MUST contain a `<saml:Conditions>` element with NotBefore and NotOnOrAfter XML attributes. The value of the NotOnOrAfter attribute SHOULD NOT exceed that specified in the request (if any) but MAY be earlier than that specified in the request (subject to policy). If no NotOnOrAfter attribute is specified in the request, the identity provider defaults to some reasonable value (subject to policy).

• The assertion SHOULD NOT contain a `<saml:AudienceRestriction>` element unless the requester explicitly asks for one in the request.

• The assertion MUST contain one and only one `<saml:AuthnStatement>` element that describes the act of authentication at the identity provider. The assertion MAY contain other SAML statements. In particular, the assertion MAY contain one or more `<saml:AttributeStatement>` elements.

2.7 Processing the SAML Response
The SAML requester validates the SAML Response according to [SAML2Core]. In particular, the requester MUST verify the signature on the `<samlp:Response>` element (if there is one).

After validating the response, the requester strips the `<samlp:Response>` element, exposing the signed assertion. The resulting assertion is a reusable security token that the requester subsequently employs at its discretion.

2.8 Use of Metadata
The use of SAML V2.0 metadata [SAML2Meta] is RECOMMENDED, but since a principal is not expected to publish metadata about itself, only the use of identity provider metadata is profiled below.

TBD
2.9 Security and Privacy Considerations

In the case where the presented X.509 certificate is a meaningless X.509 certificate, the subject is required to authenticate to the identity provider by some other means (as opposed to X.509 authentication). Although the method of authentication is unspecified, it is expected that a typical scenario will involve a subject who possesses a username/password. In that case, an application-level authentication method (such as HTTP basic authentication [HTTPBasicAuth]) or message-level authentication (such as WS-Security Username Token Profile [WSSUsername]) may be used.

2.9.1 Comparison with Web Browser SSO

There are numerous differences between this profile and the SAML V2.0 Holder-of-Key Web Browser SSO Profile [SAML2HoKSSO]:

- The Binding
- The Issuer
- The Validity Period
- The Audience

Web Browser SSO typically uses an HTTP binding (such as HTTP POST) but in the non-browser case profiled here, the SAML SOAP Binding is often used. This of course opens the door to other methods of authentication, such as WS-Security Username Token Profile [WSSUsername] mentioned previously.

In Web Browser SSO, the SAML requester is a SAML service provider, so the value of the <saml:Issuer> element is the entityID of the service provider. In the non-browser case profiled here, the requester is the subject, and so the value of the <saml:Issuer> element is the subject DN of the presented X.509 certificate.

A conforming identity provider implementation builds on these two features of the non-browser use case. First of all, since the request is typically over SOAP, the identity provider accepts self-requests at a different endpoint (than Web Browser SSO). Secondly, the value of the <saml:Issuer> element in all self-requests is a DN, and that DN must match the subject DN of the presented X.509 certificate. This is how the IdP knows the requester is the subject.

In Web Browser SSO, the time validity of a bearer assertion is very short, on the order of minutes. In the non-browser use case, the time validity of a holder-of-key assertion may be longer lived, on the order of hours. The actual validity period is of course subject to policy at the identity provider and according to the desire of the requesting subject.

And finally, in Web Browser SSO, the <saml:Audience> element in the response MUST contain the entityID of the service provider. In the non-browser use case, however, there is no <saml:Audience> element unless the subject specifically requests a targeted assertion.

2.9.2 Policy issues

It is important to note that the identity provider likely possesses no metadata for the subject. Whereas in Web Browser SSO the identity provider uses the service provider's entityID as a key into metadata [SAML2Meta], but in the non-browser case metadata is nonexistent. Thus the identity provider issues the response directly to the subject. In essence, the effective policy at the identity provider is: “Give the subject what she wants.” That is, the identity provider issues signed, holder-of-key authentication assertions directly to the subject.

The identity provider may issue attributes to the subject at its discretion. A candidate attribute for auto-inclusion in the response is the eduPersonScopedAffiliation attribute [MACEAttrib], a benign...
attribute that reveals little about the user. The identity provider may issue other attributes as well, but a
prudent identity provider will only issue identity attributes as requested.
3 SAML V2.0 Self-AuthnRequest Profile

3.1 Required Information


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 associated with every <saml:Assertion> element issued under this profile.

Description: Given below.

Updates: Implements the abstract SAML V2.0 Holder-of-Key Self-Request Profile in section 2.

3.2 Profile Description

This SAML V2.0 Self-AuthnRequest Profile describes the issuing of a <samlp:AuthnRequest> element in the special case where the requester is the subject. That is, the subject self-issues a SAML authentication request for a holder-of-key authentication assertion.

Unless otherwise specified, all of the requirements of the SAML V2.0 Holder-of-Key Self-Request Profile (section 2) apply to this profile. In particular, note that that specification relies on the SAML V2.0 Holder-of-Key Assertion Profile [SAML2HoK].

3.3 Issuing the Authentication Request

This specification is a profile of the Authentication Request Protocol specified in section 3.4 of [SAML2Core]. It is assumed that the reader is familiar with that protocol specification.

The IsPassive and ForceAuthn attributes on the <samlp:AuthnRequest> element MUST be set to true. The AssertionConsumerServiceIndex attribute MUST be omitted.

The request MAY contain a <saml:Conditions> element. In particular, the requester MAY include NotBefore and NotOnOrAfter attributes that indicate the desired lifetime of the requested holder-of-key assertion. Moreover, the <saml:Conditions> element MAY contain one or more <saml:Audience> elements that indicate the intended relying parties.

If a <saml:Subject> element is included in the request, a <saml:NameID> element SHOULD NOT be included. The identity provider is therefore free to assert any subject name identifier it wishes. For example, the identity provider MAY assert an identifier that the relying party may subsequently use as the subject of an attribute query.

If a <saml:Subject> element is included in the request, a <saml:SubjectConfirmation> element MAY be included. Otherwise, the identity provider is free to bind any available X.509 data to the authentication assertion, subject to the requirements of [SAML2HoK].

3.4 Issuing the Authentication Response

The identity provider processes the <samlp:AuthnRequest> element and issues a <samlp:Response> element conforming to the SAML V2.0 Holder-of-Key Self-Request Profile in section 2. Since the IsPassive attribute on the <samlp:AuthnRequest> element is true, the identity
provider MUST NOT interact with the client at the application level. Likewise, since the ForceAuthn attribute is true, if the identity provider can not freshly authenticate the subject (without meeting the constraints of the IsPassive attribute), the identity provider MUST return an error.

If the request contains a <saml:Audience> element, the holder-of-key assertion MUST contain a <saml:Audience> element with the same value. Otherwise a <saml:Audience> element SHOULD NOT be included in the holder-of-key assertion as specified in section 2.6.

3.5 Example

To illustrate, suppose a subject self-issues the following <samlp:AuthnRequest> element:

```xml
<samlp:AuthnRequest
    xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
    IsPassive="true" ForceAuthn="true"
    ID="someId1217706521181"
    IssueInstant="2008-08-02T19:48:41.181Z" Version="2.0">
    <saml:Issuer
        Format="urn:oasis:names:tc:SAML:1.1:nameid-format:X509SubjectName">
        emailAddress=some-address@host.org,CN=Joana Trindade,OU=GSoC 2008,O=GSoC 2008,L=Some-City,ST=Some-State,C=BR
    </saml:Issuer>
    <samlp:NameIDPolicy AllowCreate="true"/>
</samlp:AuthnRequest>
```

As a result, the identity provider issues a SAML response containing a holder-of-key authentication assertion with the following <saml:SubjectConfirmation> element:

```xml
<saml:SubjectConfirmation
    Method="urn:oasis:names:tc:SAML:2.0:cm:holder-of-key">
    <saml:SubjectConfirmationData
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:type="saml:KeyInfoConfirmationDataType">
        <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
            <ds:X509Data>
                <ds:X509Certificate>
                    MIIDuDCCAqACCQCJZK8wF0xVXJANbgkqhkiG9w0BAQQFADBnTbLMAkGA1UEBhMCQ1IxZeARBgNV
                    BAQwVTJEMEMGCCGSHBwkMBQAwEgYDVQQIDAbJXN0b3Jrc2VydCBhZGRj
                    BgNVBAUwDjQGMR2dGM2A8GzIEgV1ZG5laXJla291bmNzdW0g
                    MTEyMzg2MS4wMDA4Mjg2MTc1
                    GzIyMDA4Mjg2MTc1
                    GzIyMDA4Mjg2MTc1
                    GzIyMDA4Mjg2MTc1
                    GzIyMDA4Mjg2MTc1
                    GzIyMDA4Mjg2MTc1
                    GzIyMDA4Mjg2MTc1
                    GzIyMDA4Mjg2MTc1
                    GzIyMDA4Mjg2MTc1
                    GzIyMDA4Mjg2MTc1
                    GzIyMDA4Mjg2MTc1
                    GzIyMDA4Mjg2MTc1
                    GzIyMDA4Mjg2MTc1
                    GzIyMDA4Mjg2MTc1
                    GzIyMDA4Mjg2MTc1
                    GzIyMDA4Mjg2MTc1
                    GzIyMDA4Mjg2MTc1
                    GzIyMDA4Mjg2MTc1
                    GzIyMDA4Mjg2MTc1
                    GzIyMDA4Mjg2MTc1
                    GzIyMDA4Mjg2MTc1
                    GzIyMDA4Mjg2MTc1
                    GzIyMDA4Mjg2MTc1
                    GzIyMDA4Mjg2MTc1
                    GzIyMDA4Mjg2MTc1
                    GzIyMDA4Mjg2MTc1
                    GzIyMDA4Mjg2MTc1
                    GzIyMDA4Mjg2MTc1
                    GzIyMDA4Mjg2MTc1
                    GzIyMDA4Mjg2MTc1
                    GzIyMDA4Mjg2MTc1
                    GzIyMDA4Mjg2MTc1
                    GzIyMDA4Mjg2MTc1
                    GzIyMDA4Mjg2MTc1
                    GzIyMDA4Mjg2MTc1
                    GzIyMDA4Mjg2MTc1
                    GzIyMDA4Mjg2MTc1
                    GzIyMDA4Mjg2MTc1
                    GzIyMDA4Mjg2MTc1
                    GzIyMDA4Mjg2MTc1
                    GzIyMDA4Mjg2MTc1
                    GzIyMDA4Mjg2MTc1
                    GzIyMDA4Mjg2MTc1
                    GzIyMDA4Mjg2MTc1
                    GzIyMDA4Mjg2MTc1
                    GzIyMDA4Mjg2MTc1
                    GzIyMDA4Mjg2MTc1
                    GzIyMDA4Mjg2MTc1
                    GzIyMDA4Mjg2MTc1
                    GzIyMDA4Mjg2MTc1
                    GzIyMDA4Mjg2MTc1
```
4 SAML V2.0 Attribute Self-Query Profile

4.1 Required Information


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 associated with every <saml:Assertion> element issued under this profile.

Description: Given below.

Updates: Implements the abstract SAML V2.0 Holder-of-Key Self-Request Profile in section 2.

4.2 Profile Description

This SAML V2.0 Attribute Self-Query Profile describes the issuing of a <samlp:AttributeQuery> element in the special case where the requester is the subject. That is, the subject self-issues a SAML attribute query for a holder-of-key attribute assertion.

Unless otherwise specified, all of the requirements of the SAML V2.0 Holder-of-Key Self-Request Profile (section 2) apply to this profile. In particular, note that that specification relies on the SAML V2.0 Holder-of-Key Assertion Profile [SAML2HoK].

4.3 Issuing the Attribute Query

This specification is a profile of the Assertion Query and Request Protocol specified in section 3.3 of [SAML2Core]. It is assumed that the reader is familiar with that protocol specification.

If a <saml:Subject> element is included in the query, a <saml:NameID> element SHOULD NOT be included. The identity provider is therefore free to assert any subject identifier it wishes.

If a <saml:Subject> element is included in the query, a <saml:SubjectConfirmation> element MAY be included. Otherwise, the identity provider is free to bind any available X.509 data to the attribute assertion, subject to the requirements of [SAML2HoK].

4.4 Issuing the Attribute Response

The identity provider processes the <samlp:AttributeQuery> element and issues a <samlp:Response> element conforming to the SAML V2.0 Holder-of-Key Self-Request Profile in section 2. The identity provider MUST NOT interact with the client at the application level. If the identity provider can not freshly authenticate the subject, the identity provider MUST return an error.

A <saml:Audience> element SHOULD NOT be included in the holder-of-key assertion as specified in section 2.6.

4.5 Example

Consider an attribute self-query involving the following <samlp:AttributeQuery> element:
Perhaps the most interesting aspect of this example is the included `<saml:SubjectConfirmation>` element. By including this element, the subject is asking for a holder-of-key attribute assertion with a matching `<saml:SubjectConfirmation>` element (in fact, a strongly matching `<saml:Subject>` element as specified in [SAML2Core]).
Appendix A. Acknowledgments

The editor 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:

• Joana M. F. da Trindade, Universidade Federal do Rio Grande do Sul, (Brazil)
• Valerio Venturi, National Institute for Nuclear Physics (INFN), Italy
## Appendix B. Revision History

<table>
<thead>
<tr>
<th>Document ID</th>
<th>Date</th>
<th>Committer</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>sstc-saml2-hok-assertion-req-draft-02</td>
<td>2 Jun 2009</td>
<td>T. Scavo</td>
<td>Define new Consent attribute value</td>
</tr>
</tbody>
</table>