SAML V2.0 Attribute Predicate Profile Version 1.0

Committee Specification 01

28 November 2011

Specification URIs

This version:

http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-attr-predicate/v1.0/cs01/sstc-saml-attr-predicate-v1.0-cs01.doc (Authoritative)
http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-attr-predicate/v1.0/cs01/sstc-saml-attr-predicate-v1.0-cs01.html
http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-attr-predicate/v1.0/cs01/sstc-saml-attr-predicate-v1.0-cs01.pdf

Previous version:

N/A

Latest version:

http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-attr-predicate/v1.0/sstc-saml-attr-predicate-v1.0.doc (Authoritative)
http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-attr-predicate/v1.0/sstc-saml-attr-predicate-v1.0.html
http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-attr-predicate/v1.0/sstc-saml-attr-predicate-v1.0.pdf

Technical Committee:

OASIS Security Services (SAML) TC

Chairs:

Thomas Hardjono (hardjono@mit.edu), M.I.T.
Nathan Klingenstein (ndk@internet2.edu), Internet2

Editor:

Franz-Stefan Preiss (FRP@zurich.ibm.com), IBM

Additional artifacts:

This prose specification is one component of a Work Product which also includes:

- XML schema and examples: http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-attr-predicate/v1.0/cs01/xml/

Related work:

This specification is related to:

- *Security Assertion Markup Language (SAML) v2.0*. OASIS Standard. http://docs.oasis-open.org/security/saml/v2.0/saml-2.0-os.zip

Abstract:

This profile provides a mechanism to allow a SAML authority to certify that a given Boolean predicate holds over one or more of a subject's attribute values, without revealing the exact values of these attributes. The profile further defines a query format for attribute predicates.

Status:

This document was last revised or approved by the OASIS Security Services (SAML) TC on the above date. The level of approval is also listed above. Check the “Latest version” location noted above for possible later revisions of this document.

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

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

Citation format:

When referencing this specification the following citation format should be used:

[Attr-Predicate-V1.0]

SAML V2.0 Attribute Predicate Profile Version 1.0. 28 November 2011. OASIS Committee Specification 01. http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-attr-predicate/v1.0/cs01/sstc-saml-attr-predicate-v1.0-cs01.html.

Notices

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

SAML attribute assertions enable a SAML authority, sometimes also referred to as an issuer, to certify to a relying party that a subject is associated with a specified set of attribute values. There are many circumstances, however, where the relying party is not interested in the exact attribute values, but merely needs to be assured that a certain condition is satisfied. For example, a SAML authority may keep its subjects' dates of birth on record, but to control access to a teenage chat room, the relying party merely needs to ensure that the requesting subject belongs to the correct age group. Not only may users be reluctant to disclose such identifying information to the chat room host, but also the host itself may prefer not to know the exact date to avoid liability claims in case a data breach occurs.

For small numbers of predicates this problem can obviously be overcome by defining a dedicated Boolean attribute for each possible predicate, which the issuer authenticates by means of a SAML attribute statement. While this may work for common age restrictions such as being younger or older than eighteen, it is impractical when the space of relevant predicates is large. For example, it is rather unlikely that a dedicated attribute will be globally agreed upon if the minimum age to sign up for a theoretical driving test is seventeen years and nine months.

As another example where attribute predicates can be useful, consider an online opinion poll hosted on a city's website and a SAML authority that certifies subjects' ZIP codes. The city may want to ensure that only legal residents of the metropolitan area can participate in the poll, but for privacy reasons may not want to keep track of the voting statistics of separate neighborhoods. Other websites may want to ensure that the subject has an email address within a particular domain (e.g., youremployer.com), but to guarantee the users' privacy, they have no interest in the exact address.

Predicates may also involve relation among multiple attributes known to the SAML authority. For example, the signup form for an obesity summer camp may restrict access to participants whose body-mass index, computed as the weight divided by the square of the height, is above a certain threshold, but does not want to force participants to disclose their exact weight, height, or body-mass index.

[All text is normative unless otherwise labeled]

1.1 Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in **[RFC2119]**.

1.2 Normative References

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

[SAML2Bind] *Bindings for the OASIS Security Assertion Markup Language (SAML) V2.0*. 15 March 2005. OASIS Standard. http://docs.oasis-open.org/security/saml/v2.0/saml-bindings-2.0-os.pdf.

[SAML2Core] *Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0*. 15 March 2005. OASIS Standard. http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf.

[SAML2Errata] *SAML V2.0 Errata*. 20 October 2009. Errata Committee Draft 04. http://docs.oasis-open.org/security/saml/v2.0/sstc-saml-approved-errata-2.0.pdf.

[SAML2Meta] *Metadata for the OASIS Security Assertion Markup Language (SAML) V2.0*. 15 March 2005. OASIS Standard. http://docs.oasis-open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf.

[SAML2Prof] *Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0*. 15 March 2005. OASIS Standard. http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf.

[XACML3] eXtensible Access Control Markup Language (XACML) Version 3.0. 10 August 2010. OASIS Committee Specification 01. http://docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-cs-01-en.pdf.

[XMLSig] D. Eastlake et al. *XML-Signature Syntax and Processing*. World Wide Web Consortium, February 2002. See http://www.w3.org/TR/xmldsig-core/.

2 Protocol

This section defines messages and processing rules to query and respond attribute predicates.

In particular, the following types and elements are defined:

- <AttributePredicate>: An element describing a Boolean predicate over one or more of a subject's attribute values.
- <AttributePredicateQuery>: An element used to query a SAML authority for the truth value of a particular subject's attribute predicate.
- <AttributePredicateStatementType>: A SAML statement type describing a statement asserting that a subject's attribute predicate has a particular truth value.

2.1 Element <AttributePredicate>

The <AttributePredicate> element describes a Boolean predicate over one or more of a subject's attribute values. The predicate is expressed by means of an <xacml:Apply> element [XACML3] that denotes application of a specified function (identified by a URI in its FunctionId attribute) to one or more arguments of the <xacml:Expression> element substitution group. The return data-type of the <xacml:Apply> element MUST be "http://www.w3.org/2001/XMLSchema#boolean". Arbitrarily complex predicates can be formulated by nesting multiple <xacml:Apply> elements.

```
<xs:element name="AttributePredicate" type="AttributePredicateType"/>
<xs:complexType name="AttributePredicateType">
  <xs:sequence>
    <xs:element ref="xacml:Apply"/>
  </xs:sequence>
  <xs:attribute name="FriendlyDescription" type="xs:string" use="optional"/>
</xs:complexType>
```

The following four members of the <xacml:Expression> element substitution group may occur in the attribute predicate:

<xacml:AttributeValue>

<xacml:AttributeDesignator>

Any <xacml:AttributeDesignator> element used MUST be of category urn:oasis:names:tc:xacml:1.0:subject-category:access-subject.

The Issuer attribute in the <xacml:AttributeDesignator> element, if present, MUST have the same value as specified in the <saml:Issuer> element of the enclosing query or assertion.

<xacml:Apply>

The attribute predicate expressed as an XACML expression. All function URIs marked as mandatory in XACML version 3.0 [XACML3] MUST be supported as value for the FunctionId XML attribute by any implementation of this profile. Functions marked as optional MAY be supported, as well as any additional self-defined functions, as long as the semantics are understood by all of the involved parties. The return data type of the outermost <xacml:Apply> element MUST be http://www.w3.org/2001/XMLSchema#boolean.

<xacml:Function>

The <xacml:AttributeSelector> and <xacml:VariableReference> elements MUST not occur in an attribute predicate.

2.2 Element <AttributePredicateQuery>

The `<AttributePredicateQuery>` element is used to make the query “Does the specified Boolean predicate hold over this subject’s attribute value(s)?”. The query additionally specifies whether the response should contain an assertion that explicitly repeats the queried predicate in an attribute predicate statement. A successful response states that the queried predicate is true.

This element is of type **AttributePredicateQueryType**, which extends **SubjectQueryAbstractType** with the addition of the following element and attribute:

`IncludePredicateInResponse` [Optional]

> A Boolean value. If “true”, the queried attribute predicate MUST be included in an attribute predicate statement in the response.

`<AttributePredicate>` [Required]

> The queried Boolean predicate over one or more of the subject’s attribute values.

A SAML authority responds to an attribute predicate query according to the rules specified in Section 2.4.

The following schema fragment defines the `<AttributePredicateQuery>` element and its **AttributePredicateQueryType** complex type:

```
<xs:element name="AttributePredicateQuery"
type="AttributePredicateQueryType"/>
<xs:complexType name="AttributePredicateQueryType">
  <xs:complexContent>
    <xs:extension base="samlp:SubjectQueryAbstractType">
      <xs:sequence>
        <xs:element ref="AttributePredicate"/>
      </xs:sequence>
      <xs:attribute name="IncludePredicateInResponse" type="xs:boolean"
use="optional"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

2.3 Type <AttributePredicateStatementType>

A `<saml:Statement>` element of type **AttributePredicateStatementType** describes a statement by the SAML authority asserting that a Boolean predicate over one or more of the assertion subject’s attribute values is true. In this document, statements of this type are referred to as *attribute predicate statements*. Attribute predicate statements MUST contain a `<Subject>` element.

The type **AttributePredicateStatementType** extends **StatementAbstractType** with the addition of the following element:

`<AttributePredicate>` [Required]

> The `<AttributePredicate>` element specifies a Boolean predicate over one or more of the assertion subject’s attribute values.

The following schema fragment defines the **AttributePredicateStatementType** complex type:

```
<xs:complexType name=”AttributePredicateStatementType”>
  <xs:complexContent>
    <xs:extension base="saml:StatementAbstractType">
      <xs:sequence>
        <xs:element ref="AttributePredicate"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

2.4 Processing Rules

A SAML authority that receives an incoming `<AttributePredicateQuery>` evaluates the truth of the queried `<AttributePredicate>` in a way that is semantically equivalent to evaluating an XACML request [XACML3] containing all known attributes of the concerned subject against an XACML policy – with rule-combining algorithm "Permit-overrides" – that contains a single applicable rule – with effect "Permit" – whose condition solely contains the queried predicate expressed as `<xacml:Apply>` element. An XACML Policy Decision Point (PDP) evaluating such request will return one of the following three evaluation decisions:

"Permit"

> In case the XACML PDP returns decision "Permit", the SAML authority MAY return a SAML response with status code `urn:oasis:names:tc:SAML:2.0:status:Success`. If the `IncludePredicateInResponse` XML attribute of the corresponding query is present and "true", an assertion with an attribute predicate statement containing an attribute predicate that is equal to the queried attribute predicate MUST be included in the response. The subject of this assertion MUST strongly match the subject of the corresponding query according to the rules specified in Section 3.3.4 of [SAML2Core]. Equality for attribute predicates is defined either as string equality (i.e., including whitespaces), or, in case the assertion is signed, as equality under the XML canonicalization method used to compute the XML Signature (see [XMLSig] for more information on canonicalization methods). Note that there may be situations where the SAML authority MAY not return an assertion containing an attribute predicate statement, even though the XACML PDP returns decision "Permit". For example, when additional policies apply or when the subject does not give permission to return the queried assertion.

"NotApplicable"

> In case the XACML PDP returns decision "NotApplicable", which arises when the SAML authority knows all required attributes but they do not satisfy the predicate, the SAML authority MAY return a response with top-level status code `urn:oasis:names:tc:SAML:2.0:status:Responder` and second-level status code `urn:oasis:names:tc:SAML:2.0:status:PredicateFalse`.

"Indeterminate"

> In case the XACML PDP returns decision "Indeterminate", the SAML authority MUST return a response with top-level status code `urn:oasis:names:tc:SAML:2.0:status:Responder` and second-level status code `urn:oasis:names:tc:SAML:2.0:status:UnknownAttrProfile`. In particular, this situation arises when an `<xacml:AttributeDesignator>` element that must be present is not known by the SAML authority with respect to the concerned subject.

If the XACML PDP returns any decision other than "Permit", then the SAML authority MUST NOT return a response with status code `urn:oasis:names:tc:SAML:2.0:status:Success`.

In the following situations a SAML authority MUST return a response with top-level status code `urn:oasis:names:tc:SAML:2.0:status:Requester` and second-level status code `urn:oasis:names:tc:SAML:2.0:status:InvalidPredicate`:

- an attribute predicate contains an `<xacml:AttributeDesignator>` element with a category different from `urn:oasis:names:tc:xacml:1.0:subject-category:access-subject`,
- an attribute predicate contains an `<xacml:AttributeDesignator>` element with the Issuer value different from the one specified in the `<saml:Issuer>` element of the enclosing query, or
- an attribute predicate contains an `<xacml:AttributeSelector>` or `<xacml:VariableReference>` element.

Table 1 summarizes the prescribed status codes and their interpretations.

Table 1 Status codes and interpretations

Evaluation Decision	Top-level `<StatusCode>`	Second-level `<StatusCode>`	Interpretation
“Permit”	“Success”	Don’t care	Attribute predicate is true
“Permit”	“Requester”	Appropriate status code	Interpretation according to second-level status code For example, additional policies apply, subject does not give permission, etc.)
Different from “Permit”	MUST NOT be “Success”	Appropriate status code	Interpretation according to top-level and second-level status code
“NotApplicable”	“Responder”	“PredicateFalse”	Attribute predicate is false
“Indeterminate”	“Requester”	“UnknownAttrProfile”	An attribute contained in the predicate is not known by the SAML authority
Don’t care	“Requester”	“InvalidPredicate”	Malformed attribute predicate

3 Attribute Predicate Profile

Section 2 defines a protocol for querying a SAML authority for the truth value of a Boolean predicate over one ore more of a subject's attribute values. This profile describes the use of this protocol with a synchronous binding, such as the SOAP binding defined in [SAML2Bind].

3.1 Profile Overview

The message exchange and basic processing rules that govern this profile are largely defined by Section 2 that defines the messages to be exchanged, in combination with the binding used to exchange the messages. Section 3.2 of [SAML2Bind] defines the binding of the message exchange to SOAP V1.1. Unless specifically noted here, all requirements defined in those specifications apply.

Figure 1 illustrates the basic template for the query profile.

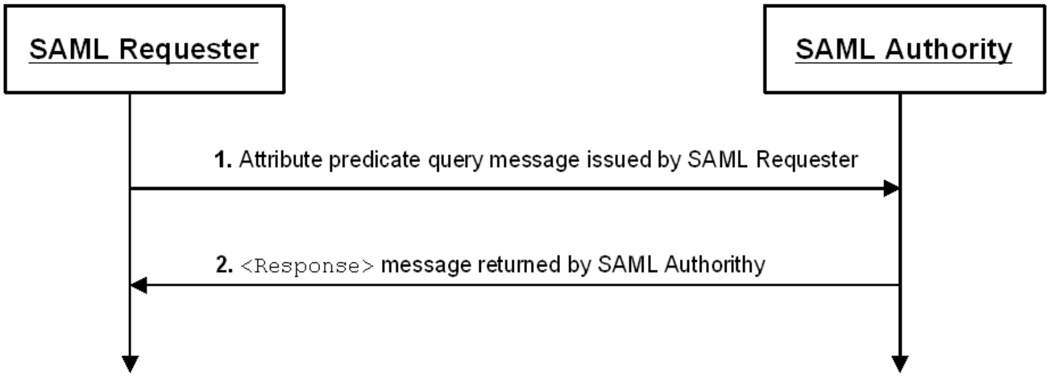
SAML Requester
SAML Authority
1. Attribute predicate query message issued by SAML Requester
2. <Response> message returned by SAML Authorithy

Figure 1

The following steps are described by the profile.

1. **Query issued by SAML Requester**

   In step 1, a SAML requester initiates the profile by sending an `<AttributePredicateQuery>` message to a SAML authority.

2. **<Response> issued by SAML Authority**

   In step 2, the responding SAML authority (after processing the query) issues a `<Response>` message to the SAML requester.

3.2 Profile Description

In the descriptions below, the following are referred to:

Attribute-Predicate Query Service

This is a query protocol endpoint at a SAML authority to which query messages are delivered.

3.2.1 Query issued by SAML Requester

To initiate the profile, a SAML requester issues an `<AttributePredicateQuery>` message to a SAML authority's attribute-predicate query service endpoint. Metadata (as in [SAML2Meta]) MAY be used to determine the location of this endpoint and the bindings supported by the SAML authority.

The SAML requester MUST use a synchronous binding, such as the SOAP binding [SAML2Bind], to send the request directly to the identity provider. The requester SHOULD authenticate itself to the SAML authority either by signing the message or using any other binding-supported mechanism.

3.2.2 <Response> issued by SAML Authority

The SAML authority MUST process the query message as defined in Section 2.4. After processing the message or upon encountering an error, the SAML authority MUST return a `<Response>` message containing an appropriate status code to the SAML requester to complete the SAML protocol exchange.

The responder SHOULD authenticate itself to the requester, either by signing the `<Response>` or using any other binding-supported mechanism.

3.3 Use of Query Protocol

3.3.1 Query Usage

The `<Issuer>` element MUST be present.

The requester SHOULD authenticate itself to the responder and ensure message integrity, either by signing the message or using a binding-specific mechanism.

3.3.2 <Response> Usage

The `<Issuer>` element MUST be present and MUST contain the unique identifier of the responding SAML authority; the `Format` attribute MUST be omitted or have a value of `urn:oasis:names:tc:SAML:2.0:nameid-format:entity`. Note that this need not necessarily match the `<Issuer>` element in the returned assertion(s).

The responder SHOULD authenticate itself to the requester and ensure message integrity, either by signing the message or using a binding-specific mechanism.

3.4 Use of Metadata

For the use of metadata, the rules specified for the Assertion Query/Request Profile in Section 6.5 of [SAML2Prof] apply.

4 Examples (non-normative)

In following example the SAML requester `requester.example.com` queries the SAML authority `idp.example.com` on whether the date of birth of subject `pseudonym12345` is before 1 January 1993.

```
<AttributePredicateQuery
  Version="2.0"
  ID="query23a0821cf186ea0a22e3818750a809b6cb3b4cda"
  IssueInstant="2011-02-28T23:59:58"
  IncludePredicateInResponse="true">

  <samla:Issuer>requester.example.com</samla:Issuer>
  <samla:Subject>
    <samla:NameID Format="urn:oasis:names:tc:SAML:2.0:nameid-format:transient">
      pseudonym12345
    </samla:NameID>
  </samla:Subject>

  <AttributePredicate FriendlyDescription="The requestor is over 18 years of age.">
    <xacml:Apply
      FunctionId="urn:oasis:names:tc:xacml:1.0:function:date-less-than-or-equal">
      <xacml:Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:date-one-and-only">
        <xacml:AttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#date"
          MustBePresent="true"
          Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
          AttributeId="urn:example:identity:birthdate"/>
      </xacml:Apply>
      <xacml:AttributeValue DataType="http://www.w3.org/2001/XMLSchema#date">
        1993-01-01
      </xacml:AttributeValue>
    </xacml:Apply>
  </AttributePredicate>

</AttributePredicateQuery>
```

If the subject's date of birth is indeed before 1 January 1993, the response of the SAML authority could be the following. (Note that the signature is not valid and cannot be successfully verified.)

```
<samlp:Response
  Version="2.0"
  ID="responsebd6996d271d23129dc2068c497ea3415f00b8b73"
  InResponseTo="query23a0821cf186ea0a22e3818750a809b6cb3b4cda"
  IssueInstant="2011-02-28T23:59:59">

  <samla:Issuer>idp.example.com</samla:Issuer>
  <samlp:Status>
    <samlp:StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Success"/>
  </samlp:Status>

  <samla:Assertion
    IssueInstant="2011-02-28T23:59:59"
    ID="assertion116fcc68e6a3feb788e5c89520b5f4eadf5ebbcf"
    Version="2.0">
    <samla:Issuer>idp.example.com</samla:Issuer>
    <ds:Signature>
      <ds:SignedInfo>
        <ds:CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#"/>
        <ds:SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#rsa-sha1"/>
        <ds:Reference URI="#assertion116fcc68e6a3feb788e5c89520b5f4eadf5ebbcf">
          <ds:Transforms>
            <ds:Transform
              Algorithm="http://www.w3.org/2000/09/xmldsig#envelopedsignature"/>
            <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
              <InclusiveNamespaces PrefixList="#default xacml samla samlp ds xsi"
              xmlns="http://www.w3.org/2001/10/xml-exc-c14n#"/>
            </ds:Transform>
          </ds:Transforms>
          <ds:DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1"/>
          <ds:DigestValue>192d3b53f6c4fda041ff36899a91422e9e9a8a2b</ds:DigestValue>
        </ds:Reference>
      </ds:SignedInfo>
      <ds:SignatureValue>
        hq4zk+ZknjggCQgZm7ea8fI79gJEsRy3E8LHDpYXWQIgZpkJN9CMLG8ENR4Nrw+n
        7iyzixBvKXX8P53BTCT4VghPBWhFYSt9tHWu/AtJfOTh6qaAsNdeCyG86jmtp3TD
        MwuL/cBUj2OtBZOQMFn7jQ9YB7klIz3RqVL+wNmeWI4=
      </ds:SignatureValue>
      <ds:KeyInfo>
        <ds:X509Data>
          <ds:X509Certificate>
            MIICyjCCAjOgAwIBAgICAnUwDQYJKoZIhvcNAQEEBQAwgakxCzAJBgNVBAYTAlVT
          </ds:X509Certificate>
        </ds:X509Data>
      </ds:KeyInfo>
    </ds:Signature>

    <samla:Subject>
      <samla:NameID Format="urn:oasis:names:tc:SAML:2.0:nameid-format:transient">
        pseudonym12345
      </samla:NameID>
    </samla:Subject>

    <samla:Statement xsi:type="ap:AttributePredicateStatementType">
      <AttributePredicate FriendlyDescription="The requestor is over 18 years of age.">
        <xacml:Apply
          FunctionId="urn:oasis:names:tc:xacml:1.0:function:date-less-than-or-equal">
          <xacml:Apply
            FunctionId="urn:oasis:names:tc:xacml:1.0:function:date-one-and-only">
            <xacml:AttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#date"
              MustBePresent="true"
              Category="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
              AttributeId="urn:example:identity:birthdate"/>
          </xacml:Apply>
          <xacml:AttributeValue DataType="http://www.w3.org/2001/XMLSchema#date">
            1993-01-01
          </xacml:AttributeValue>
        </xacml:Apply>
      </AttributePredicate>
    </samla:Statement>

  </samla:Assertion>

</samlp:Response>
```

5 Conformance

An attribute predicate requester can claim conformance with the Attribute Predicate Profile if it

- generates `<AttributePredicateQuery>` messages conforming to Section 2.2,
- transmits such queries and receives responses as described in Section 3, and
- interprets the response according to Section 2.4.

A SAML authority can claim conformance with the Attribute Predicate Profile if it

- can receive `<AttributePredicateQuery>` messages conforming to Section 3 and
- responds to such queries according to Section 2.4.

Appendix A. Acknowledgements

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

Participants:

- Abbie Barbier, Bank of America
- Scott Cantor, Internet2
- Thomas Hardjono, M.I.T.
- Frederick Hirsch, Nokia Corporation
- Nathan Klingenstein, Internet2
- Chad La Joie, Internet2
- Hal Lockhart, Oracle
- Gregory Neven, IBM
- Thinh Nguyenphu, Nokia Siemens Networks GmbH & Co. KG
- Franz-Stefan Preiss, IBM
- Anil Saldhana, Red Hat
- Emily Xu, Oracle

Appendix B. Revision History

Revision	Date	Editor	Changes Made
WD 01	17 May 2011	Gregory Neven and Franz-Stefan Preiss	Initial version
WD 02	28 June 2011	Franz-Stefan Preiss	Addressing committee comments by • moving content to Section ‘Protocol’ • adding Section ‘Attribute Predicate Profile’
WD 03	30 June 2011	Franz-Stefan Preiss	Revising conformance and appendix