Security Assertions Markup Language

Core Assertion Architecture

Phillip Hallam-Baker	*VeriSign*
Tim Moses	*Entrust*
Bob Morgan	*University of Washington*
Carlisle Adams	*Entrust*
Charles Knouse	*Oblix*
David Orchard	*Jamcracker*
Eve Maler	*Sun*
Irving Reid	*Baltimore*
Jeff Hodges	*Oblix*
Marlena Erdos	*Tivoli*
Nigel Edwards	*Hewlett Packard*
Prateek Mishra	*Netegrity*

Draft Version 0.10: July 13th 2001

Security Assertions Markup Language

Version 0.10

Table Of Contents

Executive Summary

This document contains two sections. Section 1 contains the text proposed by the Core Assertions & Protocol group for the Core Assertions section of the SAML. Section 2 contains references to the material cited in the text.

1 XML Assertion and Request Syntax

1.1 Namespaces

In this document, certain namespace prefixes represent certain namespaces.

All SAML protocol elements are defined using XML schema **[XML-Schema1][XML-Schema2]**. For clarity unqualified elements in schema definitions are in the XML schema namespace:

```
xmlns="http://www.w3.org/2001/XMLSchema"
```

References to Security Assertion Markup Language schema defined herein use the prefix "s0" and are in the namespace:

```
xmlns:s0="http://www.oasis.org/tbs/1066-12-25/"[PHB1]
```

This namespace is also used for unqualified elements in message protocol examples.

The SAML schema specification uses some elements already defined in the XML Signature namespace. The "XML Signature namespace" is represented by the prefix `ds` and is declared as:

```
xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
```

The "XML Signature schema" is defined in **[XML-SIG-XSD]** and the `<ds:KeyInfo>` elem (and all of its contents) are defined in **[XML-SIG]**§4.4.

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.oasis.org/tbs/1066-12-25/"
       xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
       xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
       xmlns:saml="http://www.oasis.org/tbs/1066-12-25/"
       xmlns="http://www.w3.org/2000/10/XMLSchema"
       elementFormDefault="unqualified">
   <import namespace=" http://www.w3.org/2000/09/xmldsig#"
          schemaLocation="xmldsig-core-schema.xsd"/>
   <annotation>
      <documentation>draft-schema-consensus-10.xsd</documentation>
   </annotation>
```

1.1.1 Basic Types

```
   <element name="AssertionID" type="saml:IDType"/>
   <simpleType name="IDType">
      <restriction base="string"/>
   </simpleType>
   <simpleType name="DecisionType">
```

```
      <restriction base="string">
         <enumeration value="Permit"/>
         <enumeration value="Deny"/>
         <enumeration value="Indeterminate"/>
      </restriction>
   </simpleType>
```

1.2 SAML Assertion

SAML specifies several different types of assertion for different purposes, these are:

Authentication Assertion
An authentication assertion asserts that the issuer has authenticated the specified subject.

Attribute Assertion
An attribute assertion asserts that the specified subject has the specified attribute(s). Attributes may be specified by means of a URI or through an extension schema that defines structured attributes.

Decision Assertion
A decision assertion reports the result of the specified authorization request.

Authorization Assertion
An authorization assertion asserts that a subject has been granted specific permissions to access one or more resources.

The different types of SAML assertion are encoded in a common XML package, which at a minimum consists of:

Basic Information.
Each assertion MUST specify a unique identifier that serves as a name for the assertion. In addition an assertion MAY specify the date and time of issue and the time interval for which the assertion is valid.

In addition an assertion MAY contain the following additional elements. An SAML client is not required to support processing of any element contained in an additional element **with the sole exception that an SAML client MUST reject any assertion containing a Conditions element that is not supported.**

Conditions.
The assertion status MAY be subject to conditions. The status of the assertion might be dependent on additional information from a validation service. The assertion may be dependent on other assertions being valid. The assertion may only be valid if the relying party is a member of a particular audience.

Advice.
Assertions MAY contain additional information as advice. The advice element MAY be used to specify the assertions that were used to make a policy decision.

The SAML assertion package is designed to facilitate reuse in other specifications. For this reason XML elements specific to the management of authentication and authorization data are expressed as claims. Possible additional applications of the assertion package format include management of embedded trust roots **[XTASS]** and authorization policy information **[XACML]**.

Figure 1 shows the class diagram of the SAMLAssertionPackage data structure. The notation used in this diagram is described in Figure 2.

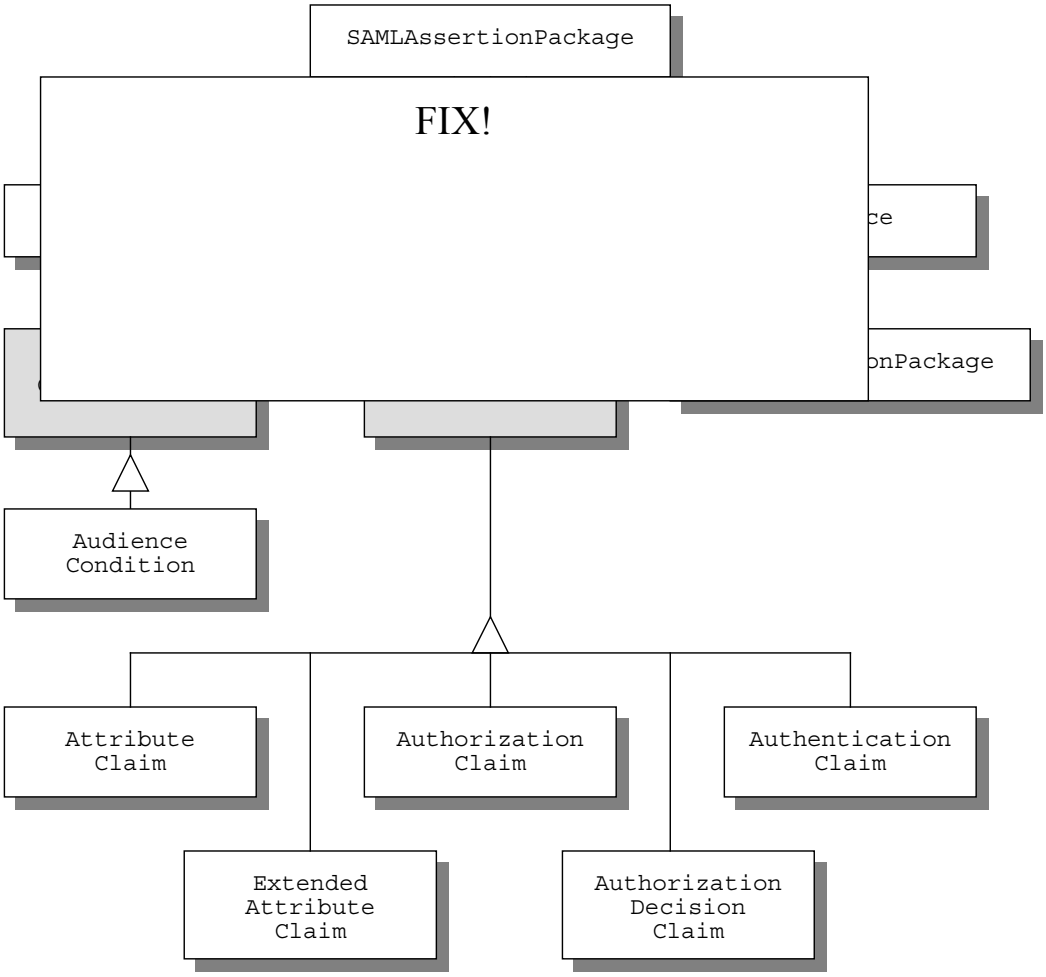

Figure 1: <SAMLAssertionPackage> Class Diagram

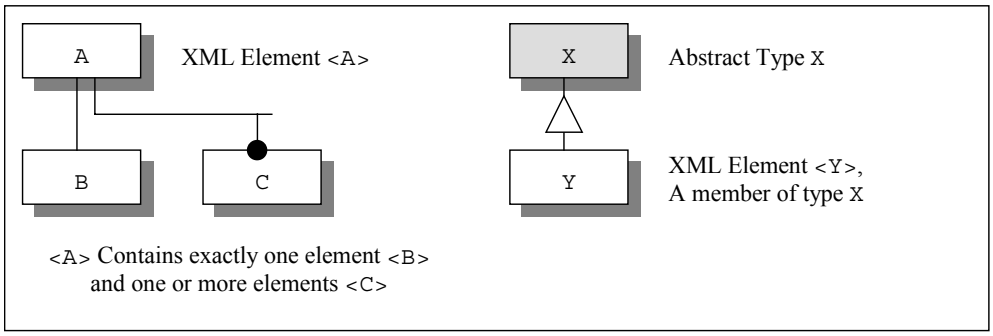

Figure 2: Key to Figure 1 and Figure 3

1.2.1 Abstract type <AssertionType>

The following schema specifies the <AssertionType> abstract type:

```
<complexType name="AssertionType" abstract="true">
   <sequence>
      <element name="Conditions" type="saml:ConditionsType" minOccurs="0"/>
```

```
            <element name="Advice" type="saml:AdviceType" minOccurs="0"/>
        </sequence>
        <attribute name="Version" type="string" use="required"/>
        <attribute name="AssertionID" type="saml:IDType" use="required"/>
        <attribute name="Issuer" type="string" use="required"/>
        <attribute name="IssueInstant" type="timeInstant" use="required"/>
    </complexType>
```

Six basic information attributes are defined; a protocol version identifier, a unique assertion identifier, an issuer identifier, the time instant of issue, the bounds of the validity interval.

1.2.1.1 Attribute `Version`

Each assertion MUST specify the SAML version identifier. The identifier for this version of SAML is the string "`1.0`".

1.2.1.2 Attribute `AssertionID`

Each assertion MUST specify exactly one unique assertion identifier. All identifiers are encoded as a Uniform Resource Identifier (URI) and are specified in full (use of relative identifiers is not permitted).

The URI is used as a *name* for the assertion and not as a *locator*. For the purposes of the SAML protocol it is only necessary to ensure that no two assertions share the same identifier. Provision of a service to resolve an identifier into an assertion is not a requirement but applications MAY specify a URL as the assertion identifier that MAY resolve to the assertion.

1.2.1.3 Attribute `Issuer`

The `Issuer` attribute specifies the issuer of the assertion by means of a URI.

1.2.1.4 Attribute `IssueInstant`

The `IssueInstant` attribute specifies the time instant of issue in Universal Coordinated Time (UTC).

1.3 Element `<AssertionSpecifier>`

The following schema specifies the `<AssertionSpecifier>` element:

```
    <element name="AssertionSpecifier" type="saml:AssertionSpecifierType"/>
    <complexType name="AssertionSpecifierType">
        <choice>
            <element ref="saml:AssertionID" />
            <element ref="saml:Assertion"/>
        </choice>
    </complexType>
```

1.4 Subject Assertion

1.4.1 Abstract Type <SubjectAssertionType

[TBS]

The following schema defines the SubjectAssertionType abstract type:

```
<complexType name="SubjectAssertionType" abstract="true">
    <complexContent>
        <extension base="saml:AssertionType">
            <sequence>
                <element ref="saml:Subject"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>
```

1.4.2 Element <Subject>

The following schema defines the <Subject> element:

```
<element name="Subject" type="saml:SubjectType"/>
<complexType name="SubjectType">
    <choice maxOccurs="unbounded">
        <element ref="saml:NameIdentifier"
                 minOccurs="0" maxOccurs="unbounded"/>
        <element ref="saml:HolderOfKey"
                 minOccurs="0" maxOccurs="unbounded"/>
        <element ref="saml:AssertionSpecifier"
                 minOccurs="0" maxOccurs="unbounded"/>
    </choice>
</complexType>
```

1.4.2.1 Element <HolderOfKey>

The following schema defines the <HolderOfKey> element:

```
<element name="HolderOfKey" type="saml:HolderOfKeyType"/>
<complexType name="HolderOfKeyType">
    <sequence>
        <element name="Protocol" type="uriReference"
                 maxOccurs="unbounded"/>
        <element name="Authdata" type="string" minOccurs="0"/>
        <element ref="ds:KeyInfo" minOccurs="0"/>
    </sequence>
</complexType>
```

1.4.2.2 Element <NameIdentifier>

The following schema defines the <NameIdentifier> element:

```
<element name="NameIdentifier" type="saml:NameIdentifierType"/>
<complexType name="NameIdentifierType">
    <sequence>
        <element name="SecurityDomain" type="string"/>
        <element name="Name" type="string"/>
    </sequence>
</complexType>
```

1.4.3 Element <AuthenticationAssertion>

The following schema defines the <AuthenticationAssertion> element:

```
<element name="AuthenticationAssertion"
       type="saml:AuthenticationAssertionType"/>
<complexType name="AuthenticationAssertionType">
   <complexContent>
      <extension base="saml:SubjectAssertionType">
         <sequence>
            <element ref="saml:AuthenticationCode"/>
            <element name="AuthenticationInstant" type="timeInstant"/>
            <element name="AuthLocale"
                  type="saml:AuthLocaleType" minOccurs="0"/>
         </sequence>
      </extension>
   </complexContent>
</complexType>
```

1.4.3.1 Element <AuthenticationCode>

The following schema defines the <AuthenticationCode> element:

```
<element name="AuthenticationCode" type="saml:AuthenticationCodeType"/>
<simpleType name="AuthenticationCodeType">
   <restriction base="string"/>
</simpleType>
```

1.4.3.2 Element <AuthLocale>

The following schema defines the <AuthLocale> type:

```
<complexType name="AuthLocaleType">
   <sequence>
      <element name="IP" type="string" minOccurs="0"/>
      <element name="DNS_Domain" type="string" minOccurs="0"/>
   </sequence>
</complexType>
```

1.4.4 Element <AuthorizationDecisionAssertion>

```
<element name="AuthorizationDecisionAssertion"
       type="saml:AuthorizationDecisionAssertionType"/>
<complexType name="AuthorizationDecisionAssertionType">
   <complexContent>
      <extension base="saml:SubjectAssertionType">
         <sequence>
            <element ref="saml:Object"/>
            <element name="Answer" type="saml:DecisionType"/>
            <element name="saml:Evidence"
                  minOccurs="0" maxOccurs="unbounded"/>
         </sequence>
      </extension>
   </complexContent>
</complexType>
```

1.4.4.1 Element <Object>

The following schema defines the <Object> element:

```
<element name="Object" type="saml:ObjectType"/>
<complexType name="ObjectType">
```

```
        <sequence>
            <element name="Resource" type="xsd:uriReference"/>
            <element name="Namespace" type="uriReference" minOccurs="0"/>
            <element name="Action" type="string" maxOccurs="unbounded"/>
        </sequence>
    </complexType>
```

1.4.4.2 Element <Evidence>

The following schema defines the <Evidence> element:

```
    <element name="Evidence" type="saml:AssertionSpecifierType"/>
```

1.4.5 Element <AttributeAssertion>

```
    <element name="AttributeAssertion"
           type="saml:AttributeAssertionType"/>
    <complexType name="AttributeAssertionType">
        <complexContent>
            <extension base="saml:SubjectAssertionType">
                <sequence>
                    <element ref="saml:Attribute" maxOccurs="unbounded"/>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
```

1.4.5.1 Element <Attribute>

The following schema defines the <Attribute> element:

```
    <element name="Attribute" type="saml:AttributeType"/>
    <complexType name="AttributeType">
        <sequence>
            <element name="AttributeName" type="string"/>
            <element name="AttributeNamespace"
                   type="uriReference" minOccurs="0"/>
            <element name="AttributeValue" type="saml:AttributeValueType"
                   minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
    </complexType>
```

1.4.5.2 Type Attribute

The following schema defines the <Attribute> element:

```
    <complexType name="AttributeValueType">
        <sequence>
            <any namespace="##any" processContents="lax"
                   minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
    </complexType>
```

1.5 Conditions

1.5.1 Element <Conditions>

Assertion Conditions are contained in the <Conditions> element. SAML applications MAY define additional elements using an extension schema. If an application encounters an element contained within a <Conditions> element that is not understood the status of the Condition MUST be considered Indeterminate.

The following schema defines the <Conditions> element:

```
<element name="Conditions" type="saml:ConditionsType"/>
<complexType name="ConditionsType">
    <sequence>
        <element name="Condition" type="saml:AbstractConditionType"
                minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
    <attribute name="NotBefore" type="timeInstant" use="optional"/>
    <attribute name="NotOnOrAfter" type="timeInstant" use="optional"/>
</complexType>

<complexType name="AbstractConditionType" abstract="true"/>
```

1.5.1.1 Attribute NotBefore and NotOnOrAfter

The NotBefore and NotOnOrAfter attributes specify limits on the validity of the assertion.

The NotBefore attribute specifies the time instant at which the validity interval begins. The NotOnOrAfter attribute specifies the time instant at which the validity interval has ended

The NotBefore and NotOnOrAfter attributes are optional. If the value is either omitted or equal to the start of the epoch it is unspecified. If the NotBefore attribute is unspecified the assertion is valid at any time before the time instant specified by the NotOnOrAfter attribute. If the NotOnOrAfter attribute is unspecified the assertion is valid from the time instant specified by the NotBefore attribute with no expiry. If neither attribute is specified the assertion is valid at any time.

In accordance with the XML Schemas Specification, all time instances are interpreted in Universal Coordinated Time unless they explicitly indicate a time zone.

Implementations MUST NOT generate time instances that specify leap seconds.

1.5.2 Element <AudienceRestrictionCondition>

Assertions MAY be addressed to a specific audience. Although a party that is outside the audience specified is capable of drawing conclusions from an assertion, the issuer explicitly makes no representation as to accuracy or trustworthiness to such a party.

- Require users of an assertion to agree to specific terms (rule book, liability caps, relying party agreement)
- Prevent clients inadvertently relying on data that does not provide a sufficient warranty for a particular purpose
- Enable sale of per-transaction insurance services.

An audience is identified by a URI that identifies to a document that describes the terms and conditions of audience membership.

Each client is configured with a set of URIs that identify the audiences that the client is a member of, for example:

`http://cp.verisign.test/cps-2000`
Client accepts the VeriSign Certification Practices Statement

`http://rule.bizexchange.test/bizexchange_ruebook`
Client accepts the provisions of the *bizexchange* rule book.

An assertion MAY specify a set of audiences to which the assertion is addressed. If the set of audiences is the empty set there is no restriction and all audiences are addressed. Otherwise the client is not entitled to rely on the assertion unless it is addressed to one or more of the audiences that the client is a member of. For example:

`http://cp.verisign.test/cps-2000/part1`
Assertion is addressed to clients that accept the provisions of a specific part of the VeriSign CPS.

In this case the client accepts a superset of the audiences to which the assertion is addressed and may rely on the assertion.

The following schema defines the <AudienceRestrictionCondition> element:

```
<element name="AudienceRestrictionCondition"
       type="saml:AudienceRestrictionConditionType"/>
<complexType name="AudienceRestrictionConditionType">
   <complexContent>
      <extension base="saml:AbstractConditionType">
         <sequence>
            <element name="Audience" type="xsd:uriReference"
                  minOccurs="0" maxOccurs="unbounded"/>
         </sequence>
      </extension>
   </complexContent>
</complexType>
```

1.6 Advice

The Advice element is a general container for any additional information that does not affect the semantics or validity of the assertion itself.

1.6.1 Element <Advice>

The <Advice> element permits evidence supporting the assertion claims to be cited, either directly (through incorporating the claims) or indirectly (by reference to the supporting assertions.

The following schema defines the <Advice> element:

```
<element name="Advice" type="saml:AdviceType"/>
<complexType name="AdviceType">
   <sequence>
      <any namespace="##any" processContents="lax"
            minOccurs="0" maxOccurs="unbounded"/>
```

```
        </sequence>
    </complexType>
</schema>
```

1.7 Schema Extension

The SAML schema is designed to support extensibility by means of XML abstract types. Extension schemas should specify the purpose of extension elements by defining them as extensions of the appropriate abstract types.

The following abstract types are defined in the schema:

Abstract Type	Purpose
`AssertionType`	Define new SAML Assertion
`SubjectAssertionType`	Define new SAML Assertion that takes a single SAML Subject as the subject.
`AbstractConditionType`	Define a new SAML Condition
`[AuthDataType???]`	??? Define new type of Auth data ???

In addition the `<Advice>` element permits arbitrary elements to be included without type restriction.

2 SAML Protocol

SAML Assertions may be generated and exchanged using a variety of protocols. The bindings section of this document describes specific means of transporting SAML assertions using existing widely deployed protocols.

SAML aware clients may in addition use the request protocol defined by the `<SAMLQuery>` and `<SAMLQueryResponse>` elements described in this section. Figure 3 shows the class diagram for the `<SAMLQuery>` element.

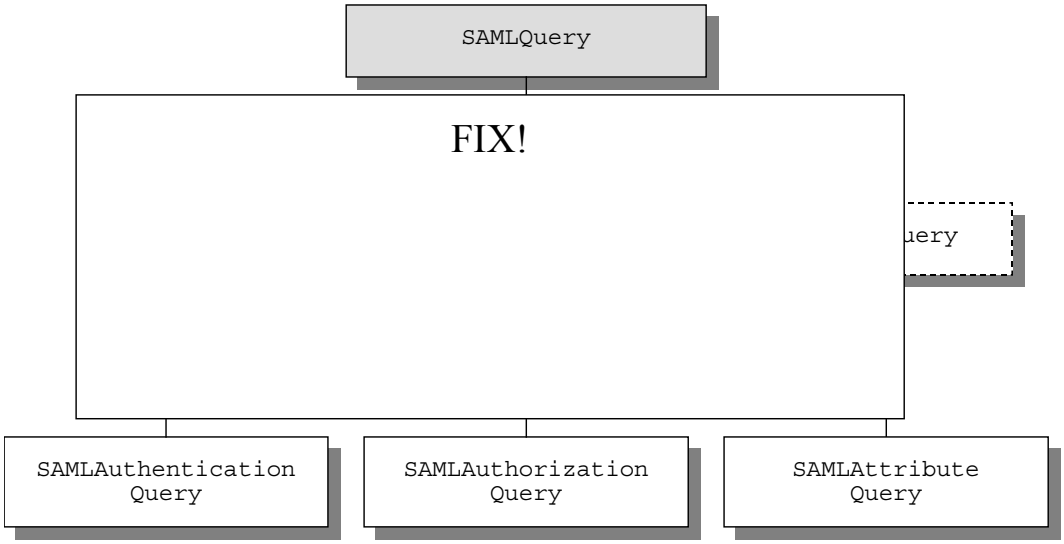

Figure 3: <SAMLQuery> Class Diagram

2.1 Namespaces

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.oasis.org/tbs/1066-12-25/protocol/"
      xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
      xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
      xmlns:saml="http://www.oasis.org/tbs/1066-12-25/"
      xmlns:samlp="http://www.oasis.org/tbs/1066-12-25/protocol/"
      xmlns="http://www.w3.org/2000/10/XMLSchema"
      elementFormDefault="unqualified">
   <import namespace="http://www.oasis.org/tbs/1066-12-25/"
         schemaLocation="draft-schema-assertion-10.xsd"/>
   <import namespace="http://www.w3.org/2000/09/xmldsig#"
         schemaLocation="xmldsig-core-schema.xsd"/>
   <annotation>
      <documentation>draft-schema-protocol-10.xsd</documentation>
   </annotation>
```

2.1.1 Type <CompletenessSpecifierType>

```
   <simpleType name="CompletenessSpecifierType">
      <restriction base="string">
         <enumeration value="Any"/>
         <enumeration value="All"/>
      </restriction>
   </simpleType>
```

2.1.2 Type <StatusCodeType>

```
   <simpleType name="StatusCodeType">
      <restriction base="string">
         <enumeration value="Success"/>
         <enumeration value="Failure"/>
         <enumeration value="Error"/>
         <enumeration value="Unknown"/>
      </restriction>
   </simpleType>
```

2.2 Request

2.2.1 Abstract Type SAMLAbstractRequestType

[PHB2]All SAML requests are extensions of the SAMLAbstractRequestType.

The following schema defines the SAMLAbstractRequestType abstract type:

```
<complexType name="SAMLAbstractRequestType" abstract="true">
   <attribute name="RequestID" type="saml:IDType" use="required"/>
   <attribute name="Version" type="string" use="required"/>
</complexType>
```

2.2.1.1 Attribute RequestID=

The RequestID attribute defines a unique identifier for the assertion request. If an assertion query specifies a RequestID value the same value MUST be returned in the response unless a Respond element of Static is specified.

2.2.1.2 Attribute Version=

The Version attribute [TBS]

2.2.2 Element <SAMLRequest>

```
<element name="SAMLRequest" type="samlp:SAMLRequestType"/>
<complexType name="SAMLRequestType">
   <complexContent>
      <extension base="samlp:SAMLAbstractRequestType">
         <choice>
            <element name="Query" type="samlp:SAMLQueryType"/>
            <element ref="saml:AssertionID" maxOccurs="unbounded"/>
         </choice>
      </extension>
   </complexContent>
</complexType>
```

2.2.3 Abstract Type SAMLQueryType

[PHB3]The SAMLQueryType type [TBS]

The following schema defines the SAMLQueryType abstract type:

```
<complexType name="SAMLQueryType" abstract="true"/>
```

2.2.4 Abstract Type SubjectQueryType

[PHB4]The SubjectQuery type extends the SubjectQuery type to specify a query with a specific subject as its principal.

following schema defines the SubjectQuery abstract type:

```
<complexType name="SubjectQueryType" abstract="true">
   <complexContent>
      <extension base="samlp:SAMLQueryType">
         <sequence>
            <element ref="saml:Subject"/>
```

```
            </sequence>
        </extension>
    </complexContent>
</complexType>
```

2.3 Authentication Query

2.3.1 Element <AuthenticationQuery>

The following schema defines the AuthenticationQuery element:

```
<complexType name="AuthenticationQueryType">
    <complexContent>
        <extension base="samlp:SubjectQueryType">
            <sequence>
                <element ref="saml:AuthenticationCode" minOccurs="0"/>
                <!--do we want more than one of these?-->
            </sequence>
        </extension>
    </complexContent>
</complexType>
```

2.4 Attribute Query

2.4.1 Element <SAMLAttributeQuery>

The following schema defines the SAMLAttributeQuery element:

```
<complexType name="AttributeQueryType">
    <complexContent>
        <extension base="samlp:SubjectQueryType">
            <sequence>
                <element ref="saml:Attribute"
                       minOccurs="0" maxOccurs="unbounded"/>
                <element name="CompletenessSpecifier"
                       type="samlp:CompletenessSpecifierType" default="All"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>
```

2.5 Authorization Query

2.5.1 Element <AuthorizationQuery>

The following schema defines the AuthorizationQuery element:

```
<element name="AuthorizationQuery" type="samlp:AuthorizationQueryType"/>
<complexType name="AuthorizationQueryType">
    <complexContent>
        <extension base="samlp:SubjectQueryType">
            <sequence>
                <element ref="saml:Evidence"
                       minOccurs="0" maxOccurs="unbounded"/>
                <element ref="saml:Object"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>
```

2.6 Response

2.6.1 Abstract Type `SAMLAbstractResponseType`

[TBS]

The following schema defines the `SAMLAbstractResponseType` abstract type:

```
    <complexType name="SAMLAbstractResponseType" abstract="true">
       <attribute name="ResponseID" type="saml:IDType" use="required"/>
       <attribute name="InResponseTo" type="saml:IDType" use="required"/>
       <attribute name="Version" type="string" use="required"/>
    </complexType>
```

2.6.2 Element `<SAMLResponse>`

[TBS]

The following schema defines the `<SAMLResponse>` element:

```
    <complexType name="SAMLResponseType">
       <complexContent>
          <extension base="samlp:SAMLAbstractResponseType">
             <sequence>
                <element ref="saml:Assertion"
                         minOccurs="0" maxOccurs="unbounded"/>
             </sequence>
             <attribute name="StatusCode" type="samlp:StatusCodeType"
                         use="required"/>
          </extension>
       </complexContent>
    </complexType>
</schema>
```

2.7 Schema Extension

The SAML schema is designed to support extensibility by means of XML abstract types. Extension schemas should specify the purpose of extension elements by defining them as extensions of the appropriate abstract types.

The following abstract types are defined in the schema:

Abstract Type	Purpose
`SAMLAbstractRequestType`	Specify a new SAML request other than a query.
`SAMLQueryType`	Specify a new SAML request that is a query.
`SubjectQueryType`	Specify a new SAML request that is a query concerning a single subject.
`SAMLAbstractResponseType`	Specify a new SAML response.

In addition the `<Advice>` element permits arbitrary elements to be included without type restriction.

3 References

[Kerberos] *TBS*

[SAML-USE] *TBS*

[PKCS1] Kaliski, B., *PKCS #1: RSA Encryption Version 2*.0, RSA Laboratories, also IETF RFC 2437, October 1998.

[RFC-2104] Krawczyk, H., Bellare, M. and R. Canetti, *HMAC: Keyed Hashing for Message Authentication*, IETF RFC 2104, February 1997.

[SOAP] D. Box, D Ehnebuske, G. Kakivaya, A. Layman, N. Mendelsohn, H. Frystyk Nielsen, S Thatte, D. Winer. *Simple Object Access Protocol (SOAP) 1.1*, W3C Note 08 May 2000, http://www.w3.org/TR/SOAP

[WSSL] E. Christensen, F. Curbera, G. Meredith, S. Weerawarana, *Web Services Description Language (WSDL) 1.0* September 25, 2000, http://msdn.microsoft.com/xml/general/wsdl.asp

[XACML] *TBS*

[XTASS] P. Hallam-Baker, *XML Trust Axiom Service Specification 1.0*, VeriSign Inc. January 2001. http://www.xmltrustcenter.org/

[XML-SIG] D. Eastlake, J. R., D. Solo, M. Bartel, J. Boyer , B. Fox , E. Simon. *XML-Signature Syntax and Processing*, World Wide Web Consortium. http://www.w3.org/TR/xmldsig-core/

[XML-SIG-XSD] XML Signature Schema available from http://www.w3.org/TR/2000/CR-xmldsig-core-20001031/xmldsig-core-schema.xsd.

[XML-Enc] *XML Encryption Specification*, In development.

[XML-Schema1] H. S. Thompson, D. Beech, M. Maloney, N. Mendelsohn. *XML Schema Part 1: Structures*, W3C Working Draft 22 September 2000, http://www.w3.org/TR/2000/WD-xmlschema-1-20000922/, latest draft at http://www.w3.org/TR/xmlschema-1/

[XML-Schema2] P. V. Biron, A. Malhotra, *XML Schema Part 2: Datatypes*; W3C Working Draft 22 September 2000, http://www.w3.org/TR/2000/WD-xmlschema-2-20000922/, latest draft at http://www.w3.org/TR/xmlschema-2/

4 Appendix

Assertion Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.oasis.org/tbs/1066-12-25/"
```

```
    xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
    xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
    xmlns:saml="http://www.oasis.org/tbs/1066-12-25/"
    xmlns="http://www.w3.org/2000/10/XMLSchema"
    elementFormDefault="unqualified">
<import namespace=" http://www.w3.org/2000/09/xmldsig#"
       schemaLocation="xmldsig-core-schema.xsd"/>
<annotation>
    <documentation>draft-schema-consensus-10.xsd</documentation>
</annotation>

<element name="AssertionID" type="saml:IDType"/>
<simpleType name="IDType">
    <restriction base="string"/>
</simpleType>
<simpleType name="DecisionType">
    <restriction base="string">
        <enumeration value="Permit"/>
        <enumeration value="Deny"/>
        <enumeration value="Indeterminate"/>
    </restriction>
</simpleType>

<complexType name="AssertionType" abstract="true">
    <sequence>
        <element name="Conditions" type="saml:ConditionsType" minOccurs="0"/>
        <element name="Advice" type="saml:AdviceType" minOccurs="0"/>
    </sequence>
    <attribute name="Version" type="string" use="required"/>
    <attribute name="AssertionID" type="saml:IDType" use="required"/>
    <attribute name="Issuer" type="string" use="required"/>
    <attribute name="IssueInstant" type="timeInstant" use="required"/>
</complexType>

<element name="AssertionSpecifier" type="saml:AssertionSpecifierType"/>
<complexType name="AssertionSpecifierType">
    <choice>
        <element ref="saml:AssertionID" />
        <element ref="saml:Assertion"/>
    </choice>
</complexType>

<complexType name="SubjectAssertionType" abstract="true">
    <complexContent>
        <extension base="saml:AssertionType">
            <sequence>
                <element ref="saml:Subject"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>

<element name="Subject" type="saml:SubjectType"/>
<complexType name="SubjectType">
    <choice maxOccurs="unbounded">
        <element ref="saml:NameIdentifier"
               minOccurs="0" maxOccurs="unbounded"/>
        <element ref="saml:HolderOfKey"
               minOccurs="0" maxOccurs="unbounded"/>
        <element ref="saml:AssertionSpecifier"
               minOccurs="0" maxOccurs="unbounded"/>
    </choice>
</complexType>
```

```
<element name="HolderOfKey" type="saml:HolderOfKeyType"/>
<complexType name="HolderOfKeyType">
    <sequence>
        <element name="Protocol" type="uriReference"
              maxOccurs="unbounded"/>
        <element name="Authdata" type="string" minOccurs="0"/>
        <element ref="ds:KeyInfo" minOccurs="0"/>
    </sequence>
</complexType>

<element name="NameIdentifier" type="saml:NameIdentifierType"/>
<complexType name="NameIdentifierType">
    <sequence>
        <element name="SecurityDomain" type="string"/>
        <element name="Name" type="string"/>
    </sequence>
</complexType>

<element name="AuthenticationAssertion"
       type="saml:AuthenticationAssertionType"/>
<complexType name="AuthenticationAssertionType">
    <complexContent>
        <extension base="saml:SubjectAssertionType">
            <sequence>
                <element ref="saml:AuthenticationCode"/>
                <element name="AuthenticationInstant" type="timeInstant"/>
                <element name="AuthLocale"
                       type="saml:AuthLocaleType" minOccurs="0"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>

<element name="AuthenticationCode" type="saml:AuthenticationCodeType"/>
<simpleType name="AuthenticationCodeType">
    <restriction base="string"/>
</simpleType>

<complexType name="AuthLocaleType">
    <sequence>
        <element name="IP" type="string" minOccurs="0"/>
        <element name="DNS_Domain" type="string" minOccurs="0"/>
    </sequence>
</complexType>

<element name="AuthorizationDecisionAssertion"
       type="saml:AuthorizationDecisionAssertionType"/>
<complexType name="AuthorizationDecisionAssertionType">
    <complexContent>
        <extension base="saml:SubjectAssertionType">
            <sequence>
                <element ref="saml:Object"/>
                <element name="Answer" type="saml:DecisionType"/>
                <element name="saml:Evidence"
                       minOccurs="0" maxOccurs="unbounded"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>

<element name="Object" type="saml:ObjectType"/>
<complexType name="ObjectType">
```

```
        <sequence>
            <element name="Resource" type="xsd:uriReference"/>
            <element name="Namespace" type="uriReference" minOccurs="0"/>
            <element name="Action" type="string" maxOccurs="unbounded"/>
        </sequence>
    </complexType>

    <element name="Evidence" type="saml:AssertionSpecifierType"/>

    <element name="AttributeAssertion"
            type="saml:AttributeAssertionType"/>
    <complexType name="AttributeAssertionType">
        <complexContent>
            <extension base="saml:SubjectAssertionType">
                <sequence>
                    <element ref="saml:Attribute" maxOccurs="unbounded"/>
                </sequence>
            </extension>
        </complexContent>
    </complexType>

    <element name="Attribute" type="saml:AttributeType"/>
    <complexType name="AttributeType">
        <sequence>
            <element name="AttributeName" type="string"/>
            <element name="AttributeNamespace"
                    type="uriReference" minOccurs="0"/>
            <element name="AttributeValue" type="saml:AttributeValueType"
                    minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
    </complexType>

    <complexType name="AttributeValueType">
        <sequence>
            <any namespace="##any" processContents="lax"
                    minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
    </complexType>

    <element name="Conditions" type="saml:ConditionsType"/>
    <complexType name="ConditionsType">
        <sequence>
            <element name="Condition" type="saml:AbstractConditionType"
                    minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
        <attribute name="NotBefore" type="timeInstant" use="optional"/>
        <attribute name="NotOnOrAfter" type="timeInstant" use="optional"/>
    </complexType>

    <complexType name="AbstractConditionType" abstract="true"/>

    <element name="AudienceRestrictionCondition"
            type="saml:AudienceRestrictionConditionType"/>
    <complexType name="AudienceRestrictionConditionType">
        <complexContent>
            <extension base="saml:AbstractConditionType">
                <sequence>
                    <element name="Audience" type="xsd:uriReference"
                            minOccurs="0" maxOccurs="unbounded"/>
                </sequence>
            </extension>
        </complexContent>
```

```
    </complexType>

    <element name="Advice" type="saml:AdviceType"/>
    <complexType name="AdviceType">
        <sequence>
            <any namespace="##any" processContents="lax"
                   minOccurs="0" maxOccurs="unbounded"/>
        </sequence>
    </complexType>
</schema>
```

Protocol Schema

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.oasis.org/tbs/1066-12-25/protocol/"
       xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
       xmlns:xsd="http://www.w3.org/2000/10/XMLSchema"
       xmlns:saml="http://www.oasis.org/tbs/1066-12-25/"
       xmlns:samlp="http://www.oasis.org/tbs/1066-12-25/protocol/"
       xmlns="http://www.w3.org/2000/10/XMLSchema"
       elementFormDefault="unqualified">
    <import namespace="http://www.oasis.org/tbs/1066-12-25/"
           schemaLocation="draft-schema-assertion-10.xsd"/>
    <import namespace="http://www.w3.org/2000/09/xmldsig#"
           schemaLocation="xmldsig-core-schema.xsd"/>
    <annotation>
        <documentation>draft-schema-protocol-10.xsd</documentation>
    </annotation>

    <simpleType name="CompletenessSpecifierType">
        <restriction base="string">
            <enumeration value="Any"/>
            <enumeration value="All"/>
        </restriction>
    </simpleType>

    <simpleType name="StatusCodeType">
        <restriction base="string">
            <enumeration value="Success"/>
            <enumeration value="Failure"/>
            <enumeration value="Error"/>
            <enumeration value="Unknown"/>
        </restriction>
    </simpleType>

    <complexType name="SAMLAbstractRequestType" abstract="true">
        <attribute name="RequestID" type="saml:IDType" use="required"/>
        <attribute name="Version" type="string" use="required"/>
    </complexType>

    <element name="SAMLRequest" type="samlp:SAMLRequestType"/>
    <complexType name="SAMLRequestType">
        <complexContent>
            <extension base="samlp:SAMLAbstractRequestType">
                <choice>
                    <element name="Query" type="samlp:SAMLQueryType"/>
                    <element ref="saml:AssertionID" maxOccurs="unbounded"/>
                </choice>
            </extension>
        </complexContent>
    </complexType>

    <complexType name="SAMLQueryType" abstract="true"/>
```

```
<complexType name="SubjectQueryType" abstract="true">
    <complexContent>
        <extension base="samlp:SAMLQueryType">
            <sequence>
                <element ref="saml:Subject"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>

<complexType name="AuthenticationQueryType">
    <complexContent>
        <extension base="samlp:SubjectQueryType">
            <sequence>
                <element ref="saml:AuthenticationCode" minOccurs="0"/>
                <!--do we want more than one of these?-->
            </sequence>
        </extension>
    </complexContent>
</complexType>

<complexType name="AttributeQueryType">
    <complexContent>
        <extension base="samlp:SubjectQueryType">
            <sequence>
                <element ref="saml:Attribute"
                       minOccurs="0" maxOccurs="unbounded"/>
                <element name="CompletenessSpecifier"
                       type="samlp:CompletenessSpecifierType" default="All"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>

<element name="AuthorizationQuery" type="samlp:AuthorizationQueryType"/>
<complexType name="AuthorizationQueryType">
    <complexContent>
        <extension base="samlp:SubjectQueryType">
            <sequence>
                <element ref="saml:Evidence"
                       minOccurs="0" maxOccurs="unbounded"/>
                <element ref="saml:Object"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>

<complexType name="SAMLAbstractResponseType" abstract="true">
    <attribute name="ResponseID" type="saml:IDType" use="required"/>
    <attribute name="InResponseTo" type="saml:IDType" use="required"/>
    <attribute name="Version" type="string" use="required"/>
</complexType>

<complexType name="SAMLResponseType">
    <complexContent>
        <extension base="samlp:SAMLAbstractResponseType">
            <sequence>
                <element ref="saml:Assertion"
                       minOccurs="0" maxOccurs="unbounded"/>
            </sequence>
            <attribute name="StatusCode" type="samlp:StatusCodeType"
                       use="required"/>
```

```
          </extension>
       </complexContent>
    </complexType>

</schema>
```

Page: 4
[PHB1] We have to align with the OASIS convention here.

Page: 16
[PHB2] asking for it to be created.

Page: 16
[PHB3] asking for it to be created.

Page: 16
[PHB4] asking for it to be created.