Shibboleth Architecture

Protocols and Profiles

10 September 2005

Document identifier:
internet2-mace-shibboleth-arch-protocols-200509

Location:
http://shibboleth.internet2.edu/shibboleth-documents.html

Editors:
Scott Cantor (cantor.2@osu.edu), The Ohio State University

Contributors:
Steven Carmody, Brown University
Marlena Erdos, Tivoli Systems, Inc.
Keith Hazelton, University of Wisconsin
Walter Hoehn, University of Memphis
RL "Bob" Morgan, University of Washington
Tom Scavo, NCSA
David Wasley, University of California

Abstract:
This specification defines the general architecture, protocols, and message formats that make up the Shibboleth web single sign-on and attribute exchange mechanism, which is built on the OASIS SAML 1.1 specification (http://www.oasis-open.org/committees/security). Readers should be familiar with that specification before reading this document.

Please submit comments to the shibboleth-dev mailing list (see http://shibboleth.internet2.edu/ for subscription details).

Table of Contents

1 Introduction

This specification defines a set of related profiles of SAML 1.1 and additional messages and protocols that make up the Shibboleth architecture. It is functionally a superset of the SAML 1.1 web browser single sign-on and attribute exchange mechanisms that incorporates additional profiles for user privacy and service-provider-first access.

Unless specifically noted, nothing in this document should be taken to conflict with the SAML 1.1 specification, or any bindings and profiles referenced within it. Readers are advised to familiarize themselves with that specification first.

1.1 Notation

This specification uses normative text to describe the use of SAML 1.1 and additional SAML profiles.

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 [RFC 2119]:

> …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:

- The prefix `saml:` stands for the SAML 1.1 assertion namespace, `urn:oasis:names:tc:SAML:1.0:assertion`
- The prefix `samlp:` stands for the SAML 1.1 request-response protocol namespace, `urn:oasis:names:tc:SAML:1.0:protocol`
- The prefix `md:` stands for the SAML 2.0 metadata namespace, `urn:oasis:names:tc:SAML:2.0:metadata`
- The prefix `saml2:` stands for the SAML 2.0 assertion namespace, `urn:oasis:names:tc:SAML:2.0:assertion`
- The prefix `ds:` stands for the W3C XML Signature namespace, `http://www.w3.org/2000/09/xmldsig#`
- The prefix `xsd:` stands for the W3C XML Schema namespace, `http://www.w3.org/2001/XMLSchema` in example listings. In schema listings, this is the default namespace and no prefix is shown.
- The prefix `xsi:` stands for the W3C XML Schema instance namespace, `http://www.w3.org/2001/XMLSchema-instance`

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

2 Architectural Overview

Broadly speaking, the Shibboleth architecture defines a set of interactions between an *identity provider* and a *service provider* to facilitate web browser single sign-on and attribute exchange.

Previous versions of this specification and the SAML 1.1 specification variously refer to these roles of identity provider and service provider as "source site" or "origin" and "destination site" or "target". This specification adopts terminology used within the Liberty ID-FF specification [LibertyProt] and the SAML 2.0 specification [SAML2Gloss].

An additional, optional component called a *WAYF service* acts independently as a possible means of identity provider discovery. The role of the WAYF can be, and often is, taken on by a service provider itself.

2.1 Identity Provider

An *identity provider* is an entity that authenticates principals and produces assertions of authentication and attribute information in accordance with the SAML Assertions and Protocols specification [SAMLCore] and the SAML browser profiles in the SAML Bindings and Profiles specification [SAMLBind]. It consists of functional components drawn from the SAML domain model, an *authentication authority* and an *attribute authority*, along with an *inter-site transfer service*, defined by the Browser profiles, and a *single sign-on service*, defined by this specification. Note that physically, the single sign-on service and inter-site transfer service MAY be the same location.

Each identity provider MUST be assigned a unique identifier, or *providerId*. The identifier MUST be a URI [RFC 2396] of no more than 1024 characters. Use of an "https" URL for this purpose may be advantageous for metadata publication (see section 3.4.2).

2.1.1 Authentication Authority

The authentication authority is a SAML-defined service that issues authentication assertions about principals to relying parties (service providers, in the case of Shibboleth). Shibboleth does not specify how authentication of principals should be performed; the authority works with the principal's authentication service so that assertions about the authentication event are issued.

The only specifically defined use of an authentication assertion in Shibboleth is in accordance with the Browser/POST and Browser/Artifact profiles. As a result, the authentication authority is NOT REQUIRED to process SAML `<samlp:Request>` messages containing `<samlp:AuthenticationQuery>` or `<saml:AssertionIDReference>` elements, but MAY choose to do so. Also note that the browser profiles do not specifically require the authentication authority to remember the assertions that it issues over an extended period of time, though this is also permitted.

2.1.2 Attribute Authority

The attribute authority is a SAML-defined service that supports a SAML protocol binding and the processing of SAML `<samlp:Request>` messages containing the `<samlp:AttributeQuery>` element. This service issues attribute assertions to service providers in a mutually authenticated fashion. Implementations typically rely on SSL/TLS [RFC 2246] or SAML message signatures to mutually authenticate the exchange.

Shibboleth additionally requires that control of attribute release to service providers be available to both administrators and principals. Therefore, a Shibboleth attribute authority MUST have the ability to authenticate requests and MUST implement some form of access control governing the release of specific attributes and values belonging to specific principals to specific requesting service providers. Subject to that constraint, any access control mechanism may be supported.

A Shibboleth attribute authority MAY implement support for <saml:SubjectConfirmation> when processing queries, but is NOT REQUIRED to do so. That is, it MAY return errors when presented with queries containing unsupported confirmation methods or when asked to produce assertions containing them.

Finally, a Shibboleth attribute authority MUST support the attribute exchange profile described in section 3.2. An attribute authority MAY also support other attribute exchange profiles that are outside the scope of this specification.

2.1.3 Single Sign-On Service

A single sign-on (SSO) service is an HTTP resource controlled by the identity provider that receives and processes authentication requests sent through the browser from service providers. The SSO service initiates the authentication process, eventually redirecting the browser to the inter-site transfer service.

The SSO service is a Shibboleth-specific service that is not defined by SAML 1.1. It supports a normative protocol to initiate SSO by a service provider, which SAML 1.1 does not define.

> **Note:** Previous versions of this specification referred to this component as the "Handle Service".

An identity provider may expose any number of SSO service endpoints. Each endpoint SHOULD be protected by SSL/TLS [RFC 2246].

2.1.4 Inter-Site Transfer Service

An inter-site transfer service is an HTTP resource controlled by the identity provider that interacts with the authentication authority to issue HTTP responses to the principal's browser adhering to the SAML Browser/POST or Browser/Artifact profiles.

In the case of the Browser/POST profile, the HTTP response contains the form controls necessary to transmit an authentication assertion inside a digitally signed <samlp:Response> message to a service provider's assertion consumer service.

In the case of the Browser/Artifact profile, the HTTP response contains a `Location` header redirecting the browser to a service provider's assertion consumer service. The redirection URL contains one or more URL-encoded SAML artifacts.

The inter-site transfer service and the SSO service MAY be located at the same HTTP endpoint, in which case the redirect mentioned in section 2.1.3 is unnecessary.

2.1.5 Artifact Resolution Service

An artifact resolution service is a SAML protocol binding endpoint controlled by the identity provider that receives requests directly from a service provider to resolve a SAML artifact into the corresponding assertion in accordance with the Browser/Artifact profile.

The service supports the processing of SAML <samlp:Request> messages containing <samlp:AssertionArtifact> elements. Implementations of this service MUST provide for mutual authentication, typically relying on SSL/TLS [RFC 2246] or SAML message signatures.

2.2 Service Provider

A *service provider* is an entity that provides a web-based service, application, or resource subject to authorization or customization on the basis of a security context established by means of a SAML

browser profile. It consists of one or more *assertion consumer services*, defined by the browser profiles, and may include an *attribute requester*.

> **Note:** Previous versions of this specification referred to these components as the "SHIRE" and "SHAR", respectively.

Each service provider MUST be assigned a unique identifier, or *providerId*. The identifier MUST be a URI [RFC 2396] of no more than 1024 characters. Use of an "https" URL for this purpose may be advantageous for metadata publication (see section 3.4.2).

2.2.1 Assertion Consumer Service

An assertion consumer service is an HTTP resource controlled by the service provider that processes form submissions adhering to the Browser/POST profile or HTTP GET requests adhering to the Browser/Artifact profile to establish a new security context for a principal. Assuming this is successful, the assertion consumer service eventually redirects the user agent to a resource hosted by the service provider.

> **Note:** [SAMLBind] refers to an assertion consumer service that supports the Browser/Artifact profile as an *artifact receiver service*. In this specification, no such distinction is made.

A service provider may expose any number of assertion consumer service endpoints. Each endpoint SHOULD be protected by SSL/TLS [RFC 2246].

2.2.2 Attribute Requester

Shibboleth supplements the SAML browser profiles with an out-of-band attribute exchange. A service provider MAY utilize a SAML protocol binding to send a SAML `<samlp:Request>` message containing a `<samlp:AttributeQuery>` element to attribute authorities and process the resulting attribute assertions. Implementations MUST provide for mutual authentication of the exchange, typically relying on SSL/TLS [RFC 2246] or SAML message signatures.

Note that in some environments where privacy is not required, a well-known principal identifier might be communicated in the authentication assertion. This may be done to make the exchange of attributes optional, or to support a non-SAML mechanism such as LDAP to obtain additional information. Also, the authentication assertion MAY itself include `<saml:AttributeStatement>` elements (or be accompanied by additional assertions that do).

A Shibboleth attribute requester MAY implement support for `<saml:SubjectConfirmation>` when submitting queries and processing assertions, but is NOT REQUIRED to do so. That is, it MAY reject assertions containing unsupported confirmation methods.

2.3 WAYF

A WAYF, or "Where are you from?", service is an optional, centralized mechanism for interactively determining a principal's identity provider. A service provider in general has no means to determine this without asking the principal or deriving the information through some user agent interaction. The WAYF is a means for service providers to collectively delegate this step to a separate entity. Service providers are NOT REQUIRED to utilize a WAYF.

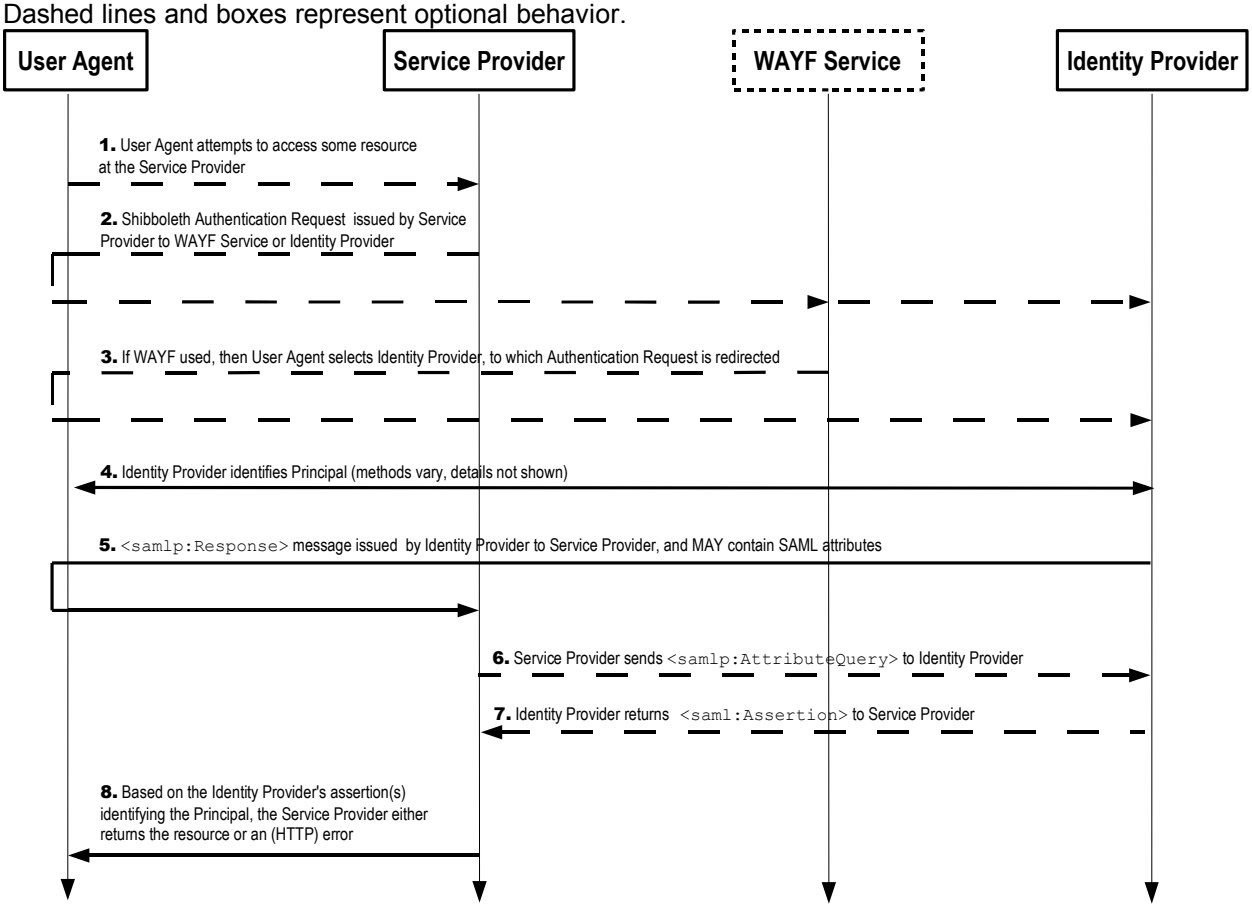
A WAYF service MUST support the Shibboleth Authentication Request profile defined in section 3.1.1. This is the same profile supported by an identity provider's SSO service. The WAYF acts as a proxy for a service provider and relays the authentication request from the service provider to the SSO service of the selected identity provider.

A WAYF service is free to interact with the principal's user agent in whatever manner it deems appropriate to determine the identity provider to which to relay the authentication request. This includes, but is not limited to, presenting lists, a search interface, heuristics based on client characteristics, etc. A WAYF service SHOULD provide some means for the user agent to cache the user's selection, perhaps using HTTP cookies, but SHOULD also provide reasonable means for the user to change the selection in the future.

2.4 Single Sign-On Overview

The following sequence diagram illustrates the set of required and optional interactions when using the Browser/POST profile. The Browser/Artifact profile replaces step 5 below with an artifact issued to the service provider followed by a back-channel request/response exchange between the service provider and identity provider. See [SAMLBind] for detailed descriptions of both profiles.

Dashed lines and boxes represent optional behavior.



1. HTTP Request to Service Provider

In step 1, the principal, via an HTTP user agent, makes an HTTP request for a secured resource at the service provider without a security context.

2. Authentication Request issued by Service Provider to WAYF or Identity Provider

In step 2, the service provider issues an authentication request and redirects the user agent to either a WAYF or directly to an identity provider. A WAYF is typically used if the service provider wishes to delegate the task of identity provider discovery (see section 2.3).

3. **WAYF redirects Authentication Request to selected Identity Provider**

   If a WAYF is used in step 2, it interacts via unspecified means with the user agent to select an identity provider to which to redirect the user agent with the service provider's authentication request.

4. **Identity Provider identifies Principal**

   In step 4, the principal is identified by the identity provider by some means outside the scope of this specification. This may require a new act of authentication, or it may reuse an existing authenticated session.

5. **Identity Provider issues <samlp:Response> or SAML Artifact(s) to Service Provider**

   In step 5, the identity provider issues a SAML `<samlp:Response>` message or one or more SAML artifacts to be delivered by the user agent to the service provider. Either the SAML 1.1 Browser/POST profile or Browser/Artifact profile may be used.

   If the Browser/POST profile is used, then either one or more assertions or an error status is passed directly through the user agent to the service provider in the response. If the Browser/Artifact profile is used, then one or more SAML artifacts are passed through the user agent to the service provider, at which point the service provider communicates directly with the identity provider to resolve the artifact(s) into assertions. This back-channel communication is not shown in the diagram. Refer to [SAMLBind] for additional details.

6. **Service Provider sends Attribute Query to Identity Provider**

   In step 6, the service provider optionally uses the subject of the authentication assertion it received in step 5 to send a `<samlp:AttributeQuery>` (inside a SAML request message) to an attribute authority associated with the identity provider.

7. **Identity Provider returns SAML Assertion to Service Provider**

   In step 7, the attribute authority associated with the identity provider processes the `<samlp:AttributeQuery>` and returns a SAML response message, possibly containing one or more assertions containing attributes that apply to the principal.

8. **Service Provider grants or denies access to Principal**

   In step 8, the service provider responds to the principal's user agent with an error, or establishes its own security context for the principal and returns the requested resource.

Note that an identity provider MAY initiate this sequence at step 5 and issue an unsolicited SAML response message or SAML artifact(s) to a service provider without the preceding steps.

Also note that in addition to steps 6 and 7 being optional, an identity provider MAY include `<saml:AttributeStatement>` elements in the assertion(s) that it returns in step 5. This is commonly referred to as "attribute push". A service provider MAY still perform step 6 at its discretion, whether or not attributes are received in step 5, although generally this is omitted, at least initially, when attributes have been pushed.

3 Protocols and Profiles

This section defines the message exchanges required of Shibboleth implementations (primarily defined by SAML 1.1), and additional profiles governing the behavior of Shibboleth components.

3.1 Authentication Request and Response Profiles

To establish a security context at a service provider, Shibboleth combines an Authentication Request profile defined in this specification with the SAML 1.1 Browser/POST or Browser/Artifact profile [SAMLBind]. An identity provider MAY initiate this process without an authentication request by directing the principal's user agent through unspecified means to its inter-site transfer service with sufficient information to create the proper HTTP response.

3.1.1 Authentication Request Profile

A Shibboleth authentication request is a URL-encoded message sent from a service provider (or another entity on its behalf, such as a WAYF service) to an identity provider's single sign-on service endpoint using the principal's user agent. Any means of causing the user agent to access the SSO service endpoint can be used; typically an HTTP redirect is used subsequent to the user agent accessing a secured resource without a valid security context.

3.1.1.1 Required Information

Identification: urn:mace:shibboleth:1.0:profiles:AuthnRequest

Contact Information: shibboleth-dev@internet2.edu

Description: Given below.

Updates: All earlier technical definitions of the Shibboleth authentication request format

3.1.1.2 Message Format and Transmission

The HTTP request to the identity provider's SSO service endpoint MUST use the GET method and MUST contain the following URL-encoded query string parameters:

`providerId`

: The unique identifier of the requesting service provider

`shire`

: The assertion consumer service endpoint at the service provider to which to deliver the authentication response

`target`

: A value specified by the service provider to be returned by the identity provider in the TARGET form control or query string of the authentication response; it SHOULD be opaque to the identity provider, but MAY be the URL of a resource accessed at the service provider

The query string MAY contain the following optional parameter:

`time`

: The current time, in seconds elapsed since midnight, January 1st, 1970, as a string of up to 10 base10 digits

A WAYF service MUST relay the parameters that it receives from a service provider unchanged to the identity provider that is ultimately selected, except that it MUST replace the `time` parameter (if present) with a value generated at the time the user agent is redirected to the identity provider's SSO service.

3.1.1.3 Processing Rules

The SSO service endpoint MUST process the supplied request and either return an error response to the user agent or attempt to fulfill the request by eventually redirecting the user agent to the inter-site transfer service.

If an error occurs, the identity provider MAY return a `<samlp:Response>` in accordance with the Browser/POST profile that contains a `<samlp:Status>` element with a `Value` other than `samlp:Success`. If the service provider only supports the use of the Browser/Artifact profile, then it is not possible to return an error indication as the Browser/Artifact profile assumes that any artifact supplied references an actual assertion. (The base SAML profiles presume successful authentication because they are identity-provider-first profiles.)

When using the Browser/POST profile, the `shire` parameter is used as the value of the `ACTION` attribute in the HTML form in the HTTP response returned by the inter-site transfer service, and is also the value placed in the `Recipient` attribute of the `<samlp:Response>` element encoded into the `SAMLResponse` form control. The `target` parameter MUST be used as the value of the `TARGET` form control whether or not an error has occurred.

When using the Browser/Artifact profile, the `shire` parameter is used as the URL prefix in the `Location` header in the HTTP redirect response returned by the inter-site transfer service. The `target` parameter MUST be used as the value of the `TARGET` query string parameter whether or not an error has occurred.

The `providerId` parameter MAY be used by the identity provider to customize the processing of the request based on its knowledge of or relationship with the service provider. Such customization might include, but is not limited to, the format of the principal's identifier to be returned in the assertion(s), the credential to use while signing the `<samlp:Response>` message, and the set of attributes to include with the authentication assertion, if any.

Note that if the service provider's identity is used as input to processing the request (which is almost always the case), then the identity provider MUST have some means to establish that the assertion consumer service endpoint in the `shire` parameter is in fact associated with the requesting service provider. Any mechanism to establish this relationship MAY be used, but some mechanism MUST be used unless the data in the authentication response is invariant with respect to the requesting service provider. The metadata profile described in section 3.4 is RECOMMENDED for this purpose.

Metadata MAY be used to determine the profile to use in returning the authentication response to the service provider. If an `<md:AssertionConsumerService>` element in metadata with a `Location` attribute corresponding to the `shire` parameter indicates support for only one of the response profiles (via the `Binding` attribute), then the identity provider MUST use this profile when returning the authentication response. If it cannot or will not use this profile, then the identity provider MUST return an error message to the user agent.

Finally, the `time` parameter MAY be used as an indicator of the freshness of the request so that replayed requests, such as might be triggered by navigation of a user agent's history list, can be detected. The `time` parameter MUST NOT be used as part of any security measures.

3.1.1.4 Example

```
https://idp.example.org/SSO?shire=https%3A%2F%2Fsp.example.com%2FShibboleth.shire&
target=https%3A%2F%2Fsp.example.com%2Fcgi-bin%2Fcoolstuff.cgi&time=1050540300&
providerId=https%3A%2F%2Fsp.example.com%2Fshibboleth
```

3.1.2 Browser/POST Authentication Response Profile

When the Browser/POST profile is used to respond to the service provider, a signed SAML response containing an authentication assertion is delivered directly to the service provider in a form POST operation. The format of the SAML response and the associated processing rules are defined primarily by the SAML Browser/POST profile in [SAMLBind].

An identity provider MAY send a response without having received an authentication request, in which case, the `TARGET` form control MUST contain a value expected to be understood by the service provider. In most cases, this SHOULD be the URL of a resource to be accessed at the service provider, but MAY contain other values by prior agreement.

Note that the identity provider MAY supply attributes within the `<samlp:Response>` message (so-called attribute push), at its discretion (this is implicitly permitted by the Browser/POST profile). However, see section 4.1.1 for additional considerations in doing so. The Browser/Artifact profile may be more suitable in such cases.

As an additional constraint, the `Issuer` attribute of any assertions included MUST be set to the unique identifier of the identity provider issuing the assertion.

Finally, any assertions included SHOULD contain a `<saml:AudienceRestrictionCondition>` with at least one `<saml:Audience>` element containing the unique identifier of the service provider.

3.1.2.1 Example

The example below shows XML that might be base64-encoded into the `SAMLResponse` form control.

```
<samlp:Response
  xmlns:samlp="urn:oasis:names:tc:SAML:1.0:protocol"
  IssueInstant="2003-04-17T00:46:02Z"
  MajorVersion="1" MinorVersion="1"
  Recipient="https://sp.example.com/Shibboleth.shire"
  ResponseID="_c7055387-af61-4fce-8b98-e2927324b306">
  <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
    <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="#_c7055387-af61-4fce-8b98-e2927324b306">
        <ds:Transforms>
          <ds:Transform
            Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature"/>
          <ds:Transform Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#">
            <InclusiveNamespaces PrefixList="#default saml samlp ds xsd 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>TCDVSuG6grhyHbzhQFWFzGrxIPE=</ds:DigestValue>
      </ds:Reference>
    </ds:SignedInfo>
    <ds:SignatureValue>
      x/GyPbzmFEe85pGD3c1aXG4Vspb9V9jGCjwcRCKrtwPS6vdVNCcY5rHaFPYWkf+5
      EIYcPzx+pX1h43SmwviCqXRjRtMANWbHLhWAptaK1ywS7gFgsD01qjyen3CP+m3D
      w6vKhaqledl0BYyrIzb4KkHO4ahNyBVXbJwqv5pUaE4=
    </ds:SignatureValue>
    <ds:KeyInfo>
      <ds:X509Data>
        <ds:X509Certificate>
          MIICyjCCAjOgAwIBAgICAnUwDQYJKoZIhvcNAQEEBQAwgakxCzAJBgNVBAYTAlVT
          MRIwEAYDVQQIEwlXaXNjb25zaW4xEDAOBgNVBAcTB01hZGlzb24xIDAeBgNVBAoT
          F1VuaXZlcnNpdHkgb2YgV2lzY29uc2luMSswKQYDVQQLEyJEaXZpc2lvbiBvZiBJ
```

```
          bmZvcm1hdGlvbiBUZWNobm9sb2d5MSUwIwYDVQQDExxIRVBLSSBTZXJ2ZXIgQ0Eg
          LS0gMjAwMjA3MDFBMB4XDTAyMDcyNjA3Mjc1MVoXDTA2MDkwNDA3Mjc1MVowgYsx
          CzAJBgNVBAYTAlVTMREwDwYDVQQIEwhNaWNoaWdhbjESMBAGA1UEBxMJQW5uIEFy
          Ym9yMQ4wDAYDVQQKEwVVQ0FJRDEcMBoGA1UEAxMTc2hpYjEuaW50ZXJuZXQyLmVk
          dTEnMCUGCSqGSIb3DQEJARYYcm9vdEBzaGliMS5pbnRlcm5ldDIuZWR1MIGfMA0G
          CSqGSIb3DQEBAQUAA4GNADCBiQKBgQDZSAb2sxvhAXnXVIVTx8vuRay+x50z7GJj
          IHRYQgIv6IqaGG04eTcyVMhoekE0b45QgvBIaOAPSZBl13R6+KYiE7x4XAWIrCP+
          c2MZVeXeTgV3Yz+USLg2Y1on+Jh4HxwkPFmZBctyXiUr6DxF8rvoP9W7O27rhRjE
          pmqOIfGTWQIDAQABox0wGzAMBgNVHRMBAf8EAjAAMAsGA1UdDwQEAwIFoDANBgkq
          hkiG9w0BAQQFAAOBgQBfDqEW+OI3jqBQHIBzhujN/PizdN7s/z4D5d3pptWDJf2n
          qgi7lFV6MDkhmTvTqBtjmNk3No7v/dnP6Hr7wHxvCCRwubnmIfZ6QZAv2FU78pLX
          8I3bsbmRAUg4UP9hH6ABVq4KQKMknxu1xQxLhpR1ylGPdiowMNTrEG8cCx3w/w==
        </ds:X509Certificate>
      </ds:X509Data>
    </ds:KeyInfo>
  </ds:Signature>
  <samlp:Status><samlp:StatusCode Value="samlp:Success"/></samlp:Status>
  <saml:Assertion
    xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"
    AssertionID="_a75adf55-01d7-40cc-929f-dbd8372ebdfc"
    IssueInstant="2003-04-17T00:46:02Z"
    Issuer="https://idp.example.org/shibboleth">
    <saml:Conditions
      NotBefore="2003-04-17T00:46:02Z"
      NotOnOrAfter="2003-04-17T00:51:02Z">
      <saml:AudienceRestrictionCondition>
        <saml:Audience>http://sp.example.com/shibboleth</saml:Audience>
      </saml:AudienceRestrictionCondition>
    </saml:Conditions>
    <saml:AuthenticationStatement
      AuthenticationInstant="2003-04-17T00:46:00Z"
      AuthenticationMethod="urn:oasis:names:tc:SAML:1.0:am:password">
      <saml:Subject>
        <saml:NameIdentifier
          Format="urn:mace:shibboleth:1.0:nameIdentifier"
          NameQualifier="https://idp.example.org/shibboleth">
          3f7b3dcf-1674-4ecd-92c8-1544f346baf8
        </saml:NameIdentifier>
        <saml:SubjectConfirmation>
          <saml:ConfirmationMethod>
            urn:oasis:names:tc:SAML:1.0:cm:bearer
          </saml:ConfirmationMethod>
        </saml:SubjectConfirmation>
      </saml:Subject>
      <saml:SubjectLocality IPAddress="127.0.0.1"/>
    </saml:AuthenticationStatement>
  </saml:Assertion>
</samlp:Response>
```

3.1.3 Browser/Artifact Authentication Response Profile

When the Browser/Artifact profile is used to respond to the service provider, one or more SAML artifacts are issued to the service provider by way of the query string of an HTTP redirect response. The format of the HTTP response and the associated processing rules are defined primarily by the SAML Browser/Artifact profile in [SAMLBind]. Note that the SAML artifact value returned in the `SAMLart` query string parameter MUST be URL-encoded.

The Browser/Artifact profile permits a variety of artifact formats to be used. Two different formats are defined by [SAMLBind], either of which MAY be used.

An identity provider MAY send a response without having received an authentication request; in such a case, the `TARGET` parameter MUST contain a value expected to be understood by the service provider. In most cases, this SHOULD be the URL of a resource to be accessed at the service provider, but MAY contain other values by prior agreement.

Upon receiving the artifact(s), the service provider uses a SAML request/response protocol binding to resolve the artifact(s) into the corresponding SAML assertion(s), in accordance with [SAMLBind].

It is RECOMMENDED that service providers enforce a single-use semantic on the artifact values they receive to prevent an attacker from interfering with the resolution of an artifact by a user agent and then resubmitting it to the service provider. If an attempt to resolve an artifact does not complete successfully, the artifact SHOULD be placed into a blocked artifact list for a period of time that exceeds a reasonable acceptance period during which the identity provider would successfully resolve the artifact. This recommendation is in addition to the existing SAML 1.1 requirement that the identity provider enforce a single-use semantic on artifact values, and matches a recommendation added to SAML 2.0 when using artifacts.

Note that the identity provider MAY supply attributes within the SAML assertions it returns in response to an artifact lookup at its discretion (this attribute push is implicitly permitted by the Browser/Artifact profile). In fact, this is typical when using this profile.

As an additional constraint, the `Issuer` attribute of any assertions returned MUST be set to the unique identifier of the identity provider issuing the assertion.

Finally, any assertions returned SHOULD contain a `<saml:AudienceRestrictionCondition>` with at least one `<saml:Audience>` element containing the unique identifier of the service provider.

3.1.3.1 Example

The example below shows a redirection URL containing a type 0x0001 SAML artifact that might be returned when using this profile. For examples of the subsequent SOAP-based exchange to obtain the assertion, refer to [SAMLBind].

```
https://sp.example.com/Shibboleth.shire?SAMLart=AAH7iBsAkCvNPMBcQlDBx%2FAlFu8FW8FM5Z
apUHYA8Nzz4nr19fBabdCU&TARGET=https%3A%2F%2Fsp.example.com%2Fcgi-bin%2Fcoolstuff.cgi
```

3.2 Attribute Exchange Profile

To support out-of-band attribute exchange from an identity provider to a service provider, Shibboleth specifies the use of the SAML request/response protocol using the `<samlp:AttributeQuery>` element as defined in [SAMLCore], along with the additional constraints and guidelines defined in this section. Other scenarios involving different actors MAY be supported by the same software components but are beyond the scope of this profile.

3.2.1 Required Information

Identification: urn:mace:shibboleth:1.0:profiles:attribute

Contact Information: shibboleth-dev@internet2.edu

Description: Given below.

Updates: All earlier technical definitions of the Shibboleth attribute syntax and exchange conventions

3.2.2 Attribute Requests

An attribute request message is a `<samlp:Request>` element containing a `<samlp:AttributeQuery>` element. The `Resource` attribute in the query MUST contain the requesting service provider's unique identifier. This is used to make up for the lack of an explicit element or attribute in SAML 1.1 to indicate the issuing service provider.

3.2.2.1 Example

The example shown does not include any surrounding context from the binding, such as a SOAP envelope.

```
<samlp:Request
  xmlns:samlp="urn:oasis:names:tc:SAML:1.0:protocol"
  IssueInstant="2004-05-25T22:46:10Z"
  MajorVersion="1" MinorVersion="1"
  RequestID="aaf2319617732113474afe114412ab72">
  <samlp:AttributeQuery Resource="https://sp.example.com/shibboleth">
    <saml:Subject
      xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion">
      <saml:NameIdentifier
        Format="urn:mace:shibboleth:1.0:nameIdentifier"
        NameQualifier="https://idp.example.org/shibboleth">
        3f7b3dcf-1674-4ecd-92c8-1544f346baf8
      </saml:NameIdentifier>
    </saml:Subject>
  </samlp:AttributeQuery>
</samlp:Request>
```

3.2.3 Attribute Responses

An attribute response is a `<samlp:Response>` element containing a `<samlp:Status>` element and zero or more `<saml:Assertion>` elements. The assertion(s), if any, SHOULD contain only attribute statements. The `Issuer` attribute of any assertions returned MUST be set to the unique identifier of the identity provider whose attribute authority issued the assertion. Any assertions returned SHOULD contain a `<saml:AudienceRestrictionCondition>` with at least one `<saml:Audience>` element containing the unique identifier of the requesting service provider.

As noted in section 2.1.2, Shibboleth attribute authorities MUST implement some form of access control over attribute release. They MAY support unauthenticated queries, but SHOULD limit the release of information in such a case, subject to administrative policy.

3.2.3.1 Example

The example shown does not include any surrounding context from the binding, such as a SOAP envelope.

```
<samlp:Response
  xmlns:samlp="urn:oasis:names:tc:SAML:1.0:protocol"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  InResponseTo="aaf2319617732113474afe114412ab72"
  IssueInstant="2004-05-25T22:46:10.940Z"
  MajorVersion="1" MinorVersion="1"
  ResponseID="b07b804c7c29ea1673004f3d6f7928ac">
  <samlp:Status>
    <samlp:StatusCode Value="samlp:Success"/>
  </samlp:Status>
  <saml:Assertion
    xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"
    AssertionID="a144e8f3adad594a9649924517abe933"
    IssueInstant="2004-05-25T22:46:10.939Z"
    MajorVersion="1" MinorVersion="1"
    Issuer="https://idp.example.org/shibboleth">
    <saml:Conditions
      NotBefore="2004-05-25T22:46:10.939Z"
      NotOnOrAfter="2004-05-25T23:16:10.939Z">
      <saml:AudienceRestrictionCondition>
        <saml:Audience>https://sp.example.com/shibboleth</saml:Audience>
      </saml:AudienceRestrictionCondition>
```

```
    </saml:Conditions>
    <saml:AttributeStatement>
      <saml:Subject>
        <saml:NameIdentifier
          Format="urn:mace:shibboleth:1.0:nameIdentifier"
          NameQualifier="https://idp.example.org/shibboleth">
          3f7b3dcf-1674-4ecd-92c8-1544f346baf8
        </saml:NameIdentifier>
      </saml:Subject>
      <saml:Attribute
        AttributeName="urn:mace:dir:attribute-def:eduPersonEntitlement"
        AttributeNamespace="urn:mace:shibboleth:1.0:attributeNamespace:uri">
        <saml:AttributeValue xsi:type="xsd:anyURI">
          urn:mace:oclc.org:100277910
        </saml:AttributeValue>
        <saml:AttributeValue xsi:type="xsd:anyURI">
          urn:mace:example.edu:exampleEntitlement
        </saml:AttributeValue>
        <saml:AttributeValue xsi:type="xsd:anyURI">
          urn:mace:incommon:entitlement:common:1
        </saml:AttributeValue>
      </saml:Attribute>
    </saml:AttributeStatement>
  </saml:Assertion>
</samlp:Response>
```

3.2.4 Attribute Naming and Syntax

SAML does not constrain the naming of attributes or the syntax of values. It is RECOMMENDED that Shibboleth attributes be identified with a URI. In such cases, the `AttributeName` XML attribute MUST contain the URI that identifies the attriibute, and the `AttributeNamespace` XML attribute SHOULD contain the value `urn:mace:shibboleth:1.0:attributeNamespace:uri`. It MAY contain a different value by prior agreement.

It is also RECOMMENDED that attribute values be expressed, when possible, as a single XML text node within the `<saml:AttributeValue>` element, using an XML Schema built-in datatype [Schema2] . In such cases, the `xsi:type` XML attribute SHOULD be used to indicate the built-in datatype that describes the allowable syntax of the value.

If the value is not from a built-in datatype, the `xsi:type` attribute MAY be used to indicate the extension type in use, but implementers are cautioned that this may require a relying party to be aware of the extension in order to process the assertion. Omitting the `xsi:type` attribute is RECOMMENDED in such cases.

See the example in section 3.2.3.1.

3.3 Transient NameIdentifier Format

SAML 1.1 identifies principals in assertions using the string-valued `<saml:NameIdentifier>` element, which contains a pair of optional XML attributes, `Format` and `NameQualifier`. See the examples in the previous sections.

Shibboleth permits any legal SAML 1.1 name identifier to be used, but also defines a special kind of identifier with the `Format` value of `urn:mace:shibboleth:1.0:nameIdentifier`. Identifiers of this format MUST satisfy the following criteria:

- The identifier has transient semantics and SHOULD be treated as an opaque and temporary value by the relying party.
- The identifier MUST be constructed in accordance with the rules for SAML identifiers (see section 1.2.3 of [SAMLCore]) and SHOULD NOT exceed a length of 256 characters.

- The `NameQualifier` attribute MUST be set to the unique identifier of the identity provider that originally created the transient identifier. The `NameQualifier` attribute MAY be omitted if it can be assumed from the context of the message containing the element (e.g. the issuer of a containing assertion or the recipient of a query).

3.4 Metadata Profile

Editor's Note: This profile has been jointly submitted with Trustgenix, Inc. to the OASIS Security Services Technical Committee and has been published as a committee draft. This section has been adapted to reference and build on the draft by specifying only Shibboleth-specific constraints.

SAML profiles (and by extension Shibboleth profiles) require agreements between system entities regarding identifiers, binding/profile support and endpoints, certificates and keys, and so forth. A metadata specification is useful for describing this information in a standardized way.

Although SAML 1.1 did not include such a specification, SAML 2.0 introduces a metadata specification in [SAML2Meta]. Subsequently, a profile of the SAML 2.0 metadata specification was developed for use by SAML 1.1 deployments [SAML1Meta]. Shibboleth identity and service providers SHOULD describe their characteristics using this profile. When doing so, specific use of these elements MUST adhere to the profile defined in [SAML1Meta]. Additional guidelines and processing rules pertaining to Shibboleth are specified below.

3.4.1 Element <md:EntitiesDescriptor>

Multiple Shibboleth entities can be collected into groups using the `<md:EntitiesDescriptor>` element. The `Name` XML attribute, if present, SHOULD be a URI, in which case the value of the attribute is a globally unique reference to the collection of entities enclosed by the element.

3.4.2 Element <md:EntityDescriptor>

A Shibboleth identity or service provider SHOULD be represented by an `<md:EntityDescriptor>` element. If used, there MUST be exactly one `<md:EntityDescriptor>` element for each provider and the unique identifier of the provider MUST be placed in the `entityID` XML attribute.

Role elements defined by this profile applicable to Shibboleth include `<md:IDPSSODescriptor>`, `<md:SPSSODescriptor>`, `<md:AuthnAuthorityDescriptor>`, and `<md:AttributeAuthorityDescriptor>`. Other elements of type **md:RoleDescriptorType** may be defined and supported, but are beyond the scope of this specification.

If a URL is used as the unique identifier of an entity, it is RECOMMENDED that resolving this URL produce a SAML metadata document containing a single `<md:EntityDescriptor>` representing that entity.

Note that metadata can vary based on the relying party in question. Resolving an entity's identifier into metadata MAY require authentication of the requester so as to produce the metadata response appropriate for that relying party. Use of an “https” scheme in the unique identifier may facilitate this.

3.4.3 Element <md:IDPSSODescriptor>

A Shibboleth identity provider MUST include the `<md:IDPSSODescriptor>` element in a metadata instance that is produced for consumption by a Shibboleth service provider. The `protocolSupportEnumeration` XML attribute MUST include at least the values:

```
urn:oasis:names:tc:SAML:1.1:protocol
urn:mace:shibboleth:1.0
```

At least one <md:SingleSignOnService> element MUST be present. At least one of the <md:SingleSignOnService> elements' Binding XML attribute MUST contain the value

```
urn:mace:shibboleth:1.0:profiles:AuthnRequest
```

and moreover, the location specified in its Location XML attribute MUST support the Authentication Request Profile defined in section 3.1.1.

3.4.4 Element <md:AuthnAuthorityDescriptor>

A Shibboleth identity provider that supports an authentication authority service as described in section 2.1.1 MUST include the <md:AuthnAuthorityDescriptor> element in its metadata if it supports lookup of assertions by SAML query or identifier. The protocolSupportEnumeration XML attribute MUST include at least the value:

```
urn:oasis:names:tc:SAML:1.1:protocol
```

3.4.5 Element <md:AttributeAuthorityDescriptor>

A Shibboleth identity provider that supports an attribute authority service as described in section 2.1.2 MUST include the <md:AttributeAuthorityDescriptor> element in a metadata instance that is produced for consumption by a Shibboleth service provider. The protocolSupportEnumeration XML attribute MUST include at least the value:

```
urn:oasis:names:tc:SAML:1.1:protocol
```

Any <saml2:Attribute> elements SHOULD follow the guidelines on attribute naming and syntax in section 3.2.4.

3.4.6 Element <md:SPSSODescriptor>

A Shibboleth service provider MUST include the <md:SPSSODescriptor> element in a metadata instance produced for consumption by a Shibboleth identity provider. The protocolSupportEnumeration XML attribute MUST include at least the value:

```
urn:oasis:names:tc:SAML:1.1:protocol
```

Any <md:RequestedAttribute> elements SHOULD follow the guidelines on attribute naming and syntax in section 3.2.4.

4 Security and Privacy Considerations

As Shibboleth is principally a set of SAML profiles, the general security and privacy considerations that apply to SAML apply to Shibboleth (see [SAMLSecure]).

4.1 Additional Browser Profile Considerations

4.1.1 Information Leakage and Impersonation

The SAML browser profiles contain a presumption that they are initiated by an identity provider. Assertion information (or an artifact) is therefore sent through the browser to service providers using locations known to be appropriate and secure.

The use of the Authentication Request profile defined in section 3.1.1 introduces the possibility of a malicious entity impersonating another service provider by identifying itself as one provider while indicating that the authentication response be delivered to the attacker instead. In the case of the POST profile, this can result in unintended leakage of personally identifying information contained within the assertion(s). In the case of the Artifact profile, the attacker could potentially impersonate the principal by immediately submitting the artifact(s) to the real service provider, who can subsequently authenticate to the identity provider to obtain the assertion.

To mitigate both attacks, it is critical for the identity provider to securely associate the assertion consumer service location to be used with the service provider to whom the assertion(s) or artifact(s) are issued. A digital signature over the authentication request would be an alternate countermeasure, but this is not supported by the Authentication Request profile.

Another source of information leakage is the `target` parameter sent with the Authentication Request URL and the `TARGET` parameter returned in both Browser profiles. This parameter is informally associated with the resource URL being requested from the service provider, but it is in fact potentially opaque to the identity provider. Exposing the resource URL releases unnecessary information about the principal's activities to the identity provider and may appear in various log files.

It is therefore RECOMMENDED that service providers utilize some kind of obfuscation, mapping, encryption, or other mechanism to prevent the exposure of resource URLs in plaintext in this parameter. Alternately, service providers MAY use a fixed value in that parameter, and maintain the state associated with the request (such as the eventual resource URL) locally by using HTTP cookies.

Finally, when user privacy in service provider interactions is a consideration or requirement, Shibboleth provides an explicit mechanism for effective anonymity when interacting with a service provider through the use of a transient identifier (see section 3.3), provided that the SAML attributes supplied in conjunction with or subsequent to it are sufficiently generic so as not to inadvertently narrow down or identify the principal. It is important to avoid facilitating coordination by service providers in correlating the principal's activity by ensuring that a different transient identifier is used across time and space. Therefore, it is RECOMMENDED that a given transient identifier not be used more than once in assertions issued by an identity provider for a principal in different executions of the Browser/POST or Browser/Artifact profiles, and that the use of a transient identifier (in queries, for example) be constrained to the service provider for which it was created.

4.1.2 Time Synchronization

The Browser/POST profile relies on tight synchronization of clocks between the identity and service providers to limit the usefulness of the bearer assertion. Additionally, assertions may be issued with expiration conditions that cannot be effectively honored if clock skew is excessive. Therefore, it is RECOMMENDED that secure time sources be used to maintain clock synchronization among all servers within the bounds usually associated with protocols like Kerberos (i.e., on the order of 5 minutes or less).

5 References

The following works are referenced directly or indirectly in the body of this specification.

5.1 Normative References

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

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

[RFC 2396] T. Berners-Lee et al. *Uniform Resource Identifiers (URI): Generic Syntax.* IETF RFC 2396, August, 1998. http://www.ietf.org/rfc/rfc2396.txt.

[SAMLCore] E. Maler et al. Assertions *and Protocols for the OASIS Security Assertion Markup Language (SAML)*. OASIS, September 2003. Document ID oasis-sstc-saml-core-1.1. http://www.oasis-open.org/committees/security/.

[SAMLBind] E. Maler et al. *Bindings and Profiles for the OASIS Security Assertion Markup Language (SAML)*. OASIS, September 2003. Document ID oasis-sstc-saml-bindings-profiles-1.1. http://www.oasis-open.org/committees/security/.

[SAML-XSD] E. Maler et al. *SAML assertion schema.* OASIS, September 2003. Document ID oasis-sstc-saml-schema-assertion-1.1. http://www.oasis-open.org/committees/security/.

[SAMLP-XSD] E. Maler et al. *SAML protocol schema*. OASIS, September 2003. Document ID oasis-sstc-saml-schema-protocol-1.1. http://www.oasis-open.org/committees/security/.

[SAMLSecure] E. Maler et al. *Security and Privacy Considerations for the OASIS Security Assertion Markup Language (SAML)*. OASIS, September 2003. Document ID oasis-sstc-saml-sec-consider-1.1. http://www.oasis-open.org/committees/security/.

[SAML2Meta] S. Cantor et al., *Metadata for the OASIS Security Assertion Markup Language (SAML) V2.0*. OASIS SSTC, March 2005. Document ID sstc-saml-metadata-2.0. See http://www.oasis-open.org/committees/security/.

[SAML1Meta-xsd] S. Cantor et al., *SAML 1.x Metadata Profile Schema*. OASIS SSTC, March 2005. Document ID sstc-saml1x-metadata. See http://www.oasis-open.org/committees/security/.

[SAML1Meta] G. Whitehead and S. Cantor, *SAML 1.x Metadata Profile*. OASIS SSTC, March 2005. Document ID sstc-saml1x-metadata-cd-01. See http://www.oasis-open.org/committees/security/.

[Schema2] P. V. Biron et al. *XML Schema Part 2: Datatypes*. World Wide Web Consortium Recommendation, May 2001. http://www.w3.org/TR/xmlschema-2/.

5.2 Non-Normative References

[SAML2Gloss] J. Hodges et al., *Glossary for the OASIS Security Assertion Markup Language (SAML) V2.0*. OASIS SSTC, March 2005. Document ID sstc-saml-glossary-2.0. See http://www.oasis-open.org/committees/security/.

[LibertyProt] J. Kemp et al., *Liberty Protocols and Schema Specification* Version 1.2, Liberty Alliance Project, August 2004, http://www.projectliberty.org/specs/v1_2/liberty-architecture-protocols-schema-v1.2.pdf.