Shibboleth Architecure

Protocols and Profiles

Working Draft 01, 25 May 2004

Document identifier:
draft-mace-shibboleth-arch-protocols-01

Location:
http://shibboleth.internet2.edu/

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

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

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, destination-site-first access, and identity provider discovery.

All Shibboleth implementations must support the required aspects of this specification to interoperate effectively.

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 IETF 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 assertion namespace, `urn:oasis:names:tc:SAML:2.0:assertion`.
- The prefix `samlp:` stands for the SAML request-response protocol namespace, `urn:oasis:names:tc:SAML:2.0:protocol`.
- 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.

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], also based on SAML, and the draft SAML 2.0 specification.

An additional non-normative component called a *WAYF service* acts independently as a means of identity provider discovery.

TBD: A sequence diagram

2.1 Identity Provider

An identity provider is an entity that authenticates principals and produces assertions of authentication and attribute information in accordance with [SAMLCore] and the SAML Browser/POST profile in [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/POST profile, and a *single sign-on service*, defined by this specification.

Each identity provider MUST be assigned a unique identifier, or *provider ID*. The identifier MUST be a URI [RFC 2396] of no more than 1024 characters.

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 profile. As a result, the authentication authority is NOT REQUIRED to support a SAML protocol binding or process SAML `<samlp:Request>` messages containing `<samlp:AuthenticationQuery>` or `<saml:AssertionIDReference>` elements. It MAY of course choose to do so. Also note that the Browser/POST profile does not specifically require the authentication authority to remember the assertions that it issues, though this is also of course 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. It issues attribute assertions to service providers, typically using 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 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.

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 and initiates the authentication process, eventually redirecting the browser to the inter-site transfer service.

This 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.

An identity provider may expose any number of SSO service endpoints. They 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 profile. The response contains the form controls necessary to transmit a short-lived authentication assertion inside a digitally signed `<samlp:Response>` message to a service provider's assertion consumer service.

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 the SAML Browser/POST profile. It consists of one or more *assertion consumer services*, defined by the Browser/POST profile, and may include an *attribute requester*. Previous versions of this specification referred to these components as the "SHIRE" and "SHAR".

Each service provider MUST be assigned a unique identifier, or *provider ID*. The identifier MUST be a URI [RFC 2396] of no more than 1024 characters.

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 SAML Browser/POST profile to establish a new security context for a principal. Assuming this is successful, it eventually redirects the browser to a resource at the service provider.

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

2.2.2 Attribute Requester

Shibboleth supplements the SAML Browser/POST profile with an out of band attribute exchange. A service provider MAY utilize a SAML protocol binding to send SAML `<samlp:Request>` messages containing the `<samlp:AttributeQuery>` element to attribute authorities and process the resulting attribute assertions, typically using SSL/TLS [RFC 2246] or SAML message signatures to mutually authenticate the exchange.

Note that in some environments where privacy is not required, a well-known principal identifier might be communicated in the authentication assertion, making the exchange of attributes optional, or to support a non-SAML mechanism such as LDAP to obtain additional information.

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

2.3 WAYF

A WAYF, or "Where are you from?", service is a 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. 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 protocol defined in section 3.1. This is the same protocol 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 browser 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.

Both service providers and WAYF services MAY use the Identity Provider Discovery profile defined in section 3.6 as a means of determining (and caching) a principal's identity provider(s).

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

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

3.1.1 Authentication Request

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 browser. Any means of causing the browser to access the SSO service endpoint can be used; typically an HTTP redirect is used subsequent to the browser accessing a secured resource without a valid security context.

3.1.1.1 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 to be returned by the identity provider in the TARGET form control of the authentication response

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 with a value generated at the time the browser is redirected to the identity provider's SSO service.

3.1.1.2 Processing Rules

The SSO service endpoint MUST process the supplied request and either issue an error to the browser or attempt to fulfill the request by eventually redirecting the browser 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`.

The `target` parameter MUST be used as the value of the `TARGET` form control in the HTTP response returned by the inter-site transfer service, whether or not an error has occurred.

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 `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 push with the authentication assertion, if any.

Note that if the service provider's identity is used as input to processing the request, 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, such as out-of-band exchange or querying of a trusted source for this information, but some mechanism MUST be used unless the data in the authentication response is invariant with respect to the requesting service provider.

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 browser's history list, can be detected. It is NOT a security measure.

3.1.1.3 Example

```
https://idp.example.org/SSO?shire=https%3A%2F%2Fsp.example.com%2FShibboleth.shire&
target=https%3A%2F%2Fsp.example.com%2Fcgi-bin%2Flogin.cgi&time=1084819377&
providerId=https%3A%2F%2Fsp.example.com%2Fshibboleth%2F
```

3.1.2 Authentication Response

The format of the authentication response and the associated processing rules are defined entirely by the SAML Browser/POST profile in [SAMLBind]. An identity provider MAY send a response without having received an authentication request; in such a 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, at its discretion (this is implicitly permitted by the Browser/POST profile).

The assertion(s) returned in the response MUST be consistent with the profiles described in sections 3.3-3.5.

3.1.2.1 Example

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

```
<Response
  IssueInstant="2003-04-17T00:46:02Z" MajorVersion="1" MinorVersion="1"
  Recipient="https://sp.example.com/Shibboleth.shire"
  ResponseID="_c7055387-af61-4fce-8b98-e2927324b306"
  xmlns="urn:oasis:names:tc:SAML:1.0:protocol"
  xmlns:samlp="urn:oasis:names:tc:SAML:1.0:protocol">
<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>
<Status><StatusCode Value="samlp:Success"/></Status>
<Assertion
  AssertionID="_a75adf55-01d7-40cc-929f-dbd8372ebdfc"
  IssueInstant="2003-04-17T00:46:02Z"
  Issuer="https://idp.example.org/shibboleth/"
  MajorVersion="1" MinorVersion="1"
  xmlns="urn:oasis:names:tc:SAML:1.0:assertion">
<Conditions NotBefore="2003-04-17T00:46:02Z" NotOnOrAfter="2003-04-17T00:51:02Z">
<AudienceRestrictionCondition>
<Audience>http://sp.example.com/shibboleth/</Audience>
</AudienceRestrictionCondition>
</Conditions>
<AuthenticationStatement
  AuthenticationInstant="2003-04-17T00:46:00Z"
  AuthenticationMethod="urn:oasis:names:tc:SAML:1.0:am:password">
<Subject>
<NameIdentifier Format="urn:mace:shibboleth:1.0:nameIdentifier"
  NameQualifier="https://idp.example.org/shibboleth/">
3F7B3DCF-1674-4ecd-92C8-1544F346BAF8
</NameIdentifier>
<SubjectConfirmation>
```

```
<ConfirmationMethod>urn:oasis:names:tc:SAML:1.0:cm:bearer</ConfirmationMethod>
</SubjectConfirmation>
</Subject>
<SubjectLocality IPAddress="127.0.0.1"/>
</AuthenticationStatement>
</Assertion>
</Response>
```

3.2 Attribute Request and Response

To support 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]. Implementations MUST support the SAML SOAP binding [SAMLBind] with SSL/TLS [RFC 2246] for mutual authentication, integrity protection, and confidentiality. They MAY implement additional bindings and security mechanisms (such as digital signatures).

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.

3.2.1 Attribute Request

An attribute request message is a `<samlp:Request>` element containing a `<samlp:AttributeQuery>` element.

Additionally, the `Resource` attribute in the query MUST contain the requesting service provider's unique identifier.

3.2.1.1 Example

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

```
<Request xmlns="urn:oasis:names:tc:SAML:1.0:protocol"
 xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"
 xmlns:samlp="urn:oasis:names:tc:SAML:1.0:protocol"
 IssueInstant="2004-05-25T22:46:10Z" MajorVersion="1" MinorVersion="1"
 RequestID="aaf2319617732113474afe114412ab72">
<AttributeQuery Resource="http://sp.example.com/shibboleth/">
<Subject xmlns="urn:oasis:names:tc:SAML:1.0:assertion">
<NameIdentifier Format="urn:mace:shibboleth:1.0:nameIdentifier"
 NameQualifier="http://idp.example.org/shibboleth/">
 082dd87d-f380-4fd6-8726-694ef2bb71e9
</NameIdentifier>
</Subject>
</AttributeQuery>
</Request>
```

3.2.2 Attribute Response

An attribute response is a `<samlp:Response>` element containing a `<samlp:Status>` and zero or more `<saml:Assertion>` elements. The assertion(s), if any, SHOULD contain only attribute statements. The assertion(s) MUST be consistent with the profiles described in sections 3.3 and 3.5.

3.2.2.1 Example

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

```
<Response xmlns="urn:oasis:names:tc:SAML:1.0:protocol"
 xmlns:saml="urn:oasis:names:tc:SAML:1.0:assertion"
 xmlns:samlp="urn:oasis:names:tc:SAML:1.0:protocol"
 InResponseTo="aaf2319617732113474afe114412ab72"
 IssueInstant="2004-05-25T22:46:10.940Z" MajorVersion="1" MinorVersion="1"
 ResponseID="b07b804c7c29ea1673004f3d6f7928ac">
<Status><StatusCode Value="samlp:Success"></StatusCode></Status>
<Assertion xmlns="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/">
<Conditions NotBefore="2004-05-25T22:46:10.939Z"
 NotOnOrAfter="2004-05-25T23:16:10.939Z">
</Conditions>
<AttributeStatement>
<Subject>
 <NameIdentifier Format="urn:mace:shibboleth:1.0:nameIdentifier"
  NameQualifier="https://idp.example.org/shibboleth/">
082dd87d-f380-4fd6-8726-694ef2bb71e9
</NameIdentifier>
</Subject>
<Attribute AttributeName="urn:mace:dir:attribute-def:eduPersonEntitlement"
 AttributeNamespace="urn:mace:shibboleth:1.0:attributeNamespace:uri">
 <AttributeValue>urn:mace:oclc.org:100277910</AttributeValue>
 <AttributeValue>urn:mace:example.edu:exampleEntitlement</AttributeValue>
 <AttributeValue>urn:mace:incommon:entitlement:common:1</AttributeValue>
</Attribute>
</AttributeStatement>
</Assertion>
</Response>
```

3.3 NameIdentifier Profile

SAML identifies principals in assertions using the `<saml:NameIdentifier>` element, which contains a pair of descriptive XML attributes, `Format` and `NameQualifier`.

Shibboleth permits any format of name identifier to be used, and also defines an additional format with the URI value of `urn:mace:shibboleth:1.0:nameIdentifier`. Identifiers of this format MUST adhere to 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.

In all cases, the `NameQualifier` attribute MUST be set to the unique identifier of the identity provider issuing the assertion containing the element.

3.4 Authentication Assertion Profile

The authentication assertions issued by Shibboleth identity providers MUST adhere to the `<saml:NameIdentifier>` profile defined in section 3.3.

Furthermore, the `Issuer` attribute MUST be set to the unique identifier of the identity provider issuing the assertion.

3.5 Attribute Assertion Profile

The attribute assertions issued by Shibboleth identity providers MUST adhere to the `<saml:NameIdentifier>` profile defined in section 3.3.

Furthermore, the `Issuer` attribute MUST be set to the unique identifier of the identity provider issuing the assertion.

SAML does not constrain the naming of attributes or the syntax of values. It is RECOMMENDED that Shibboleth attributes be identified with a URI [RFC 2396]. In such a case, 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 as simple strings when possible.

3.6 Identity Provider Discovery Profile

[LibertyBind] defines an "introduction" profile by which a service provider can discover which identity providers a principal is using with the Browser SSO profile. In deployments having more than one identity provider, service providers need a means to discover which identity provider(s) a principal uses. The discovery profile relies on a cookie that is written in a domain that is common between identity providers and service providers in a deployment. The domain that the deployment predetermines is known as the common domain in this profile, and the cookie containing the list of identity providers is known as the common domain cookie.

Shibboleth specifies the use of this profile, with the following changes. These changes are consistent with the version of the profile expected to be adopted in the SAML 2.0 specification in the future.

- The name of the cookie MUST be _saml_idp.
- The format of the cookie value MUST be a set of one or more base-64 encoded URI values separated by a single space character. Each URI is the unique identifier of an identity provider. The final set of values is then URL encoded.

The profile is unchanged in all other respects.

4 Security and Privacy Considerations

5 References

The following works are cited in the body of this specification.

5.1 Normative References

[LibertyBind]	J. Kemp et al., *Liberty Bindings and Profiles Specification* Version 1.2, Liberty Alliance Project, August 2004, http://www.projectliberty.org/specs/v1_2/liberty-architecture-bindings-profiles-v1.2.pdf.
[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/.

5.2 Non-Normative References

[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.