Liberty ID-WSF Profiles for Liberty enabled User Agents and Devices

Version: 1.0

Editors:
Robert Aarts, Nokia Corporation

Abstract:

User agents or devices, i.e. personal computers, mobile terminals, etc., participate in ID-WSF transactions in various ways. This document specifies profiles for some cases where user agents or devices act as an ID-WSF entity, i.e. execute software that implements at least parts of the ID-WSF specifications.

Filename: liberty-idwsf-client-profiles-v1.0.pdf

Notice

This document has been prepared by Sponsors of the Liberty Alliance. Permission is hereby granted to use the document solely for the purpose of implementing the Specification. No rights are granted to prepare derivative works of this Specification. Entities seeking permission to reproduce portions of this document for other uses must contact the Liberty Alliance to determine whether an appropriate license for such use is available.

Implementation of certain elements of this document may require licenses under third party intellectual property rights, including without limitation, patent rights. The Sponsors of and any other contributors to the Specification are not, and shall not be held responsible in any manner for identifying or failing to identify any or all such third party intellectual property rights. **This Specification is provided "AS IS", and no participant in the Liberty Alliance makes any warranty of any kind, express or implied, including any implied warranties of merchantability, non-infringement of third party intellectual property rights, and fitness for a particular purpose.** Implementors of this Specification are advised to review the Liberty Alliance Project's website (http://www.projectliberty.org/) for information concerning any Necessary Claims Disclosure Notices that have been received by the Liberty Alliance Management Board.

Liberty Alliance Project
Licensing Administrator
c/o IEEE-ISTO
445 Hoes Lane
Piscataway, NJ 08855-1331, USA
info@projectliberty.org

Contents

1. Notation and Conventions

This specification uses schema documents conforming to W3C XML Schema (see [Schema1]) and normative text to describe the syntax and semantics of XML-encoded messages.

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

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.

Namespaces

- The prefix disco: represents the namespace defined in [LibertyDisco].
- The prefix lib: represents the namespace defined in [LibertyProtSchema].
- The prefix sa: represents the namespace defined in [LibertyAuthn].
- The prefix sec: represents the namespace defined in [LibertySecMech].
- S: represents the namespace defined in [SOAPv1.1]

2. Overview

The ID-WSF specifications define a number of protocols that enable any party to act as a *Web Service Consumer*, a *Web Service Provider*, or both. When user agents or devices wish to act in these roles, some particular issues need to be addressed and hence additional specifications are useful to guarantee interoperability. The Liberty Alliance specifies the ID-WSF Authentication Service by which a WSC on a user agent or device may authenticate to an identity provider, and LibertyPAOS to enable a user agent or device to act as a WSP. Also, whenever a WSC or WSP acts as a user agent it typically represents only a very small number of users, hence there are some particular considerations regarding privacy.

User agents and devices that send or consume protocol messages specified in the ID-WSF (or ID-FF) specifications are called *Liberty enabled User Agents and Devices*, abbreviated as *LUAD*. The defining characteristic of a LUAD is that it is closely associated with one user (or a few users, such as a family); the LUAD represents that user. This is very different from a web-site that acts as WSC or WSP and may represent thousands of users. In addition, a LUAD is often, but certainly not always, *not* a highly-available HTTP server, unlike web-site based WSCs and WSPs.

To illustrate some of the issues we briefly sketch out a scenario where a LUAD acts as a WSC in a typical ID-WSF setting. The following, as well as the remainder of this document, assumes familiarity with the Liberty ID-WSF specifications, especially the Discovery Service and Security Mechanisms.

Any WSC that wishes to contact an ID-WSF WSP requires a `ResourceOffering` and often some security tokens. A WSC typically obtains these from a Liberty ID-WSF Discovery Service (discovery service). However, the discovery service is a WSP too, so for the WSC to make a request to the discovery service, it needs a `<disco:ResourceOffering>` and security tokens for the discovery service.

A WSC can get such discovery service specific information when it acts as an SP in the Liberty Identity Federation Framework during a single-sign-on transaction; the identity provider can insert the necessary information to contact the discovery service in a `<lib:AuthnResponse>`. This process is informally known as "bootstrapping ID-WSF".

But a LUAD-WSC is not a web-site that acts as SP. So when the LUAD-WSC needs to contact the discovery service it needs somehow to contact a party that can issue the `<disco:ResourceOffering>` and tokens needed. Here we recommend that the LUAD-WSC obtains this information through the the Liberty ID-WSF Authentication Service ([LibertyAuthn]) offered by an identity provider.

The identity provider will need to authenticate the LUAD – this is similar to the identity provider authenticating Principals that use a browser. As the LUAD-WSC is not a full-blown browser, however, it may not be able to present a login form.

The identity provider and LUAD should use a protocol for authentication. The use of [LibertyAuthn] is recommended for this purpose.

Once the LUAD-WSC can make requests to the discovery service it can ask the discovery service for descriptions and tokens for a particular identity service type (a WSP). If the WSP that is referred to in the discovery service response requires security tokens (such as a `<sec:ResourceAccessStatement>`) the discovery service will create such tokens. Normally such tokens include a `providerID` for the WSC and require that the WSC can authenticate as that provider to the WSP, perhaps by signing the request with a particular key. A LUAD-WSC however does not have a `providerID`, as a `providerID` could compromise the privacy of the LUAD user: the LUAD-WSC would show the same `providerID` to various WSPs allowing the WSPs to collude about the LUAD-WSC and hence about the user. Thus the content of security tokens should be profiled for various situations.

In summary, this document then specifies how *LUAD* implementations should utilize the various Liberty Alliance specifications in order to enable particular scenarios while ensuring a high degree of interoperability, security and privacy. The following sections specify and discuss profiles for particular uses of a LUAD. Note that in each section, profiles are defined for both the LUAD as well as for the providers that (wish to) interact with the LUAD.

3. LUAD-WSC Profile

A LUAD-WSC will often need to authenticate to a provider; for example when that LUAD-WSC wants to make a request to a discovery service. The discovery service may have been set up to require a security token; web-site based WSCs typically obtain such a token during a authentication transaction with an identity provider associated to that discovery service. But with a LUAD-WSC there may not be an associated browsing session, hence no interaction with an identity provider has occurred and the WSC cannot have a valid security token for the discovery service. In another typical scenario the WSP is not an ID-WSF WSP, i.e. not an "identity providing" service but an "identity consuming" service, much like an ID-FF service provider (here we abbreviate those non-ID-WSF Web Service Providers as *wSP* to indicate that these are a subclass of SPs). A LUAD-WSC that requests service from such wSPs may need to obtain ID-FF authentication assertions that will be presented as security tokens to the wSP.

As the LUAD represents at most a few users, the LUAD should not use a single authentication identity towards different providers. To achieve the required level of security and privacy the LUAD and provider must carefully choose the authentication mechanism and nature of credentials.

A LUAD-WSC implementation must adhere to the following rules:

1. The LUAD-WSC SHOULD avoid being traceable across providers. Hence, the LUAD SHOULD NOT authenticate to different providers using a single credential.
   Note:
   This implies that if a LUAD-WSC employs message level confidentiality protection, a different signing key should be used in communication with each individual provider.

2. If a LUAD-WSC is required to authenticate to a provider directly, because it does not have or cannot obtain security tokens, the LUAD-WSC SHOULD authenticate using [LibertyAuthn].
   Note:
   This applies to situations where the LUAD *itself* needs to assert its identity to a provider – typically only when a LUAD authenticates to an identity provider. In most cases a LUAD-WSC can obtain (bearer) security tokens from a Liberty ID-WSF Discovery Service and would include these tokens in the message to the WSP.

3. A LUAD-WSC SHOULD use the ID-WSF Authentication Service specified in [LibertyAuthn] to obtain security tokens from an identity provider; these tokens can then be used when submitting a `<disco:Query>` to a Discovery Service.

4. A LUAD-WSC that wishes to interact with a WSP SHOULD support at least the `urn:liberty:security:2003-08:TLS:Beare` security mechanism as specified in [LibertySecMech].

Note:

Note that these rules *do* allow the LUAD to authenticate to a provider, using a client certificate. However, that same certificate should not be used to authenticate to another provider. For example a LUAD-WSC could use its certificate to authenticate to a discovery service or an identity provider (to both if both interfaces are offered by one provider) but not then to another WSP.

3.1. Rules for WSPs that offer service to LUADs

ID-WSF compliant WSPs that register with a discovery service SHOULD support at least the `urn:liberty:security:2003-08:TLS:Bearer` security mechanism as specified in [LibertySecMech].

3.2. Examples

See [LibertyAuthn] for examples of interactions of a LUAD-WSC.

4. LUAD acting as WSP

A WSP that is deployed on a LUAD is again not very different from a network WSP. One issue for a client-WSP is reachability: a LUAD is typically not acting as a HTTP/SOAP server, may be behind a firewall, and does not have a fixed IP address.

A second issue is that a LUAD-WSP, by definition, offers service for only one, or a few, Principals. Hence, the LUAD-WSP cannot have a *service provider* identity. Normally a WSP needs to offer a `providerID` and `metadata` that WSCs use to construct requests. A LUAD-WSP should not have a `providerID` and hence cannot publish metadata. Metadata and signing keys make the client traceable to different WSCs, compromising the privacy of the LUAD user.

Note:

A Liberty ID-WSF Discovery Service hosted on a LUAD has to satisfy additional rules (see next section).

4.1. LUAD-WSP profile

A LUAD-WSP must adhere to the following rules:

1. It is RECOMMENDED that LUADs that are not normally reachable expose ID-WSP web services over LibertyPAOS
   Note:
   Note that future versions of the ID-WSF specifications may include SOAP bindings for alternative approaches, such as SIP.

2. The LUAD-WSP SHOULD avoid being traceable. If the WSP uses message level confidentiality protection a different signing key for communications with different WSCs SHOULD be used.

3. As the LUAD-WSP is not an entity different from the Principal it represents, it should not have a `providerID`. A discovery service cannot issue a `ResourceOffering` for entities that do not have a `providerID`. Hence, A LUAD-WSP SHOULD NOT register with a Discovery Service.

An ID-WSF WSC that requests services from a LUAD-WSP must adhere to the following rules:

1. If authentication of the WSP is needed it is RECOMMENDED that SP/WSCs authenticate the LUAD-WSP using ID-FF, presumably before making a ID-WSF request to the PAOS-exposed WSP.

See [LibertyPAOS] for an example of interaction with a LUAD-WSP. Another example is given in Section 5.

5. LUAD implementations of a Discovery Service

A LUAD implementation of a discovery service, i.e. a LUAD-DS, can be useful as a discovery service can inform parties in its immediate proximity about identity services for the user of the LUAD. For example a LUAD-DS could inform a mall entrance about a personal profile service, or inform a parking exit about a payment service. As with any LUAD-WSP implementations there are some issues around traceability of the client, but in a discovery service these problems are more important as a discovery service very likely must issue signed security tokens to parties that subsequently will submit those tokens to a WSP.

5.1. LUAD-DS Profile

An ID-WSF discovery service that executes at a LUAD must adhere to the following rules:

1. The LUAD-DS implementation SHOULD adhere to the rules defined for LUAD-WSP implementations.

2. The key that the LUAD-DS uses to sign security tokens SHOULD be unique for each WSP that registers with the LUAD-DS. The LUAD-DS SHOULD inform the WSP about the key when the WSP registers with the LUAD-DS, i.e. the LUAD should include the key in the `disco:ModifyResponse` as specified in [LibertyDisco].
   When the LUAD-DS sends key material it MUST ensure Transport Layer Channel Protection, and in the presence of intermediaries MUST also ensure Message Confidentiality Protection, using one of the mechanisms specified in [LibertySecMech].

References

Normative

[LibertyBindProf] Cantor, Scott, Kemp, John, eds. "Liberty ID-FF Bindings and Profiles Specification," Version 1.2-errata-v1.0, (18 April 2004). *http://www.projectliberty.org/specs*

[LibertyDisco] Sergent, Jonathan, eds. "Liberty ID-WSF Discovery Service Specification," Version 1.0-08, Liberty Alliance Project (24 July 2003). *http://www.projectliberty.org/specs*

[LibertyInteract] Aarts, Robert, eds. "Liberty ID-WSF Interaction Service Specification," Version 1.0, Liberty Alliance Project (12 November 2003). *http://www.projectliberty.org/specs*

[LibertyMetadata] Davis, Peter, eds. "Liberty Metadata Description and Discovery Specification," Version 1.0, Liberty Alliance Project (12 November 2003). *http://www.projectliberty.org/specs*

[LibertyPAOS] Aarts, Robert, eds. "Liberty Reverse HTTP Binding for SOAP Specification," Version 1.0, Liberty Alliance Project (12 November 2003). *http://www.projectliberty.org/specs*

[LibertyProtSchema] Cantor, Scott, Kemp, John, eds. "Liberty ID-FF Protocols and Schema Specification," Version 1.2, Liberty Alliance Project (20 January 2004). *http://www.projectliberty.org/specs*

[LibertySecMech] Ellison, Gary, eds. "Liberty ID-WSF Security Mechanisms," Version 1.0, Liberty Alliance Project (12 November 2003). *http://www.projectliberty.org/specs*

[LibertyAuthn] Hodges, Jeff, Aarts, Robert, eds. " Liberty ID-WSF Authentication Service Specification ," Version 1.0-16, Liberty Alliance Project (26 Feb 2004). *http://www.projectliberty.org/specs/ [http://www.projectliberty.org/specs/]*

[LibertySOAPBinding] Hodges, Jeff, Aarts, Robert, eds. " Liberty ID-WSF SOAP Binding Specification ," Version 1.0, Liberty Alliance Project (12 November 2003). *http://www.projectliberty.org/specs*

[RFC2119] Bradner, S., eds. "Key words for use in RFCs to Indicate Requirement Levels," RFC 2119, The Internet Engineering Task Force (March 1997). *http://www.ietf.org/rfc/rfc2119.txt [March 1997]*.

[RFC3066] Alvestrand, H., eds. (January 2001). "Tags for the Identification of Languages," RFC 3066., Internet Engineering Task Force *http://www.ietf.org/rfc/rfc3066.txt [January 2001]*.

[RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., Berners-Lee, T., eds. (June 1999). "Hypertext Transfer Protocol – HTTP/1.1," RFC 2616, The Internet Engineering Task Force *http://www.ietf.org/rfc/rfc2616.txt [June 1999]*.

[Schema1] Thompson, Henry S., Beech, David, Maloney, Murray, Mendelsohn, Noah, eds. (May 2002). "XML Schema Part 1: Structures," Recommendation, World Wide Web Consortium *http://www.w3.org/TR/xmlschema-1/*

[SOAPv1.1] "Simple Object Access Protocol (SOAP) 1.1," Box, Don, Ehnebuske, David , Kakivaya, Gopal, Layman, Andrew, Mendelsohn, Noah, Nielsen, Henrik Frystyk, Winer, Dave, eds. World Wide Web Consortium W3C Note (08 May 2000). *http://www.w3.org/TR/2000/NOTE-SOAP-20000508/*

Informative

[LibertyIDPP] Kellomaki, Sampo, eds. "Liberty Identity Personal Profile Service Specification," Version 1.0, Liberty Alliance Project (12 November 2003). *http://www.projectliberty.org/specs*