Liberty ID-FF 1.2 Errata

Version: 1.0

Editors:
John Kemp, IEEE-ISTO
Peter Thompson, IEEE-ISTO
Darryl Champagne, IEEE-ISTO

Abstract:

This document contains errata items pertaining to the Liberty ID-FF 1.2 specification set.

Filename: draft-liberty-idff-1.2-errata-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.

Copyright © 2004 ActivCard; America Online, Inc.; American Express Travel Related Services; Axalto; Bank of America Corporation; Bell Canada; Cingular Wireless; Cisco Systems, Inc.; Communicator, Inc.; Deloitte & Touche LLP; Earthlink, Inc.; Electronic Data Systems, Inc.; Entrust, Inc.; Epok, Inc.; Ericsson; Fidelity Investments; France Telecom; Gemplus; General Motors; Hewlett-Packard Company; i2 Technologies, Inc.; Internet2; Intuit Inc.; MasterCard International; NEC Corporation; Netegrity, Inc.; NeuStar, Inc.; Nextel Communications; Nippon Telegraph and Telephone Corporation; Nokia Corporation; Novell, Inc.; NTT DoCoMo, Inc.; OneName Corporation; Openwave Systems Inc.; Phaos Technology; Ping Identity Corporation; PricewaterhouseCoopers LLP; RegistryPro, Inc.; RSA Security Inc; Sabre Holdings Corporation; SAP AG; SchlumbergerSema; Sigaba; SK Telecom; Sony Corporation; Sun Microsystems, Inc.; Symlabs, Inc.; Trustgenix; United Airlines; VeriSign, Inc.; Visa International; Vodafone Group Plc; Wave Systems. All rights reserved.

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

Contents

1. Introduction 4
2. Abbreviations 5
3. Target Specifications 6
4. Editorial Errata 7
5. Substantive Errata 20
References 26

1. Introduction

This document lists errata in the Liberty ID-FF v1.2 specification set. The specification set targeted by this errata document is listed in section 2 below. This is not an authoritative document, nor a final version, but a precursor for changes that will likely be included in a future revision of the targeted specifications.

The ID-FF v1.2 protocols, as initially specified, contained certain material errors, collectively referred to as *errata*. Readers of the Liberty ID-FF v1.2 specification set should note the errata in this document, and incorporate it into their reading of the specifications.

Additionally, implementers of the affected specifications should use the Liberty schemata listed below, in place of those affected by the specified errata.

- Filename: liberty-idff-protocols-schema-1.2-errata-v1.0.xsd
- Filename: liberty-metadata-1.0-errata-v1.0.xsd

2. Abbreviations

The following abbreviations are used in this document:

SE - Substantive Errata designator

EE - Editorial Errata designator

CR - Change Request number (included for internal reference only).

3. Target Specifications

- [1] Liberty ID-FF Protocols & Schema Specification
  http://www.projectliberty.org/specs/liberty-idff-protocols-schema-v1.2.pdf

- [2] Liberty ID-FF Bindings & Profiles Specification
  http://www.projectliberty.org/specs/liberty-idff-bindings-profiles-v1.2.pdf

- [3] Liberty Metadata Description & Discovery Specification
  http://www.projectliberty.org/specs/liberty-metadata-v1.0.pdf

- [4] Liberty Authentication Context Specification
  http://www.projectliberty.org/specs/liberty-authentication-context-v1.2.pdf

4. Editorial Errata

4.1. [EE1] Default value for NameQualifier attribute

4.1.1. Summary

The `NameQualifier` attribute, present on name identifiers is most often set to the same provider identifier that is the subject of the assertion. An optimization of this scheme would allow the `NameQualifier` to be omitted in this case.

4.1.2. Resolution

1. Change line 534 to read:
   Liberty name identifier elements are directly based on saml:NameIdentifierType. Liberty ID-FF use of SAML-defined attributes is as follows - it should be noted that NameQualifier and Format attributes are more specifically defined in Liberty messages than in SAML (see [SAMLCore11]) and their omission has specific meaning:

2. Deleted lines 537-541.

4.2. [EE2] Default value for Format

4.2.1. Summary

The `Format` attribute, present on name identifiers is most often set set to the URN *urn:liberty:iff:nameid:federated*. An optimization of this scheme would allow the `Format` to be omitted in this case.

4.2.2. Resolution

1. Line 542 of [1] changes to read:
   `Format` [Optional]

2. Lines 543, 544 of [1] change to read:
   Indicates the format, semantics, and processing rules associated with the identifier. If this attribute is omitted, then the value *urn:liberty:nameid:federated* is assumed. Otherwise, one of these four values MUST be present:

3. line 574 of [1] changes to read:
   SHOULD be omitted, or else it MUST contain a value of *urn:liberty:iff:nameid:federated* (of the formats defined in this specification, only *federated* name identifiers sometimes require encryption).

4.3. [EE3] Optionality of IDPProvidedNameIdentifier

4.3.1. Summary

4.3.2. Resolution

1. Line 532,533 of [1] change to read:
   The type **`SubjectType`**, extended from **`saml:SubjectType`**, is used to include the optional `<IDPProvidedNameIdentifier>` element in subject statements.

2. Lines 765-769 of [1] change to read:
   When including a Principal's federated identity in the response, the `<Subject>` element MUST include a `<saml:NameIdentifier>`. If the service provider or affiliation group has set its own identifier for the Principal, then the `<saml:NameIdentifier>` MUST be set to that value. The identifier set by the identity provider MUST then be included in the `<IDPProvidedNameIdentifier>`.

3. line 594 of [1] changes to read:
   ```
   <xs:element ref="IDPProvidedNameIdentifier" minOccurs="0"/>
   ```

4.4. [EE4] Incorrect URI references for Authentication Context

4.4.1. Summary

In two places, the previous version of Liberty Authentication Context is referenced.

4.4.2. Resolution

1. Line 635 of [1] changes to read:
   `<saml:AuthenticationStatement>` MUST be *urn:liberty:ac:2003-08*

2. Line 638 of [1] changes to read:
   urn:liberty:ac:2003-08, the service provider MUST refer to the `<AuthnContext>` element and ignore the `saml:AuthenticationMethod` attribute.

4.5. [EE5] Use of the resource URL in RelayState

4.5.1. Summary

4.5.2. Resolution

1. Insert after line 717 of [1]:
   Optionally, a `<RelayState>` with a value understood by mutual agreement between the identity provider and service provider so that the service provider knows how to handle subsequent interactions with the Principal after SSO. This MAY be the URL of a resource at the service provider.

4.6. [EE6] Use of substitutionGroup AssertionType causes instance validation errors when using saml:Advice

4.6.1. Summary

The saml:Advice element contains a choice element which can contain a saml:Assertion, or an <any namespace="##other"...>. As the lib:Assertion is defined as a member of the substitutionGroup for saml:Assertion, this allows a non-deterministic document instance to be created which will fail validation - if an Advice element contains a lib:Assertion, then this may match either the saml:Assertion (because of the substitutionGroup) or the <any namespace="##other"...>.

4.6.2. Resolution

Modify line 521 of [1] to read:

```
<xs:element name="Assertion" type="AssertionType"/>
```

4.7. [EE7] URL-encoding of the Consent Attribute

4.7.1. Summary

The consent attribute was previously added to several messages. However, it was not added to the list of elements present in those messages that should be URL-encoded when using the HTTP redirect profiles.

4.7.2. Resolution

1. Insert after line 464 of [2]:
   ```
   consent="[consent]"
   ```

2. Modify line 489 of [2] (see [EE9] for a further change to this line):
   ```
   AuthnContextComparison, RelayState, ProxyCount, consent.
   ```

3. Modify line 494 of [2]:
   ```
   &consent=urn:liberty:consent:obtained&ProviderID=http%3A%2F%2Fsp.example.com%2Fliberty%
   2F&ForceAuthn=true
   ```

4. Insert after line 511 of [2]:
   ```
   consent="[consent]"
   ```

5. Modify line 523 of [2] to read:
   ```
   NameIdentifier, consent.
   ```

6. Insert after line 531 of [2]:
   ```
   consent="[consent]"
   ```

7. Modify line 546 of [2] to read:
   ```
   SessionIndex, RelayState, consent.
   ```

4.8. [EE8] Cookie encoding

4.8.1. Summary

There are some issues of interoperability when working with the Identity Provider Introduction Profile, which uses a common-domain cookie.

4.8.2. Resolution

1. Insert after line 1904 of [2]:
   The cookie SHOULD be URL-encoded.
   Cookie syntax should be in accordance with [RFC2965] or [NetscapeCookie].
   D1 The cookie MAY be either session or persistent. This choice may be made within an identity federation network, but should apply uniformly to all providers in the network (see [4] for more details on cookies).

2. Delete lines 1905, 1906 of [2]

4.9. [EE9] URL-encoding the IDPList element

4.9.1. Summary

The IDPList element contains a repeating group of elements, which makes it difficult to URL-encode.

4.9.2. Resolution

1. Insert after line 481 of [2]:

```
<lib:IDPList>
   <lib:IDPEntries>[IDPEntries]</lib:IDPEntries>
   <lib:GetComplete>[GetComplete]</lib:GetComplete>
</lib:IDPList>
```

2. Modify line 489 of [2] to read:
   `AuthnContextComparison, RelayState, ProxyCount, IDPEntries, GetComplete, consent.`

3. Insert after line 489 of [2]:
   The `<IDPEntries>` element may contain multiple `<IDPEntry>` elements, each of which may contain multiple pieces of data (`<ProviderID>`, `<ProviderName>` and `<Loc>`). The `<IDPEntries>` element MUST be URL-encoded by taking only the `<ProviderID>` element from each individual `<IDPEntry>` element, and concatenating them in a space-separated string, as in the following example:

```
... &IDPEntries=http%3A%2F%2Fidp1.com%2Fliberty%2F%20http%3A%2F%2Fidp2.com%2Fliberty%2F ...
```

   D1 The recipient of such a URL-encoded list of `<ProviderID>` elements may obtain the remainder of the information present in the original `<IDPEntry>` by accessing metadata for the individual providers referenced in the URL-encoded list.

4. Add references for the following:
   [NetscapeCookie] eds. "Persistent Client State HTTP Cookies," Netscape *http://wp.netscape.com/newsref/std/cookie_spec.html*

   [RFC2965] Kristol, D., Montulli, L., eds. (October 2000). "HTTP State Management Mechanism," RFC 2965., Internet Engineering Task Force *http://www.ietf.org/rfc/rfc2965.txt [October 2000]*.

4.10. [EE10] Missing Security Considerations

4.10.1. Summary

The entire 'Security Considerations' section was omitted due to editorial error, from the public document

4.10.2. Resolution

Insert the following text after line 2028 of [2]:

4.10.2.1. Security Considerations

4.10.2.1.1. Introduction

This section describes security considerations associated with Liberty protocols for identity federation, single sign-on, federation termination, and single logout.

Liberty protocols, schemas, bindings, and profiles inherit and use extensively the SAML protocols. Therefore, the security considerations published along with the SAML specification have direct relevance (see [SAMLCore], [SAMLBind], and [SAMLSec]). Throughout this section if, for any reason, a specific consideration or countermeasure does not apply or differs, notice of this fact is made; and a description of alternatives is supplied, where possible.

4.10.2.1.2. General Requirements

4.10.2.1.2.1. Security of SSL and TLS

SSL and TLS utilize a suite of possible cipher suites. The security of the SSL or TLS session depends on the chosen cipher suite. An entity (that is, a user agent, service provider, or identity provider) that terminates an SSL or TLS connection needs to offer (or accept) suitable cipher suites during the handshake. The following list of TLS 1.0 cipher suites (or their SSL 3.0 equivalent) is recommended.

- `TLS_RSA_WITH_RC4_128_SHA`
- `TLS_RSA_WITH_3DES_EDE_CBC_SHA`
- `TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA`

The above list is not exhaustive. The recommended cipher suites are among the most commonly used. Note: New cipher suites are added as they are standardized and should be considered for inclusion if they have sufficiently strong security properties. For example, it is anticipated that the AES-based cipher suites being standardized in the IETF will be widely adopted and deployed.

4.10.2.1.2.2. Security Implementation

The suitable implementation of security protocols is necessary to maintain the security of a system, including

- Secure random or pseudo-random number generation
- Secure storage

4.10.2.1.3. Classification of Threats

4.10.2.1.3.1. Threat Model

For an analysis of threat classifications, an Internet threat model has been used. In other words, the threat model assumes that intermediary and end-systems participating in Liberty protocol exchanges have not been compromised. However, where possible, the consequences and containment properties of a compromised system entity are described and countermeasures are suggested to bolster the security posture so that the exposure from a security breach is minimized.

Given the nature of the Internet, the assumption is made that deployment is across the global Internet and, therefore, crosses multiple administrative boundaries. Thus, an assumption is also made that the adversary has the capacity to engage in both passive and active attacks (see 4.3.3).

4.10.2.1.3.2. Rogue and Spurious Entities

Attackers may be classified based on their capabilities and the roles that they play in launching attacks on a Liberty system as follows:

- **Rogue Entities:** Entities that misuse their privileges. The rogue actors may be Principals, user agents, service providers, or identity providers. A rogue Principal is a legitimate participant who attempts to escalate its privileges or masquerade as another system Principal. A rogue user agent may, for instance, misuse the relationships between its associated Principals and an identity provider to launch certain attacks. Similarly, a rogue service provider may be able to exploit the relationship that it has either with a Principal or with an identity provider to launch certain attacks.

- **Spurious Entities:** Entities that masquerade as a legitimate entity or are completely unknown to the system. The spurious actors include Principals, user agents (i.e., user agents without associated legitimate Liberty Principals), service providers, or identity providers. A spurious service provider may, for instance, pretend to be a service provider that has a legitimate relationship with an identity provider. Similarly, a spurious Principal may be one that pretends to be a legitimate Principal that has a relationship with either a service provider or an identity provider.

4.10.2.1.3.3. Active and Passive Attackers

Both rogue and spurious entities may launch active or passive attacks on the system. In a passive attack the attacker does not inject traffic or modify traffic in any way. Such an attacker usually passively monitors the traffic flow, and the information that is obtained in that flow may be used at a later time. An active attacker, on the other hand, is capable of modifying existing traffic as well as injecting new traffic into the system.

4.10.2.1.3.4. Scenarios

The following scenarios describe possible attacks:

- **Collusion: The secret cooperation between two or more Liberty entities to launch an attack, for example,**

  - Collusion between Principal and service provider
  - Collusion between Principal and identity provider
  - Collusion between identity provider and service provider

- Collusion among two or more Principals
- Collusion between two or more service providers
- Collusion between two or more identity providers

- **Denial-of-Service Attacks:** The prevention of authorized access to a system resource or the delaying of system operations and functions.
- **Man-in-the-Middle Attacks:** A form of active wiretapping attack in which the attacker intercepts and selectively modifies communicated data to masquerade as one or more of the entities involved in a communication association.
- **Replay Attacks:** An attack in which a valid data transmission is maliciously or fraudulently repeated, either by the originator or by an adversary who intercepts the data and retransmits it, possibly as part of a masquerade attack.
- **Session Hijacking:** A form of active wiretapping in which the attacker seizes control of a previously established communication association.

4.10.2.1.4. Threat Scenarios and Countermeasures

In this section, threats that may apply to all the Liberty profiles are considered first. Threats that are specific to individual profiles are then considered. In each discussion the threat is described as well as the countermeasures that exist in the profile or the additional countermeasures that may be implemented to mitigate the threat.

4.10.2.1.4.1. Common Threats for All Profiles

Threat: Request messages sent in cleartext

Description: Most profile protocol exchanges do not mandate that all exchanges commence over a secure communication channel. This lack of transport security potentially exposes requests and responses to both passive and active attacks.

One obvious manifestation is when the initial contact is not over a secure transport and the Liberty profile begins to exchange messages by carrying the request message back to the user agent in the location header of a redirect.

Another such manifestation could be a request or response message which carries a URI that may be resolved on a subsequent exchange, for instance lib:AuthnContextClassRef. If this URI were to specify a less or insecure transport, then the exchange may be vulnerable to the types of attacks described above.

Countermeasure: Ensure that points of entry to Liberty protocol exchanges utilize the https URL `<scheme>` and that all interactions for that profile consistently exchange messages over `https`.

Threat: Malicious redirects into identity or service provider targets

Description: A spurious entity could issue a redirect to a user agent so that the user agent would access a resource that disrupts single sign-on. For example, an attacker could redirect the user agent to a logout resource of a service provider causing the Principal to be logged out of all existing authentication sessions.

Countermeasure: Access to resources that produce side effects could be specified with a transient qualifier that must correspond to the current authentication session. Alternatively, a confirmation dialog could be interposed that relies on a transient qualifier with similar semantics.

Threat: Relay state tampering or fabrication

Description: Some of the messages may carry a `<lib:RelayState>` element, which is recommended to be integrity-protected by the producer and optionally confidentiality-protected. If these practices are not followed, an adversary

could trigger unwanted side effects. In addition, by not confidentiality-protecting the value of this element, a legitimate system entity could inadvertently expose information to the identity provider or a passive attacker.

Countermeasure: Follow the recommended practice of confidentiality- and integrity-protecting the `<lib:RelayState>` data. Note: Because the value of this element is both produced and consumed by the same system entity, symmetric cryptographic primitives could be utilized.

4.10.2.1.4.2. Single Sign-On and Federation

4.10.2.1.4.2.1. Common Interactions for All Single Sign-On and Federation Profiles

Threat: `<lib:AuthnRequest>` sent over insecure channel

Description: It is recommended that the initial exchange to access the intersite transfer service be conducted over a TLS-secured transport. Not following this recommendation can expose the exchange to both passive and active attacks.

Countermeasure: Deploy the intersite transfer service under an https scheme.

Threat: Unsigned `<lib:AuthnRequest>` message

Description: The signature element of an `<lib:AuthnRequest>` is optional and thus the absence of the signature could pose a threat to the identity provider or even the targeted service provider. For example, a spurious system entity could generate an unsigned `<lib:AuthnRequest>` and redirect the user agent to the identity provider. The identity provider must then consume resources.

Countermeasure: Sign the `<lib:AuthnRequest>`. The IDP can also verify the identity of the Principal in the absence of a signed request.

Threat: Replay of an authentication assertion

Description: After obtaining a valid assertion from an identity provider, either legitimately or surreptitiously, the entity replays the assertion to the Service at a later time. A digital signature must cover the entire assertion, thus elements within the assertion cannot be corrupted without detection during the mandatory verification step. However, it is possible to fabricate an `<lib:AuthnResponse>` with the valid assertion.

Countermeasure: The issuer should sign `<lib:AuthnResponse>` messages. Signing binds the `<samlp:IssueInstant>` of the response message to the assertion it contains. This binding accords the relying party the opportunity to temporally judge the response. Additionally, a valid signature over the response binds the `<samlp:InResponseTo>` element to the corresponding `<lib:AuthnRequest>`. (Specifying a short period that the authentication assertion can be relied upon will minimize, but not mitigate this threat. Binding the `<lib:AssertionId>` to the request `<samlp:InResponseTo>` element may also be handy.)

Threat: Fabricated `<lib:AuthnResponse>` denial of service

Description: An attacker captures the `<samlp:RequestID>` sent in an `<lib:AuthnRequest>` message by a service provider to an identity provider, and sends several spurious `<lib:AuthnResponse>` messages to the service provider with the same `<samlp:InResponseTo>`. Because the `<samlp:InResponseTo>` matches a `<samlp:RequestID>` that the service provider had used, the service provider goes through the process of validating the signature in the message. Thus, it is subject to a denial of service attack.

Countermeasure: A secure communication channel should be established before transferring requests and responses.

Threat: Collusion between two Principals

Description: After getting an artifact or <lib:AuthnResponse> in step 6 (see 3.2.1), a legitimate Principal A could pass this artifact or <lib:AuthnResponse> on to another Principal, B. Principal B is now able to use the artifact or <lib:AuthnResponse>, while the actual authentication happened via Principal A.

Countermeasure: Implementations where this threat is a concern MUST use the <saml:AuthenticationLocality> in the authentication statement. The IP address that Principal B uses would be different from the IP address within the <saml:AuthenticationLocality>. This countermeasure may not suffice when the user agent is behind a firewall or proxy server. IP spoofing may also circumvent this countermeasure.

Threat: Stolen artifact and subsequent Principal impersonation

Description: See Section 4.1.1.9.1 in [SAMLBind]

Countermeasure: Identity providers MUST enforce a policy of one-time retrieval of the assertion corresponding to an artifact so that a stolen artifact can be used only once. Implementations where this threat is a concern MUST use the <saml:AuthenticationLocality> in the authentication statement. The IP address of a spurious user agent that attempts to use the stolen artifact would be different from IP address within the <saml:AuthenticationLocality>. The service provider may then be able to detect that the IP addresses differ. This countermeasure may not suffice when the user agent is behind a firewall or proxy server. IP address spoofing may also circumvent this countermeasure.

Threat: Stolen assertion and subsequent Principal impersonation

Description: See Section 4.1.1.9.1 in [SAMLBind]

Countermeasure: Refer to the previous threat for requirements.

Threat: Rogue service provider uses artifact or assertion to impersonate Principal at a different service provider

Description: Because the <lib:AuthnResponse> contains the <lib:ProviderID>, this threat is not possible.

Countermeasure: None

Threat: Rogue identity provider impersonates Principal at a service provider

Description: Because the Principal trusts the identity provider, it is assumed that the identity provider does not misuse the Principal's trust.

Countermeasure: None

Threat: Rogue user attempts to impersonate currently logged-in legitimate Principal and thereby gain access to protected resources.

Description: Once a Principal is successfully logged into an identity provider, subsequent <AuthnRequest> messages from different service providers concerning that Principal will not necessarily cause the Principal to be reauthenticated. Principals must, however, be authenticated unless the identity provider can determine that an <AuthnRequest> is associated not only with the Principal's identity, but also with a validly authenticated identity provider session for that Principal.

Countermeasure: In implementations where this threat is a concern, identity providers MUST maintain state information concerning active sessions, and MUST validate the correspondence between an <AuthnRequest> and an active session before issuing an <AuthnResponse> without first authenticating the Principal. Cookies posted by identity providers MAY be used to support this validation process, though Liberty does not mandate a cookie-based approach.

4.10.2.1.4.2.2. Liberty-Enabled Client and Proxy Profile

Threat: Intercepted <lib:AuthnRequestEnvelope> and <lib:AuthnResponse> and subsequent Principal impersonation.

Description: A spurious system entity can interject itself as a man-in-the-middle (MITM) between the user agent (LECP) and a legitimate service provider, where it acts in the service provider role in interactions with the LECP, and in the user agent role in interactions with the legitimate service provider. In this way, as a first step, the MITM is able to intercept the service provider's <lib:AuthnRequestEnvelope> (step 3 of section 3.2.5) and substitute any URL of its choosing for the <lib:AssertionConsumerServiceURL> value before forwarding the <lib:AuthnRequestEnvelope> on to the LECP. Typically, the MITM will insert a URL value that points back to itself. Then, if the LECP subsequently receives a <lib:AuthnResponseEnvelope> from the identity provider (step 6 in section 3.2.5) and subsequently sends the contained <lib:AuthnResponse> to the <lib:AssertionConsumerServiceURL> received from the MITM, the MITM will be able to masquerade as the Principal at the legitimate service provider.

Countermeasure: The identity provider specifies to the LECP the address to which the LECP must send the <lib:AuthnResponse>. The <lib:AssertionConsumerServiceURL> in the <lib:AuthnResponseEnvelope> element is for this purpose. This URL value is among the metadata that identity and service providers must exchange in the process of establishing their operational relationship (see sections 3.1 and 3.1.3).

4.10.2.1.4.2.3. Federation

Threat: Collusion among service providers can violate privacy of the Principal

Description: When a group of service providers collude to share the <lib:IDPProvidedNameIdentifier> of a Principal, they can track and in general compromise the privacy of the principal. More generally, this threat exists for any common data (e.g. phone number) shared by rogue system entities.

Countermeasure: The <lib:IDPProvidedNameIdentifier> is required to be unique for each identity provider to service provider relationship. However, this requirement does not eliminate the threat when there are rogue participants under the Principal's identity federation. The only protection is for Principals to be cautious when they choose service providers and understand their privacy policies.

Threat: Poorly generated name identifiers may compromise privacy

Description: The federation protocol mandates that the <lib:NameIdentifier> elements be unique within a Principal's federated identities. The name identifiers exchanged are pseudonyms and, to maintain the privacy of the Principal, should be resistant to guessing or derivation attacks.

Countermeasure: Name identifiers should be constructed using pseudo-random values that have no discernable correspondence with the Principal's identifier (or name) used by the entity that generates the name identifier.

4.10.2.1.4.3. Name Registration

No known threats.

4.10.2.1.4.4. Federation Termination: HTTP-Redirect-Based Profile

Threat: Attacker can monitor and disrupt termination

Description: During the initial steps, a passive attacker can collect the <lib:FederationTerminationNotification> information when it is issued in the redirect. This threat is possible because the first and second steps are not required to use https as the URL scheme. An active attacker may be able to intercept and modify the message conveyed in step 2 because the digital signature only covers a portion of the message. This initial exchange also exposes the name identifier. Exposing these data poses a privacy threat.

Countermeasure: All exchanges should be conducted over a secure transport such as SSL or TLS.

4.10.2.1.4.5. Single Logout: HTTP-Redirect-Based Profile

Threat: Passive attacker can collect a Principal's name identifier

Description: During the initial steps, a passive attacker can collect the `<lib:LogoutRequest>` information when it is issued in the redirect. Exposing these data poses a privacy threat.

Countermeasure: All exchanges should be conducted over a secure transport such as SSL or TLS.

Threat: Unsigned `<lib:LogoutRequest>` message

Description: An Unsigned `<lib:LogoutRequest>` could be injected by a spurious system entity thus denying service to the Principal. Assuming that the NameIdentifier can be deduced or derived then it is conceivable that the user agent could be directed to deliver a fabricated `<lib:LogoutRequest>` message.

Countermeasure: Sign the `<lib:LogoutRequest>` message. The identity provider can also verify the identity of a Principal in the absence of a signed request.

4.10.2.1.4.6. Identity Provider Introduction

No known threats.

4.11. [EE11] AuthenticatingAuthority maxOccurs

4.11.1. Summary

The AuthenticatingAuthority element should have maxOccurs="unbounded" in the AuthenticationContextStatement-Type.

4.11.2. Resolution

1. line 670 of [4] changes to read:
   ```
   <xs:element ref="AuthenticatingAuthority" minOccurs="0" maxOccurs="unbounded"/>
   ```

4.12. [EE12] Specifying SAML Version

4.12.1. Summary

The version of SAML is ambiguous in several instances.

4.12.2. Resolution

1. line 59 of [1] changes to read:
   The prefix saml: stands for the SAML 1.1 assertion namespace (urn:oasis:names:tc:SAML:1.0:assertion).

2. line 60 of [1] changes to read:
   The prefix samlp: stands for the SAML 1.1 protocol namespace (urn:oasis:names:tc:SAML:1.0:protocol).

3. line 160 to 163 of [1] changes to read:
   Most protocol messages and assertions used in the protocols defined in this specification are defined in the Liberty ID-FF namespace, and are therefore assigned the 1.2 version designation. A notable exception is when the SSO artifact profile is used, in which case pure SAML 1.1 Request and Response, elements are exchanged when dereferencing the artifact. These messages have a 1.1 version designation because they are SAML 1.1 protocol messages. In contrast, the Assertion element inside the message is a Liberty assertion and therefore carries the 1.2 designation.

4. line 168 of [1] changes to read:
   If the element or its type is in a SAML 1.1 namespace (urn:oasis:names:tc:SAML:1.0:assertion or urn:oasis:names:tc:SAML:1.0:protocol), then the values MUST be 1 and 1 respectively.

4.13. [EE13] Additional Values for Consent Attribute

4.13.1. Summary

Additional values for the Consent Attribute need to be added.

4.13.2. Resolution

1. line 252 of [1] changes to read:
   The following values are defined for this attribute:

2. lines appended after 252 of [1] to read:
   * urn:liberty:consent:obtained:prior indicates that a user's consent has been obtained by the sender of the message at some point prior to this action. If the message sender uses this value, they SHOULD sign the message such that the signature covers this attribute.
   * urn:liberty:consent:obtained:current:implicit indicates that a user's consent has been implicitly obtained by the sender of the message during this action as part of a broader indication of consent. If the message sender uses this value, they SHOULD sign the message such that the signature covers this attribute. Implicit consent is typically more proximal to the action in time and presentation than prior consent, such as part of a session of activities.
   D1 * urn:liberty:consent:obtained:current:explicit indicates that a user's consent has been explicitly obtained by the sender of the message during this action. If the message sender uses this value, they SHOULD sign the message such that the signature covers this attribute.

4.14. [EE14] Incorrect section reference

4.14.1. Summary

Incorrect section reference in [1].

4.14.2. Resolution

1. line 884-885 of [1] changes to read:
   Specify whether the <AuthnRequest> element sent from the service provider to the identity provider via the intermediary is wrapped in an <AuthnRequestEnvelope>. See Section 3.2.3.

2. line 886-887 of [1] changes to read:
   Specify whether the <AuthnResponse> element sent from the identity provider to the service provider via the intermediary is wrapped in an <AuthnResponseEnvelope>. See Section 3.2.4.

4.15. [EE15] Added Maximum as allowed AuthnContextComparison

4.15.1. Summary

Need to allow an SP to specify a maximum authentication context, in addition to equal, minimum, or better.

4.15.2. Resolution

1. line 346 of [1] changes to read:
   If set to "exact", then the identity provider is asked to match at least one of the specified <AuthnContext> elements exactly. This can also be set to "minimum", which asks that the identity provider use a context that he feels is at least as good as any specified in the <AuthnContext> or "better", which means that they can use any context better than any that were supplied, or "maximum", which means to use a context that is at most as strong as any specified. If not specified, this is assumed to be "exact".

2. Insert after line 802 of [1]:
   If <AuthnContextComparison> is specified and set to maximum, then the resulting authentication statement in the assertion (if any) MUST be as strong as possible (as deemed by the identity provider) without exceeding the strength of at least one of the authentication contexts specified.

4.16. [EE16] Incorrect indication of ID-FF version 1.2 support in LECP profile

4.16.1. Summary

The URI that implementers of the LECP profile use to indicate version 1.2 message support is incorrect, and inconsistent with other indications of ID-FF version 1.2 support.

4.16.2. Resolution

Line 1031 of [2] should read:

with the URI `urn:liberty:iff:2003-08` if it supports version 1.2 requests and knows that the identity providers

4.17. [EE17] Deprecated SAML 1.0 artifact URN used.

4.17.1. Summary

The SAML 1.1 deprecated URN urn:oasis:names:tc:SAML:1.0:cm:artifact-01 was used instead of urn:oasis:names:tc:SAML:1.0:cm:artifact.

4.17.2. Resolution

Line 275 of [3] should read:

```
urn:oasis:names:tc:SAML:1.0:cm:artifact
```

5. Substantive Errata

5.1. [SE1] Single Logout Inconsistencies

5.1.1. Summary

There are a number of issues related to the concept of session, and the ability for a service provider to properly handle a single logout request.

5.1.2. Resolution

1. Modify lines 472-475 of [1] to read:
   Identity providers MUST include a `SessionIndex` attribute in resulting authentication statements, which is used to aid the identity provider in managing multiple sessions with the Principal. Subsequent messages from the service provider to the identity provider that are session-dependent MUST include this `SessionIndex` attribute.

2. Modify line 627 of [1] to read:
   `SessionIndex` [Required]

3. Modify line 650 of [1] to read:
   `<xs:attribute name="SessionIndex" type="xs:string" use="required"/>`

4. lines appended after 1313 of [1] to read:
   The Principal may have established authenticated sessions both with the identity provider, and individual service providers, based on authentication assertions supplied by the identity provider.

5. line 1314-1315 of [1] changes to read:
   When the Principal invokes the single logout process at a service provider, the service provider MUST send a <LogoutRequest> message to the identity provider that provided the authentication service related to that session at the service provider.

6. line 1316-1319 of [1] changes to read:
   When either the Principal invokes a logout at the identity provider or a service provider sends a logout request to the identity provider specifying that Principal, the identity provider MUST send a <LogoutRequest> message to each service provider to which it provided authentication assertions under its current session with the Principal, with the exception of the service provider that sent the <LogoutRequest> message to the identity provider.

7. lines appended after 1319 of [1] to read:
   The following rules apply to identity providers involved in authentication proxying:

8. line 1327-1330 of [1] changes to read:
   The <LogoutRequest> message indicates to the message receiver that a Principal's session was terminated. The message includes an optional SessionIndex element that MUST be specified if and only if the authentication statement in the assertion that the service provider used in establishing the session with the Principal. This message SHOULD be signed.

9. Insert after line 1353 of [1]:
   `NotOnOrAfter` [Optional]
   This attribute is used to specify the time instant at which the recipient may expire a logout request from the sender. The sender MAY set this value equal to the `NotOnOrAfter` attribute of a previously-supplied assertion, if one was specified.

10. Modify line 1363 of [1] to read:
    `<xs:element name="SessionIndex" type="xs:string" minOccurs="0" maxOccurs="unbounded"/>`

11. Insert after line 1366 of [1]:
```
<xs:attribute name="NotOnOrAfter" type="xs:dateTime" minOccurs="0"/>
```

12. line 1431 of [1] changes to read:
Terminate the Principal's current session as specified by the <saml:NameIdentifier> element, and any <SessionIndex> present in the logout request message.

13. Insert after line 1431 of [1]:
When constructing a logout request message, the identity provider MUST set the value of the *NotOnOrAfter* attribute of the message to a time value, indicating an expiration time for the message. See ??? for more details.

14. Modify lines 1438-1441 of [1] to read:
When the service provider receives the `<LogoutRequest>` message, the service provider MUST validate the identity provider's signature contained in the `<ds:Signature>` element. If the signature is that of the identity provider that provided the authentication for the Principal's current session, the service provider MUST invalidate the Principal's session(s) referred to by the `<saml:NameIdentifier>` element, and any `SessionIndex` elements supplied in the message.

15. Insert after line 1441 of [1]:
The service provider MUST apply the logout request message to any assertion that meets the following conditions, *even if the assertion arrives after the logout request*:

    1. The `SessionIndex` of the assertion matches one specified in the logout request.
    2. The assertion would otherwise be valid
    3. The logout request has not yet expired (determined by examining the `NotOnOrAfter` attribute on the message).

5.2. [SE2] ID-FF 1.2 Backward Compatibility

5.2.1. Summary

One problem with this is the change between v1.1 and v1.2 in the usage of the saml:NameIdentifier NameQualifier and Format attributes. Imagine an SP that implements both v1.1 and v1.2, that does a RegisterNameIdentifier using the v1.1 protocol and includes values for NameQualifier and Format. If the SP (or IDP) subsequently initiates a Logout or FedTerm using the v1.2 protocol, the request will have a ProviderID/AffiliationID in the NameQualifier and a federation policy value in the Format. The recipient's attempt to match the NameIdentifier to an existing federation would then fail.

5.2.2. Resolution

1. Append after 248 of [1]:
3.1.11.1 Deprecation of ID-FF 1.1 Name Identifier Practices
Previous ID-FF specifications were not explicit about the use (or lack of use) of the SAML Format and NameQualifier XML attributes in Liberty <NameIdentifier> elements. Whereas this specification explicitly profiles the use of those attributes in a strict fashion, previous implementations have been found to make use of these attributes for their own purposes. Convention has been for implementations to reproduce whatever values they received from other implementations.
D1 This specification deprecates the non-standard use of these attributes and implementations MUST follow the rules laid out in this specification in communicating all new identity federations created between version 1.2 providers. However, 1.2 providers SHOULD maintain interoperability with older providers by honoring and reproducing non-standard values for these attributes when communicating with older providers regarding

existing identity federations. But, it is RECOMMENDED that 1.2 implementations provide tools by which these older identity federations can be "upgraded" to the behavior outlined in this specification by means of the Name Registration Protocol when both parties to a federation support the 1.2 specification (see Section 3.3). Finally, when establishing new federated identifiers with 1.1 or earlier providers, 1.2 implementations MUST omit the Format and NameQualifier attributes in the 1.1 protocol messages they send. This is possible because all such identifiers have federated semantics and do not involve affiliations. Thus, the presence of the format values in this specification unambigiously indicates a 1.2 federation and the absence of those values indicates a pre-1.2 federation in all protocol messages.

2. Modify lines 455-456 of [1] to read:
In either case, the Format attribute MUST either be urn:liberty:iff:nameid:federated, or in the case of a non-1.2 identity federation may contain a non-standard Format value.

3. Modify lines 542-544 of [1] to read:
Format [Required Except for Pre-1.2 Federations]
Indicates the format, semantics, and processing rules associated with the identifier. If this attribute is omitted, or if a value other than those listed below is used, then the value urn:liberty:iff:nameid:federated is effectively assumed, but no other 1.2 semantics are to be implied, including use of NameQualifier as described below. That is, such an identifier is assumed to be a pre-1.2 federated identifier and the NameQualifier attribute, if present at all, is to be treated as literal data. Otherwise, for federations established between 1.2 providers, one of these four values MUST be present:

4. Append after line 558 of [1]:
NameQualifier [Optional]
Generally used to indicate the unique identifier of the service provider or affiliation group for or by whom the name identifier was created. Unless otherwise specified, the service provider's or affiliation group's unique identifier MUST be placed in this attribute to disambiguate the identifier from any other identity federations the principal may have with the identity provider.
D1 An issuer MAY omit this attribute if the containing element is in the subject of a statement in an assertion that contains a single <saml:Audience> equal to the unique identifier of the value that this attribute would have had. This typically means that an assertion issued during SSO to a service provider that is not acting as part of an affiliation group can omit this attribute. The above interpretation of this attribute applies ONLY if the Format attribute contains one of the four values listed above. Otherwise, the identifier MUST be assumed to represent a pre-1.2 identity federation and MUST be treated only as literal data.

5. Modify lines 574-575 of [1] to read:
MUST contain a value of urn:liberty:iff:nameid:federated (of the formats defined in this specification, only federated name identifiers sometimes require encryption), excepting that Pre-1.2 federated identifiers being encrypted MAY omit this attribute or use another unspecified value, as described earlier.

6. Modify lines 585 of [1] to read:
Nonce [Optional]

7. Modify lines 586-588 of [1] to read:
MAY contain a unique pseudorandom value. The probability of two randomly chosen values being identical MUST be less than or equal to 2^-128 and SHOULD be less than or equal to 2^-160. Receiving providers MAY use this value to prevent unacceptable reuse of the same encrypted identifier.

8. line 769 of [1] changes to read:
If a one-time identifier is being used, or if the service provider or affiliation group has not set its own identifier for the Principal, then the <saml:NameIdentifier> MUST contain the identifier set by the identity provider for the Principal. The <IDPProvidedNameIdentifier> SHOULD be omitted in this case, since it would contain identical information.

9. Modify lines 1091-1097 of [1] to read:
During federation, the identity provider generates an opaque value that serves as the initial name identifier that both the service provider and the identity provider use in referring to the Principal when communicating with each other. This name identifier is termed the <IDPProvidedNameIdentifier>.
Subsequent to federation, the service provider MAY register a different opaque value with the identity provider. This opaque value is termed the <SPProvidedNameIdentifier>. Until the service provider registers a different name, the identity provider will use <IDPProvidedNameIdentifier> to refer to the Principal when communicating with the service provider.

10. Append after line 1108 of [1] to read:
* Only federated identifiers (and by extension pre-1.2 identifiers which are always federated) can be replaced and set with this protocol. Encrypted or one-time identifiers MUST NOT be used.
* The Format attribute in the newly registered identifier element MUST be urn:liberty:iff:nameid:federated. The Format attribute in the <OldProvidedNameIdentifier> element MUST either be urn:liberty:iff:nameid:federated, or MAY be omitted or set to a non-standard value if a pre-1.2 federation is being updated. This also applies to the <SPProvidedNameIdentifier> sent by an IdP or the <IDPProvidedNameIdentifier> sent by an SP if referencing a pre-1.2 federation.
D1 * Similarly, the NameQualifier attribute in the newly registered identifier MUST be the service provider's unique identifier, an affiliation group's unique identifier, or omitted (implying the service provider's identifier). The other elements either follow the same rules, or if a pre-1.2 federation, MUST be treated as literal data.

11. Modify lines 1232-1236 of [1] to read:
In all of the name identifier elements in the request and response messages of this protocol, if the Principal's identity federation is between the identity provider and an affiliation group in which the service provider is a member, then the NameQualifier attribute MUST contain the unique identifier of the affiliation group. Otherwise, it MUST contain the unique identifier of the service provider. This attribute MUST be used by the providers to identify the specific identity federation to be modified. The exception as noted previously is if the old identifier(s) represent a pre-1.2 federation being updated with the request.

12. Modify lines 1306-1309 of [1] to read:
If the Principal's identity federation was between the identity provider and an affiliation group in which the service provider is a member, then the NameQualifier attribute MUST contain the unique identifier of the affiliation group. Otherwise, it MUST contain the unique identifier of the service provider. This attribute MUST be used by the providers to identify the specific identity federation being terminated. The exception is a pre-1.2 federated identifier, which can be recognized by a missing or non-standard Format attribute.

13. Modify lines 1418-1421 of [1] to read:
If the Principal's identity federation is between the identity provider and an affiliation group in which the service provider is a member, then the NameQualifier attribute MUST contain the unique identifier of the affiliation group. Otherwise, it MUST contain the unique identifier of the service provider. This attribute MUST be used by the providers to identify the specific identity federation of the Principal who is logging out. The exception is a pre-1.2 federated identifier, which can be recognized by a missing or non-standard Format attribute.

14. Modify line 1497 of [1] to read:
The responding provider MUST return a <NameIdentifierMappingResponse> message if the signature is valid.

5.3. [SE3] RNI Protocol Changes

5.3.1. Summary

Need to clarify the defaults for NameIdentifier, and change existing specifications to match the RNI protocol.

5.3.2. Resolution

1. Modify line 1122 of [1] to read:
   For an IdP-initiated request, the new name identifier the service provider should use when communicating with the identity provider. For an SP-initiated request, the original name identifier established by the IdP for the SP to use when communicatng with it.

2. Modify lines 1123-1124 of [1] to read:
   SPProvidedNameIdentifier [Optional]
   For an SP-initiated request, this is required, and is the new name identifier the identity provider should use when communicating to the service provider. For an IdP- initiated request, this is the name identifier established by the SP for the IdP to use when communicating with it, or if none exists, MUST be omitted.

3. Modify lines 1126-1130 of [1] to read:
   In the case of either provider choosing to request a change of provided name identifiers, this element holds the previous version set by that provider. For a service provider making their first name change following federation, the <OldProvidedNameIdentifier> will contain the current <IDPProvidedNameIdentifier>.

4. Modify lines 1222-1225 of [1] to read:
   If the request includes an <IDPProvidedNameIdentifier> for which no federation exists between the service provider and the identity provider, the identity provider MUST respond with a <samlp:Status> element containing a second-level <samlp:StatusCode> of lib:FederationDoesNotExist. Otherwise, the identity provider MUST use <SPProvidedNameIdentifier> when subsequently communicating to the service provider regarding this Principal.

5.4. [SE4] Metadata changes

5.4.1. Summary

Updates needed for the Metadata specification.

5.4.2. Resolution

1. Modify lines 86-88 of [3] to read:
   As a single instance document describing an affiliation (a set of entities) collectively identified as providerID (located with the EntityDescriptor parent), which in turn enumerates each entity member by it's own providerID and maintained by an entity referenced by its affiliationOwnerID. Each member's metadata is then located by the methods provided in this specification.

2. Modify lines 358-363 of [3] to read:
   protocolSupportEnumeration [Required]
   Describes the protocol release supported by the entity described by providerID. NMTOKENS type allows the enumeration of a set of Liberty ID-FF protocol releases which the interfaces described within MUST support. The datatype of the tokens MUST be URNs (presently http://projectliberty.org/schemas/core/2002/12 for release ID-FF 1.1 and urn:liberty:iff:2003-08 for release ID-FF 1.2). Subsequent releases of ID-FF shall express protocol support using the defined nameSpace attribute of the corresponding ID-FF schema.

3. Appended after line 400 of [3]:
   NameIdentifierMappingProtocolProfile [Optional, 0-many] of type anyURI, which indicates the profile of the NameIdentifierMapping protocol supported by the Provider.

4. Modify lines 403-404 of [3] to read:
   NameIdentifierMappingEncryptionProfile [optional, 0-many] of type anyURI, which indicates the encryption profiles supported by the Provider.

5. Modify lines 434-436 of [3] to read:
   name="NameIdentifierMappingProtocolProfile" type="xs:anyURI"/>
   <xs:element minOccurs="0" maxOccurs="unbounded" name="NameIdentifierMappingEncryptionProfile" type="xs:anyURI "/>

6. Append after line 497 of [3]:
   IDFFSOAPAuthnProtocolProfile [Optional, 0-many] of type anyURI describes the profile of this protocol supported the identity provider as defined in [LibertyAuthn].

7. Modify lines 507-508 of [3] to read:
   <xs:element maxOccurs="unbounded" minOccurs="0" name="IntroductionNotificationProtocolProfile" type="xs:anyURI"/>
   <xs:elementname="IDFFSOAPAuthnProtocolProfile" type="xs:anyURI" minOccurs ="0" maxOccurs="unbounded"/>

8. Modify line 583-589 of [3] to read:
   The AffiliationDescriptor element describes a group of entities, identified collectively by providerID (located within EntityDescriptor), as an enumeration of providerID's. The uniqueness constraints for providerID also apply for providerID in this context, such that it MUST be unique across all Liberty entities with which the affiliation expects to interact, including other affiliations and providers. Therefore, it MUST NOT be the providerID of any of the members of the affiliation, and SHOULD be unique across the set of providerID's with which the affiliation expects to interact. It is the responsibility of the entity represented by affiliationOwnerID to administer this identifier, and thus, its members and uniqueness.

9. Deleted lines 591-592 of [3]:

10. Deleted line 617 of [3]:

References

[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/]*