Liberty ID-WSF Security and Privacy Overview

Version: 1.0

Editors:
Susan Landau, Sun Microsystems

Contributors:
Carolina Canales Venezuela, Ericsson
Gary Ellison, Sun Microsystems
Jeff Hodges, Sun Microsystems
Sampo Kellomäki, Symlabs
John Kemp, IEEE-ISTO
John Linn, RSA Laboratories
Peter Thompson, IEEE-ISTO

Abstract:

This document provides an overview of the security and privacy issues in ID-WSF technology and briefly explains potential security and privacy ramifications of the technology used in ID-WSF. This is not a normative document. The intended audience for this document is implementers of the Liberty Identity Web Services Framework (ID-WSF). It is assumed that the audience is familiar with the Liberty Identity Federation Framework.

Filename: liberty-idwsf-security-privacy-overview-v1.0.pdf

Notice

Implementation of certain elements of this Specification 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. Introduction

1.1. Audience

The intended audience for this document is implementers and deployers of the Liberty Identity Web Services Framework (ID-WSF) and presents guidance for service interface specifications for identity services. It is assumed that the audience is familiar with the Liberty Identity Federation Framework [LibertyIDFFOverview].

1.2. Goals

This document provides a non-normative overview of the security and privacy issues in ID-WSF technology and briefly explains potential security and privacy ramifications of the technology used in ID-WSF.

There are a number of related documents. [LibertySecMech] is a normative document that specifies security protocol profiles including authentication and authorization in Liberty identity services. [LibertyDisco], [LibertyInteract], and [LibertyMetadata] are normative document specifying respectively the protocols used in Discovery Service, Interaction Service, and Metadata description and discovery protocols. [LibertyTrustModels] provides an extensive discussion of the trust models used in Liberty, while [LibertyPrivacy] presents privacy best practices for Liberty - enabled providers.

1.3. Document Structure

The Liberty Alliance Project is an undertaking by a group of companies to develop a set of open, technical specifications for web services. The first step, now completed, is the Liberty Identity Federation Framework, a set of specifications enabling single sign-on using federated network identity. The Liberty Identity Federation Framework provides specifications for associating, connecting, and binding multiple accounts for a given Principal at various Liberty Alliance sites within a Circle of Trust. This document is concerned with Identity Services, which is an abstract notion of a web service that acts upon some resource to obtain information about an identity, update information about an identity, or perform some action for the benefit of an identity. The Liberty Identity Web Services Framework (ID-WSF) is a set of specifications for creating, using, and updating various aspects of identities.

Security and privacy protection in ID-WSF are enforced through several mechanisms:

1. Via general facilities provided at the application layers, and
2. Within each Liberty component, there are application-specific facilities for securing and privacy-protecting data and services.

This document first discusses general security requirements and the issues of authentication and authorization and gives a brief discussion of threat models. Then the document introduces the architectural elements comprising the ID-WSF and discusses the various mechanisms that enhance security and privacy in these components of the ID-WSF: Discovery Service, Interaction Service, and data services. Some more general security issues, including privacy, are then discussed.

1.4. Definitions

Definitions for Liberty-specific terms can be found in the Liberty Glossary [LibertyGlossary]. Security is highly dependent on precise implementation of protocols and for this reason, definitions of a number of the terms used are presented.

Attribute A distinct characteristic of a Principal. A Principal's characteristics are said to describe the Principal.

Attribute Provider The attribute provider (AP) provides Identity Personal Profile (ID-PP) information. Sometimes called an ID-PP provider, the AP is a ID-WSF web services that hosts the ID-PP.

Federate To link or bind two or more entities together.

Identity The essence of an entity and often described by its characteristics.

Identity Provider A Liberty-enabled entity that creates, maintains, and manages identity information for Principals and provides Principal authentication to other service providers within an authentication domain.

Identity Service A particular type of web service that acts upon some resource to retrieve information about an identity or group of identities (e.g., calendars in order to schedule a meeting), update information about an identity or group of identities, or perform some action for an identity or group of identities.

Invocation Identity The subject of a SAML assertion, party requesting service when message is processed.

Non-Transitive Proxy Capability The ability to act for another entity based on Trusted Authority policy. The capability is not transferable.

Policy Decision Point A system entity that evaluates decision requests in light of applicable policy and information describing the requesting entity or entities and renders an authorization decision.

Policy Enforcement Point A system entity that performs access control by making decision requests and enforcing authorization decisions. If the authorization decision is pushed to the PEP there will be no need for it to create a request.

Principal A Principal is an entity that can acquire a federated identity, that is capable of making decisions, and to which authenticated actions are done on its behalf. Examples of principals include an individual user, a group of individuals, a corporation, other legal entities, or a component of the Liberty architecture.

Proxy An entity authorized to act for another.

Recipient An entity that receives a message which is the ultimate processor of the message.

SAML Authority An abstract system entity in the SAML domain model that issues assertions. See [SAMLGloss].

Sender Initial SOAP sender. A sender is a proxy when its identity differs from the invocation identity.

Service Invocation responder, providing a service. Ultimate message processor.

Service instance An instantiation of a particular type of identity service.

Service Provider An entity that provides services and/or goods to Principals.

Trusted Authority A Trusted Third Party that issues and vouches for assertions.

Web Service A service that uses Internet protocols to provide a service designed to be used by programs.

Web Service Consumer (WSC) An entity that uses a web service to access data.

Web Service Provider (WSP) An entity that provides data via a web service.

1.5. What is a Security Policy?

Security needs a clear set of rules that enable the system's administrators to understand what is protected and what is not. A security policy is a set of rules and practices specifying the who, what, when, why, where, and how of access to system resources by system entities (often, but not always, involving or acting on behalf of people). Significant portions of security policies are implemented via security services, which are processing or communication services that are provided by a system to give a special type of protection to system resources [OASISGloss].

In the Liberty context of web services in a distributed environment, two particular aspects of a security policy warrant special note: authentication and authorization. Authentication is the process of confirming a system's entity's asserted identity with a specified, or understood, level of confidence [OASISGloss]. There are variety of methods for doing this. Techniques for authenticating people include account number and PIN and username and password (really two versions of the same technique), which are typically considered a weak form of authentication; challenge-response is a stronger form. The SSL/TLS "handshake protocol" is a cryptographic protocol mechanism for authenticating processing entities; it establishes server-side (and client-side) authentication at the beginning of a SSL/TLS session. In the distributed architecture of Liberty Identity Web Services, authentication is extremely important and we discuss various aspects below.

Authorization is the process of determining which types of activities an entity can perform. If access is to be limited, authorization only makes sense in the context of authenticating an entity. Depending upon the level of authentication, the entity will have authorization to perform different types of activities [OASISGloss].

2. General Security and Privacy Mechanisms for Liberty Identity Web Services Framework

This section provides discussion and guidance related to the distributed security and privacy mechanisms in the Liberty ID-WSF protocols. It emphasizes inter-component aspects as embodied in the ID-WSF architecture; aspects oriented to individual Liberty services will be considered in the next section.

Security in the Liberty Framework is layered. Liberty protocols are constructed with extensive security mechanisms. Furthermore they build upon various Internet protocols that themselves have embedded security mechanisms [LibertyInteract].

Table 1 generally summarizes the security mechanisms incorporated in the Liberty specifications, and thus in Liberty-enabled implementations, across two axis: channel security and message security. It also generally summarizes the security-oriented processing requirements placed on Liberty implementations.

Table 1. Liberty security mechanisms

Security Mechanism	**Channel Security**	**Message Security** **(for Requests, Assertions)**
Confidentiality	Required	Optional
Per-message data integrity	Required	Required
Transaction integrity (requests protected against replay and responses checked that they correspond with requests)	–	Required
Peer-entity authentication	Identity provider - Required Service provider - Optional	–
Data origin authentication	–	Required
Nonrepudiation	–	Required

Channel security addresses how communication between identity providers, service providers, and user agents is protected. Liberty implementations must use TLS1.0 or SSL3.0 for channel security, although other communication security protocols may also be employed (for example, IPSec) if their security characteristics are equivalent to TLS or SSL. Note: TLS, SSL, and equivalent protocols provide confidentiality and integrity protection to communications between parties as well as authentication.

Critical points of channel security include the following:

- In terms of authentication, service providers are required to authenticate identity providers using identity provider server-side certificates. Identity providers have the option to require authentication of service providers using service provider client-side certificates.

- The authenticated identity of an identity provider must be presented to a user before the user presents personal authentication data to that identity provider.

Message security addresses security mechanisms applied to the discrete Liberty protocol messages passed between identity providers, service providers, and user agents. These messages are exchanged across the communication channels whose security characteristics were just discussed.

Critical points of message security include the following:

- Liberty protocol messages and some of their components are generally required to be digitally signed and verified. Signing and verifying messages provide data integrity, data origin authentication, and a basis for nonrepudiation. Therefore, identity providers and service providers are required to use key pairs that are distinct from the key pairs applied for TLS and SSL channel protection and that are suitable for long-term signatures.

- In transactions between service providers and identity providers, requests are required to be protected against replay, and received responses are required to be checked for correct correspondence with issued requests. Time-based assurance of freshness may be employed. These techniques provide transaction integrity.

To federate, providers are required to join communities of trust such as PKI or Kerberos frameworks, or to establish bilateral agreements. These should include obtaining X.509 credentials, establishing and managing trusted public keys, and managing life cycles of corresponding credentials.

Many of the security mechanisms mentioned above, for example, SSL and TLS, have dependencies upon, or interact with, other network services and/or facilities such as the DNS, time services, firewalls, etc. These latter services and/or facilities have their own security considerations upon which Liberty-enabled systems are thus dependent [LibertyIDFFOverview].

2.1. Establishing Trust

Web services is about sharing information. Liberty specifications aim for enabling a networked world in which individuals and businesses can engage in virtually any on-line transaction without compromising security or privacy of vital identity information. In order for interoperating Liberty components to be able to do so, Liberty-enabled entities must establish a "trust relationship." In the original version of the Liberty Identity Federation Framework, federations, established through business and/or legal agreements combined with an out-of-band exchange of shared secret keys or public-key certificates, exemplified a strong and direct trust model. This model of trust does not scale well and is too limited to accomplish web services. A more flexible way of establishing trust is needed. This is done through Brokered Trust and Community Trust models. We present a brief discussion here; more detail on establishing trust among Liberty components can be found in the Liberty Trust Models Guidelines document [LibertyTrustModels].

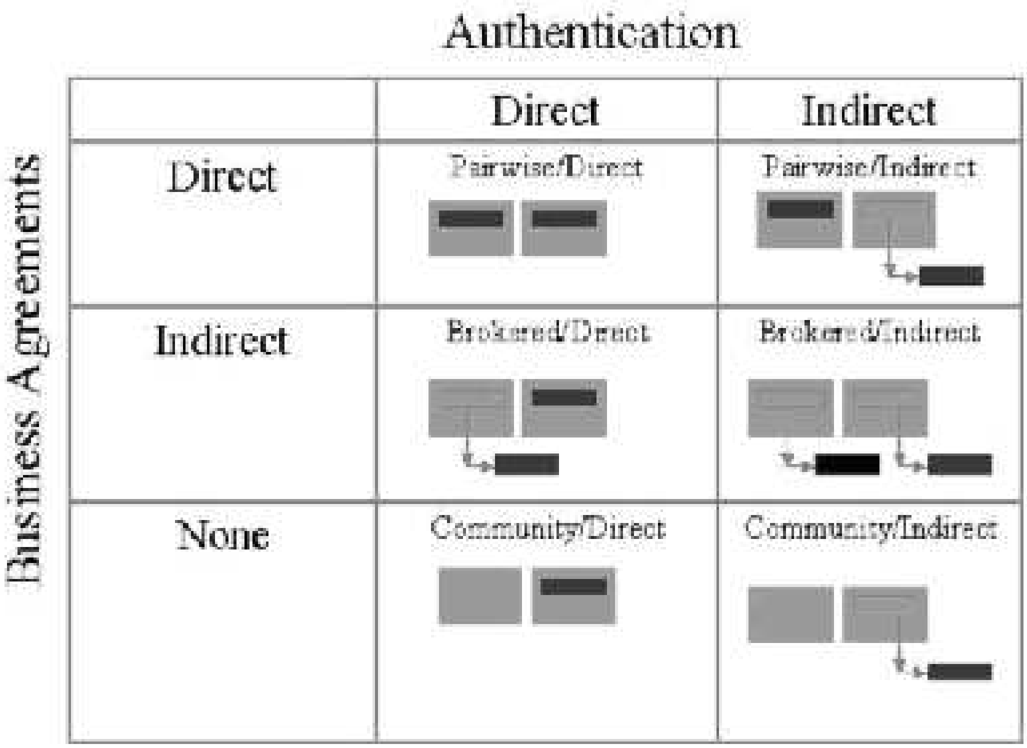
2.2. Authentication

Authentication is the act of confirming a system entity's asserted identity with a specified, or understood, level of confidence. It is dependent on a number of things: the type of credentials being provided, the authentication of the entity providing it (if it is not the asserted owner), etc.

The simplest case occurs when a Principal presents credentials to an identity provider. The identity provider decides whether or not to authenticate the Principal based on the credentials provided by the Principal and the identity provider's own authentication policy.

A more complex scenario occurs when a service provider receives an authentication of a Principal from an identity provider. In this case, the service provider must look at the authentication context: the information additional to the authentication assertion itself that the service provider may require before it makes an entitlements decision. This may include information about the identity provider and its mechanisms. The service provider decides whether to accept the Principal's authentication context as sufficient based on the service provider's authentication policy (note that the service provider will need to authenticate the identity provider).

Brokered Trust models come into play when federation and/or authentication transactions span multiple administrative domains. They require the availability of appropriate intermediaries in order to construct a path to federate a user's relationship and/or to authenticate a particular session. For example, Brokered Trust may be applicable when a service provider associated with identity provider A receives an assertion to be processed from identity provider B, with which the service provider has had no prior relationship. The assertion is a piece of data produced by a SAML authority about an authentication of a subject, attributed information about a subject, or authorization permissions applying to the subject about a particular resource [SAMLGloss]. The service provider must decide whether to trust identity provider B's assertion (which, for simplicity, we will assume is an authentication assertion, though in fact it could be any of assertions mentioned). Trust is determined through a combination of business trust, based on business agreements, and authentication trust, based on cryptographic assertion infrastructure.



	Direct	Indirect
Direct	Pairwise/Direct	Pairwise/Indirect
Indirect	Brokered/Direct	Brokered/Indirect
None	Community/Direct	Community/Indirect

Figure 1. Liberty Protection Schema

In Brokered Trust, there is no direct business agreement between the entities. In our example, identity provider B has no direct relationship with the service provider. There are two possible cases for Brokered Trust: either there is a business agreement between the service provider and an intermediary and the intermediary has a direct business relationship with identity provider B (this can used transitively), or there is not. The business relationship between the service provider and the intermediary allows the intermediary to act as an agent for the service provider. The latter case enables the dynamic establishment of business trust. This is accomplished through the authentication of service entities using Metadata documents.

Community Trust models use membership in a community defined by a cryptographic infrastructure as a basis for enabling federation and/or authentication. Public Key Infrastructure, Kerberos realms and inter-realm relationships, and PGP webs of trust are all examples of such infrastructures.

It is also possible to develop business relationships without authentication infrastructures. That approach is out of scope in the context of Liberty.

In the physical world, authentication is established through physical artifacts or other characteristics of a claimant possibly including the claimant's demonstrated knowledge of private information. Authentication in the on-line world is typically based on cryptographic mechanisms. As observed earlier, there are different mechanisms depending on whether one is authenticating Principals (human) or processing entities. In the Liberty context, Principals are authenticated by identity providers, which determine the means by which they choose to authenticate the Principal. The technique an identity provider uses for authenticating a Principal is not within the scope of Liberty specifications. However, Liberty does specify the transport mechanism for the authentication interchanges. Communications from Principals to Liberty-enabled sites must be integrity protected and confidentiality must be ensured. Liberty-enabled sites must use SSL 3.0 or TLS 1.0 for conducting communications with Principals. Note that the security of the SSL or TLS session depends on the chosen ciphersuite; Liberty specifications recommend the use of at least a 112-bit symmetric key. More details may be found in the normative [LibertySecMech].

Different identity providers will choose different technologies, follow different processes, and be bound by different legal obligations with respect to how they authenticate Principals. The choices that an identity provider makes for authentication mechanisms will be driven in large part by the requirements of the service providers with which the identity provider has affiliated itself. These requirements will be determined by the nature of the service (that is, the sensitivity of the exchanged information, the associated financial value, the service provider's risk tolerance, etc.) that the service provider will be providing to the Principal. If the service provider is to place sufficient confidence in the authentication assertions it receives from an identity provider, it will be necessary for the service provider to know which technologies, protocols, and processes were used for the authentication. With this knowledge and the authentication of the provider of the assertion, the service provider will be better able to make an informed decision regarding what services the subject of the authentication assertion should be allowed to access.

Authentication context is a combination of:

1. Initial user identification mechanisms (e.g., face-to-face, online, shared secret).
2. Mechanisms for minimizing compromise of a Principal's credentials (e.g., credential renewal frequency, client-side key generation).
3. Mechanisms for storing and protecting credentials.
4. Authentication mechanism (e.g., password, challenge-response).

Clearly, not all authentication assertions are the same. One can think of the aspects listed above as serving as axes in a multi-dimensional grid and an authentication assertion has values defined by its coordinates in the space. Liberty can ease the job of assessing and composing authentication assertions by defining particular authentication contexts that are representative of current technologies and practices among identity providers. The [LibertyAuthnContext] document delineates the more common of these contexts. By identifying the authentication context as a Liberty class and giving it a unique identifier, this simplifies the service provider's authentication task.

Liberty specifications require authentication of processing entities. There are two cases to consider. In the absence of active intermediaries in the message path, Peer Entity Authentication mechanisms suffice to ensure the confidentiality and integrity of the message exchange. Authentication of both sender and recipient is required. SSL 3.0 or TLS 1.0 and X.509 client and server-side certificates (see [KPIX-WG] for information on the X.509 Public-Key Infrastructure) can be used for this. If, however, active intermediaries are present, the sender must use message authentication. Therefore the sender must authenticate the messaging layer either by using Web Services Security SOAP Message Security, X.509 token profile sender authentication or Web Services Security SOAP Message Security, SAML token profile sender authentication; normative specifications can be found in [LibertySecMech]. In both cases, the recipient receives an assertion binding the sender to the key, and the sender provides proof of possession of the key by signing elements of the message.

Identity services are invoked by requesters. Under certain circumstances, the Web Services Framework allows two separate identities for a given request: the *invocation identity* and the *sender identity* (see [LibertyDisco]). Typically the identity of the message sender is to be treated as the invocation identity, in which case there is no need for a distinction between the invocation identity and the sender identity. The candidate mechanism to convey identity information is client-side X.509 certificates based authentication over a SSL/TLS connection. Generally this protocol framework may rely upon the authentication mechanism of the underlying transfer or transport protocol binding to convey the sender's identity.

For scenarios where the sender's messages are passing through one or more intermediaries, the sender must explicitly convey its identity to the recipient. This is done through a Web Services Security token. A security token is a representation of security-related information that is used to represent and substantiate a claim. An unsigned security token must be transported through SSL/TLS.

For the cryptographic mechanisms described above to work properly, private and shared secret keys must be secured. Loss of key—private or shared secret—completely compromises the security systems based on cryptographic mechanisms. This means that sensitive processing functions must be performed within systems designed to satisfy appropriate assurance requirements and systems should be operated and managed in accordance with appropriate security practices.

Public keys need not be protected against disclosure but must be protected for integrity purposes. Effective use of a public key for signature validation requires that the key be associated with a trust anchor acceptable to the relying party. This can either be through direct knowledge of the key by the relying party or by successful validation of a correct—and timely—certification path. Secure operation of a signature-based architecture like Liberty ID-WSF requires that a relying party's set of trust anchors be correctly managed. Validation steps (including, e.g., revocation checking) should be correctly performed before accepting a signature as representing its presumed signer. Careless use of the public-key infrastructure invalidates the protections provided by the Liberty Framework security protocol specifications.

In addition to secure processing at the levels of cryptographic operations and trust validation, secure operation of the ID-WSF protocols also requires that the processing rules defined in their specifications be fully and correctly implemented. Security protocols are often fragile and a minor change to a protocol can completely invalidate its security mechanisms. Liberty ID-WSF implementers should ensure that the protocol processing modules they employ are fully conformant with the Liberty protocol specifications.

2.3. Authorization

Access to the attribute data managed by Liberty ID-WSF-based deployments is mediated according to two classes of authorization policies: policies established by Liberty processing components and policies established by the individual Principals with whom the attribute data is associated. Before access to protected data is granted, constraints of ALL applicable policies must be satisfied. Liberty implementers must ensure that suitable policy management interfaces are available to administrators and to Principals. The type and scope of interfaces provided may vary in different operational environments.

Authorization depends on the combination of a securely managed authentication system and securely managed data describing authorization policy (e.g.,. in the form of Access Control Lists (ACLs)) for protected resources [LibertySecMech].

Identity services may be accessed by system entities. The access may be direct or with the assistance of an active intermediary. To access an identity service, a system entity must interact with a specific service instance service that exposes some resource. Identity services are ultimately responsible for the security and privacy of the Principal's information. We believe that they are therefore the right point to enforce access control policies.

The authorization decision to access an identity service offering a specific resource may be made locally (at the entity hosting the resource) or remotely. But regardless of whether the policy decision point (PDP) is distributed or not, a policy enforcement point (PEP) must always be directly implemented by the entity hosting or exposing the resource.

(The authentication context for the Principal and the identity provider's authentication of itself to the service provider are PEPs, e.g., gateways to the resource being managed.) In most cases, the service requester directly interacts with the identity service. Thus the identity service may implement both the PEP and the PDP. Under these circumstances, at a minimum, the authorization decision should be based on the authenticated identity of the service requester and the resource for which access is being requested.

An identity service may rely upon a trusted third party (TTP) to make coarse policy decisions. It is also likely that the TTP will act as a Policy Information Point (PIP) that can convey information regarding the resource and the policy it maintains. This scenario might occur if the Principal is unable to actively authenticate to the identity service. One example of this is when the TTP provides a bridge function to introduce new participants to the identity service. If the TTP acts as a PIP, the result of any such decision made by the TTP must be presented to the entity hosting the identity service. Of course, a decision by a TTP does not preclude the identity service from making additional policy decisions based on other criteria.

Our definition of an identity service enables the possibility of a service performing an action for the benefit of an identity. To appreciate the possibilities this idea suggest one must recognize scenarios whereby peer entities may need to represent or perform actions on behalf of other system entities. From the point of view of authorization, in the case where the invocation identity and the sender identity are distinct, the identity service makes a decision to deny or grant access to the resource based on either or both of these identities.

Identity services relying on authorization decision assertions provided by the TTP must maintain accurate policy data at the TTP and must trust the TTP to correctly reflect that data in the assertions it generates. The Liberty ID-WSF specification enables a TTP to act as an information source to obtain assertions demonstrating the session context of the interacting Principal. The TTP must enforce any access control policies pertaining to the resource which the requester is attempting to locate. If, according to the TTP's policies, the requester is not permitted to access the resource, a failure indication should be returned.

The Liberty ID-WSF also incorporates an Interaction Service that enables providers to engage in direct interactions with the Principals responsible for requested attributes. Authorization policies should be specifiable in a manner that allows these facilities to be invoked as needed, either at the level of confirming that a user is currently logged on to a Liberty identity provider or, more strongly, obtaining explicit approval for access to designated attributes.

Note that a browser-based user agent interacting with some service provider does not necessarily imply that the service provider will use the user identity as the invocation identity. In some cases, the identity of the service provider may be used for invocation.

2.4. Threats

The Liberty Alliance specifications seek to enable individuals and businesses to engage in virtually any transaction without compromising the privacy and security of vital identity information. Liberty specifications have been designed to protect against:

- Eavesdropping: Information within the message is viewable by an unauthorized user.
- Replay attack: A message is sent in which includes portions of another message in order to gain access to otherwise unauthorized information.
- Message Insertion/Deletion/Modification: The message is altered by inserting/deleting/modifying information and is mistaken by the receiver as having been sent as is by the original sender.
- Message Spoofing: The message is written and sent in such a way as to make it appear as having come from a different sender.
- Man-in-the-Middle attack: An attack in which an intermediary poses as the other party to the real sender and receiver in order to fool both parties.

These attacks are prevented through a combination of the authentication and authorization requirements discussed above; see also [LibertySecMech]. There are also a number of security vulnerabilities and risks that are out of scope for the Liberty specifications. These include denial-of-service attacks at the network level, host penetration/access, traffic analysis, timing attacks (computing the amount of time a computation takes in order to determine other information, such as key bits).

3. Security Functions Required for Privacy

Recall that a security policy is a set of rules and practices specifying the who, what, when, why, where, and how of access to system resources by system entities (often, but not always, involving or acting on behalf of people). Considering privacy purely from a security vantage point, privacy is a security policy applied to an individual, or, in the Liberty context, a Principal. Of course, privacy is much broader than such a definition. One can easily find databases with excellent security policies that are nonetheless privacy invasive (any secured database that contains non-relevant personal information, e.g., a research medical database containing the patient's social security number). However, in the context of the Liberty Identity Web Services Framework, where the issue is designing technical specifications for the secure sharing of Principal attribute data, the model that "privacy is security policy applied to a Principal" is a useful model for privacy protections.

The security functions most relevant to privacy are:

- Authentication of the Principal and/or any other entities that could perform policy management tasks (policy definition, modification, etc.).
- Authentication of attribute requesters.
- Policy integrity in transit (at the moment of policy definition, modification or any other kind of policy management operation).
- Policy integrity in storage.
- Attribute confidentiality in transit (response from the attribute provider to the service provider).
- Attribute confidentiality in storage.
- Attribute integrity in storage and transit.
- Policy management authorization.
- Audit capability: maintenance of transaction records in secure storage.
- Avoiding collusion between identity provider and service provider.
- Data aggregation.

A number of the security functions above, including Principal authentication, attribute requester authentication, attribute confidentiality in transit, attribute integrity in transit, and some aspects of avoiding collusion between the identity provider and service provider, fall within the scope of the Liberty specifications. But a number of security issues concern Principal data residing at a provider. These include policy integrity in storage, attribute integrity and confidentiality in storage, audit capability, other aspects of collusion between identity provider and service provider, and data aggregation. There is an important point to note here: Liberty specifications enable Principal's privacy but they do not ensure it. The Liberty Alliance recommends that Liberty-enabled providers satisfy a baseline set of fair information practices, including:

- Notice. Public-facing Liberty-enabled providers should provide the Principal clear notice of who is collecting the information, how they are collecting it (e.g., directly or through cookies, etc.), whether they disclose this information to other entities, etc.

- Choice. Public-facing Liberty-enabled providers should offer Principals choice, to the extent appropriate given the circumstances, regarding how Personally Identifiable Information (PII) is collected and used beyond the use for which the information was provided. Providers should allow Principals to review, verify, or modify consents previously given. Liberty-enabled providers should provide for "usage directives" for data through contractual arrangements or through the use of Rights Expression Languages.

- Principal Access to Personally Identifiable Information (PII). Consistent with, and as required by, relevant law, public-facing Liberty-enabled providers that maintain PII should offer a Principal reasonable access to view the non-proprietary PII that it collects from the Principal or maintains about the Principal.

- Correctness. Public-facing Liberty-enabled provider should permit Principals the opportunity to review and correct PII that the entities store.

- Relevance. Liberty-enabled providers should use PII for the purpose for which it was collected and consistent with the uses for which the Principal has consented.

- Timeliness. Liberty-enabled providers should retain PII only so long as is necessary or requested and consistent with a retention policy accepted by the Principal.

- Complaint Resolution. Liberty-enabled providers should offer a complaint resolution mechanism for Principals who believe their PII has been mishandled.

- Security. Liberty-enabled providers should provide an adequate level of security for PII.

Following these practices will ensure secure and private handling of Principal data at a provider site. (A more detailed discussion of privacy best practices for Liberty-enabled sites, from which the above has been excerpted, can be found in [LibertyPrivacy]).

This brings up an important aspect of Liberty specifications, which are for the (secure) exchange of information between system entities. There are no Liberty specifications about data storage at a system entity. The Liberty privacy best practices include such recommendations but these are best practices *recommendations* only. Furthermore, they are necessarily non-normative, as are any recommendations in this document.

That is a general issue about the security functions described above. The Liberty specifications provide various security mechanisms that help protect the Principal's privacy. Table 1 presents an overview of these mechanisms, which are described in much greater detail in the normative document [LibertySecMech]. Liberty specifications require authentication for anyone acting for a Principal and for any entity requesting or consuming attribute information. For security and privacy, the Liberty specifications specify encryption of Principal data during message transport. Through the appropriate use of nonces, the protections provide security against unauthorized parties accessing data about the Principal through a replay attack. Through the use of pseudonymity, the specifications protect against collusion between WSPs and WSCs who may hold the Principal's attribute information. These requirements provide a high degree of security and thus privacy for the data transmission. But the Liberty specifications must be used in conjunction with business and legal agreements between deploying entities. It is expected that these entities will adhere to their business and legal agreements, including stated privacy policies. But these entities may not adhere to their contracts. In that case, the issue is out of scope for Liberty, which is, after all, a set of technical specifications for data exchange. Instead such a situation is appropriately handled by the legal system.

4. ID-WSF Architectural Elements

An Identity Service is a particular type of web service that acts upon some resource to either retrieve information about an identity, update information about an identity, or perform some action for the benefit of an identity. A resource is either data related to some identity or a service acting for the benefit of some identity [LibertySecMech].

In the current document we assume that the Principal has already registered with an identity provider. The Principal may have done so through a commercial portal or she may have been automatically enrolled through her employer. Nothing precludes the Principal from having several identity providers. Principals, in fact, typically have many identities: as an employee, as a <spouse, parent, child>, as a member of several distinct civic groups (e.g., membership in a political party, membership in service organizations), etc. It is expected that many people will have more than one identity provider, perhaps one through work and several personal ones. In an ID-WSF, the Principal uses services: ordering and arranging for a gift to be shipped (the shipping address already being known to the shipping company), scheduling a meeting with several colleagues, arranging a trip, authorizing an insurance company to view patient treatment information.

ID-WSF consists of a number of distinct elements (see [LibertyIDWSFOverview]) that together form a framework of web services. There are several types of system entities: Web Service Providers (WSP), which host web services such as a profile service (see below), Web Service Consumers (WSC), which, with appropriate authentication and authorization, can access a user's web services by communicating with the WSP's endpoint (the targeted entity that contains the resource), and Discovery Service (DS), which is a web service typically hosted by an identity provider that enables a WSC to determine which WSP provides the needed service. Each of these elements has its own facilities for security and privacy protection.

The ID-WSF SOAP Binding provides a SOAP-based invocation framework for identity services. It defines SOAP Header blocks and processing rules enabling the invocation of identity services via SOAP requests and responses. Additionally, a usage directive container is defined for those implementations that wish to use an existing rights language to specify the required service and data usage policies. The Discovery Service defines a core identity service that enables various entities (e.g., service providers) to dynamically discover a user's registered identity service. The Discovery Service also functions as security token service, issuing security tokens to the requester that the requester will use in the request to the discovered identity service.

4.1. Discovery Service

The first step in Liberty Identity Web Services is to determine where the needed resources are located: which provider holds the Principal's credit-card information, which server stores the Principal's calendar, which provider stores the Principal's travel preferences. The Discovery Service presents an interface for consumers of identity services to locate resource offerings. Entities place resource offerings—information describing the location of different types of information about Principals—in a discovery resource. Thus the Discovery Service is essentially a web service interface for "discovery resources," each of which can be viewed as a registry of resource offerings.

The Discovery Service provides ResourceIDs, a URI used to identify a particular resource. For example, a Principal wants to make airline reservations. Through a *Query* operation a WSC can determine with which resource (WSP) a Principal stores her travel preferences (e.g., client sends a *Query(resource(identity, airlinePrefs))* to a DS. The DS responds with *QueryResponse* message that includes the information as to which WSP handles that resource requested—airlinePrefs— for that identity. Or if a Principal wants to make a purchase over the Internet, a WSC would send a *Query(identity, WalletServ)* to discover which WSP holds the Principal's wallet data. The two *DiscoveryUpdate* operations, *Modify* and *ModifyResponse*, enable maintenance of a discovery resource, accommodating inserts and removals of resource offerings.

The Query operation enables a requester to obtain an enumeration of ResourceOffering elements. The set of results is dependent on the local access policy of the discovery resource

Because a provider hosting a Discovery Service may also be fulfilling other roles for an identity (such as a Policy Decision Point or an Authentication Authority), the QueryResponse operation also functions as a security token service, providing the requester with an efficient means of obtaining security tokens that may be necessary to invoke service instances returned in the DisoveryLookupResponse. If security tokens (currently this is a WS security token, but this type is extensible to other types of security tokens) are provided within the QueryResponse, they will be in the *Credentials* element of the response. As the Discovery Service provider may have to perform significant work for each result in the response, especially if security tokens will be generated, responders should construct a QueryRespone to be as qualified as possible. The Discovery Service provider should provide security tokens if it knows that these tokens will be necessary and it is able to provide them based on the security token included in the request.

Four policy-related directives are defined for *ModifyResponse*: *AuthenticateRequester*, *AuthorizeRequester*, *AuthenticateSessionContext*, and *EncryptResourceID*. If *AuthenticateRequester* is specified for a resource, then the discovery service provider should include a SAML assertion containing an *Authentication* statement (as defined in [LibertySecMech]) in any future QueryResponse message for the resource. This is to enable the client sending the Query message to authenticate to the service instance hosting the resource. If the AuthorizeRequester directive is specified for a resource, then the discovery service provider should include a SAML assertion containing a Resource Access Statement (as defined in LibertyID-WSFSecurity) in any future QueryResponse for the resource. The *AuthenticateSessionContext* directive is identical to the *AuthorizeRequester* directive with the single change being the appropriate statement is SessionContextStatement. If credentials are provided in response to these directives, they must comply with the processing rules defined in the normative [LibertySecMech]. If the EncryptResourceID is included, the discovery service must not reveal the unencrypted ResourceID to the clients (e.g., when returning it in a QueryResponse). If the discovery service is unwilling to do this (e.g., this violates the discovery service's policy), the discovery service must fail the Modify request.

Previously we mentioned the notion of conveying both a *sender identity* and an *invocation identity*. In doing so the ID-WSF framework accommodates a restricted (non-transitive) proxy capability whereby a consumer of an identity service (the intermediate system entity or proxy) can act on behalf of another system entity (the subject) to access an identity service (the recipient). To be granted the right to proxy for a subject, the intermediate system entity may need to interact with a Trusted Authority. Based on the Authority's access control policies, the intermediate system may generate and distribute a token authorizing the intermediary to act on behalf of the subject to the recipient. This protocol framework can only convey authoritative information regarding the identities communicated to other system entities. Even with the involvement of an authority playing the roles of Policy Administration Point and Policy Decision Point, the recipient must still implement some degree of policy decisions and enforcement [LibertySecMech]

The Discovery Service is usually hosted at the identity provider since that is the only way that the WSC has of discovering the Discovery Service itself. This discovery is done through the WSC acting as a ID-FF service provider and obtaining a SAML AttributeStatement containing the resource offering (a DiscoveryResourceOffering) from the ID-FF identity provider. In order to prevent the WSC from colluding and determining information about the Principal's identity, the resource offering (the ResourceID) must be sent encrypted using a key encrypted with the public key of the provider hosting the resource. For privacy reasons, this encrypted key must exhibit nonce-like characteristics.

4.2. Interaction Service

An identity service may sometimes need to interact with the owner of the resource that it is exposing, for example, to collect attribute values or to obtain permission to share the data with a Web Service Consumer. The Interaction Service specification is an ID-WSF specification that defines schemas and profiles that enable a Web Service Provider to interact with the owner of the resource that is exposed by that WSP. The Interaction Service (IS) allows its clients (services) to indirectly query a resource owner for consent, authorization decisions, etc. By definition, the IS is capable of interacting with the Principal at any time, for example, by using special protocols, mechanisms, or channels such as instant messaging, WAP Push, etc. The IS accepts requests to present some information and questions to a Principal and is responsible for "rendering" a "form" to the Principal; to do so, the IS must know about the Principal's capabilities and preferences. The IS returns the answer of the Principal in a response that contains the parameters and values of the request.

The `InteractionRequest` element allows the requester to define a "form" that the IS will try to present to the Principal. The Interaction Service allows for Principal signing of the response. This `InteractionRequest` may contain the optional `ds:KeyInfo` element, a public signing key that the sender has for the Principal. If `ds:KeyInfo` is present, then the attribute signed, which has two possible values, strict and lax, must also be present; strict means that the sender wants a positive response only if the response contains a signed statement from the Principal. The signature must be done using the private key associated with ds:KeyInfo. Although in general InteractionRequests may contain more than one query, in the case where the "signed" attribute is present, the InteractionRequest should contain only a single query. If the InteractionRequest contains the signed attribute, then the Inquiry element of the InteractionRequest must contain an id attribute, a nonce that will be used in the signing.

If the InteractionRequest requests signing, then the recipient should attempt to obtain a signed InteractionStatement from the Principal. If the value of the signed attribute is "strict," then the recipient must respond with an InteractionResponse that contains either an InteractionStatement or a status element with its code attribute set to is:notSigned. If an InteractionResponse contains a signed InteractionStatement, the recipient must verify the signature and discard the response if the signature cannot be verified. The recipient must verify that the id attribute of the signed Inquiry matches the id of the corresponding request Inquiry.

The IS query and response is done through a series of HTTP Redirects. For example, when the resource owner is visiting (where *visiting* is short for having used a HTTP User Agent to send a HTTP request to) the WSC, there are three possibilities for the WSP to contact the resource owner: (i) WSP interacts with the Principal by requesting the WSC to redirect the user-agent, (ii) WSP interacts with the Principal by requesting the WSC to pose an inquiry, and (iii) WSP interacts with the Principal by requesting the IS to pose an inquiry (this last case works for all other cases also). The first case is handled in the following way (see [LibertyInteract] for the other two cases):

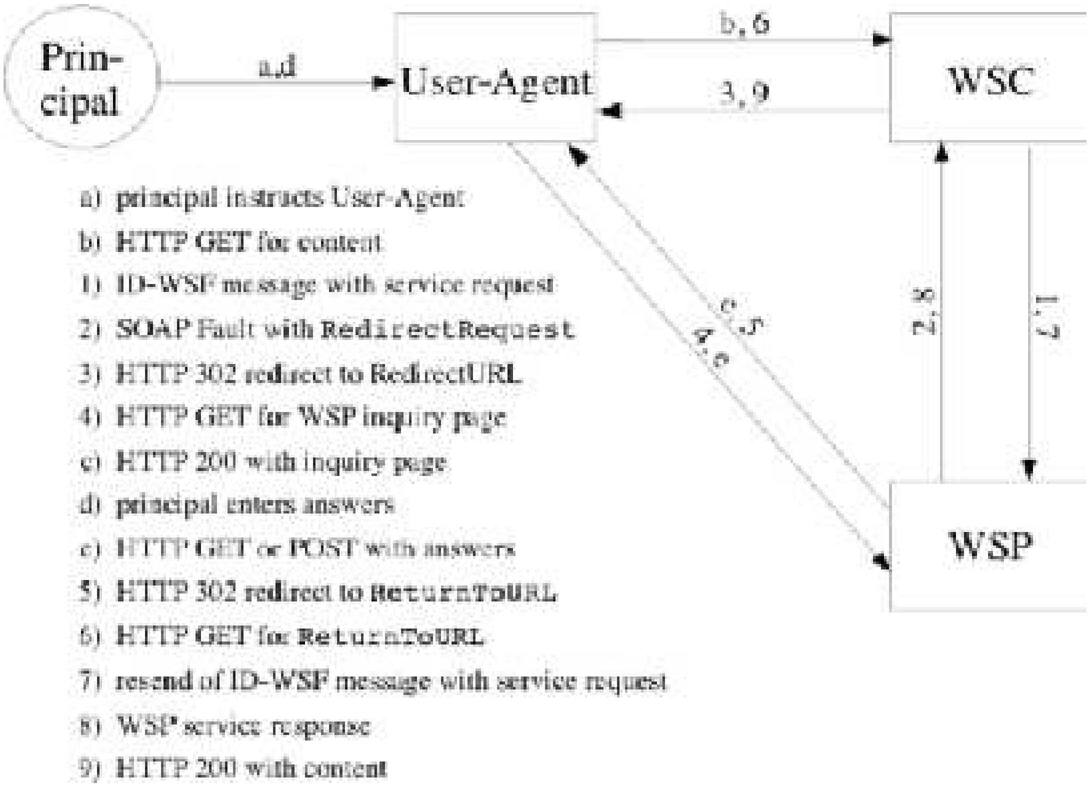

Prin-
cipal
a,d
User-Agent
b,6
3,9
WSC
c,5
4,e
2,8
1,7
WSP
a) principal instructs User-Agent
b) HTTP GET for content
1) ID-WSF message with service request
2) SOAP Fault with RedirectRequest
3) HTTP 302 redirect to RedirectURL
4) HTTP GET for WSP inquiry page
c) HTTP 200 with inquiry page
d) principal enters answers
e) HTTP GET or POST with answers
5) HTTP 302 redirect to ReturnToURL
6) HTTP GET for ReturnToURL
7) resend of ID-WSF message with service request
8) WSP service response
9) HTTP 200 with content

Figure 2. WSP Interaction via WSC redirect of user-agent

To ensure privacy and security of Principal data, the Liberty specifications require that the information be returned to the correct source. Thus in step 2, the recipient of a RedirectRequest must verify that the redirectURL points to the WSP, i.e. the host in the URL must be the same as the host to which the WSC sent its service request. If this is not the case, the recipient must ignore the RedirectRequest. In step 3, the user agent must be associated with the ID-WSF message that caused the RedirectRequest and the WSC must append a ReturntoURL parameter to the redirectURL with the URL to which the WSC wants the user agent directed back. In Step 4, the WSP must verify that the host in the URL is the same as the host to which the WSP the RedirectRequest. The WSP should verify that the identity of the user is the owner of the resource that was targeted in the original ID-WSF request. The other two cases similarly verify that the information is sent to the correct location.

An InteractionRequest is responded to with an InteractionResponse. If the InteractionResponse contains a signed InteractionStatement, the recipient must verify the signature and must discard the response if the signature cannot be verified. The recipient must verify that the id attribute of the signed Inquiry corresponds to the id of initiating request.

It is worth recalling that privacy has many meanings, and in addition to keeping PII private, privacy can also denote the right to be left alone. While Web Services connotes the idea of *always available* and is based on that capability, nonetheless, there are undoubtedly times when the Principal would prefer to "left alone."IS has an optional attribute whose purpose is to protect this aspect of the Principal’s privacy; this is the *interact* attribute. The interact attribute may be set at doNotInteract, which indicates that the recipient must not interact with the resource owner, or doNotInteractforData, which indicates that the recipient may not request data (e.g., Personal Profile data) but may request a responses (e.g., for obtaining consent).

An important security issue to consider is that the Interaction Service is effectively acting to its client WSCs as a proxy for the Principal. Thus the IS should be trusted by those clients. This is especially the case when such a WSC is itself a WSP that needs to obtain consent or permissions.

There is no general possibility for an IS to prove on-line that it did indeed obtain the response from the Principal. But of course the IS can—and should—authenticate the Principal and then save the proof of authentication, such as an assertion. There is little point in forwarding such assertion to the WSC as proof, as ID-FF authentication assertion will contain the NameIdentifier of the Principal as it is known to the IS, not to the WSC (for pseudonymity purposes, this name is encrypted). An IS that is closely associated with an identity provider (i.e., has the same providerID as the identity provider) could issue an assertion that states the Principal as known to the WSC was present.

It does not suffice to know that a Principal was present at the IS. There remains the possibility that the IS modified the Principal’s response. One solution to this threat is to have the Principal sign the response with a private key for which the invoking WSC has a public key associated with that Principal.

For the Redirect Profile these considerations do not apply, as parties that need to interact with a resource owner do so themselves. It is again important that the WSP authenticates the Principal. Although the information in these redirects is not particularly valuable, it is nonetheless recommended that secure connections be used so that intruders cannot replay a request. This risk is reduced if WSPs require that all ID-WSF requests are signed and/or authenticate WSCs. All participants should protect themselves against replay attacks by checking for recently-used messageIDs, etc.

The Principal has a risk that an IS, or for that matter, any WSP, may misrepresent him. That is, of course, an out-of-band issue. Nonetheless, we observe that IS providers should make efforts to induce trust in the Principal by offering transaction logs, by employing sufficiently strong authentication methods, etc. [LibertyInteract].

4.3. Data Services

Web services provide data services to computers and networked devices. In the current context, a data service is a web service that supports the storage and update of specific data attributes regarding a Principal. The Liberty Personal Profile Service and the Liberty Data Service Template are two examples of data services; the Personal Profile Service provides profile information regarding a Principal while the Data Services Template provides protocols for querying and modifying data attributes while implementing a data service using ID-WSF. Although the Personal Profile Service is actually part of the Liberty Identity Services Interface Specification, for completeness, we include it here.

4.3.1. Personal Profile Service

The Liberty Personal Profile Service, ID Personal Profile, is a service that handles identity information for a Principal; the service provides identity attribute data structured in containers (containers are sets of related attributes, e.g., street address, town, city, postal zip, country may form the address container). Typically a Principal will have several identities. The Principal may choose not to have these identities linked. All of a Principal's ID Personal Profiles may, however, be registered with a Discovery Service.

The attribute data may be carefully validated (more likely if the information is from an HR database) but it need not be. A Principal may list different values for an attribute in different ID Personal Profile services (e.g., different choice of personal title in work and personal ID Personal Profile services, different photo for personal and work ID Personal Profile services). Because there may be multiple hosts for a single Principal's ID Personal Profiles, data synchronization between these various hosts is infeasible. In any case, such synchronization is quite possibly not desired. It is neither expected nor necessary that all attributes of an ID Personal Profile service be populated.

There are no Liberty ID-WSF requirements on how data actually resides at an ID Personal Profile service. Thus data may be stored at the service, it may be computed on the fly, it may be kept on a backend system. Although the Liberty ID Personal Profile specification is defined in terms of XML, that does not mean that data at the ID Personal Profile service must actually be kept in XML format. The ID Personal Profiles are queried by or updated by clients, typically a service provider, acting on behalf of a Principal. An ID Personal Profile is not required to report the same results to two instances of the same query unless the query is being made by the same client and no update (a modify or out-of-band update) of the data has occurred in the interim [LibertyIDPP].

4.3.2. Data Service Template

The ID-WSF Data Service Template provides two protocols, one for querying data attributes of a Principal and one for modifying data attributes when implementing a data service on a Liberty ID-WSF. The `Description` element contains one or more `SecurityMechID URI`s, which identify the security mechanims supported by the service instance. It is expected that authentication will be used, though there is one `SecurityMechID URI` for the case in which no authentication of the client is required (see [LibertyDisco] or [LibertySecMech] for further details).The query must identity the Principal and the data being queried.

The request message must state the resource it wishes to access (e.g., the Personal Profile of a certain Principal) as well as more specified information about exactly what data it wishes to access (e.g., telephone number). Both data requests and data modifys support multiple operations in a single message, but in a single request all the operations must be of the same type, e.g., all requests or all modifications. The response message includes a status element that indicates whether the processing of the request succeeded [LibertyDST]. To protect the user's privacy, the `ResourceID` may be encrypted. A non-predictable nonce must be used in the encryption so as that the discovery service client does not have a persistent reusable identifier. This prevents collusion between a web services client and another that otherwise could be issued the same `ResourceID` by the Discovery Service.

The data services template includes an optional attribute, ACC (Attribute Collection Context), which describes the context or mechanism used in collecting the data. This informs the service provider asking for the data as to whether any validation of the data has occurred. Three attributes of ACC are of particular note:

1. `acc:challenge` attribute documents that a challenge mechanism has been used to validate the data (e.g., an email sent to an address and a reply received or an SMS message sent to a mobile phone and the message contained a WAP URL to be clicked).

2. `acc:secondarydocuments` that the value has been validated from secondary documents (e.g.„ a an address from an electric bill),

3. `acc:primarydocuments` that the value has been validated from primary documents (e.g., name and identification number from a passport)..

4.4. Metadata

In the original documents for Liberty ID-FF specifications, the Liberty Alliance protocols dealt only with the exchange of Principal data. Metadata to enable the linking of Liberty entities was handled out of band. This was quite limiting. If two entities wished to communicate without previous awareness of membership in a common trust infrastructure, there were three possible outcomes:

1. The entities communicate insecurely without authentication.
2. The entities transfer data enabling them to perform authentication.
3. The entities do not interoperate.

The first option does not fit the Liberty Alliance paradigm of secure data exchange and the third option is unduly restrictive. Liberty ID-WSF includes protocols for two Liberty-compliant entities to exchange metadata in real time, thus enabling ad-hoc interaction between entities. The information to enable this interaction is published in a "metadata document."

There are two ways to publish metadata document locations: via a "well-known location" or via DNS. In either case, metadata should always be transported securely, e.g., via SSL/TLS to ensure integrity. Parties relying on the metadata should process the SSL/TLS certificate presented by the server through normal validation processes.

Trust establishment of the metadata will be based on at least one of the following:

1. DNS Signatures (recommended).
2. TLS Server authentication (recommended).
3. Metadata `ds:signature` (strongly recommended).

It is suggested that entities publish their resource records in signed zones using DNSSEC such that relying parties may establish certain trust decisions based on these signatures. If DNS Signatures are present, relying parties must validate the signature.

Trust of the metadata document and trust of the entity described by it can be achieved in several ways:

1. Trust derived from the signature of the zone from which the metadata location URI was resolved, ensuring accuracy of the metadata document location. This should be done as described in DNSSEC.
2. Trust derived from the signature processing of the metadata document itself, ensuing the integrity of the XML document. This is especially important in the case of local caching.
   Metadata documents should be signed, either by a certificate issued to the subject of the document, or by another trusted party. Consumers of metadata documents must validate signatures on initial retrieval as well as *each time* the document is retrieved from a local cache. This is to detect any document tampering. Trust derived from the SSL/TLS negotiation of the metadata delivery URI, ensuring the identity of the publisher of the metadata. Consumers of metadata documents should consider the trust inherited from the issuer of the SSL/TLS certificate. Since publication URLs are not always located in the domain of the provider of the subject of the metadata document, consumers of documents should anticipate certificates whose subject is the provider.
   Since the basis of this trust may not be available against a cached document, in this case other trust mechanisms should be used.

Post processing of the metadata document should include at least one of these processes.

It is important that consumers of metadata documents observe the `validUntil`, which indicates the expiration date and time of the node and its descendants, and `cacheDuration` of documents. In both cases, the most restrictive value of the value must be used if there are conflicting directives. It is recommended that publishers of metadata documents express document expiration at the `EntityDescriptor` level only and not on the child nodes.

5. ID-WSF Security and Privacy Policy Capabilities

The members of the Liberty Alliance envision a networked world across in which individuals and businesses are able to engage in virtually any on-line transaction without endangering the privacy and security of vital identity information. The key objectives of the Alliance are to enable consumers to protect the privacy and security of their network identity information, to enable businesses to maintain and manage their customer relationships without third-party participation, to provide a single sign-on standard that includes decentralized authentication and authorization from multiple providers, and to create a network identity infrastructure that supports all current and emerging network technologies [LibertyIDFFOverview]. Below we give some non-service-specific security and privacy guidance.

5.1. Usage Directives

The Liberty ID-WSF architecture incorporates a usage directive facility that allows requesters to designate the use they intend for requested data, and allows providers to designate the permitted uses of released data. While it is intended that this facility can be leveraged to integrate processing of privacy policies into Liberty ID-WSF protocol exchanges, the usage directives' scope is not confined to this purpose. The Liberty architecture provides a general means for interacting parties to exchange policy statements, and is suitable for use with various policy expression languages. In order to apply the usage directive facility effectively, implementers responsible for a set of interoperating Liberty components must agree on a common set of supported policies, and on the expression language to use to represent those policies.

For example a WSC may include usage directives in a request sent to a WSP, known as request usage directives. Request usage directives may include information about the WSC, the purpose of the request, whether there is intent to share any returned information with other parties, and so forth. Request usage directives will be evaluated at the WSP against any applicable policies governing the requested information in order to determine whether the intended usage of the requested information complies. If so, then the WSP will reply to the request with the requested information, and the WSP may include usage directives of its own in the response. These response usage directives stipulate what the WSC may do with the returned information, for example whether the information may be shared with other parties.

Incorporating request usage directives as a factor in policy decisions at a WSP will influence the policy expression language used by the WSP to define site-specific policies. This is by virtue of the usage directives themselves being expressed in some language. The site-specific policies do not necessarily need to be expressed in the same language as the request usage directives. But if they differ, it must be possible to create an effective mapping between the expression languages.

Incorporating usage directives cannot ensure a Principal's privacy since the requester, the WSC, might request information using an attestation of adherence to a strict privacy policy, and subsequently not adhere to its stated policy. That situation is, however, out of scope for the Liberty specifications.

Implementations of Identity Services should provide mechanisms to enable deployments to customize the policies that control the distribution of a Principal's attributes. Policies cover the circumstances/conditions under which the Principal attributes are provided to a requesting service provider/WSC.

Although on first thought it might seem that Principals should define the policies for their personally-identifiable information (PII), in many cases the identity provider should also play a central role in this determination. This is because Principals may not be prepared to define policies to control their privacy information in instances where they have not fully understood the privacy implications. Such situations include:

- Some attributes that are used for formal identification purposes, as the legal name, require a close control of privacy and the Principal may not be aware of this need.
- Some attribute values can be deduced from the combinations of other attributes value (date of birth from age and birthday) and the identity provider's policies should be defined considering it.

- In situations where the Principal has the right to expect full anonymity, their identity can often be determined from an unexpectedly small set of attributes (e.g., date of birth, date of hospitalization, type of medical treatment, postal code). In cases such as these, the identity provider needs to understand what policies are necessary in order to properly protect the Principal's anonymity.

The attribute provider needs to define some basic/default policies to protect Principal's privacy. These rules should be written in such a way that a Principal has to consciously choose not to use these rules (that is, the Principal has to "opt-in" for a weaker privacy policy).

There may be other reasons that the attribute provider or the entity managing the attribute provider infrastructure (e.g., telecom operator etc.) defines its own policies. Besides these policies (Principal's, attribute provider), other policies may be needed in order to cover legal issues of the jurisdiction. Since different kind of policies may occur for the same attributes, a priority mechanism is needed for cases in which those policies are contradictory. Thus it can be decided which policy has a higher priority and is thus applied.

In various jurisdictions, service providers may be required to let the Principal exercise first right of control over information she chooses to share with the attribute provider. In this case, the Principal must actively define the attributes that the attribute provider can host, and the attribute provider needs an explicit consent from the Principal for service creation. The Principal may select the set of attributes that each attribute provider holds so that certain attributes are only hosted at attribute providers controlled by the Principal (or which the Principal especially trusts). Because of this, a given instance of a (e.g., ID-Personal Profile Service) may not offer the complete set of user attributes.

The ID-WSF Discovery Service supports this functionality by means of the "options" feature.

The definition of policies to safeguard the Principal's privacy is not only applicable to the attributes but also to the use of the specific identity service. That is, there will be policies to determine if the service provider can use the identity service. Some of these policies may be based on the Principal whose information is being requested (e.g., the ID-Personal Profile service as a whole might be denied to a service provider if this is looking for some VIP Principal).

5.1.1. Policy Applicability

Defined policies may apply to a specific attribute, they can apply to a container so that the policy is applicable to all the attributes within that container, or they can even apply to the whole set of attributes so that a particular service provider cannot access any of the Principal's attributes. Moreover, the attribute provider's policies or legal policies may be defined in such a way so that certain service providers do not have access to the service. This means that there can be two kinds of policies:

- Those defined for the usage of the identity service ("**service privacy**"); this is, the resources that can be returned by the DS to the requesting service provider.
- Those defined for the access to Principal information ("**Principal privacy**"); this is the attributes that can be returned by the attribute provider to the requesting service provider.

It is perfectly reasonable to evaluate the policies from a higher definition level to a lower, e.g., policies at container level will first be evaluated and if that policy is satisfied, then the policies for the attributes of that container will be evaluated. For example, there may be a policy allowing access to the Address container, but with restrictions on Street Address, allowing only Postal Code, Locality or City, State or Provence, and Country to be sent in the answer.

5.1.2. Policy Decision and Enforcement

The policies concerning **service privacy** have to be checked (policy decision) and executed when there is any request to the Discovery Service. The policy decision and enforcement is executed before sending the information on the attribute provider holding the Principal attributes and therefore the Discovery Service acts as a policy enforcement point (it could act as well as Policy Decision Point but the decision could be delegated to other entity controlling the service policies).

The policies concerning **Principal's privacy** must be executed when there is any attribute request to the attribute provider. The policy decision and enforcement is executed before sending requested information about the Principal's attributes. Therefore the attribute provider acts as a Policy Enforcement Point (it could act as well as Policy Decision Point but the decision could be delegated to other entity controlling the Principal's policies).

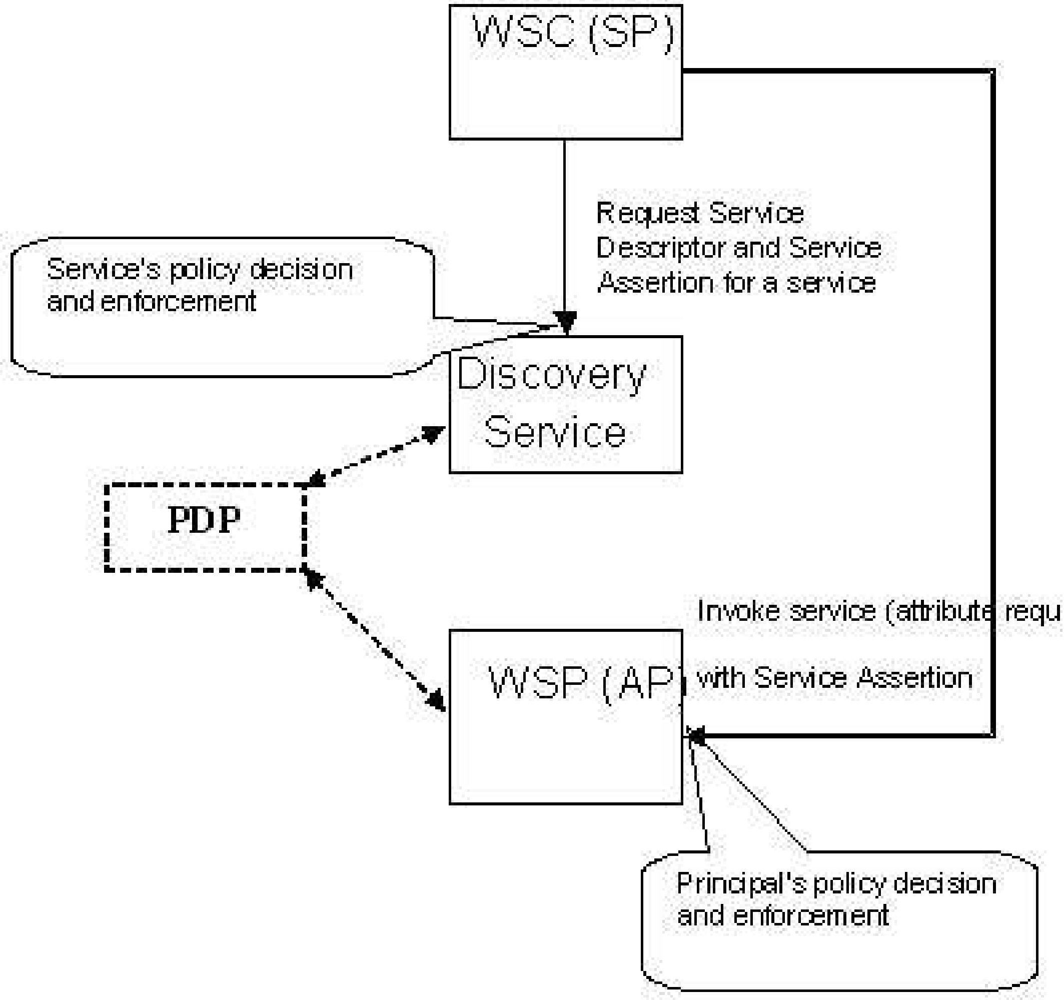

WSC (SP)
Request Service Descriptor and Service Assertion for a service
Service's policy decision and enforcement
Discovery Service
PDP
Invoke service (attribute requ
with Service Assertion
WSP (AP)
Principal's policy decision and enforcement

Figure 3. Privacy-enforcing Decision Point

When controlling the access to the whole set of attributes of certain Principals (e.g. some service provider doesn't have access to the attribute provider if the request is on a VIP Principal), the policies can be regarded as:

- Policies controlling the access to the services (for a specific Principal) and in this case the policies are enforced in the DS.

- Policies controlling the access to the attributes (the whole set) of a Principal and in this case the policies are enforced in the attribute provider.

References

Informative

[LibertyAuthnContext] Madsen, Paul , eds. "Liberty ID-FF Authentication Context Specification," Version 1.2, Liberty Alliance Project (12 November 2003). *http://www.projectliberty.org/specs*

[LibertyDST] Kainulainen, Jukka, Ranganathan, Aravindan, eds. "Liberty ID-WSF Data Services Template Specification," Version 1.0, Liberty Alliance Project (12 November 2003). *http://www.projectliberty.org/specs*

[LibertyDisco] Sergent, Jonathan, eds. "Liberty ID-WSF Discovery Service Specification," Version 1.0, Liberty Alliance Project (12 November 2003). *http://www.projectliberty.org/specs*

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

[LibertySecMech] Ellison, Gary, eds. "Liberty ID-WSF Security Mechanisms," Version 1.0, Liberty Alliance Project (12 November 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*

[LibertyTrustModels] Linn, John, eds. "LibertyTrust Models Guideline," Version 1.0, Liberty Alliance Project (12 November 2003). *http://www.projectliberty.org/specs*

[LibertyIDFFOverview] Wason, Thomas, eds. "Liberty ID-FF Architecture Overview," Version 1.2, Liberty Alliance Project (12 November 2003). *http://www.projectliberty.org/specs*

[LibertyGlossary] Wason, Thomas, eds. "Liberty Technical Glossary," Version 1.2, Liberty Alliance Project (12 November 2003). *http://www.projectliberty.org/specs*

[LibertyIDWSFOverview] Tourzan, Johnathan, eds. "Liberty ID-WSF Architecture Overview," Version 1.0, Liberty Alliance Project ( ). *http://www.projectliberty/specs*

[LibertyPrivacy] Korentayer, E., eds. (14 April 2003). "Project Liberty Privacy and Security Best Practices," Release 1.1, Liberty Alliance Project *http://www.projectliberty.org/specs/Project_Liberty_Best_Practices4.14.03.pdf*

[OASISGloss] Hodges, J., eds. (2003). "OASIS Security Services TC: Glossary," Organization for the Advancement of Structured Information Standards *http://www.oasis-open.org/committees/security/#documents*

[KPIX-WG] Arsenault, A., Diversinet, , Turner, S., eds. (July 2002). IETF *http://www.ietf.org/internet-drafts/draft-ietf-pkix-roadmap-09.txt* "Internet X.509 Public Key Infrastructure Roadmap," Draft version 0.9,

[wss-sms] Hallam-Baker, P., Kaler, C., Monzillo, R., Nadalin, A., eds. (June 30, 2003). Organization for the Advancement of Structured Information Standards *http://www.oasis-open.org/committees/download.php/2757/WSS-SOAPMessageSecurity-14-063003-merged.pdf* "Web Services Security: SOAP Message Security," Draft WSS-SOAPMessageSecurity-14-063003,

[SAMLGloss] Hodges, J., Maler, E., eds. (05 November 2002). "Glossary for the OASIS Security Assertion Markup Language (SAML)," Version 1.0, OASIS Standard, Organization for the Advancement of Structured Information Standards *http://www.oasis-open.org/committees/security/#documents*