Internet Single Sign-On Systems

Research Report

Author:
Radovan Semančík

Date:
May 2005

Version:
1.0

Abstract:
This document describes the requirements and general principles of Internet Single Sign-On systems. The general model of Internet SSO system is described as well as User-Persona-Account relationships. The Liberty ID-FF, WS-Federation, Shibboleth, SXIP and LID specifications are considered and of these specifications suitability for implementing an Internet SSO system is evaluated.

nLight, s.r.o.
Súľovská 34
Bratislava, Slovakia
www.nLight.sk

Table of Contents

1 Introduction

Applications based on the HTTP and HTML are the most commonly used mechanisms for providing live content on the Internet. These applications are under the control of many organizations and user management is done by each application independently of the others. As there must be an independent authentication system for each application, this led to the use of the simplest and least expensive authentication system: password authentication. Apart from being simplest and cheapest, the password authentication is also weak, and the large number of systems makes it difficult for a user to maintain good password management procedures.

The use of independent strong authentication systems in each Internet application directly is inefficient and may even prove infeasible. The solution may be the outsourcing of authentication services to trusted third parties. For such authentication services providers it may be economically feasible to implement stronger authentication techniques and maintain better security procedures.

This document provides overview of systems, that allow user to authenticate on one Internet site and use services on the other Internet site. These systems are referred to as *Single Sign-On (SSO)* systems, because they allow single authentication for multiple services. Only the Single Sign-On systems that can efficiently operate in the Internet environment are considered in this document. The requirements for the Internet SSO systems are defined in the section 1.1, generic principles and models are described in section 2 and the existing Internet SSO systems are described and evaluated in section 3. Section 4 summarizes and concludes the document.

1.1 Requirements

The requirements for web application Simplified Sign-On system for the usage on the Internet are defined as follows:

- It must be based on open protocols and standards.
- It must support cross-organization operation. It must make no assumption that the actors are under single organizational control or that they follow same procedures or policies (except for the SSO protocols itself).
- It must provide a mechanism to securely share user attributes across organizational boundaries.
- It must support privacy features. The user that posses the data must be able make decision what data and under what circumstances to share. There may be operational or collected data, that are not directly submitted by user (e.g. usage patterns). The system must make it difficult to misuse that data, for example by correlating them with other similar data without user consent.
- It must support standard web browser, that implements current versions of HTTP, HTML and accompanying standards. It must not need to install custom software components to the end user system or extend browser functionality in any non-standard way.

2 Internet SSO Systems Principles

The goal of Simplified Sign-On system is to securely transfer user's identity, attributes and current authentication status of a user from source site (Identity Provider) to the destination site (Service Provider). For example, user may have established authentication session with source site. The SSO system is used to transfer that session status to the target site, so that that site can establish similar session with the user (Figure 1). The source and target sessions need not to be same, they may differ in session tracking mechanism (cookies, URL parameters, etc.), there may be different user identifiers, policies or any other session parameters. The session at source site may not even exist at the start of the SSO process, it may be created by source site on demand.

The trust relationship must be established between source and target sites for a source site to trust the target site's requests and for a target site to trust the source site's identity statements. Establishing and maintaining this trust relationship is out-of-band for most SSO systems. The described Internet SSO systems follow the proxy-based true SSO model according to [1].

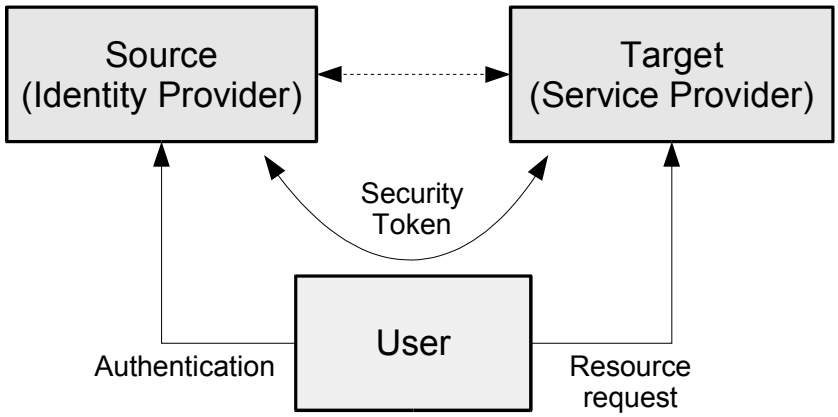

Figure 1: Simplified Sign-On system architecture

2.1 Terminology

Different SSO systems use different terminology to describe their operation. For the purpose of comparing these systems we first define common terminology:

Source site or *Identity Provider* is a system, that has some information regarding user (authenticated session, attributes, etc.) and is willing to transfer that information to other sites (may be subject to policy).

Target site or *Service Provider* is a system, that is willing to receive information about user. Note that receiving a specific information does not imply trust in that information.

User refers to the physical person who is interacting with the computer system. The User is a virtual entity for a computer system, that is represented by persona (or personae).

Persona is a digital representation of user's characteristics. User may maintain several personae that may be more or less related to each other. The characteristics of persona are represented in the form of *attributes*.

Account is a data structure that is usually kept in the computer system databases. The account is used for access control purposes, storing attributes, credentials, etc. Account is usually used as a persistent storage for (partial) persona attributes, but it also may be unrelated to any physical user or persona (e.g. system account).

Persona Identifier is a value of persona attribute that uniquely identify a persona in a given scope or context. In the most common case these identifiers are in form of short strings representing a username, but may have hierarchical structure of LDAP distinguished name or may be just a random binary value.

Pseudonym is an alternate identifier for a persona. Pseudonyms belongs to specific persona and are typically long-term identifiers. The persona to pseudonym mapping is in most cases a private information, the persona to which a specific pseudonym belongs is not publicly known.

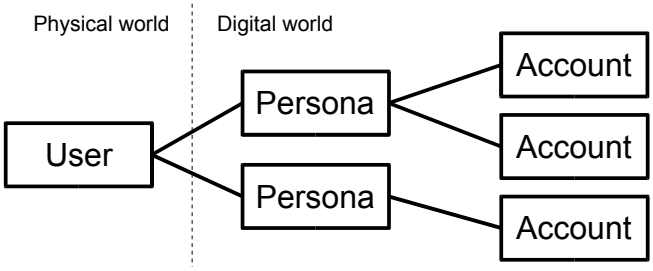

Figure 2 Simple User - Persona - Account structure

For the purpose of considering SSO systems we will limit the terminology model to focus on human-to-computer interactions only, as this makes the terminology considerably simpler.

Simple User-Persona-Account structure is depicted on Figure 2. This is just an example of tree-like persona structure. The data stored in accounts may itself act as a persona (or

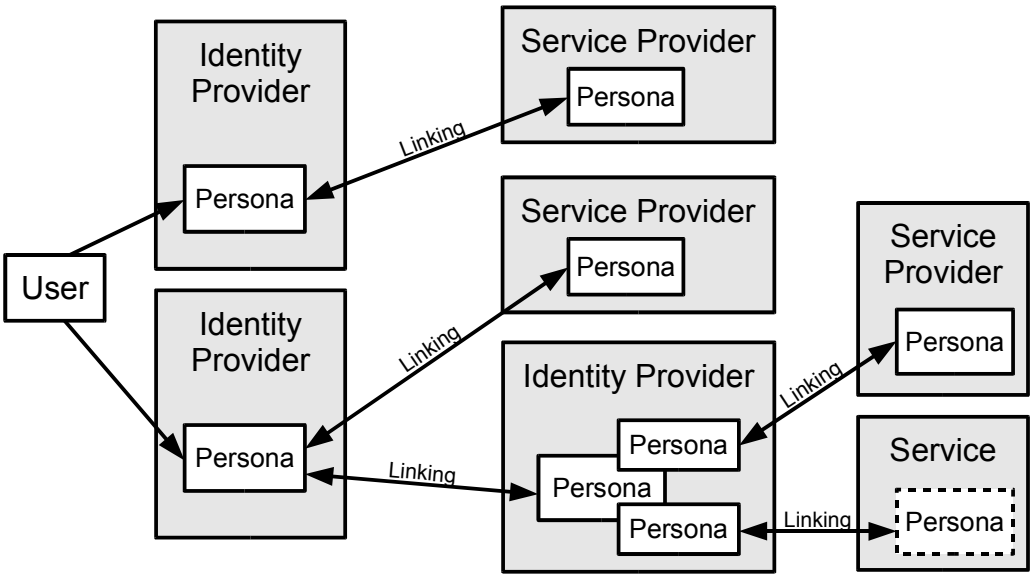

Figure 3 Complex User - Persona - Account structure

personae) and the structure may get much more complicated. Example of such structure is illustrated on Figure 3. Note that persona is an abstract representation of characteristics and that in practical implementation any persistent persona will need an account-like structure (or structures), where the data will be physically stored.

2.2 User Identifiers and Attributes

The persona identifier may be presented to the target site in different ways:

- *Direct Linking*: Provide to the target site the same persona identifier (e.g. username) as the user has established with the source site. The source site usually provides the same attribute values to all target sites.
- *Indirect Linking*: Provide a pseudonym for a persona or a whole new persona to the target site. The pseudonym is an identifier that is different as the primary persona identifier established with the source site, but is fixed in time for the same persona and the same target site. Indirect linking may be used to implement pseudonymity [2].
- *Transient linking*: Do not provide identifier or provide an temporary anonymous handle valid for a single session or a part of session. Transient linking is used in anonymity scenarios [2].

In addition to the persona identifier, source site may also provide other persona attributes. These attributes may be for example user's personal data (first name, last name), attributes used in authorization decisions (privileges, roles) or pointers to user's personal services (calendar service). Note that the user's pseudonym may also be regarded as a regular attribute by some SSO systems.

2.3 Message Exchange

All considered SSO systems employs similar mechanism to transfer authentication status from source to destination site. In all these cases, browser redirection or form processing capabilities are used to transfer security tokens between sites. The process is illustrated on Figure 4 and it consists of following steps:

1. The user **requests resource** on target system (service provider). For this example we assume that there was no prior user interaction with the target site.
2. Target site does not recognize the user (has no valid session for the user/persona). The target site constructs the **authentication request** and returns it to the user's browser in the response. The response is returned in the form of HTTP redirect or HTML form, that will redirect user interaction to the source (identity provider) site.
3. The **authentication request** is received by source site (identity provider). The source site processes the request, and applies any relevant policy.
4. The source site may **authenticate** the user, if not already authenticated or if any policy or the request requires re-authentication.
5. The source site constructs the **authentication response**, which contains the results of persona identity evaluation. The authentication response may contain a security token, that will prove persona identifier and/or attributes to the target site. The authentication response is returned in the HTTP response to the user's browser in a form of HTTP redirect or HTML form, that will redirect user interaction back to the target (service provider) site.
6. The authentication response is received by the target site. The response is processed and the security token is evaluated. For the response and token processing it may be necessary to contact source site directly (6a), for example to resolve references in the

response. Note that the token itself may be passed by reference in the authentication response and it may be needed to dereference it by direct communication to the source site. After the response and any related security tokens and processed the persona identifier and/or attributes are determined.

7. The target site applies any relevant policies to the original access request (step 1) combined with the information determined in step 6. If the request is allowed, the target site will in most cases establish a local session with the user's browser. The local session will help avoid quite significant overhead of future re-authentications.

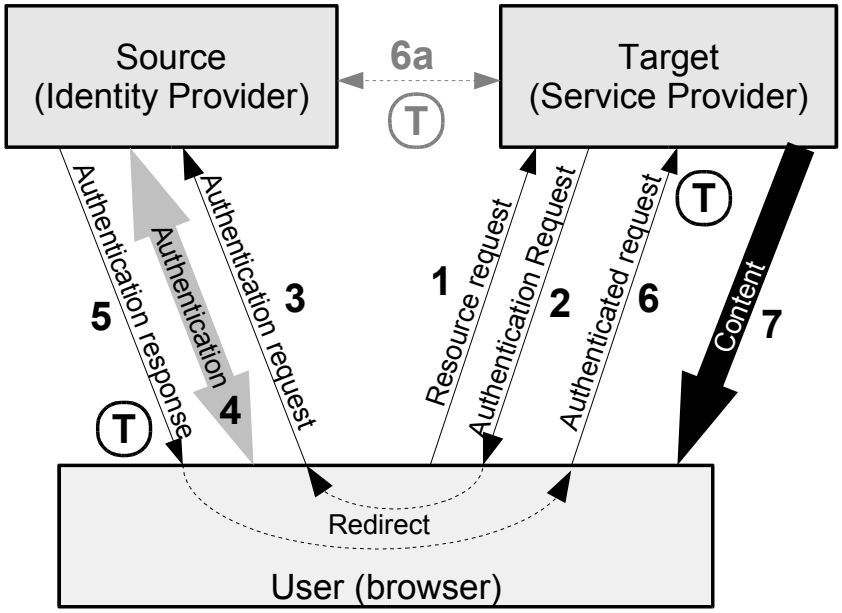

Figure 4: Generic SSO message exchange

3 SSO Systems Overview

The web SSO systems considered in this document are summarized in the following table:

System	*Origin*	*Version*	*Date*
Liberty ID-FF	Liberty Alliance Project	1.2	Nov 12, 2003
WS-Federation	BEA, IBM, Microsoft, RSA Security, Verisign	1.0	Jul 8, 2003
LID	NetMesh	3 Jan 2005	Jan 3, 2005
SXIP	SXIP Identity	1.0.4	Sep 23, 2004
Shibboleth	Internet2 Project	Working Draft 09	Feb 28, 2005

These are the most advanced systems that at least partially comply with re requirements specified in section 1.1. There are other similar systems that will not be considered in detail:

- *Kerberos* is a SSO system based on symmetric cryptography. It has limited cross-domain capabilities and due to the use of symmetric mechanisms is not suitable for the Internet environment. Kerberos is not directly applicable for cross-domain web SSO.
- *Microsoft passport* is a system that is very similar to the considered SSO systems. The Passport design and deployment has many drawbacks [3] and the vendor announced the end of the system marketing.
- *Yale University Central Authentication Service* is an authentication and SSO system. It has few interesting features (e.g. proxy support), but does not provide privacy and attribute exchange.
- *xns.org (Identity Commons)* is an initiative to the create a framework for trusted electronic communications based on XRI and XDI specifications. But sufficient technical documentation was not available for further study of the system.
- *X.509 Public Key Infrastructure (PKI)* is a system based on public key certificates. In its current form the certificate can be used as a universal global identifier, that may not efficiently preserve user privacy.
- *Digital Credentials* [4] is a cryptographic system based on secret-key certificates that was designed to overcome some PKI problems. There is no complete specification for an Internet digital credential-based SSO system at the time of this writing, and it is likely that such a system will require the modification of standard Internet browser to work efficiently.

The following sections describe principles and methods that different SSO systems employ and how it fulfill the requirements defined earlier. Only an overview of the systems architecture is given and it is focused on SSO features only. Attribute services are considered only marginally. Persona linking methods and overall fitness of the system for the Internet environment is evaluated.

3.1 Liberty Identity Federation Framework (Liberty ID-FF)

The Liberty Alliance Project is a group of industry and non-commercial organizations whose objective is to prepare an open standard for the network identity systems. Decentralization and openness are the main goals of the alliance, their effort aims at providing federated identity. The Liberty Identity Federation Framework (ID-FF) specification [5][6] set is a product of the first phase of the Liberty Alliance Project. It defines a SSO system with support for federated identities.

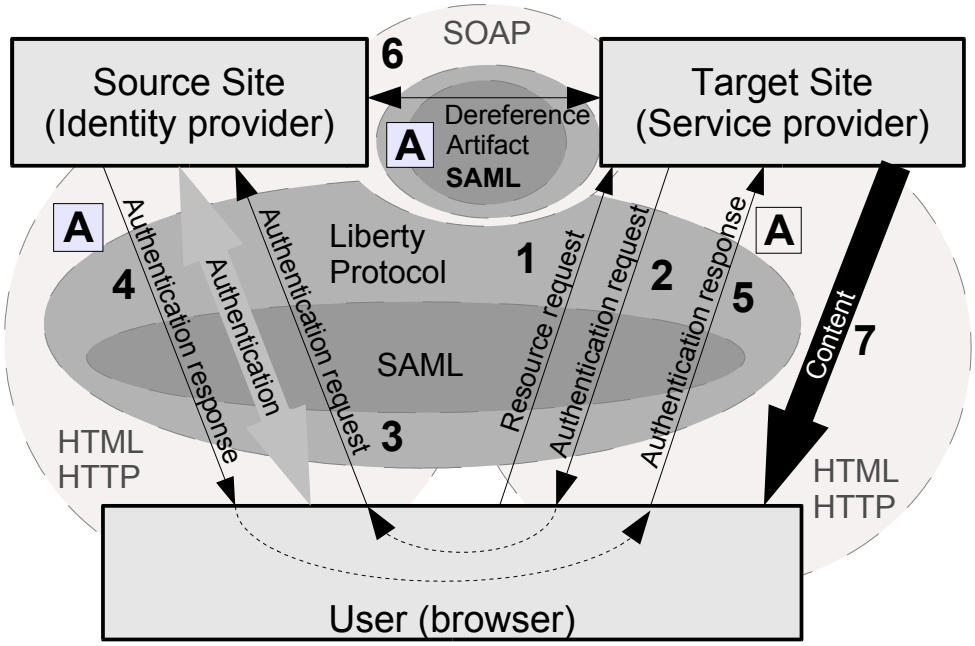

Figure 5 Liberty Artifact profile for single sign-on

The Liberty ID-FF follow the generic model described in section 2 and use Security Assertion Markup Language (SAML) [7] assertions as a security tokens. Assertions are either passed directly in the HTML forms or are referenced by smaller artifacts. Liberty ID-FF specifications define several profiles for different combinations of these techniques. The example interaction using the Liberty Artifact Profile is depicted on Figure 5.

The Liberty ID-FF supports pseudonymity as a default behavior. When creating persona identifiers for target systems (`NameIdentifier` tag in SAML assertions), it is required to be a pseudo-random value that have no discernible correspondence with the original persona identifier. The linking itself is carried out by associating pseudonyms, not primary persona

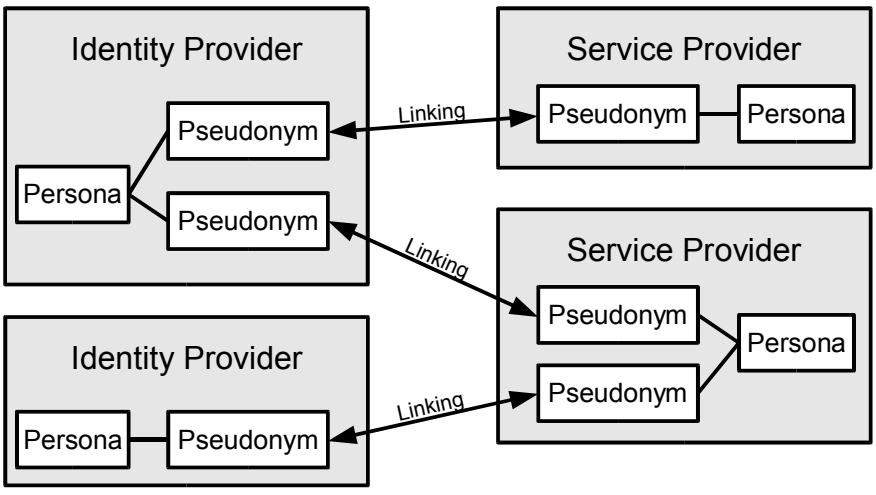

Figure 6 Liberty persona linking

identifiers[1], as illustrated on Figure 6. It is also possible to make linking chains by linking accounts managed by different Identity Providers, which will enable to form a higher-level structure similar to Certificate Authority cross-signing in X.509 based PKIs. The relations are always implemented by linking a pair of identifiers (pseudonyms) combined with the provider identifiers, so there is no need for a global persona identifier space.

3.1.1 Discussion

The Liberty ID-FF system is build on top of SAML and is dependent on it. The modification of Liberty specification for other, non-SAML security tokens may be difficult.

The Liberty specifications mandates the use of pseudonyms by default. This requirement may help to enforce good privacy features to all Liberty-compliant implementations.

3.2 Web Services Federation Language (WS-Federation)

The Web Services Federation Language (WS-Federation) specification [8] define a mechanism for identity federation. It's primary focus is on federating services (computer-to-computer interactions), but it also addresses SSO issues in WS-Federation: Passive Requestor Profile specification [9]. The WS-Federation specifications builds on other documents, especially on WS-Trust [10] and WS-Security [11] and are published as a public drafts.

The WS-Federation Passive Requestor Profile uses the general mechanisms described in the section 2. The exchanged messages are XML formated according to the WS-Trust specification, with several additional service attributes. The sites may use URL references instead of direct message exchange. The example message exchange is illustrated on Figure 7.

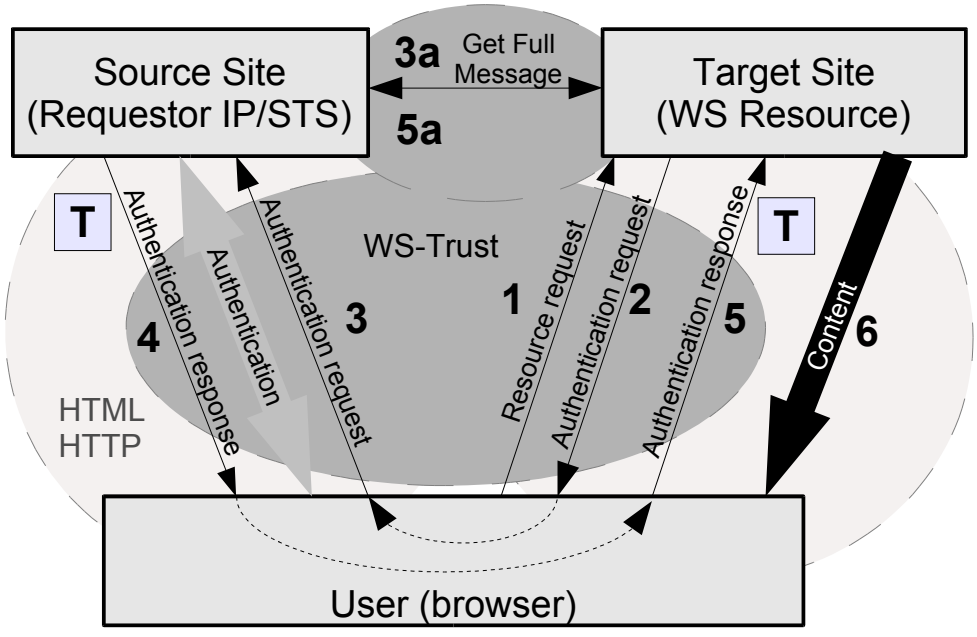

Figure 7 WS-Federation passive requestor profile single sign-on

1 This is required for Identity Providers and recommended for Service Providers.

No specific format for security token is mandated by WS-Federation specifications, the security token formats are specified by WSS: SOAP Message Security (WS-Security) profiles. At the time of writing this document, the security tokens described in the following table were defined.

Security Token	*Description*
Username	Provided for simple username/password authentication
X509	Provides authentication by X.509 certificate
REL	Provides support for Rights Expression Language
SAML	Provides support for SAML assertions

The source site (Requestor IP/STS) may also provide attribute and pseudonym services. However, the use of pseudonyms is not mandatory and no strict pseudonym models are defined. Some of the possible persona linking scenarios are illustrated on Figure 8.

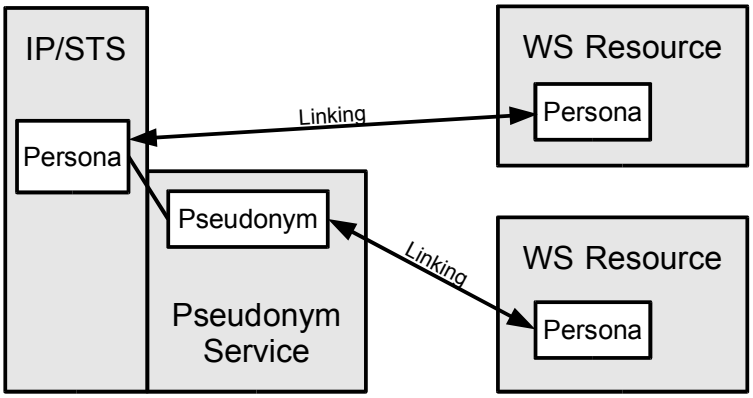

Figure 8 WS-Federation persona linking

3.2.1 Discussion

The WS-Security specifications leaves a lot of details to the implementer's decisions and to be defined by the service policy. Although that is good for flexibility, it brings additional degree of uncertainity to the system. The implementing systems may not be interoperable by implementing different subsets of specifications and/or using non-compatible policies.

The privacy decisions (e.g. use of pseudonyms) is left to the implementers. This may in practice lead to the implementations, that will not adhere to the best practice and the level of privacy in WS-Federation-compliant systems may be lower (in average).

3.3 Shibboleth

Shibboleth [12] is a web single sign-on and attribute exchange system built on SAML specifications [7]. It follow the model described in section 2. It uses a modified SAML protocol for communications and SAML assertions as security tokens (Figure 9). Shibboleth adds optional WAYF service for identity provider selection.

The specifications does not limit the use of NameIdentifier types in the SAML assertion, but defines Shibboleth-specific transient name identifier. Shibboleth specification recommends that if such transient identifier is used, it should be used only once. Transient identifiers can be used for transient persona linking, and may be used for subsequent attribute exchange using shibboleth attribute services. This model is illustrated on Figure 10.

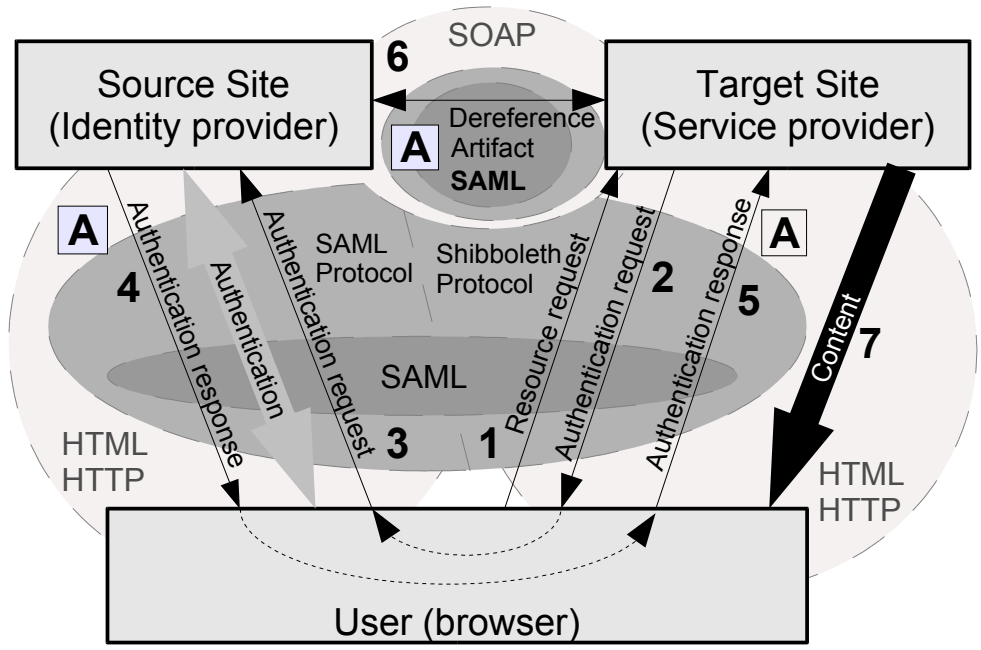

Figure 9 Shibboleth Single Sign-On

3.3.1 Discussion

The system implemented using Shibboleth specifications will need to specify a lot of local details, e.g. name identifier types, linking policies, etc. This type of flexibility may lead to situation that two shibboleth-compliant implementations will not interoperate.

The shibboleth will depend on other specification to define persistent persona linking, if such will be needed. The use of transient name identifiers allow good degree of privacy, but for any practical purpose it will require a solid attribute service.

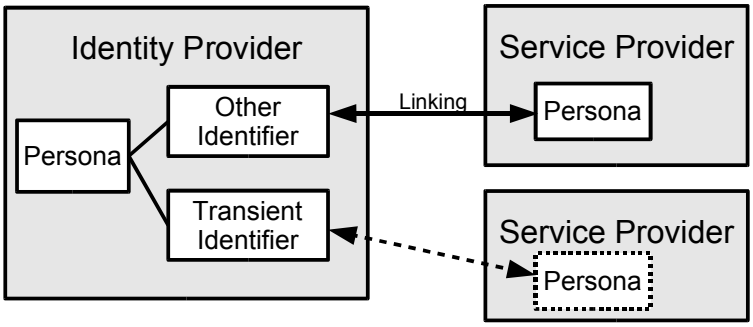

Figure 10 Shibboleth persona linking

3.4 SXIP Network

SXIP Network is open identity network that is based on the Simple eXtensible Identity Protocol (SXIP) [13]. The SXIP Network provides SSO and attribute services to the participating sites. The SSO mechanisms follow the principle similar to the one described in section 2, but it introduces a central Rootsite that manages global persona identifiers. The SXIP SSO message exchange is illustrated on Figure 11. The SXIP protocol defines two communication options:

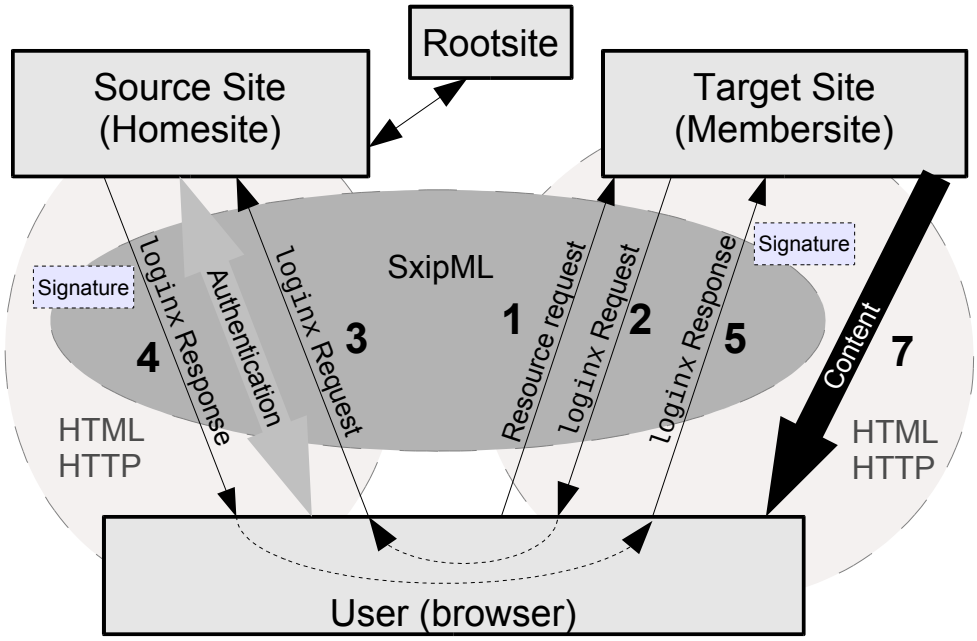

Figure 11 SXIP Network Login (`loginx` command)

- *Simple Commands* provide simple, parameter-based method. The protocol data are transferred in HTTP parameters or HTML form fields. This method does not include any security token or any other form of intra-protocol security measure.
- *XML Commands* provide richer, XML-based interface. The exchanged messages are represented in SXIP Markup Language (SxipML) [14]. This method uses XML Digital Signature [15] element as a security token, but it is not included in all messages.

Personae are identified by 64-bit Globally Unique Persona Identifier (GUPI) assigned by the Rootsite during persona registration process. The GUPI is used as an universal identifier for a persona at all target sites (membersites). This model is illustrated on Figure 12.

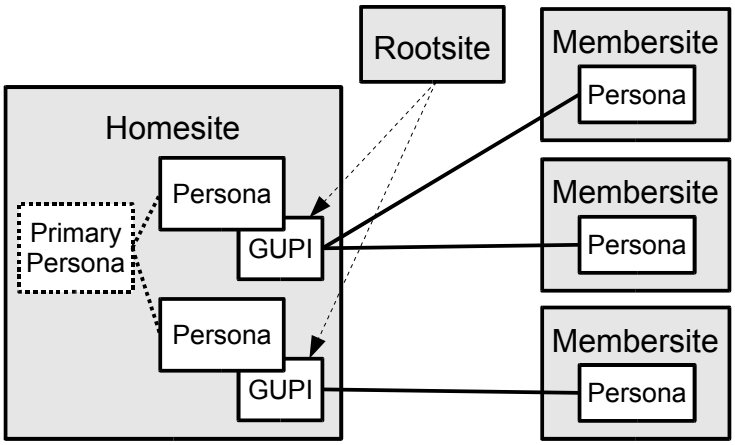

Figure 12 SXIP persona linking

3.4.1 Discussion

Protocol variation using simple commands provides only minimal security. No intra-protocol security token is passed, the security is left to the external means. Even the use of HTTPS does not add any real security to the simple command exchange. The simple command exchange should be considered dangerous for most deployments and all simple (non-xml) SXIP commands should be disabled in production deployments.

Some of the SXIP XML commands include XML digital signature element, that should protect the integrity of SxipML message. However, no specific methods or guidelines are documented for creation and validation of these signatures.

The `storex` and `fetchx` command messages are not authenticated, which may lead to the implementations that may allow anyone reading or setting arbitrary persona attributes.

The use of globally unique GUPI at several membersites makes it easy for the membersites to collude and correlate persona activities at several sites. This is partially mitigated by the use of different personae for different membersites. However, in the extreme case each membersite will require separate persona (and GUPI) to avoid possibility of collusion. As the GUPIs are assigned by rootsite and the assignment is governed by the rootsite policies, this approach may be unconvenient or maybe even unfeasible.

The GUPI is assigned to the persona on a specific homesite by rootsite. This assignment is claimed by `authDelegation` SxipML element. The `authDelegation` element is signed by Rootsite and includes an expiration time. The correct setting of expiration time it is the only way of limiting the validity GUPI delegation. If the expiration time interval is long (more that few hours), it will considerably limit the ability to migrate persona from one homesite to the other. If the expiration time interval is short (minutes or hours), it may reveal some GUPI usage patterns to the Rootsite.

3.5 Lightweight Identity (LID)

Lightweight Identity (LID) [16] is a web-based SSO system, that also allows sharing of additional data. LID is implemented using mechanisms similar to the one described in the section 2. The GNU Privacy Guard (GPG) [17] signatures on message parameters are used as a security tokens. The example message exchange during SSO interaction is depicted on Figure 13.

The credential supplied in SSO approval message is a GPG signature of response parameters. The credential is verified by the target site by getting corresponding public key using LID URL and validating the signature.

The LID documentation mentions pseudonyms, but these are in fact separate personae that may have been linked by unspecified means. The model of LID persona linking is depicted on Figure 14. Note that the only option to make association between source and target persona is direct linking, because the accounts on the target sites use LID URL as an account identifier. The indirect linking may occur on the source persona level, when linking several personae by SSO or attribute value delegation, but this process is not sufficiently documented.

3.5.1 Discussion

LID URL as an identifier may leak information, especially in self-hosting scenario as proposed by LID documentation. For detailed explanations see section 3.6.3.

GPG public key validation is left on simple “callback” method. No other method is mandated by the LID documentation (although it is allowed). The described simple method can be dangerous when using HTTP protocol, for example due to the DNS attacks [18] (note that

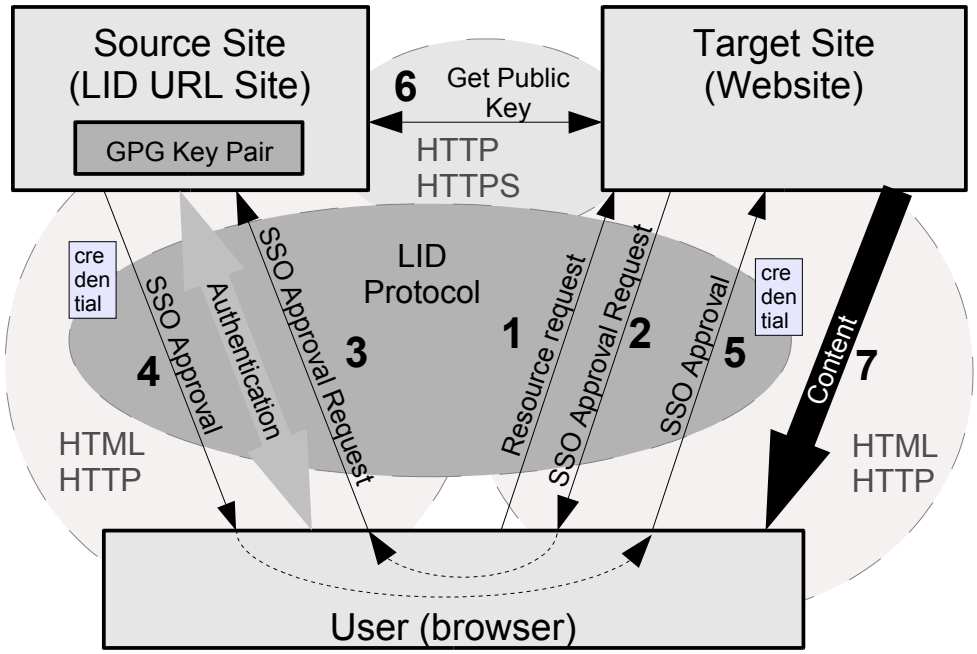

Figure 13 LID Single Sign-On

DNS attacks may interfere with other LID functions, if LID URL specifies `http` scheme). While using HTTPS method to get public key, using SSL/TLS brings a dependency on X.509 PKI. The result is that LID uses two different PKI systems (X.509 and GPG), that are in principle and features very similar, but not compatible. Both of these systems must work properly for a safe LID operation: a problem in either one may result in the compromise of LID security as a whole. Validation of GPG public key using the GPG web-of-trust feature might be more desirable in this case, but such a validation may not be trivial and is not specified in the LID documentation.

LID pseudonyms are created as a different LID URLs. Getting a pseudonym that is indistinguishable from primary LID URL may not be easy, as it will frequently lead to getting a new DNS domain or hosting space in existing domain and installing a LID software. This

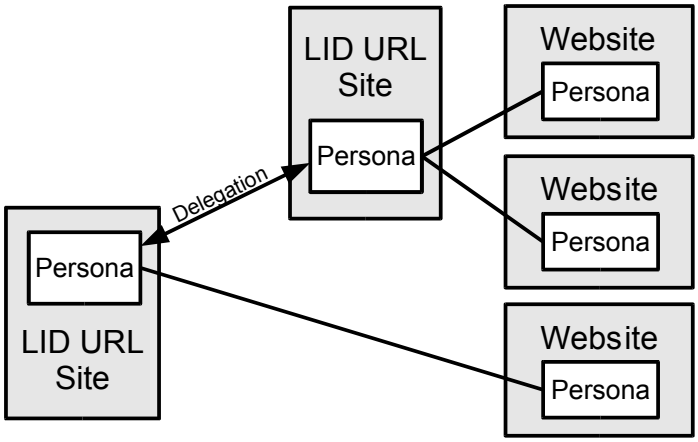

Figure 14 LID persona linking

process is difficult to automate and for that case it may be difficult to implement good pseudonymity or anonymity functions using LID specifications.

All user attributes has to be stored under the LID URL control. That may be undesirable as this means that there exists a centralized point that has all the information about user (or at least a persona). The LID documentation mentions, that the attribute processing may be delegated from pseudonym to the primary URL, but such a delegation will result in the pseudonym LID URL management software to see the attribute value anyway. There is no mechanisms to refer requesting website to other URL for the attribute value retrieval.

3.6 Summary

The considered web SSO systems are similar in the generic SSO mechanism, but are different in the following areas:

- The *method of identifying* personae, the way of generating and assigning identifiers. Global identifiers are better suited for tightly-coupled systems that are same organizational control or share common policies. Local identifiers are better suited for loose-coupled systems that cross organizational boundaries.
- The *method of persona linking* in different target and source systems. The use of direct and indirect linking methods, or transient linking, may be useful in different environment. The direct linking is desirable only when user privacy is not a concern, it is not well suited for the Internet environment. The pseudonymity of indirect linking or anonymity of transient linking is better suited to privacy-sensitive environments. The transient linking SSO case will in practice require secure and interoperable attribute service.
- The *level of detail* that is specified in the documents and the freedom that is left for system implementers. Where few details are specified, the mechanism may provide much flexibility for different environment, but specific deployments on Internet scale will require better specification to be interoperable. The profile documents will be needed for these specification to work well on the Internet. The specifications that provide much details may not be flexible for any environment, but may be immediately implemented and deployed with high probability that different implementations will be interoperable.

The following table summarizes features of considered SSO systems and the next subsections provides discussion on some aspects of SSO system's architecture and design.

System	***Security Tokens***	***Linking Method***	***Persona Identifier***	***Extensible to Web Services***
Liberty ID-FF	SAML	Indirect	Local	Yes
WS-Federation	WS-Trust: Username, X.509, REL, SAML	Not specified	Not specified	Yes
LID	GPG Signature	Direct	Global (URL)	No
SXIP	Simple: None XML: XML digital signature	Direct	Global (GUPI)	No
Shibboleth	SAML	Transient, other	Transient, other	No

3.6.1 Common Issues

The Internet SSO systems that follows the described method usually use short-lived security tokens based on public-key cryptography. The common drawback of these systems is the ability of source site to track the user's log-ons on service providers site. While the source site cannot track the user's activity at these sites, the log-ons itself may provide sufficient information to potentially violate user's privacy.

The general property of single sign-on systems is the ability to trivially impersonate the user [1]. The source site that exploits this ability may use the target's site interface to read the target site persona attributes.

3.6.2 Persona Identifiers

The Single Sign-On systems need a way to link several personae. The linking is implemented by associating persona identifiers on different systems. There are two approaches to the management of persona identifiers:

- *Global persona identifiers.* The persona identifiers are allocated by central authority that guarantees global uniqueness of the identifier. The examples of these identifiers are LID URL or SXIP GUPI. The global uniqueness of the identifier allows direct linking of personae on the global (Internet) scale.
- *Local persona identifiers.* The persona identifiers are allocated by the system, where the persona originated. These identifiers may be unique only in the scope of the source system. For the purposes of persona or account linking, the target site must accept the identifier in this form or (more frequently) apply appropriate identifier mapping.

While the direct linking and global persona identifiers may be the easiest scenario, global identifiers shared by many sites may be used to correlate user activities on several systems and thus reveal personal information without user consent. To overcome this problem, lower level virtual personae (with different identifiers) may be used as pseudonyms. If this approach is deployed in the Internet scale, the persona management may become difficult and may need automation. The automatic pseudonym persona management is technically close to the indirect linking scenario, and the indirect linking may be considered as better approach for the Internet environment.

Global persona identifier is not required in indirect linking scenarios. The link is defined as a pair of persona identifiers, each unique in the context of own system. The identifiers combined with the peer system identifier (need not be globally unique) is sufficient to uniquely define a link on both source and target system.

3.6.3 Self Hosting of Source Sites

One way of storing identity information is to host a source site on a system, that is under user's sole control. That may be a separate computer system or a dedicated portion of shared system, which in common case is represented by URLs that the user controls.

As this concept may seem attractive from the privacy point of view, it in fact may be undesirable in the practice:

- URLs of self-hosted source site may leak information. Parts of personal information may be direct part of the URL, for example domain name or path prefix may contain user's name. Additional information may be leaked by DNS records (e.g. SOA record), IP addresses or DNS and IP address databases (e.g. Internet Registries).
- The maintaining of site security on the operating system and application level is a difficult and never ending process. It is not likely that common user will have necessary knowledge and skills to implement and maintain all required security controls. In the case

of outsourcing the system administration or security maintenance of the hosting system, the user is no longer in sole control of the data.

- The trust relation between source site and target site may not be unidirectional. For example in electronic banking scenarios the bank may be made responsible to enforce sufficient authentication level. For this reason, the bank (target site) must trust the the authentication provider (source site) to authenticate user according to the agreed regulations. The trust to the source site in the self-hosting scenario may be questionable, and the high number of self-hosting sites with whom to establish trust may make the process unfeasible.

The self-hosting scenario is technically proxy-based true SSO system [1], but may be regarded a local true SSO system from the organizational control point of view. The self-hosting of source sites brings only one advantage: control over the stored data. But the control over data is lost when transmitted to other sites and even control of the stored data itself may be questionable. The self-hosting scenario will in most cases likely lower the privacy and/or security level.

Note that the self-hosting scenario as described here refer to the hosting of data for single user (may be represented by several personae). It does not include so called attribute providers, that store data for many users but do not provide user authentication.

4 Conclusion

This document described the generic model for Internet Single Sign-On mechanisms and provided an overview of existing Internet SSO systems. Each system was considered and its fitness for the Internet environment was evaluated.

The Liberty Alliance ID-FF and the WS-Federation were found as the most advanced and flexible Internet SSO systems. These two systems are suitable for general Internet use.

The WS-Federation specifications are quite generic, lack a considerable amount of details and the early WS-Federation implementations may have interoperability problems. However, the WS-Federation may become a good platform for SSO services in the future, extended to the web services area as well.

The Liberty Alliance ID-FF specifies an practical SSO system build on SAML specifications. The level of detail is sufficient for good interoperability, the dependency on SAML is reasonable in the Internet environment. The Liberty Alliance also specifies extensions to ID-FF for web services environments (ID-WSF).

The Shibboleth specifications also depends in SAML, but the level of details is considerable lower compared the the Liberty case. It is expected that the specific Shibboleth implementation will supply additional details. The Shibboleth system is suitable for large communities, that are mostly composed of independent organizations (e.g. academic community).

The SXIP and LID SSO systems in it's current state are not well suitable for the Internet environment. They provide only minimal privacy features, use global identifiers and feature limited standards support. These systems may be suitable for closed communities or for the environments where security and privacy is not a concern.

The common properties of Internet SSO system, especially the ability to impersonate the user and track user log-ons at target sites may pose a threat to user privacy. The source site must be trusted by both the user and service providers.

While all the evaluated systems use similar mechanisms, their properties vary considerably. Especially the use of persona identifiers and pseudonyms as well as the use of attribute services will require further study for the SSO systems to be deployed in secure and privacy-supporting manner.

Bibliography

[1] Pashalidis, A., Mitchell, C.: A taxonomy of single sign-on systems, Information Security and Privacy, ACISP 2003, 2003

[2] Pfitzmann, A., Köhntopp, M.,: Anonymity, Unobservability, Pseudonymity, and Identity Management A Proposal for Terminology, Designing Privacy Enhancing Technologies, International Workshop on Design Issues in Anonymity and Unobservability, 2000

[3] Semančík, R.: Internet Applications Security, Written part of Ph.D. exam, 2002

[4] Brands, S.: Rethinking Public Key Infrastructures and Digital Certificates, 2000, ISBN 0-262-02491-8

[5] Cantor, S., Kemp, J.: Liberty Bindings and Profiles Specification, Liberty Alliance Project Specification, 2003

[6] Cantor, S., Kemp, J., Champagne, D.: Liberty Bindings and Profiles Specification, Liberty Alliance Project Specification, 2003

[7] Maler, E., Mishra, P., Philpott, R., et.al.: Assertions and Protocol for the OASIS Security Assertion Markup Language, OASIS Standard, 2003

[8] Bajaj, S., et.al.: Web Services Federation Language (WSFederation), BEA, IBM, Microsoft, RSA Security, Verisign, 2003

[9] Bajaj, S., et.al.: WS-Federation: Passive Requestor Profile, BEA, IBM, Microsoft, RSA Security, Verisign, 2003

[10] Anderson, S., et.al.: Web Services Trust Language (WS-Trust), , 2005

[11] Nadalin, A., et.al.: Web Services Security: SOAP Message Security, OASIS Standard, 2004

[12] Shibboleth Architecture Protocols and Profiles, http://shibboleth.internet2.edu/shibboleth-documents.html

[13] The Simple eXtensible Identity Protocol (SXIP) Reference, https://sxip.net/archive/specs/sxip-reference.pdf

[14] The Sxip Markup Language (SxipML), https://sxip.net/archive/specs/sxipml-spec.pdf

[15] Bartel, M., et.al.: XML-Signature Syntax and Processing, W3C Recomendation, 2002

[16] Ernst, J.: Light-Weight Identity, NetMesh Inc., 2005

[17] The GNU Privacy Guard, http://www.gnupg.org/

[18] Bellovin, S. M.,: Using the Domain Name System for System Breakin, 5th USENIX UNIX Security Symposiu, 1995