Web Single Sign-On Systems

Shakir James , scj1@cse.wustl.edu

Abstract:

Currently, many web applications require users to register for a new account. With the proliferation of web applications, it has become impractical to expect users to remember different usernames and passwords for each application. Web Single Sign-On (Web SSO) protocols allow users to use a single username and password to access different applications. This paper examines three Web SSO protocols: SAML Web Browser SSO Profile, WS-Federation Passive Requestor Profile, and OpenID.

Keywords:

Single Sign-On, Web Single Sign-On, Web SSO, Brower-based SSO, Federated Identity Management, Identity Federation, SAML, WS-Federation, OpenID

Table of Contents

1. Introduction

Today the proliferation of web applications force users to remember multiple authentication credentials (usernames and passwords) for each application. Faced with the impractical task of remembering multiple credentials, users reuse the same passwords, pick weak passwords, or keep a list of all usernames and passwords. Managing multiple authentication credentials is annoying for users and weakens security for the authentication system. Web Single Sign-On (Web SSO) systems allow a single username and password to be used for different web applications. For the user, Web SSO systems help to create what is called a *federated identity*.

Federated identity management benefits both the user and the application provider. Users only remember one username and password, so they do not have to suffer from *password-amnesia*. Application providers also reduce their user management cost. They neither need to support a redundant registration process nor deal with one-time users creating many orphan accounts. Therefore, all stakeholders benefit from a Web SSO standard. Microsoft's Passport protocol [Microsoft01] was the first attempt at creating a Web SSO standard, but the protocol was never widely deployed by non-Microsoft vendors. The fact that major security flaws in Passport were found by [Kormann00] may have hindered its adoption.

Passport is not the only Web SSO protocol that had vulnerabilities, flaws in others protocols were identified by [Pfitzmann03, Groß03]. Although Web SSO is conceptually similar to Single Sign-On (SSO) protocols like Kerberos, they must operate within limitations of commercial browsers. Web SSO systems are proxy-based true SSO systems [Pashalidis03]. Web SSO protocols are also called browser-based or *zero-footprint* protocols since they operate within the constraints of web browsers. Furthermore, messages exchanged in browser-based protocols must be encapsulated in Hypertext Transfer Protocol (HTTP) supported by all commercial browsers since users should not be expected to install software to support a new protocol. Moreover, browser cookies are tied to a single domain so cannot be used for the multi-domain SSO (MDSSO).

MDSSO refers to the case where SSO occurs between security domains operated by disparate organizations. For example, in the web-based MDSSO, Alice uses her browser to book her flight at airlineinc.com and is transparently logged in at hotelinc.com to reserve her room. A trust relationship must exist between the two domains to support this federated authentication since the domains may have different security polices. This paper does not cover the trust establishment problem. It also does not focus on the federation of attribute and authorization data. It deals with federated authentication of users in the web-based MDSSO case.

Security analysis of web-based MDSSO protocols demands unconventional requirements and models, and to date only one protocol has been proved cryptographically secure [Groß05]. In the paper, we focus on the three-party authentication protocols for Web SSO since these protocols have been analyzed by [Groß03, Groß05]. The three parties include a *service provider*(SP), *identity provider*(IP), and *user agent*(UA). The UA is the web browser operated by the user, the IP authenticates the UA, and the SP is the web application.

1.1 Related Work

In contrast to MDSSO web-based systems, some Web SSO systems provide SSO within a single organization. They are called Web Initial Sign-On (WebISO) systems. The Internet2 WebISO Working Group [WebISO03] attempted to standardize such systems, but the project is currently inactive. WebISO systems typically provide a web-based component to an organization's existing single sign-on infrastructure, which currently does not support web-based authentication. Organizations have direct control over their security policies, authentication procedures, and direct access to their user database. Thus, cross-domain issues such as establishing trust relationships are not a major concern in WebISO systems. Many universities have developed and deployed custom WebISO systems [WebISO03]. Also notable is another Internet2 project called Shibboleth [Shibboleth07]. Shibboleth handles both the web-based MDSSO and SSO within an organization, typically a university, and is implements SAML v1.1.

In addition to federation authentication, MDSSO federated identity management also involves attribute-based authorization, pseudo-identifiers for privacy, single logout, and IP discovery. Typically most Web SSO protocols address these problems in conjunction with federated authentication. A mechanism for establishing trust relationships between organizations is also covered by SSO protocols.

1.2 Organization

Section 2 of the paper provides an overview of the Security Assertion Markup Language (SAML) standard as a frame of reference for presenting other standards in later sections. Section 3 describes the SAML Web Browser SSO profile in particular. The Web Services Federation Language (WS-Federation) Passive Requestor profile, a Web SSO secure [Groß05] Web SSO protocol, is described in section 4. OpenID, an open-source Web SSO protocol, is presented in Section 5. We finish with some closing remarks in section 6.

Back to Table of Contents

2. SAML

The Security Services Technical Committee (SSTC) of the Organization for Advancement of Structured Information Standards (OASIS) developed Security Association Markup Language (SAML). OASIS is a global organization that produces more web standards than any other organization [OASIS]. SAML is based on Extensible Markup Language (XML) and is used for communicating authentication and authorization data between web domains. It is arguably the de facto standard for federated identity management. Successful SAML implementations exist in industry, government, and academia [Wisniewski05]. Many identity management products support SAML. Moreover, other standards such as the Internet2 Shibboleth project, Liberty Alliance, OASIS Web Services Security (WS-Security), and eXtensible Access Control Markup Language (XACML) are based on SAML [Wisniewski05].

The modular design of the SAML framework allows its components to be combined to support a wide variety of deployment scenarios. SAML consists of *core*, *bindings*, and *profiles* components. Figure 1 shows the relationship between the SAML components. The profile component describes the context in which SAML is used, and bindings specify the protocol used to encapsulate SAML messages. Bindings and profiles are based on the SAML core, which defines the format of messages and the generic request/response protocols.

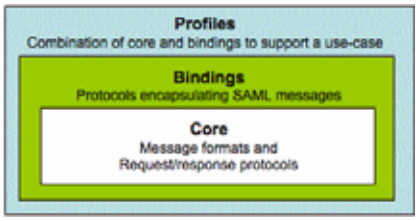

Figure 1: Relationship between profiles, bindings, and the core in the SAML framework.

2.1 Core

The core consists of *security assertions* that define the syntax and semantic of messages, and general request/response *protocols* for transferring assertions [Cantor05]. Assertions are XML packages that carry SAML *statements* about the user. For example, authentication assertion may contain statements saying how and when a user was authenticated. The request/response protocol is also specified in XML. In the protocol, a request is a query for an assertion and a response returns either the assertion or an error. For example, a request may be for a principal to be authenticated and the response will be an authentication assertion saying whether authentication was successful. The encapsulation of the SAML core, assertions and protocols, in another common underlying protocol is called a binding.

2.2 Bindings

SAML bindings specify how SAML protocol messages map to other common protocols such as Simple Object Access Protocol (SOAP) or HTTP [Cantor05]. Bindings use standard communications and messaging protocols to allow autonomous SAML-compliant systems to securely transfer messages. Either SAML or the underlying protocol supports mutual authentication, message integrity and confidentiality. For example, in the SOAP binding, either SOAP or SAML can be secured. XML signatures and encryption are used for application level security and TLS/SSL for transport layer security. The core and bindings define a SAML use-case called a profile.

2.3 Profiles

SAML profiles specify how SAML core and bindings are used within a specific application [Hodges06]. For example, the Web Browser SSO profile uses the Authentication request protocol and bindings for HTTP and SOAP. Many types of SAML profiles exist, but SSO specific profiles include the Web Browser SSO Profile, Enhanced Client or Proxy (ECP) Profile, Identity Provider Discovery Profile, Single Logout Profile, and Name Identifier Management Profile [Hughes05]. Although SAML defines many profiles, Web SSO was the primary reason for developing the standard [Lockhart06a].

Back to Table of Contents

3. SAML Web Browser SSO Profile

In SAML v1.1, Web SSO was implemented in two separate profiles: the Browser Artifact and Brower POST profile. In SAML v2.0, a single Web Browser SSO profile was created. The new profile incorporates the two older profiles as bindings. The HTTP Redirect, HTTP POST, and HTTP Artifact in conjunction with the Authentication Request protocol implements the Web Browser SSO profile [Hughes05]. Each binding defines a different means of encapsulating the authentication assertions. In particular, Extensible Hypertext Mark-up Language (XHTML) forms transport request/respond messages by value and by reference in the HTTP POST and HTTP Artifact binding respectively.

3.1 Framework

Although the binding determines the actual messages, the exchange follows a generic model irrespective of the choice of binding [Lockhart06a]. Figure 2 depicts this general exchange pattern.

(1) The UA attempts to access a resource at the SP. Assuming the UA is not authenticated at the SP, the SP determines the IP of the UA. (The Identity Discovery profile can be used to determine the IP of a UA.)
(2) The UA conveys the authentication request message on behalf of the SP to the IP.
(3) By some unstated method, the IP authenticates the UA.
(4) The IP responds to the SP by relaying the message through the UA.
(5) The SP either grants or denies access to resource based on the IP response.

To transfer the message through the UA to the SP/IP the HTTP POST, HTTP Artifact, or HTTP Redirect binding can be used. However, the HTTP Redirect binding cannot be used in (5) because of the length of the response [Lockhart06a]. For IP-initiated authentication, the exchange begins at (4).

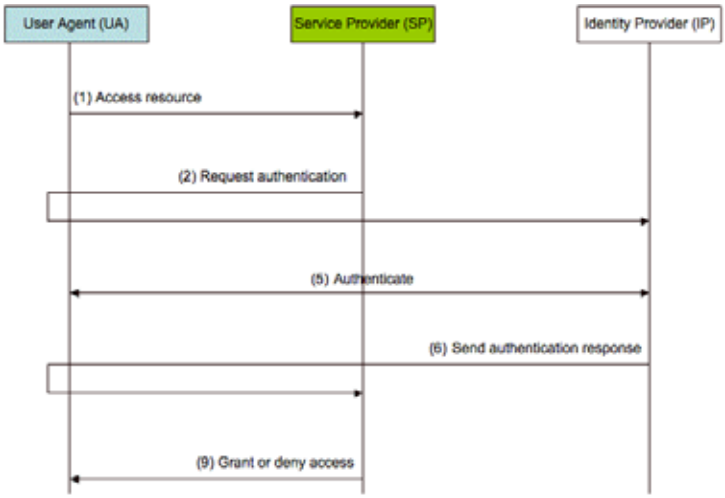

Figure 2: SAML message exchange model for achieving Web SSO.

As a real world example, in Figure 3, we describe the deployment of the Web SSO profile using the HTTP Arti fact binding. An artifact is a reference to a message. An artifact can be resolved to get the content of the message.

(1) The UA attempts to access a resource at the SP.
(2) To request SSO service at the IP, the SP issues a request artifact (a reference to the request message) to the ID using the UA as a middleman.
(3) The IP asks the SP to resolve the request artifact.
(4) The SP responds with a message containing the original request message [Cantor05a].
(5) The ID authenticates the UA
(6) The ID sends a response artifact to the SP again using the UA as a middleman.
(7) The SP asks the ID to resolve the response artifact.
(8) The ID returns the content of the original response.
(9) The SP grants or denies access to the resource based on the IP response.

For security, the specification recommends the artifact be transfer to/from the UA using a secure channel (SSL/TLS) and the SP and IP use source authentication before sending the contents of the original message.

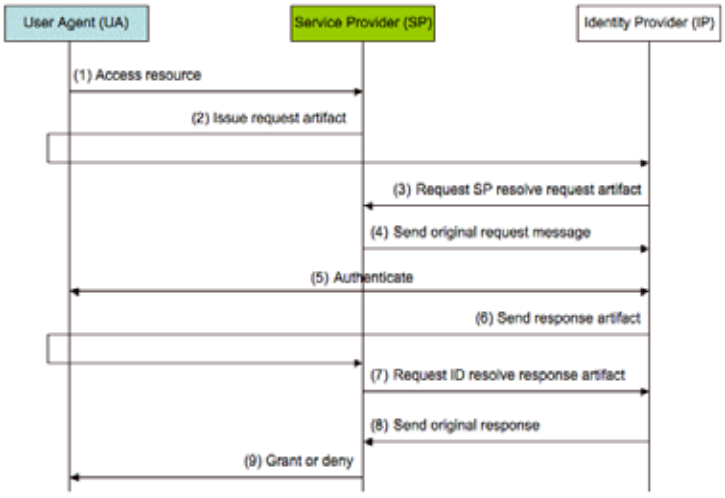

Figure 3: Specific model for achieving Web SSO using the Artifact binding.

3.2 Security Issues

A security analysis of the Browser/Artifact profile in SAML v1.1 was presented by [Groß03]. Groß identifies several vulnerabilities in the protocol including man-in-the-middle, message leakage, and replay attacks. He offers counter-measures, and recommends the use of "secure channels such as SSL 3.0 or TLS 1.0 with unilateral authentication" [Groß03]. Groß also notes that secure channels prevent man-in-the-middle and replay attacks, but not message leakage. Although he describes the protocol as the "most carefully designed browser-based protocols in federated identity management," he concludes that its security vulnerabilities make it unsuitable for use as an industry standard.

The SSTC responded to the [Groß03] in [SSTC05]. They incorporated some recommendations made by [Groß03] into SAML V2.0, and added counter-measures to threats [Groß03] identified. In general, the SSTC argues that vulnerabilities in underlying protocols, which SAML operates above, are not weaknesses in the SAML specification per se. They also point out that the SAML specification recommends using /TLS secure channels, which counter two of the three attacks identified in [Groß03].

Back to Table of Contents

4. WS-Federation Passive Requestor Profile

Similar to SAML, the WS-* specifications is a modular architecture that aims to provide a flexible and extendable framework to solve the general web security problem [Lockhart06]. The Web Services Federation Language (WS-Federation) is part of the WS-* specifications that addresses the Web SSO problem for both browser-based applications and web services in particular [IBM07]. IBM and Microsoft and other companies developed the standard. The Passive Requestor profile targets browser-based SSO and other web-enabled devices, and the Active requestor profile focuses on SOAP-based web applications. Like SAML, WS-Federation can be combined with different transport and application layer protocols to

implement various Web SSO scenarios.

Also like the SAML Web Browser SSO profile, WS-Federation is based on different subcomponents. WS-Federatio n uses WS-Security framework, WS-Trust, and WS-SecurityPolicy [Goodner07]. WS-Security, standardized by OASIS, provides the mechanism for securing SOAP messages by using security tokens. WS-SecurityPolicy allows services to describe their security requirements, and WS-Trust defines the Security Token Service protocol for requesting/issues *security tokens*. WS-Federation extends the Security Token Service (STS) model of WS- Trust.

4.1 Framework

WS-Federation builds on the WS-Trust STS model [Nadalin07] to provide, among other things, a Web SSO service. In the STS model, a security token is needed for parti es to communicate. The STS assigns these security tokens [Lockhart06]. The UA first queries the SP to determine its security requirements before trying to access a resource [Goodner07]. The UA checks if it has a valid token using WS-SecurityPolicy expressions. If the UA does not have a valid token, it request one from the STS. Since the STS is a web service, it also has its own security requirement. As a result, the UA must present a security token to the STS before requesting a token for the SP.

In the WS-federation Passive Requestor profile, the STS plays the role of the IP authenticating the UA [Bajaj03]. The HTTP POST protocol is used to transfer security tokens. The message exchange pattern is shown in Figure 4.

(1) The UA tries to access a resource at the SP.
(2) Assuming the UA does not have a valid token, the UA is redirected to the IP.
(3) The IP authenticates the UA by assigning it a valid token.
(4) The UA present the access token to the SP to gain access to the resource.

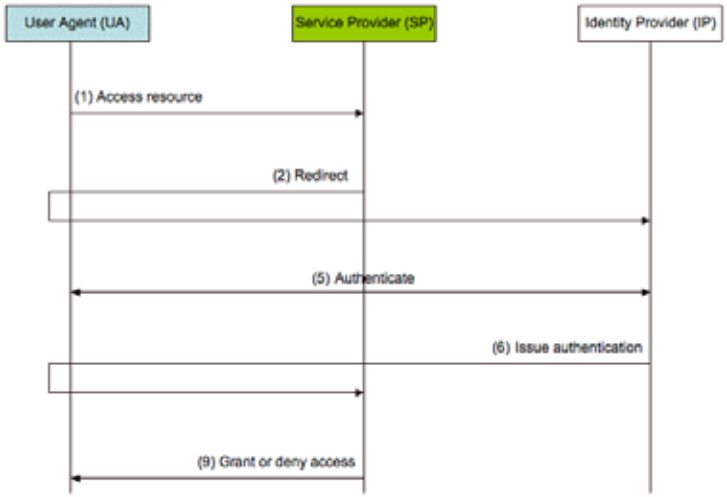

Figure 4: Model for achieving Web SSO with the WS-federation Passive Requestor profile.

4.2 Security Issues

[Groß05] analyzed the WS-federation Passive Requestor profile and proved it was cryptographically secure. Their work was significant because it was the first formal proof of any browser-based federation protocol [Groß05]. They showed formally that the profile provides authentication and successfully establishes a secure channel. In their proof, they created new models for user and browser, and they treated the secure SSL as a "blackbox submodule."

Back to Table of Contents

---

5. OpenID

Unlike SAML and WS-Federation, OpenID is an open-source project with a community driven standardization pro cess. Brad Fitzpatrick originally invented the protocol for his blogging application. OpenID Foundation whose mandate is to "enable and protect whatever is created by the community" now supports the community. [OpenID] Major industry vendors also support OpenID. For example, AOL is an IP or OpenID and Microsoft's identity management software supports O penID.

SAML and WS-Federation are flexible business-oriented specifications. They give implementers many options for customized deployments. On the other hand, OpenID is a user-centric, lightweight protocol. OpenID does not abstract away any details from its specification; it offers a direct solution to the Web SSO problem.

5.1 Framework

OpenID employs a decentralized architecture that takes advantage of the existing Domain Names Server (DNS) service. Anyone with a domain name can chose to be her own OpenID IP; users without their own domain name can register with companies they trust. OpenID effectively piggybacks on DNS to solve the IP discovery problem together with Web SSO. In the authentication protocol, users are assigned identifiers that are based on a domain name. For example, A lice has registered the OpenID alice@id-service.com with her IP. When she tries to access a protected resource at bob-blog.com she supplies her OpenID alice@id-service.com. Then bob -blog.com contacts id-service.com to authenticate Alice. To associate multiple OpenID identifiers to a single domain, OpenID uses the Yardis protocol. [Miller06].

Figure 5 shows how OpenID without the Yardis protocol extension implements Web SSO.

(1) The UA attempts to access a secured resource at the SP.
(2) The SP requests the user's OpenID.
(3) The UA supplies the OpenID to the SP.
(4) The SP determines the location of the IP based on the OpenID and sends an authorization request to the IP by redirecting the UA to the IP.
(5) If necessary, the IP authenticates the UA.
(6) The IP sends a signed authentication response to the SP via the UA.
(7) The SP grants or denies access to the UA.

After (3), the SP can optionally establish a shared-secret key using Diffie-Hellman for signing and verific ation of messages.

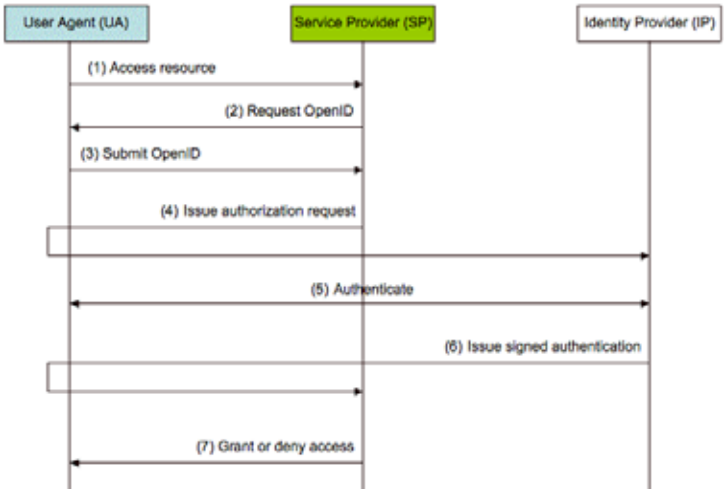

Figure 5: OpenID model for achieving Web SSO.

5.2 Security Issues

Unlike SAML, the security of OpenID protocol has not been analyzed. What is more, the OpenID community does not present a document considering the security issues with the protocol. The specification documents mandates and recommends some security features such as the use of SSL/TLS, but a self-contained document is not available.

Phishing attacks are the major security issue in the OpenID protocol. In a typical phishing attack a malicious SP redirects a user to a fake IP that steals the user's password. Another phishing attack involves a malicious SP masquerading as a trusted SP by hiding behind a redirect server. The OpenID specification mandates that the SP verifies a successful authentication message from an IP [Recordon06]. Verification involves checking the domain name of the IP, ensuring the message is not a duplicate (by using nonces), and verifying the message signature. Other attacks such as man-in-middle and replay attacks are also possible but use of HTTP over SSL/TLS can mitigate these threats.

Back to Table of Contents

6. Summary

SAML and WS-Federation are both relatively mature standards. Major industrial players support them and many identity products support them [OASIS07, IBM07]. Both protocols have been independently analyzed and their specifications are well documented. One of these standards will arguably dominate the Web SSO arena.

On the other hand, OpenID has support from Microsoft and AOL, so it is a serious "standard". And its apparent weaknesses in terms of maturity and formalism may be its strength. Being a lightweight protocol OpenID has a significantly lower barrier to entry that the other standards. Thus, many online projects are supporting OpenID. Also, the dynamic nature of the open-source community, specifically its less rigid model for standardization, may help the standard stay one step ahead of the other industry standards.

Back to Table of Contents

References

[Bajaj03] Bajaj, S., et al., "WS-Federation: Passive Requestor Profile Version 1.0," July 2003, http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-fedpass/ws-fedpa

[Cantor05a] Cantor, S., et al., "Bindings for the OASIS Security Assertion Markup Language (SAML) V2.0. OASIS," March 2005, http://docs.oasis-open.org/security/saml/v2.0/saml-bindings-2.0-os.pdf

[Cantor05] Cantor, S., et al., "Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0. OASIS Standard," March 2005, http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf

[Goodner07] Goodner, M., "Understanding WS-Federation Version 1.0," May 2007, http://msdn2.microsoft.com/en-us/library/bb498017.aspx

[Groß03] Groß, T., "Security analysis of the SAML single sign-on browser/artifact profile", Computer Security Applications Conference, 2003. Proceedings. 19th Annual , vol., no. 298-307, 8-12 Dec. 2003, http://ieeexplore.ieee.org/iel5/8882/28060/01254334.pdf?isnumber=28060[]=STD&arnumber=1254334&arnumber=12543 34&arSt=+298&ared=+307&arAuthor=Gross

[Groß05] Groβ, T., Pfitzmann, B., and Sadeghi, A., "Proving a WS-federation passive requestor profile with a browser model", In Proceedings of the 2005 Workshop on Secure We Services (Fairfax, VA, USA, November 11 - 11, 2005). SWS '05. ACM, New York, NY, 54-64. http://doi.acm.org/10.1145/1103022.1103034

[Hodge06] Hodges, J., "How to Study and Learn SAML", September 2006, http://identitymeme.org/doc/draft-hodges-learning-saml-00.html

[Hughes05] Hughes, J., et al., "Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0," OASIS, March 2005, http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf

[IBM07] IBM developerWorks, "Web Services Federation Language website," May 2007, http://www.ibm.com/developerworks/library/specification/ws-fed/

[Kormann00] Kormann, D. P. and Rubin, A. D., "Risks of the Passport single signon protocol," Computer Networks, 33(1–6) :51–58, June 2000.

[Lockhart06] Lockhart, H., et al., "Security Assertion Markup Language (SAML) V2.0 Technical Overview. Working Draft 10, " October 2006, http://www.oasis-open.org/committees/download.php/20645/sstc-saml-tech-overview-2%200-draft-10.pdf

[Lockhart06] Lockhart, H., "Web Services Federation Language (WS-Federation) Version 1.1," December 2006, http://download.boulder.ibm.com/ibmdl/pub/software/dw/specs/ws-fed/WS-Federation-V1-1B.pdf

[Microsoft01] Microsoft Corporation: Various .NET Passport documentation (started 1999), in particular Technical Overview , Sept. 2001, and SDK 2 .1 Documentation; http://www.passport.comand http://msdn.microsoft.com/

[Miller06] Miller, J., "Yadis Specification Version 1.0," March 2006, http://yadis.org/papers/yadis-v1.0.pdf

[Nadalin07] Nadalin, A., et al., "WS-Trust 1.3," OASIS Standard, March 2007, http://docs.oasis-open.org/ws-sx/ws-trust/200512/ws-trust-1.3-os.pdf

[OASIS] About OASIS, Organization for the Advancement of Structured Information Standards website. http://www.oasis-open.org/

[OASIS07] OASIS Security Services Security Assertion Markup Language (SAML) TC website, November 2007, http://www.oasis-open.org/committees/security/

[OpenID] OpenID webstie. http://openid.net/

[Pashalidis03] Pashalidis, A., Mitchell, C., "A taxonomy of single sign-on systems," Proceedings, volume 2727 of Lecture Notes in Computer Science, Springer-Verlag, Berlin, July 200 249-264.

[Pfitzmann03] B. Pfitzmann03 and M. Waidner, "Analysis of Liberty single-signon with enabled clients. IEEE Internet Computing," 7(6):38–44, 2003.

[Recordon06] Recordon, D. and Fitzpatrick, B., "OpenID Authentication 1.1," May 2006, http://openid.net/specs/openid-authentication-1_1.html

[Shibboleth07] Shibboleth Internet2 Project, October 2007, http://shibboleth.internet2.edu/

[SSTC05] SSTC Response to "Security Analysis of the SAML Single Sign-on Browser/Artifact Profile", January 2005 http://www.oasis-open.org/committees/download.php/11191/sstc-gross-sec-analysis-response-01.pdf

[Wisniewski05] Wisniewski T., et al., "SAML V2.0 Executive Overview," March 2005, http://xml.coverpages.org/SAML-ExecOverviewV206-11785-20050310.pdf

[WebISO03] Web Initial Sign-on (WebISO) website, October 2003, http://middleware.internet2.edu/webiso/

Back to Table of Contents

List of Acronyms

DNS	Domain Names Server
HTTP	Hypertext Transfer Protocol
ID	Identity provider
MDSSO	Multi-domain Single Sign-On
OASIS	Organization for the Advancement of Structured Information Standards
SAML	Security Assertion Markup Language
SOAP	Simple Object Access Protocol
SP	Service provider
SSL	Secure Sockets Layer
SSO	Single Sign-On
SSTC	Security Services Technical Committee of the Organization
STS	Security Token Service
TLS	Transport Layer Security
UA	User agent
URI	Unique resource identifier
WebISO	Web Initial Single Sign-On
Web SSO	Web Single Sign-On
WS-Federation	Web Services Federation Language
WS-Security	Web Services Security
XHTML	Extensible Hypertext Mark-up Language
XML	Extensible Markup Language

Back to Table of Contents

Last Modified: December, 2007
This paper is available at: http://www.cse.wustl.edu/~jain/index.html