Architecture Patterns for a Ubiquitous Identity Management System

Anders Fongen
Norwegian Defence Research Establishment
Norway
anders.fongen@ffi.no

***Abstract*—The design of an Identity Management System (IdM) must strike a balance between protocol overhead, software footprint and security level in order to operate successfully under the resource constraints found in mobile and wireless systems. But, what is good for a constrained system is also good for everyone else, in the sense that reduced overhead benefits all business application processing. This paper contributes to the discussion of IdM construction by suggesting patterns that preserves existing investments, maintains adaptability, scalability and modularity of the IdM. It also provides a discussion where security level is balanced with other non-functional requirements, which is seen less often in security research. A prototypical IdM systems built upon the proposed principles is also presented to some detail.**

Keywords*-*Identity management

I. Introduction

Identity Management (IdM) refers to the arrangement of manual procedures and software components which are needed to identify and control the use of computing resources. IdM also supports the privacy and integrity of data.

IdM involves tasks like key and certificate generation, role and attribute management, authentication operations, access control and auditing. Together, the IdM comprises a large set of distributed software components and a number of networking protocols. Besides, the components of an IdM will interface to business components and its management procedures will interface with procedures involving matter of law, human resources and business ethics [1].

Consequently, it is crucial to the successful deployment of an IdM that certain design principles are observed. The purpose of this paper is to present a set of design guidelines which serves as *design patterns* for the construction of an IdM.

Identity Management should maintain the following set of design patterns:

- Use existing Public Key Infrastructure (PKI)
- Federate domains for guest access
- Roles matter, not identity
- Domains are autonomous
- Avoid belt-and-suspenders protocols
- Trust has a lifetime
- Limit the unconditional trust

By following this list of rules the identity management system requires less connectivity and bandwidth, and consequently is better fit for a mobile and wireless computing environment. The identity management system becomes applicable to a wider range of environments, thus the use of the word ”ubiquitous” in the title.

The remainder of the paper is organized as follows: Section II explores briefly the design patterns just listed. Section III presents a short survey of existing IdM architectures. Sections IV and V present the Gismo IdM system and its protocols, followed by a section with some conclusive remarks.

II. Candidate Design Patterns

A. Use Existing PKI

In most organizations, there are formal procedures related to employee and inventory information. Quality of that information is crucial in order to detect fraud and theft. Some organization have also implemented a Public Key Infrastructure (PKI) (or are planning to do so) for the purpose of public key management. A PKI in operation will be the result of a long planning process, complicated software deployment and configuration, and the development of several new managerial interfaces between the HR and IT departments. An operational PKI represents a significant investment that should be built upon when an IdM is being developed.

B. Federate Domains for Guest Access

Back then, there was the idea of a PKI which could operate on a very large scale, e.g., for every citizen of a nation, and serve a large number of applications. Today, a national PKI is believed to provide keys only for limited communication between citizens and public sector. Other PKIs will provide keys for banks, other for Internet shopping and again, others for professional communication.

IdMs have the potential to bridge the gaps between different *domains* of key administration, meaning that they can manage trust relations between domains in an articulated manner. Domain federations allow subjects to bring their credentials across domains for controlled access and trust.

C. Roles Matter, not Identity

The rule in “traditional” user management in standalone computers has been never to grant privileges directly to subjects. Subjects should be assigned to *groups*, and groups given access rights. *Role-based* or *attribute-based* access

ISBN:978-1-61208-114-4

control [2] is built on this idea, which is several decades old and well proven.

This separation makes lots of sense in a distributed environment. It means that only the IdM service needs to maintain actual identities, whereas the providers of business services maintain the mapping between access rights and *roles* or *attributes*.

In a domain federation, this separation is crucial. Although some IdM systems for domain federations provide mapping between user names on different systems (hopefully for legacy reasons only), the only scalable approach is to allow the users to be represented by a set of roles/attributes.

D. Domains are Autonomous

All domains of identity management wish to be autonomous. They establish identification procedures based on their own business and security policies, according to national legislation and the ethics of their profession. They will determine what services will be made available to residents and guests of the domain. They decide by themselves the access rights that are associated with subject attributes. *Domain federations should not impose federated authorities*.

Another matter of domain autonomy is role (or attribute) *privacy*. The attributes associated with a subject may be of sensitive nature, since they may reveal information about the subject's authority. Consequently, the domain must be in control of how attributes are exposed inside and outside the domain [3].

E. Avoid belt-and-suspenders protocols

The network cost associated with the operation of a PKI is substantial, and inhibits this operation in parts of the network where the bandwidth is narrow or the connectivity is episodic [4]. Networks with such conditions include wireless mobile networks (MANET) and military tactical networks. Wireless networks are more exposed to intrusion attacks than a wired network. Ironically, the parts of the network that really need the protection that a PKI could offer, are thus the parts least suited to use it!

Consequently, the networking protocols (and the security policies they result from) must ensure that the network capacity requirements does not exceed the expected performance of the technology in place. This may require a closer inspection of the risk estimate, and some belt-and-suspenders security requirements may have to be relieved.

F. Trust has a lifetime

This pattern is firmly related to the previous paragraph. It is a matter of reducing the network traffic through a "trust has a lifetime" decision. For example, a validated public key is believed to be valid for some duration, and will not need to be revalidated in this period. This principle is well established through the distribution interval of certificate revocation lists (CRLs).

This principle reduces the number of necessary operations from both the client and the server to the security services. They do not longer need to receive credentials and validation information for each business operations, since this information can be cached and re-used for a while.

G. Limit the unconditional trust

The last design pattern is related to the number of *trust anchors*. A trust anchor is a subject whose signature is unconditionally trusted. All trust relationships are derived from a trust anchor through a chain of signatures. The security of the entire system collapses if a trust anchor gets compromised. Therefore, the number of trust anchors should be low for the sake of system security and robustness [5].

III. Existing IdM Architectures

The proposed design is related to the SAML 2.0 architecture for federated identity management [6] and the WS-Security [7] and WS-Trust standards [8], but this model aims to provide better answers to the challenges of mobile and tactical environments.

Based on a survey of existing models for federated identity management like Liberty Alliance [9], Shibboleth [10], and OpenID [11], it is an observation that they are *not* well suited for low-bandwidth, mobile or disadvantaged networks for the following reasons:

- They require much connectivity, in the sense that every new connection with a service involves operations on the identity provision servers.
- They require a coordinated replication of user registries, so that an excessive amount of work is needed to maintain user information in a highly dynamic network.

The same survey also indicates that these approaches to identity federation create rather strong coupling between the security domains; they either require mapping between local user identities, or mapping between local and federated identities. Both approaches could be replaced by an RBAC (role based access control) [2] arrangement that removes the need for replicated user identities in order to weaken the coupling between the domains.

Please observe that the term "federated" in this paper refers to federation of servers from different communities with different security requirements. The term "federation" as used in the related literature may refer to a group of servers in the same domain, in which case coordination is a much simpler problem.

IV. The Gismo Architecture

Following the guidelines given in Section II, a IdM prototype was built for the purpose of experimentation. The prototype has been implemented in Java for operation in a Web Services environment. The protocol data units have accordingly been coded in XML syntax, to the extent possible using suitable XML standards (SAML, WS-security, WS-addressing, etc.).

The functional components of the Gismo[1] IdM and their relations are shown in Figure 1.

A. The Domain

In the context of this project, the term “Domain” means a population of services and subjects with the following set of properties:

- Members (services and subjects) belong to one domain only
- All members of a domain share the same *Certificate Authority* (CA) and *trust anchor*.

B. Community of Interest

Inside a domain, there are one or more *Communities of Interest* (COI). For each COI, there is one *Identity Provider* (IdP). Members of a COI are services and subjects, which can be members of several COIs (inside the same domain). Two subjects can have authenticated communication (client-server or message exchange) if they are members of the same COI, or members of two COIs with a *trust relationship*.

C. The Identity Statement

The Identity Statement (IS) is similar to a public key certificate in the sense that it attests a binding between a public key and the identity information of the “owner” of the private key. In addition, the IS contains a set of roles/attributes associated with the represented identity.

The identity statements are issued and signed by the identity provider, and are therefore valid only inside the COI served by that IdP.

There is *no revocation checking* associated with identity statements. An IS is therefore meant to be short-lived, i.e., expire after a duration comparable to the issue interval of certificate revocation lists.

D. The Identity Provider

The Identity Provider (IdP) is a CA-like service which issues identity statements for members of the COI. Upon requests from subjects, their IS are issued and returned to the clients for use in different authentication procedures.

Another important task for an IdP is to provide identity statements for *guests*. If a subject sends an IS issued by an IdP with which there exists a trust relationship, a *guest IS* is issued. The guest IS contains the same information as the original IS, except that attributes may have been added or removed. It also bears a new signature.

E. The Authentication Protocol

Several authentication protocols have been proposed under the Gismo IdM project, with the goal to reduce the number of protocol round trips and to explore the relation between network cost and risk.

Protection against replay attack in authentication protocols is quite costly, since it requires the service to remember previous requests (identified by e.g., nonces) for the maximum allowed clock skew period, *also during a crash* (i.e., across “incarnations”). This is a hard problem, since lightweight service platforms (like embedded systems) may not be able to offer the transactional stable storage which is needed to implement this mechanism.

Under the conditions that the service is stateless, i.e., a request is not altering the state of the system (i.e., a lookup service), replay protection is not needed, provided that only the intended client can read the reply. The authentication protocol may under such circumstances simply encrypt the reply with the public key of the client to achieve this effect.

Another matter is the number of protocol round trip. During a separate authentication phase, client and service can mutually authenticate themselves before the actual service call is made. A more effective approach would be to piggyback the client authentication in the service request, and the service authentication in the response, as shown in Figure 3. This reduces the number of round trips, but the risk remains that a mere request to a fraudulent service may compromise sensitive information. This is (in the author’s opinion) a far-fetched risk: An attacker who is able to stage such an advanced attack would benefit more from simple eavesdrop than a “hit and run” attack (a fraudulent service which is not able to authenticate itself would trigger an intrusion alarm and subsequent hunt for the intruder).

Under other conditions, e.g., a protected and authenticated conversation, a more traditional approach would still be the best choice where mutual authentication and session key exchange takes place before the information flow starts.

F. Cross-COI Operation

An important property of an IdM architecture is the ability to offer services to members of a different organization in a well controlled manner. This property is an important part of the Gismo IdM and is based on *guest IS* to indicate the approval of a guest identity, and the *cross-COI IS* to indicate the trust relationship between to COIs. Together with an RBAC/ABAC based access control framework, guest may be given access under a fine-grained policy.

Trust relationships between two COIs are expressed by a pair of IS where they attest each other’s public keys and identities. These *cross-COI ISes* links the signature on an IS from a remote COI to the trust anchor (the CA) of the local COI, and conveys the delegation of trust from the local IdP to the remote IdP.

[1] “Gismo” is the acronym for the Norwegian expression “Fundamental IT security for mobile operations”

G. Proof of validity

Members of a COI trust the CA of the domain, i.e., the CA is their *trust anchor*. They also need to trust the IdP, since the identity statements bear its signature. The IdP may be declared as a trust anchor, too, but there are good reasons (mentioned in Sect. II-G) why the number of trust anchors should be kept to a minimum

The trust in the IdP could be derived from the CA through a PKI-style *validation* of the IdP's certificate, which is not a desirable solution for reasons of network economy and architectural coupling.

Rather, it is a preferred solution that the IdP is the only central service that the members know about, and that the IdP itself can provide a "proof of validity" for its key and certificate. Given this proof, any member can conclude that the key of the IdP is authentic and not revoked at the moment.

The proof of validity (POV) may have several forms, depending on whether the CA is the direct or indirect issuer of the IdP's certificate. It should contain all certificates from (and including) the IdP's certificate and up to (not including) the trust anchor (normally the root CA). It should also provide proof that none of the certificates on this list is revoked at the moment.

The proof of non-revocation cannot be a revocation list, since it is not possible to provide positive information in it, only negative. What is needed is a positive revocation status (meaning not revoked), which can be the output of a *validation server*, e.g., one that is based on the SCVP or OCSP protocols. These responses must be signed with a key that is attested by the trust anchor through a signature chain.

The CA could issue an SCVP response on a regular basis which the IdP could hand out on demand, but that would require a custom built CA and a violation to the rule in Sect. II-A. Standard PKI services must be used, which would likely be the signed and timestamped output from certificate status providers (using OCSP) if available. If the trust anchor refuses to issue revocation status in any other form than through CRLs then one is out of luck and needs to declare the IdP as the trust anchor for the members of the COI.

H. Attribute Protection

Subject attributes in an IS (elsewhere also called *roles*) are name/value pairs which can describe any aspect of the subject. It can be used to store the subject's native language in order to improve the user interface of a service etc., or describe the subject's authorizations for access control support.

Attributes may contain sensitive information which should be adequately protected. The ultimate protection is for the IdP to issue an IS for the purpose of one particular service, encrypted with the public key of this service. On the other hand, this arrangement makes the IS non-cacheable and requires frequent connection to the IdP, effectively making it into a single point of failure.

The Gismo IdM approach is taking a middle road. An IS issued for use in a COI should be cacheable and be used for all services and conversations withing the COI until the IS expires. When an IdP receives an IS from a guest who is requesting a guest IS, only attributes marked for export are copied into the guest IS, the other are removed. Since there exists a trust relationship between these two IdPs it is reasonable to trust a "foreign" IdP to do this honestly and correctly. It is also reasonable to allow services and subjects in the same COI to share attribute knowledge, since the COI membership of shared goals and shared responsibility also implies a level of trust (and since they might obtain this information anyway through listening on the shared data links).

V. Protocol and Data Structure Details

At this point the design principles and the main functional components of the Gismo IdM have been explained, and the paper will commence with a description of the data structures and protocols in greater detail.

A. The Identity Statement

As previously described in Section IV-C, the authentication mechanisms relies heavily on the data structures called *Identity Statement* (IS). Formally, the identity statement of principal x signed by the IdP of COI a is denoted $(Id_x)_a$ and has this structure:

$$(Id_x)_a = Name_x + PublicKey_x + Attributes_x + Timestamp + Serialnumber + Signature_a$$

$Attributes_x$ denotes a set of name-value pairs which describes the roles etc. of the subject. It may be used for access control purposes. $Signature_a$ indicates that the entire statement is signed by the IdP of CIO a. The IdP of COI a will from now on be denoted IdP_a.

In the proposed system, the identity statement is formatted according the the SAML 2.0 syntax requirements, which means that it is coded in XML. The SAML assertion is used in a so-called "Holder of Key" mode, which means that the authentication process requires a demonstration of the private key corresponding to the public key bound in the identity statement.

B. Identity Statement Issuance

The discussion in Section IV-H identified the need to protect subject attributes outside the Community of Interest (COI), which means that only members of a COI should not be allowed to ask the Identity Provider (IdP) for an IS regarding a COI member.

There is no easy way to distinguish a member from a non-member (without a costly authentication phase). The design choice has therefore been to issue an IS only to the subject itself. This requires a straightforward SSL-based

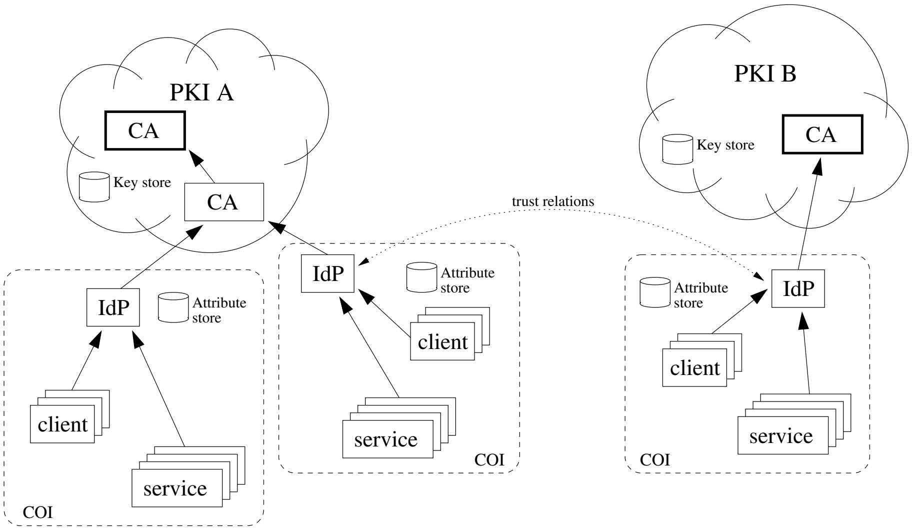

Figure 1. The functional components of a federated IdM. Observe that the IdP serves one single COI, and the trust relations are formed between COIs, not domains. Key management is handled by the PKI whereas the attribute management is done by the IdPs on the COI level

client authentication based on the client key pair related to the IS. An alternative approach would be to encrypt the IS with the subject's public key, but that is less flexible towards future policy changes.

A part of the service semantics is that the subject's key pair is *validated* before the IS is issued. If the key pair is generated by a PKI (as suggested in Section II-A) the IdP should use the available PKI-based validation mechanisms for this purpose, and deny the IS request if the key is invalidated or revoked.

C. Issuance of Guest Identity Statement

The IdP is responsible for the issuance of guest identity statements as explained in Section IV-D. Presented with $(Id_x)_a$, the IdP_b (IdP of COI b) can issue the identity statement $(Id_x)_b$ provided that there exists a trust relationship between COI b and a expressed by an identity statement issued by IdP_b with IdP_a as the *subject*. This is called a cross-COI IS and expressed as $(Id_a)_b$. With the guest IS $(IdP_x)_b$, the subject x which is a member of COI a, can authenticate itself to members (e.g., services) of COI b.

For two-way authentication in a guest COI, e.g., for the client from COI a to trust the signed response from a member of COI b, the reverse cross-COI IS is needed, termed $(Id_b)_a$, to link the signature key to the client's trust anchor. Therefore, $(Id_b)_a$ is included in the response of the guest IS issuance. $(Id_b)_a$ is issued to IdP_b by IdP_a (as a normal IS issue) and stored by IdP_b for the purpose of guest IS issue.

Figure 2 illustrates the guest IS issuing protocol as a two stage process involving two IdPs. Key validation takes place only in the first stage. The required proof of validity (Section IV-G) is assumed to have been issued at an earlier occasion.

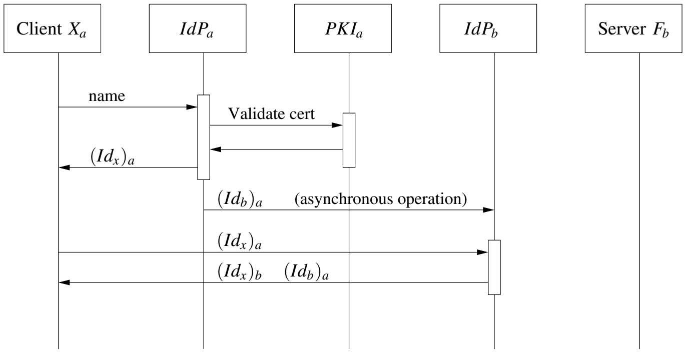

Figure 2. The identity statement issuing protocol. The IdP of COI A, termed IdP_a, issues a "native" identity statement to the client, which is given to IdP_b which in turn issues a guest identity statement. The term PKI_a denotes a set of certificate validation services in domain a.

TABLE I
ABBREVIATIONS USED IN THE FIGURES

Client X_a	Client X of COI a
IdP_a	Identity provider of COI a
PKI_a	Validation services in domain a
Server F_b	Server F in COI b
$(Id_x)_a$	Identity statement for identity x, issued by IdP_a
$(msg)S_x$	Message msg signed with private key of x
$(msg)E_x$	Message msg encrypted with public key of x

D. The Authentication Protocol

Section IV-E provides a discussion on the effectiveness of authentication protocols. The Gismo IdM offers a range of authentication protocols with different properties, two of which are presented in this paper. Figure 3 shows a protocol suited for a server with the necessary resources to implement replay protection. The data elements needed for mutual authentication (signature, timestamp, nonce, servername) are

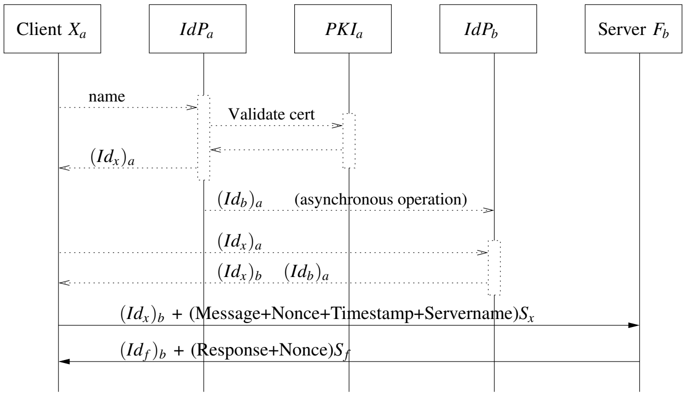
Client X_a
IdP_a
PKI_a
IdP_b
Server F_b
name
Validate cert
$(Id_x)_a$
$(Id_b)_a$ (asynchronous operation)
$(Id_x)_a$
$(Id_x)_b$ $(Id_b)_a$
$(Id_x)_b$ + (Message+Nonce+Timestamp+Servername)S_x
$(Id_f)_b$ + (Response+Nonce)S_f

Figure 3. The authentication protocol for the stateful service. Both the request and response are signed with the sender's private key as a part of authentication process. A timestamp, a nonce and the server's name is included for replay protection.

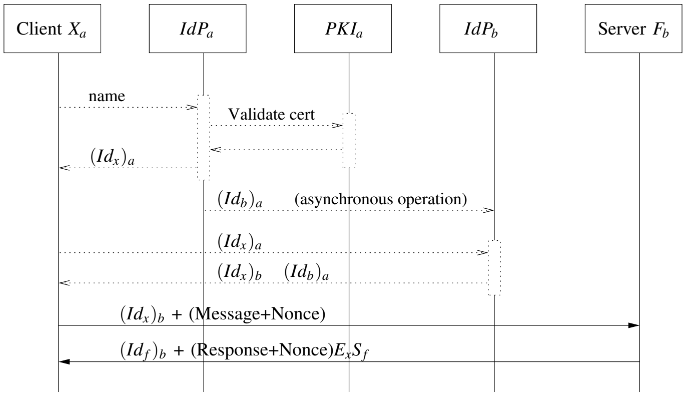

Figure 4. The authentication protocol for the stateless service. There is no replay attack protection since they are not considered as threats, but the response need to be protected for reasons of response replay and information compromise.

piggybacked on the request and response messages in order to save a protocol round trip. The remaining security risk, which results from this choice is marginal, as pointed out in Section IV-E.

Figure 4 illustrates the much simpler authentication to a stateless service. All requests are processed since they do not alter the system state (other than consume resources), but the authentication requirements are enforced through the encryption of the response. The response is signed for the purpose of server authentication, and includes a nonce for protection against response replay. The nonce is not remembered across invocations and introduces no state space in the sever.

E. Notes on Implementation

The Gismo IdM is implemented as a proof of concept in Java and targeted for web services use. It employs relevant WS standards (SAML, WS-Security, WS-Addressing, etc.) and implements the authentication protocols as *WS Message Handlers*, including the client interface to the IdP services.

VI. Summary and Conclusions

In the course of this paper, a slightly different approach to the construction of IdM has been made, where a balance between security level and other non-functional requirements have been sought. A prototypical system built upon the proposed principles has been presented in some details.

Ongoing efforts on the Gismo IdM includes integration with Role Based Access Control, combining IdM with arrangements for object type enforcement and principles of least privilege. Future plans include porting the software to the Android platform for study of its performance in mobile computing environments.

References

[1] N. Delessy, E. B. Fernandez, and M. M. Larrondo-Petrie, "A pattern language for identity management," in *Proceedings of the International Multi-Conference on Computing in the Global Information Technology*. Washington, DC, USA: IEEE Computer Society, 2007, pp. 31–31.

[2] R. Sandhu, D. Ferraiolo, and R. Kuhn, "The NIST model for role-based access control: towards a unified standard," in *RBAC '00: Proceedings of the fifth ACM workshop on Role-based access control*. New York, NY, USA: ACM, 2000, pp. 47–63.

[3] A. Bhargav-Spantzel, A. C. Squicciarini, and E. Bertino, "Establishing and protecting digital identity in federation systems," *J. Comput. Secur.*, vol. 14, pp. 269–300, May 2006.

[4] A. Fongen, "Scalability analysis of selected certificate validation scenarios," in *IEEE MILCOM*, San Diego, CA, USA, Nov. 2010, pp. 1–7.

[5] C. Wallace and G. Beier, "Practical and secure trust anchor management and usage," in *Proceedings of the 9th Symposium on Identity and Trust on the Internet*, ser. IDTRUST '10. ACM, 2010, pp. 97–107.

[6] N. Ragouzis, J. Hughes, R. Philpott, E. Maler, P. Madsen, and T. Scavo, *Security Assertion Markup Language (SAML) V2.0 Technical Overview*, OASIS Committee Draft, March 2008.

[7] K. Lawrence and C. Kaler, *Web Services Security: SOAP Message Security 1.1*, OASIS Standard Specification, 2004.

[8] ——, *WS-Trust 1.4*, OASIS Standard, 2009. [Online]. Available: http://docs.oasis-open.org/ws-sx/ws-trust/v1.4/os/ws-trust-1.4-spec-os.pdf [retrieved November 9, 2010]

[9] "The Libery Alliance." [Online]. Available: http://www.projectliberty.org/ [retrieved November 9, 2010]

[10] "Shibboleth." [Online]. Available: http://shibboleth.internet2.edu/ [retrieved November 9, 2010]

[11] "OpenID." [Online]. Available: http://openid.net/ [retrieved November 9, 2010]