Towards Enabling Web 2.0 Content Sharing Beyond Walled Gardens

San-Tsai Sun and Kirstie Hawkey and Konstantin Beznosov
University of British Columbia
Vancouver, Canada
{*santsais,hawkey,beznosov*}@*ece.ubc.ca*

***Abstract*—Web 2.0 users have many choices of content-hosting or application-service providers (CSPs). It can be difficult for a user to share content with a set of real-life friends and associates; intended viewers of the content may have different CSP memberships than the content sharer. Web 2.0 users need usable mechanisms for sharing their content with each other in a controlled manner across boundaries of CSPs. In this position paper, we discuss the problem users face and propose a solution that builds upon the existing secret-link mechanism. Our proposed solution does not require users to setup another account on each CSP to view shared content and does not require any special software being installed. The mechanisms for content hosting and sharing are separated; CSPs do not need to change their existing access-control mechanisms.**

***Keywords*-Web 2.0 content sharing; access control; trust management;**

I. Introduction

In Web 2.0, the user is both content consumer and provider of the Web [1]. Users own their personal data (e.g., profile, friend lists, content), and should have freedom to control who has access to that data globally, regardless of where the data is stored and what access-control mechanisms are provided by service providers. As illustrated in Fig. 1, the current Web is site centric. Personal content is created and resides on different content-hosting or application-service providers (CSPs). Each service provider forms an administrative domain by defining the membership of users, permissions on protected resources, and access-control policies. A web user has to maintain a separate copy of identity, personal data, and relationships for each service provider.

Sharing beyond administrative domains refers to the capability for a user who is not the member of an administrative domain to be able to access protected resources in that domain. In this paper, we use the term "*walled garden*" to refer to such an administrative domain defined by a service provider. Since each walled garden controls its own set of users and employs a different access-control mechanism to protect personal content, it is difficult to share personal content beyond and across those walled gardens.

In this position paper, we present an approach which allows content sharing beyond walled gardens. We first present the limitations of current Web 2.0 content sharing mechanisms and discuss the requirements for a solution. We then present our proposed approach, including our user centric model of Web 2.0 sharing and our system architecture. Finally, we conclude with a discussion of our current and future work.

II. Limitations of Current Web 2.0 Content Sharing Mechanisms

Web 2.0 users need usable mechanisms for sharing their content with each other in a controlled manner across boundaries of CSPs. Personal content sharing is currently available in limited forms. There are three primary content sharing mechanisms offered by CSPs: making content public, the walled garden approach, and using secret-links. Making user content public is obviously inadequate for controlled sharing. We will discuss each of the other two approaches next.

In the *walled-garden* approach, the user who "owns" content can grant permissions directly to other users (or user groups) within the same CSP, the walled garden. This approach is easy to implement and use. Its main limitation is that not all content users are necessarily registered with a CSP. Therefore, users outside of that CSP cannot be granted selective access. Even within the same walled garden, the resource requester and owner might not be known to each other, increasing the challenge of controlled sharing for both the owners and consumers of content.

To enable controlled sharing beyond walled gardens, some CSPs (e.g., Google, Facebook, Flickr) use the *secret-link* mechanism. A secret-link is a hard to guess URL that uniquely identifies a shared resource (e.g., http://spreadsheets.google.com/ccc?key=px0Ox4z1SIE). When a resource owner shares personal content using a secret-link, the corresponding CSP makes the shared resource publicly accessible, and creates a special URL for that resource. Anyone who knows a secret-link can access the content identified by that link. To share a specific personal content, a resource owner sends (sometimes with the aid of the CSP) the secret link via email to select users. The message recipients view the shared content by clicking on the link. Secret-link is easy to use for both owners and users, and it provides a certain degree of control over sharing since only those who obtained (or guessed) the link can access the content. The main limitation of this approach is that the secret-link can be forwarded or

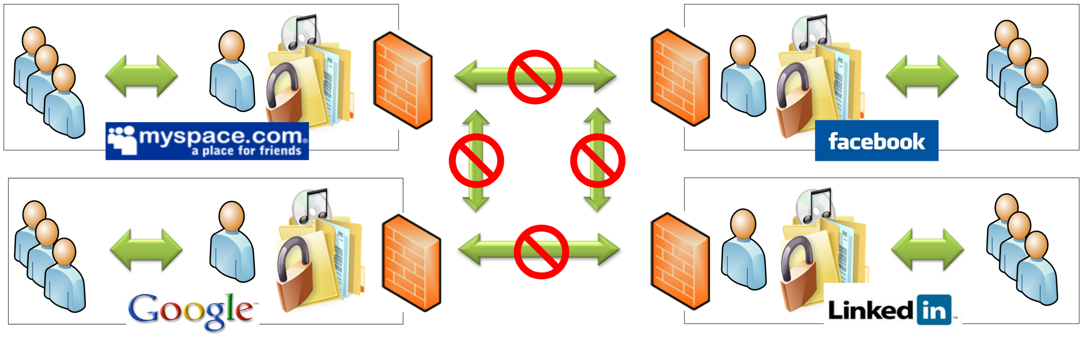

Figure 1. Site-centric walled garden Web.

otherwise "leaked" to unauthorized users. In other words, if sharing content using secret-links can be viewed as a capability-based access control, then unrestricted capability delegation becomes an issue. Some CSPs, such as Google Picasa, provide a "sign-in required to view" option when sending a secret-link. However, this requires the recipients to have an account with the CSP in order to view the shared content, essentially reducing the approach to a walled garden.

Throughout this paper, we will use the following scenario of content sharing to illustrate our discussion:

Alice is a girl scout in the Colonial Coast Adventures (CCA) club, which is a certified member of the Girl Scout Association (GSA). Jenny is another CCA member and Mary is her mother. Alice and Jenny took pictures at a scout training event. The CCA policy is that pictures of training events can only be seen by CCA troop members and their parents. Alice would like to use her favorite photo web site to share her photos online and is willing to share them with other troop members and their parents. Alice use MyPhoto.com, but neither Jenny nor Mary is a registered member of that site and Alice does not directly know Mary. In order for Mary to access Alice's photos for this event, Mary has to prove that she is Jenny's parent and that Jenny is a CCA member.

III. Developing Requirements for Sharing Content Beyond Walled Gardens

In order to develop requirements for a mechanism to share content beyond walled gardens, we reviewed existing literature to understand user file sharing practices and identify the breakdowns users encounter when sharing. As we need to design a solution that can be operated across administrative boundaries, we also studied existing solutions in federated identity management and distributed authorization systems.

A. File Sharing Practices

Olson et al. [2] studied what is shared and with whom in order to explore preferences for general information sharing. Their focus was on understanding how people might abstract the details of sharing into high-level of classes of recipients and information which are treated similarly. Voida et al.[3] studied the sharing practices of 10 research organization employees and identified the types of content they shared and with whom, the mechanisms they use to share, and how much control over the information they grant to the sharing recipients. One of the most important findings related to our work is that *Email is the most common mechanism for sharing* (45%) (followed by network folders (16%) and posting content to a web site (11%)). The study also identified the breakdowns that users have experienced in their own file sharing experiences including (1) difficulties in selecting a sharing mechanism with desired features that are available to all sharing participants, (2) forgetting what files had been shared and with whom, and (3) knowing when new content was made available. Similarly, Whalen et al. [4] studied 200 employees of a US-based research institution and investigated file sharing practice in both work and personal contexts. Most of their results confirm the findings of Voida et al. [3] In addition, they identified the factors that influence the choice of sharing method used, the frequency of file sharing and permission management, types of errors users encounter during permission management, and how such errors were detected.

To lower the barrier for users and CSPs to adopt our proposed solution, we investigated what sharing solutions are provided by current CSPs. We found that the secret-link approach is the most widely deployed mechanism provided by current CSPs to enable sharing beyond administrative domains.

B. Federated Identity Management

One of the limitations of the walled-garden controls for sharing is the inconvenience for users of having different user identifiers with different CSPs. Users end up maintaining multiple identities and corresponding passwords at multiple sites [5], which leads to weaker passwords and/or password re-use across accounts. A possible solution is the use a technology that would allow re-use of the same user identity across multiple CSPs. Federated identity management enables information about a user in one CSP to be provided to other CSPs in a federation; OpenID [6] is one such solution. Federated identity solutions enable cross-domain single sign-on and remove the need for users to keep identifiers and passwords at individual CSPs. Identity providers (IdPs) supply user information, while CSPs "consume" provided identity and mediate accesses based on this information. Except for OpenID, current federated identity solutions require pre-established relationships between IdPs and CSPs in a federation, which essentially form a larger walled garden by aggregating existing ones.

OpenID [6] is an open protocol for user-centric identity management and web-based single-sign-on. OpenID uses URI as end user's identifier, which acts as a universal user account and is valid across all unrelated CSPs. OpenID is user-centric in the sense that users are free to choose their own OpenID identifier and identity providers. OpenID is promising, but has had some problem with adoption because most CSPs prefer to maintain as many of their own registered users as possible, building "walls" to protect their subscriber base. Currently, there are many OpenID IdPs, but not enough CSPs that use it. To facilitate adoption of OpenID, more value services could be added to OpenID identity providers. Additionally, the usability of the OpenID scheme could be improved. OpenID uses a URI as user's identifier, but Web users perceive a URI as a "web addresses" instead of a personal identifier. In the context of Web content sharing, users rarely know the URIs of whom they want to share their content with, but they do tend to know each other's e-mail addresses.

C. Distributed Authorization

Traditional access control mechanisms (e.g., access control matrix, Role-Based Access Control (RBAC) [7]) make authorizations based on the identity of the request . However, in decentralized environments such as the Web, the content owner and the requestor often are unknown to each other (e.g., Alice does not know Mary or other scout members).

There is much research that has addressed the problem of authorization within distributed environments. Such systems includes trust management systems (e.g., SPKI/SDSI [8], RT [9]), ABLP Logic-based systems (e.g., ABLP [10]), XML standard-based systems (e.g., SAML, XACML), and semantic web-based systems (e.g., Rei [11]). Among them, RT [9] is the most expressive and concise language that combines the strength of RBAC [7]) and trust-management (TM) systems [12] for representing credential and policy in decentralized environment.

D. Summary of Requirements

As discussed, existing distributed authorization systems provide expressive policy languages for expressing credentials and access policies, sound algorithms for discovering credential chains, and inference engines for deriving authorization decisions. However, because Web 2.0 access policies for personal contents are authored by users without special technical skills, and are enforced by mutual-untrusted walled gardens, there are many issues remaining to address. The main issues are *usability* and *inter-operability*. The expressive power of a policy language must be balanced with usability. An average internet user must be able to comprehend the language to ensure that an access policy matches the owner's sharing intention.

For sharing content on the Web, the system should not require any special software to be installed or require any public key, X.509 or SPKI/SDSI certificates to be possessed— a user is assumed to be equipped only with a web browser. To share personal content beyond walled gardens, credential and access policy that are authored in one policy provider should be able to be employed to protect personal content residing on multiple CSPs. The data owner should have freedom to choose policy providers, and the access polices should follow the owner to wherever she goes. Because a user's credential and access-policy are considered to be private to the user who issuing them, authorization decisions should be derived without credentials to be exposed to other users.

In addition to usability and inter-operability, *granularity* of control and *accountability* should be considered as well. Content created by Web users is diverse and sometimes complex; the content owner should be able to specify access-control in a fine-grained format. For example, owners might want to protect a photo in a web album, an event in a calendar, or even a paragraph within a blog. For accountability, the owner should be able to know which data is being accessed, by who and when, and be able to revoke an authorization at anytime if necessary.

IV. Our Proposed Approach

One intuitive solution for sharing beyond walled gardens is to improve the existing secret-link sharing mechanism. To assert the user's ownership of an email address, a CSP could prompt the user to enter an email and password when accessing a shared content. If the email has not been set up in the CSP yet, the CSP could send a confirmation link to the prompted email account, and ask the user to click the confirmation link to ensure the user is the owner of the prompted email account. This solution is trivial to develop and does not require CSPs to tear down their guarded walls. However, this solution is not user-centric, and does not work across walled gardens. The content owner has to organize a "contact list" for each CSP, and the user has to set up an account and password for each CSP to view the shared content. Another limitation of this solution is that it does not support attribute-based grouping of recipients.

Attribute-based access control determines whether a request should be granted based on attributes of the resource requestor (e.g., all colleagues with @ece.ubc.ca e-mail address; all undergraduate students from the academic

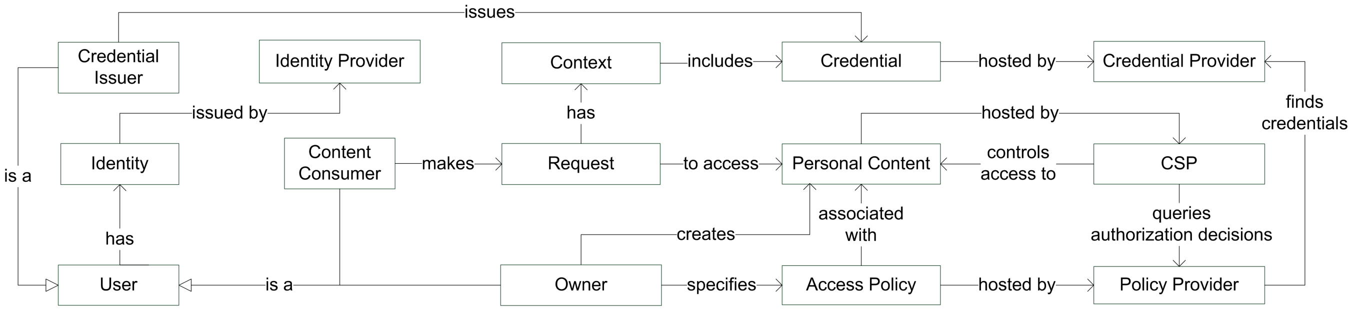

Figure 2. Web 2.0 user-centric content sharing model.

program; all those with whom Alice already shared these pictures). The strength of attribute-based access control is empowered by the concept of *delegation*, which provides a flexible way for a user to delegate authority to another user who is in better position in defining attribute of other users. For example, Alice might trust the CCA scout troop to define its girl scout members, even if some members are unknown to Alice. A content owner might want to use one attribute to make inference about another attribute (e.g., Alice defines all girl scouts of CCA as her friends). The owner might also want to delegate to unknown users based on their certified attributes. For example, Alice trusts GSA to define its member troops and also delegate the authority over "girl scouts" attribute to those troop members.

A. Web 2.0 User-centric Content Sharing Model

Current access-control mechanisms provided by CSPs are site-centric. To enable secure sharing beyond walled gardens, we framed our design in a proposed Web 2.0 user-centric content sharing model as illustrated in Figure 2. In this model, a *user* is not only a content *owner* and *consumer*, but a *credential* issuer as well. A user enrolls a set of *identities* (e.g., user name/password) from multiple *identity providers* to represent themselves when accessing shared content and constructing access polices. A content owner creates personal content on CSPs and associates that content with access-control polices that are hosted by a *policy provider*. To access shared content, a content consumer chooses an appropriate identity to make a *request*. Each request contains the identity provided by the consumer and a corresponding set of *context* information. Context information is the meta-data of a request, such as user-specific profile attributes, current location, date/time of the request and user *credentials*. A credential is an assertion of certified user-attribute from another individual user or an organization authority. To mediate accesses, a CSP requests authorization decisions from a policy provider to protect shared content. The policy provider then acts as a policy decision point (PDP), which responds with authorization decisions based on the context of the request and a set of pre-defined credentials and access policies.

User-centric access control requires user-centric authentication and authorization. For user-centric authentication, the user/owner should be able to control their own identities, and is free from choosing when and where to use it. For user-centric authorization, the content owner is the author of access policies. Access-control decisions are based on the policy associated with the protected content and credentials issued by multiple trusted individual or organization users. The content owner has the freedom to choose policy providers that host policies and trusted authorities that issue credentials. In a user-centric Web, access policy follows the user. One access policy hosted in one policy provider should be able to be enforced to protect shared content residing on different CSPs.

B. System Architecture

Figure 3 illustrates the system architecture of our proposed Web 2.0 content sharing system and data flows among the main players in the system. The main idea of our approach is to shift secret-link sending and access-control functions to OpenID$_{email}$ providers. An OpenID$_{email}$ provider is an OpenID identity provider that is augmented with two additional components. The first component is an OpenID extension, which was originally proposed by Ben Adida [13]; we labeled this *OpenID*$_{email}$. It extends the existing OpenID protocol to enable OpenID identity providers to use email as an alternative identifier. Using email addresses as user accounts is common, and users are often prompted for an email address as a user name. For instance, major service providers (e.g., Google, Yahoo, AOL) use email addresses as user accounts to associate provided services. Thus, the user experience for registering and entering CSPs that support OpenID$_{email}$ would be the same as they experience today.

The second component is the role-based trust-management policy service (RTPs), which provides services for internet users to organize their credentials and polices and services for CSPs to make access decisions. In Web 2.0, attributes or roles of a user are often certified or asserted by other individuals or organizations other than the content owner herself. The notation of trust allows a user to delegate authority to another user who is in a better

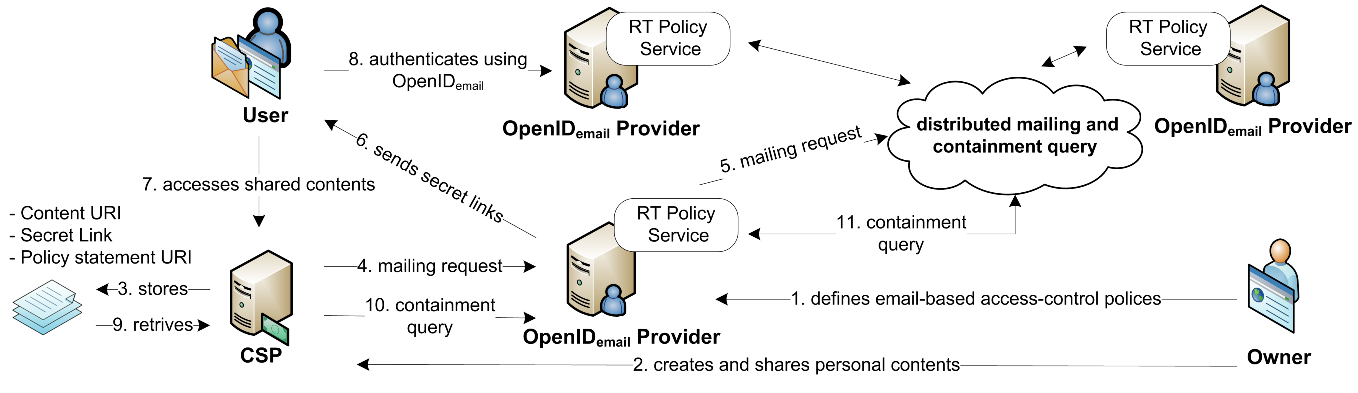
8. authenticates using OpenID$_{email}$
RT Policy Service
RT Policy Service
User
OpenID$_{email}$ Provider
distributed mailing and containment query
OpenID$_{email}$ Provider
6. sends secret links
5. mailing request
7. accesses shared contents
RT Policy Service
11. containment query
- Content URI
- Secret Link
- Policy statement URI
4. mailing request
3. stores
9. retrives
10. containment query
1. defines email-based access-control polices
CSP
OpenID$_{email}$ Provider
Owner
2. creates and shares personal contents

Figure 3. System architecture of the proposed Web 2.0 content sharing system.

position to define roles of other users, i.e., the user trusts another user's judgement on the role. RTPS makes use of the RT [9] framework, which can express selective use of capacities and delegation of these capacities. An entity in RTPS is a uniquely identified user (by email address), and a role is a set of entities. A user can delegate other trusted users to define the members of a role. Thus, a RTPS role can function as a delegation-enabled contact list, which is then used by CSPs for sending secret-links and for deriving authorization decisions.

To share personal content, the content owner specifies a content c and a set of roles R as the recipients of c. The service provider (CSP) then generates a link l based on the resource c, records the tuple (c, R, l), and submits the tuple to the RTPS the content owner belongs to. RTPS then constructs a set of destination email addresses $E = \{\ e \mid e \in r,\ \ r \in R\}$ by performing a nontrivial *distributed membership query* to other RTPS. For each destination email address $e \in E$, RTPS sends the l to the email box identified by e.

To access a shared resource, the resource requester clicks on the link l in their email box e, which in turn, presents the l to CSP. Based on l, CSP lookups the stored tuple (c, R, l) to identifies the resource c and roles R. CSP determines whether the request should be granted by performing a *distributed containment query* to the RTPS the R belongs to. A containment query $\mathcal{Q}$ takes the form $R \sqsupseteq e$, in which R is a set of roles and e is an email owned by the requester U. The access to resource c is granted only if $\mathcal{Q}$ holds.

V. Conclusions and Future Work

In this position paper, we have described the problem of Web 2.0 content sharing across walled gardens and discussed the requirements for a solution. We briefly presented our proposed approach for secure Web 2.0 content sharing across CSPs. With our approach, the user experiences for content sharing are similar to the existing secret-link sharing mechanism, with additional benefits of contact-list reuse and delegation across CSPs. Secret-links can be forwarded, as long as the person who clicks on the link has membership in an allowed list. Content requesters do not need to setup an account on each CSP and do not require any special software installed to view shared content. The functionalities for content sharing using secret-link are shifted from CSPs to OpenID$_{email}$ providers. CSPs do not need to change their existing user management and access-control mechanisms. In addition, policy statements are URI-addressable, and same access policies can be reused and enforced across CSPs.

We are currently evaluating our approach on a theoretical basis through threat analysis and proof of the protocol. The main open problem we are investigating is how to refine the notation of trust-management to reflect the semi-trusted nature of Web 2.0. For instance, trust relations in RT language are transitive (i.e., A trusts B, and B trusts C, implies A trusts C). However, it might not be the case in the real world. For example, Alice may trust Jenny to define her parents; but she may not trust others to which Jenny may delegate authority over the parent attribute. In addition, a user's trust might be abused by others, either maliciously or unintentionally. For example, Jenny might include her friend Bob as her parent in order to allow Bob to access the shared CCA photos. Finally, the confidentiality of credentials and policies should be protected as well. For example, CCA might not want everyone to know the email addresses of its scout members.

Once we have refined and validated our approach, we will implement it and conduct an empirical evaluation with users. We are in the process of building a system we named OpenRT, which will enable Facebook users to share their web albums securely with other non-Facebook users. We plan to conduct usability studies on OpenRT to ensure our proposed system is usable for Web 2.0 users.

We also plan to investigate the feasibility for OpenID to function without relying on redirection between CSPs and OpenID identity providers. Phishing attacks on federated identity protocols are a looming threat. OpenID and other similar protocols (e.g., Google AuthSub [14], AOL OpenAuth [15]) may cause users to become accustomed to being redirected to identity provider websites for authentication. If users do not verify the authenticity of these websites before entering their credentials (and they usually do not [16], [17]), phishing attacks are possible. To prevent phishing attacks, users must confirm the authenticity of an identity provider before entering their credentials. Existing research on authenticating web-sites to users include security indicator [18], secure bookmarks for known websites [19], [20], [21], and automated detection and blacklisting of known phishing sites [22]. However, studies suggest that security indicators are ineffective at preventing phishing attacks [17], [16], and blacklisting known phishing sites still suffers from high rate of false-positives and false-negatives [23]. Even with improved security indicators, users still tend to ignore them [16]. How to make OpenID protocol more resilient to phishing attacks is an important task of our future work.

References

[1] T. Oreilly, "What is Web 2.0: Design patterns and business models for the next generation of software," *Communications and Strategies, No. 1, p. 17*, 2007.

[2] J. S. Olson, J. Grudin, and E. Horvitz, "A study of preferences for sharing and privacy," in *CHI '05 extended abstracts on Human factors in computing systems (CHI '05)*. New York, NY, USA: ACM, 2005, pp. 1985–1988.

[3] S. Voida, W. K. Edwards, M. W. Newman, R. E. Grinter, and N. Ducheneaut, "Share and share alike: exploring the user interface affordances of file sharing," in *Proceedings of the SIGCHI conference on Human Factors in computing systems CHI '06:*. New York, NY, USA: ACM, 2006, pp. 221–230.

[4] T. Whalen, "Supporting file sharing through improved awareness," Ph.D. Dissertation, Dalhousie University, Canada, 2008. [Online]. Available: http://www.proquest.com/

[5] D. Florencio and C. Herley, "A large-scale study of web password habits," in *WWW '07: Proceedings of the 16th international conference on World Wide Web*. New York, NY, USA: ACM, 2007, pp. 657–666.

[6] D. Recordon and B. Fitzpatrick, "OpenID authentication 2.0 - final," http://openid.net/specs/openid-authentication-2_0.html, December 2007. [Online]. Available: http://openid.net/specs/openid-authentication-2_0.html

[7] R. Sandhu, E. Coyne, H. Feinstein, and C. Youman, "Role-based access control models," *IEEE Computer*, vol. 29, no. 2, pp. 38–47, 1996.

[8] C. M. Ellison, B. Frantz, B. Lampson, R. Rivest, B. Thomas, and T. Ylonen, "SPKI certificate theory," September 1999. [Online]. Available: http://www.ietf.org/rfc/rfc2693.txt

[9] N. Li, J. C. Mitchell, and W. H. Winsborough, "Design of a role-based trust-management framework," in *SP '02: Proceedings of the 2002 IEEE Symposium on Security and Privacy*, 2002, p. 114.

[10] M. Abadi, M. Burrows, B. Lampson, and G. Plotkin, "A calculus for access control in distributed systems," *ACM Transactions on Programming Languages and Systems*, vol. 15, no. 4, pp. 706–734, 1993. [Online]. Available: http://citeseer.nj.nec.com/64113.html

[11] L. Kagal, T. Finin, and A. Joshi, "A policy based approach to security for the semantic web," in *Proceedings of the 2nd international semantic web conference - ISWC 2003*, vol. 2870. Bangkok, Thailand: LNCS, September 26 2003.

[12] M. Blaze, J. Feigenbaum, and J. Lacy, "Decentralized trust management," in *the 1996 IEEE Symposium on Security and Privacy*, Washington DC, USA, 1996, pp. 164–173.

[13] B. Adida, "EmID: Web authentication by email address," in *The Proceedings of Web 2.0 Security and Privacy Workshop 2008*, Oakland, California, USA, 2008.

[14] Google Inc., "Authsub authentication for web applications," http://code.google.com/apis/accounts/docs/AuthSub.html, December 2008. [Online]. Available: http://code.google.com/apis/accounts/docs/AuthSub.html

[15] AOL LLC., "AOL Open Authentication API (OpenAuth)," http://dev.aol.com/api/openauth, January 2008. [Online]. Available: http://dev.aol.com/api/openauth

[16] S. E. Schechter, R. Dhamija, A. Ozment, and I. Fischer, "The emperor's new security indicators," in *Proceedings of the 2007 IEEE Symposium on Security and Privacy*. Washington, DC, USA: IEEE Computer Society, 2007, pp. 51–65.

[17] R. Dhamija, J. D. Tygar, and M. Hearst, "Why phishing works," in *CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems*. New York, NY, USA: ACM, 2006, pp. 581–590.

[18] A. Herzberg and A. Jbara, "Security and identification indicators for browsers against spoofing and phishing attacks," *ACM Trans. Interet Technology.*, vol. 8, no. 4, pp. 1–36, 2008.

[19] R. Dhamija and J. D. Tygar, "The battle against phishing: Dynamic security skins," in *SOUPS '05: Proceedings of the 2005 symposium on Usable privacy and security*. New York, NY, USA: ACM, 2005, pp. 77–88.

[20] M. Wu, R. C. Miller, and G. Little, "Web wallet: preventing phishing attacks by revealing user intentions," in *SOUPS '06: Proceedings of the second symposium on Usable privacy and security*. New York, NY, USA: ACM, 2006, pp. 102–113.

[21] K.-P. Yee and K. Sitaker, "Passpet: convenient password management and phishing protection," in *SOUPS '06: Proceedings of the second symposium on Usable privacy and security*. New York, NY, USA: ACM, 2006, pp. 32–43.

[22] Earthlink Inc., "Earthlink toolbar: scambloker for windows users," 2008. [Online]. Available: http://www.earthlink.net/software/domore.faces?tab=toolbar

[23] Y. Zhang, S. Egelma, L. Cranor, and J. Hong, "Phinding phish: Evaluating anti-phishing tools," in *Proceedings of the 14th Annual Network and Distibuted System Security Symposium (NDSS 2007)*, 2007.