2009 33rd Annual IEEE International Computer Software and Applications Conference

A Framework for Enabling User-controlled Persona in Online Social Networks

Dongwan Shin, Rodrigo Lopes, William Claycomb
Computer Science and Engineering
New Mexico Tech
Socorro, NM 87801, USA
{*doshin,rodrigo,billc*}*@nmt.edu*

Gail-Joon Ahn
School of Computing and Informatics
Arizona State University
Tempe, AZ 85287, USA
Gail-Joon.Ahn@asu.edu

***Abstract*—As the use of personal information in social network sites seems manifold, including the representation of an individual's digital persona (or social role) and identification, so does the abuse or misuse of the information. The issue of privacy is critically important in this context. In this paper we present a novel framework for enabling user-controlled sharing of sensitive personal information for better privacy protection in current online social networks. Specifically, the framework called U-Control is proposed to facilitate digital persona and privacy management (DPPM) in a user-centric way that it can satisfy diverse privacy requirements and specification, and social network environments. We discuss the design of a security system based on the proposed framework. Finally we discuss a proof-of-concept implementation, along with performance evaluation.**

***Keywords*-social network; privacy; user centricity;**

I. Introduction

Many online social networking (SN) sites have emerged recently and become the central places for social activities. The fundamental building block for the proper operation of such sites is personal information; most SN sites collect and process information regarding their entities, typically individuals, and offer a variety of features such as personalization, affinity sharing, accelerated networking, and novel services [1]. Therefore, SN sites can create a central repository of personal information, which is persistent and cumulative [2]. Consequently, marketers, school officials, government agencies, and online predators can collect data about users through online SN sites. We strongly believe that one of the most challenging problems in SN sites is related to this issue, *privacy*, and it must be addressed immediately.

However, the support for user privacy protection in online SN systems have been limited so far. As a result, this loss of control often makes us exposed to a bewildering excess of intentional and unintended consequences, including criminal activities ranging from identity theft to online and physical stalking; from embarrassment to price discrimination and blackmailing. Just as the evolution of computing has enabled such capabilities of digital society, there must also be a solution that protects the key informational enabler of online SN systems and provides systematic mechanisms to share such information in a more controlled and secure way. We refer to such management as Digital Persona and Privacy Management (DPPM).

In this paper we present a novel framework for enabling user-controlled sharing of sensitive personal information for better privacy in current online SN sites. Specifically, the framework called *U-Control* is proposed to facilitate DPPM in a user-centric way that it can satisfy diverse privacy requirements and specification in social network environments. The basic notion of user-centricity is to give users, not organizations, a larger degree of control over personal information, and it has been used in the federated identity management (FIM) domain to provide a better mechanism for upholding user privacy over identity attributes [3], [4]. To support user-centricity in online SN sites, our framework focuses on 1) ontology-based privacy attribute management for classifying requested privacy attributes and rating tolerability of those attributes by different privacy factors and 2) authenticated dictionary-based selective disclosure and sharing of personal attributes to allow a user to submit a subset of his attributes requested while preserving his privacy on others. The design of a security system based on the proposed framework is discussed. Finally we discuss a proof-of-concept implementation, along with performance evaluation.

The rest of this paper is organized as follows. Section 2 discusses our framework, followed by the discussion of our design and implementation in Section 3 & 4, respectively. Section 5 presents the performance evaluation and Section 6 discusses related works. Section 7 concludes this paper.

II. The Framework: U-Control

We have identified three fundamental services required to support and manage user persona and privacy in online SN systems: identity attribute management, privacy preference management, and selective attribute sharing. We propose a framework called U-Control that enables those services, as shown in Figure 1.

A. Privacy Attribute Management

In order to support privacy preference management, privacy rating of numerical scale of 1 to 5 is given to each user

0730-3157/09 $25.00 © 2009 IEEE
DOI 10.1109/COMPSAC.2009.46

IEEE computer society

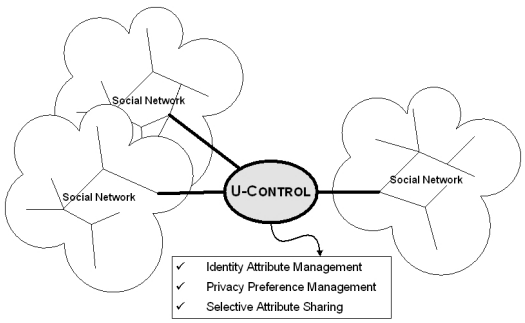

Figure 1. U-Control Framework

attribute in the privacy attribute ontology.[1] The privacy rating 1 is least sensitive and 5 is most sensitive. It should reflect social consensus on privacy. This allows a user to know how other people consider about values of privacy attributes, and it is especially helpful when the user is vaguely familiar with the risk of requested attributes. The privacy rating has the following important characteristics:

- Data Type: Generally speaking, from service requester's point of view, there are two types of data used by service providers: personal data and non-personal data. Given a personal data, one can allude to contact information, financial information, medical information, and so on.
- Data User: Each service requester describes who can use the personal information. We may distinguish various types of user such as owner, collaborator, family, third party, and so on.
- Data Usage: It claims the intention of usage, indicating which data is used or is going to be used.
- Data Validity: Each party (data user) in the system may have different retention rights on the data. The validity element can be either absolute time or relative time. This particular limit for a particular data user should be defined along with ontology.

Common privacy rating can be different among specific social groups, such as age groups and ethnic groups. Each user may set an allowable limit of privacy rating, so that she can be notified when the requested attributes exceed her tolerable limit. In such case, the user may choose her action from: i) reject the request, ii) grant the request, and iii) initiate negotiation with the service provider and revise the manifest to be more tolerable. As for reflecting personal value of privacy, each user should be able to personalize the privacy attribute ontology. Personalization can be done by overriding common privacy rating by her own rating.

Privacy rating to each privacy attribute can be divided into privacy factors which indicate different aspects of damage caused by disclosure. Privacy factors may include:

[1]Due to the page limitation we omit the detailed discussion of our privacy ontology. Please refer to [5] for more information.

Table I
AN EXAMPLE OF RATINGS ON PRIVACY ATTRIBUTES

attribute	parent class	personality	identifiability	financial
name	identity	3	5	4
family members	social.family	4	4	2
income related	financial status	3	1	4
SSN related	identity	3	5	5
hobby	life	4	1	2

- Personality factor: representing seriousness of how much disclosure could embarrass the user's life (such as age, address, education history, and so on).
- Financial factor: representing seriousness of financial damage to the user (such as credit card number, bank account number, social security number, and so on).
- Identifiability factor: representing how much an attribute or combination of attributes has potential risk of the user's identity to be disclosed (such as name, e-mail address, login account name of a service provider, and so on).

Table I illustrates an example of ratings on various privacy attributes. Note that the three privacy factors cover basic principles, and they can be extended to cover more practical necessities such as composite privacy attributes. For instance, the composition of name, address and phone number is often used for verifying person's identity, hence it should be rated higher than each single attribute.

Our privacy attribute management scheme can be realized in the three phases of exchanging user attributes over online social networking sites.

- (Request): the social network (SN) site presents the user a personal information manifest on personal attributes which are necessary for carrying out service for the user. The manifest includes a list of personal attributes, purpose of usage, expiration, and a list of SN sites in the circle of trust (CoT), namely those who will share the information;
- (Evaluation): the personal information manifest is evaluated by a system at the SN site, which reviews the requested personal attributes, and evaluates sensitiveness of the attributes and risk of releasing those attributes. The user has predefined his/her tolerable upper-limit of sensitiveness and risk. If the request exceeds the upper limit, then the user is alerted and encouraged to demand revision of the manifest to the SN site;
- (Negotiation): if the personal information request is intolerable to the user at some point, she may decline release of part of personal attributes, or deny sharing with some of the listed SN sites. The user may choose a safer release scheme, such as downgrading from identity-disclosed release to identity-hidden release, or restricting to pseudonym-based information sharing within the CoT. Such restrictions could result in deterioration of service quality.

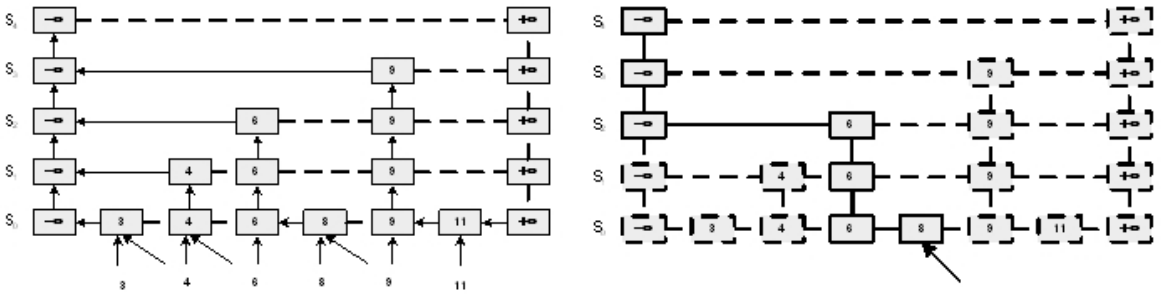

Figure 2. Commutative hash computation (left) & values needed to authenticate a search result(right)

B. Selective Attribute Disclosure and Sharing

Authenticated dictionaries (ADTs) have been primarily studied and used in the context of certificate revocation in public key infrastructure (PKI), especially to implement certificate revocation lists (CRLs). One of the best known examples is based on the Merkle hash tree [6]. Recently, ADT-based approaches to designing a credential system that allows users to selectively disclose their attributes have been made in [7], [8], and they are based on skip lists [9], [10]. A skip list is a data structure that allows the effective search and update of elements within a set. It supports three operations on a set of elements: find(x), insert(x), and delete(x), where x is an element of the set.

To construct a skip list, we need to order elements and form the first list using the ordered elements. Subsequent lists are built on top of the list by selecting randomly some of the elements from the list immediately below. This will be repeated until there is only one element. Two special symbols, $-\infty$ and $+\infty$, represent the lower and higher boundaries in each list, and the last list consists of only these two symbols. An example of the skip list is shown in Figure 2. Searching an element starts at the lower boundary symbol on the top list and continues to the right until the element is found or an higher element. If the element is lower, we will descend to the element immediately below; else we will descend to the element below the previous symbol in the list. The search ends when we find our target at the bottom list or two consecutive elements at the bottom list in which the first is lower than our target, and the second is larger. The latter shows that the element being searched is not in the list. The data structure described above is efficient for search, whose cost is $\mathcal{O}$(log n).

To ensure the integrity, a commutative hashing function called f is used; that is, it is a hashing function that takes two values and returns the same hash independent of the order the values are given. We apply f to every element that is dependent on its previous element, as shown in Figure 2. The lower boundary element on the top list will contain a tag as a result of the function on the element that actually depends on the full list. This last tag will be signed and used to verify the authenticity of the skip list. The commutative hashing is used to allow the function to be computed independently of the order in which both parameters of the function are entered, making the verification process more efficient, and the right part of Figure 2 shows the values needed to authenticate the result of a search.

We use an ADT to represent a credential holding user attributes. Further, the credential allows the user to disclose a subset of his attributes to a verifier. Specifically, personal attributes and corresponding random values are hashed, ordered, and stored in the skip list as elements. Hence, ADT will not contain any user information in it. Additionally, to prevent an offline dictionary attack on the hash value based on the limited domain values of some attributes, personal attributes are salted. The running time complexity of ADT is $\mathcal{O}$(log n) for both verification and update, thereby making its implementation very efficient. Issuing and showing credentials are as follows. After establishing a pseudonym, the user requests the issuer to issue a credential containing the attributes the issuer can assert about the user. The issuer will calculate the hash value of each of the attributes and a corresponding random value that will be included in the credential and then order them by the hash values. From here, ADT can be built, with the last node signed by the issuer's private key. The issuer then sends the credential to the user along with attribute and random value pairs. To show a credential to a verifier, upon the request for personal attributes required for a service, the user sends the set of pairs of requested attributes and random values, and corresponding hash values for each of the attributes to allow the verifier to verify the attribute. This set of attributes corresponds to the actual value of the attributes. The verifier will hash the attributes and corresponding random values to verify that they are in fact part of ADT.

III. System Architecture

The system architecture based on the proposed framework is shown in Figure 3. It shows both functional component based system architecture (in Figure 3-A) and operational system architectures (in Figure 3-B). The first is composed

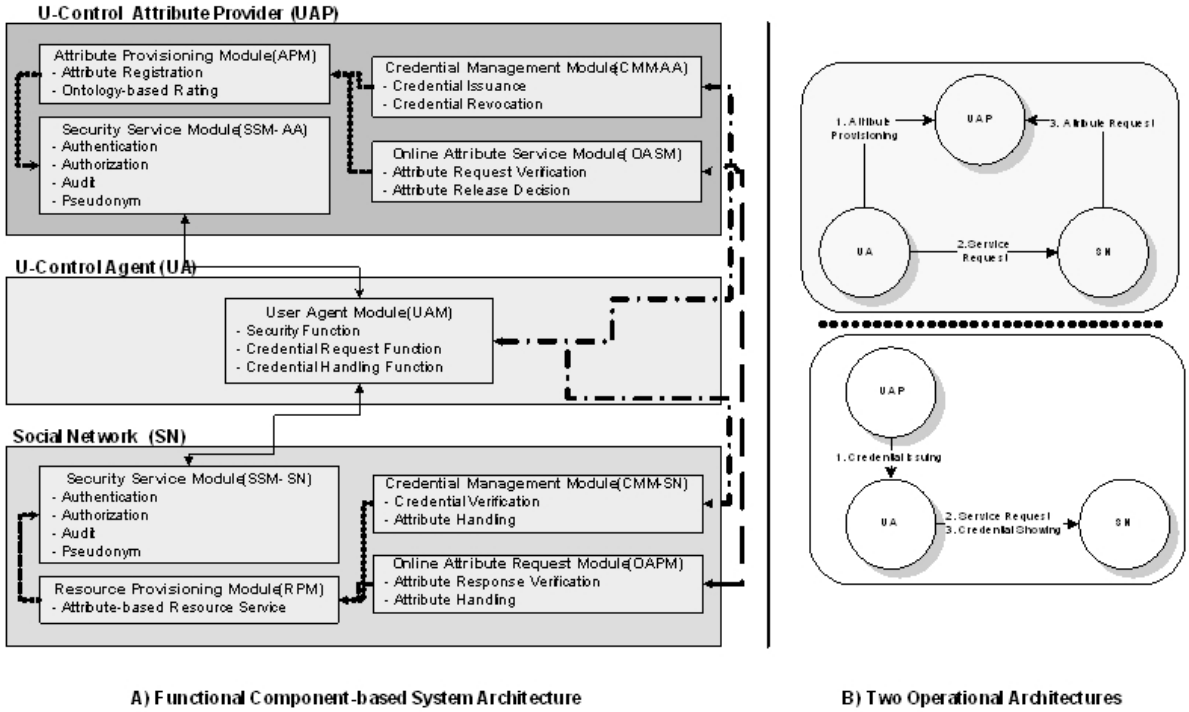

Figure 3. U-Control System Architecture

of three systems, namely a U-Control attribute provider, a U-Control agent, and a social network.

- U-Control attribute provider (UAP): this system has four functional modules; security service module (SSM-AA), attribute provisioning module (APM), credential management module (CMM-AA), and online attribute service module (OASM). SSM-AA supports strong security services like authentication, access control, and audit. In addition to them, it supports the establishment of a pseudonym between the user and itself. APM depends on (shown as the dotted arrow) SSM-AA, and it supports attribute registration and ontology-based privacy attribute rating service, after verifying the user's identity and establishing a pseudonym. The remaining two modules are required to support the selective sharing of personal attributes provisioned in UAP; CMM-AA offers the functional features such as issuing a credential to the user. OASM is needed so that UAP can release the user's personal attributes requested from the social network, and the decision should be made based on the user's privacy preferences configured using APM.
- U-Control agent (UA): this component has one functional module called user agent module (UAM). It supports security functions such as user authentication and secure communication, which are negotiated between UA and UAP, or UA and the social network. It also supports the credential request function when the user wants to request a credential. The credential handling function which is also supported by this module allows the user to store, retrieve, select, and send the credential he want to use in transactions with UAP, or the social network.
- Social Network (SN): upon the service request from the user, this system requires the user to first authenticate himself and then decide the scheme to present attributes required to obtain the service. It has four functional modules; security service module (SSM-SN), resource provisioning module (RPM), credential management module (CMM-SN), and online attribute request module (OARM). These modules are essentially functional counterparts to the four modules in UAP.

IV. Prototype Design and Implementation

We designed and implemented an ADT-based credential system of U-Control. Personal attributes are specified based on the ontology discussed in the previous section. They are strongly typed and represented as a concatenation of the attribute type and attribute value, separated by a colon. As a container for personal attributes in ADT, we decided to use a four-pointer node. The node will contain pointers to all its neighbors in the skip list: up, down, left, and right. Additionally, the ADT-based credential is represented as an XML file for the readability and extensibility.

An attribute can be proved to be an element of the credential by searching it inside the skip list and retrieving the path function values. The U-Control agent system is responsible for doing this on behalf of the user. The user will send only the attribute's type/value, a salting random value, and the corresponding values in the path that allow the re-computation of the signature element. An interesting observation is that the user will most likely disclose several attributes, and these attributes are likely to have overlapping path function values. In such cases, the user only needs to send each repeating value once.

The standard widget toolkit (SWT)[2] was used for developing the prototype UI for issuing, using, and verifying ADT-based credentials in U-Control. The CMM-AA module of UAP was implemented that allows the construction and issuance of credentials. For the present work, the issued credential is written to a file in an XML format, so that the user can add the file to the client module as a credential. In a fully distributed functional system, the XML contents generated would be embedded in a web services request, for example, and transmitted to the client. The CMM-SN module of SN was also implemented to test the functionality required for the verification purpose. This module reads the XML proof request generated by the UA and verifies all the attributes disclosed. The UAM module of UA includes a card picker and an attribute selector. The list of attributes displayed on the attribute selector matches the selected card on the card picker. In addition, it also includes a graphical representation of ADT that shows the full dictionary. Finally, it includes a simple proof request viewer/editor that shows the request built for the currently selected attributes. The editor allows that request to be manually changed, working a valuable tool to build invalid requests in an attempt to fool the verifier and test our current prototype system.

V. Performance Evaluation

We tested and analyzed the performance of our prototype. Several aspects such as attribute proof size and credential generation time were considered to investigate their impacts on the system in terms of performance and extensibility. The experiment to check the average proof size was conducted based on the the number of attributes contained and the number of attributes to be proved. The credential sizes that were generated have from 1 to 100 attributes, randomly selected from a pool of possible attribute types, with no repetitions. The result shows that the average proof time is easily predictable based on a logarithm function because the proof size is directly related to the path size between the signed node and the attribute node being proved.

Having the average proof size for a single attribute, we further experimented to observe the impact of overlapping proof path, which results from multiple attributes to be proved. Figure 4 shows the performance gaining due to overlapping proof path when multiple attributes are proved. It further shows the percentage ratio of the actual proof size over the expected proof size, demonstrating a larger gain as the credential size grows.

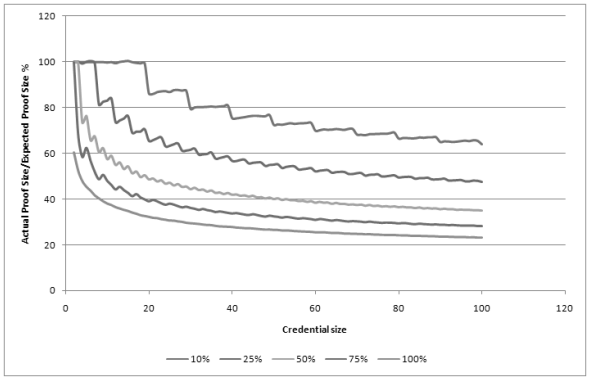

Figure 4. Performance gaining due to overlapping proof path

Verification time is of more importance because a SN site may have to process many simultaneous verifications for many different clients at the same time, while proof generation at client side is not expected to be as intensive. Generating and verifying a proof both have, on average, a complexity of $\mathcal{O}$(v log(n)), where v is the number of attributes to be disclosed and n is the number of attributes contained in the credential. Generating a credential has the most expensive performance of $\mathcal{O}(n^2)$, but this would only happen if all the attributes are present in every list. On average generating a dictionary will take $\mathcal{O}$ (n log(n)), where n is the number of attributes in the credential.

VI. Related Works

The identifiability of information is quite challenging in a sense that even online SN sites that do not expose their users' identities may provide enough information to identify the profile's owner. According to a recent study [11], for instance, a 15 percent overlap of personal information is observed in two of the major social networking sites, which allows sophisticated viewers to know more about an individual than she may want them to. In addition, since individuals often re-use the same or similar photos across different sites, an identified face can be used to identify a pseudonym profile with the same or similar face on another site. The possible recipients of personally identifiable/identified information are social network hosting sites and third party application hosting sites that may abuse or misuse the information. Our current approach is related to this aspect in that it is based on the notion of user centricity [12], [4] in sharing user profile in a sense that the user will have the ultimate authority to share which data in her profile with what SN sites, thus providing the user more control on her profile.

[2]http://www.eclipse.org/swt/

A credential system is a system in which a user can obtain credentials from one organization and demonstrate possession of them to other organizations, and several credential systems have been proposed for achieving user privacy in literature [13], [14], [15]. Chaum's approach to designing the digital cash system [15], based on blind signature techniques, was one of them, also called an anonymous credential system. One major disadvantage of using this system is that a trusted third party is always required that all participating entities are dependent upon.

Similar to Chaum's system, but a more advanced scheme to design an anonymous credential system was presented by Brands [13]. His credential system could support many features such as expressions of any satisfiable proposition from proposition logic, limitation on the number of times a credential may be used, revocable anonymity, and discouragement of lending credentials. Camenisch et al. [14] proposed a credential system that relies on proofs of knowledge like Brands' system. One of the main disadvantages in these credential systems is related to the computational aspect of their cryptographic primitives using number theory and zero-knowledge proof (ZKP).

VII. Conclusion

In this paper we discussed a novel framework called U-Control for enabling user-controlled sharing of sensitive personal information for better privacy protection in current online SN sites. Specifically, the framework is proposed to facilitate digital persona and privacy management (DPPM) in a user-centric way that it can satisfy diverse privacy requirements and specification, and social network environments. We discussed the design of a security system based on the proposed framework. Finally we discussed a proof-of-concept implementation of selective attribute sharing components, along with their performance evaluation.

Acknowledgment

This work was partially supported by the grants from Sandia National Laboratories (PASP10), National Science Foundation (NSF-IIS-0242393 and NSF-CNS-0831360), and Department of Energy Early Career Principal Investigator Award (DE-FG02-03ER25565).

References

[1] R. Gross, A. Acquisti, and H. J. H. III, "Information revelation and privacy in online social networks," in *Proceedings of the 2005 ACM Workshop on Privacy in the Electronic Society*, Alexandria, VA, November 7 2005.

[2] F. B. Vigas, "Blogger's expectations of privacy and accountability: An initial survey," *Journal of Computer-Mediated Communication*, vol. 10, no. 3, 2005.

[3] G.-J. Ahn and J. Lam, "Managing privacy preferences for federated identity management," in *Proceedings of the 2005 Workshop on Digital Identity Management*, Alexandria, VA, USA, November 11 2005, pp. 28–36.

[4] D. Shin, G.-J. Ahn, and P. Shenoy, "Ensuring information assurance in federated identity management," in *Proceedings of the 23rd IEEE International Performance Computing and Communications Conference*, Phoenix, Arizona, April 14-17 2004.

[5] M. Iwaihara, K. Murakami, G.-J. Ahn, and M. Yoshikawa, "Risk evaluation for personal identity management based on privacy attribute ontology," in *Proceedings of the 27th International Conference on Conceptual Modeling*, Barcelona, Spain, October 20-23 2008.

[6] R. C. Merkle, "A digital signature based on a conventional encryption function," in *Advances in Cryptology - CRYPTO 87*, Santa Barbara, CA, August 16-20 1987, pp. 369–378.

[7] D. Shin and R. Lopes, "Enabling interoperable and selective data sharing among social networks sites," in *Proceedings of the 3rd International Workshop on Trusted Collaboration*, Orlando, Florida, November 13 2008.

[8] D. Shin, R. Lopes, and W. Claycomb, "Authenticated dictionary-based attribute sharing in federated identity management," in *Proceedings of the 6th International Conferecne on Information Technology: New Generation*, Las Vegas, Nevada, April 27-29 2009.

[9] A. Anagnostopoulos, M. T. Goodrich, and R. Tamassia, "Persistent authenticated dictionaries and their applications," in *Proceedings of 4th International Conference on Information Security*, Malaga, Spain, October 1-3 2001.

[10] M. T. Goodrich, R. Tamassia, and A. Schwerin, "Implementation of an authenticated dictionary with skip lists and commutative hashing," in *DISCEX II*, 2001.

[11] H. Liu and P. Maes, "Interestmap: Harvesting social network profiles for recommendations," in *Proceedings of IUI Beyond Personalization 2005: A Workshop on the Next Stage of Recommender Systems Research*, San Diego, CA, January 9 2005.

[12] P. Shenoy, D. Shin, and G.-J. Ahn, "Towards ia-aware web services for federated identity management," in *Proceedings of the IASTED International Conference on Communication, Network, and Information Security*, New York City, USA, December 10-12 2003.

[13] S. Brands, *Rethinking Public Key Infrastructure and Digital Certificates - Building in Privacy*. MIT Press, 2000.

[14] J. Camenisch and E. V. Herreweghen, "Design and implementation of the idemix anonymous credential system," in *Proceedings of 9th ACM Conference on Computer and Communication Security*, Alexandria, VA, November 7-11 2002.

[15] D. Chaum, "Security without identification: Transaction systems to make big brother obsolete," *Communications of the ACM*, vol. 28, no. 10, pp. 1030–1044, 1985.