This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2009 proceedings

Privacy-enhanced User-Centric Identity Management

Gail-Joon Ahn [†]
Arizona State University
gahn@asu.edu

Moonam Ko and Mohamed Shehab
University of North Carolina at Charlotte
{mnko,mshehab}@uncc.edu

***Abstract*—User-centric identity management approaches have received significant attention for managing private and critical identity attributes from the user's perspective. User-centric identity management allows users to control their own digital identities. Users are allowed to select their credentials when responding to an authentication or attribute requester and it gives users more rights and responsibility over their identity information. However, current user-centric approaches mainly focus on interoperable architectures between existing identity management systems and privacy issues have not been considered in depth. In this paper, we propose a category-based privacy preference approach to enhance the privacy of user-centric identity management systems. In addition, we present our proof-of-concept prototype of our approach in the Identity Metasystem.**

I. Introduction

Federated identity enables organizations to securely recognize and leverage user identities managed by trusted organizations within or across a circle of trust (CoT). Furthermore, identity federation allows organizations to securely share user profiles with trusted organizations. Since identity federation facilitates the voluminous exchange of sensitive user information, privacy concern associated with such exchanges is an important issue in federated identity management which have been addressed by several research projects [4], [5], [21], [23].

Recently, user-centric identity management approaches have received significant attention for managing private and critical identity attributes. User-centric identity management allows users to control their own digital identities. Users are allowed to select their credentials when responding to authenticator or attribute requester, this gives users more rights and responsibility over their identity information. However, current user-centric approaches mainly focus on interoperable architectures between existing identity management systems without considering privacy issues in depth. In this paper, we propose a category-based privacy preference approach to enhance the privacy of user-centric identity management systems based on Identity Metasystem [8], [20].

The rest of this paper is organized as follows. Section II overviews the user-centric identity management with Microsoft's approaches and discusses the related technologies. Section III discusses the privacy concerns in Identity Metasystem and our category-based privacy preference management. Section IV describes our implementation details. Section V concludes the paper.

II. Background and Related Technologies

In this section, we first start with the discussion of user-centric identity management and briefly explain the Identity Metasystem and Microsoft's CardSpace followed by the privacy technologies.

The user-centric identity management shifts the control of digital identity attributes from organizations to the users by putting the users into the middle of transactions between identity providers and relying parties. By allowing a user to control their own digital identities, the user can decide which identity attributes are needed to share with other trusted parties and under what circumstance. As the users have more rights and responsibilities over their identity information, it provides better protection of the user's private information.

The Identity Metasystem is an interoperable architecture for digital identity management [8]. The architecture of the Identity Metasystem is designed based on the "Laws of Identity" which are intended to codify a set of fundamental principles to which any universally adopted, sustainable identity architecture must conform [7]. Instead of replacing the current identity management systems, the Identity Metasystem provides seamless interoperability between existing and future identity management systems using WS-* web services which is a set of specifications built on the web service platform [20]. In this architecture, the subjects are usually users and each user's digital identities are represented by a visual "information card" in the client user interface [15]. An information card generally contains the card name, card image, a list of claims, and card issuer information. The list of claims in the information card includes pieces of information about user and assertions generated by the card issuer. The Windows CardSpace (InfoCard) [19], which is a component of the Microsoft .NET Framework version 3.0, is an implementation of the Identity Metasystem. It provides the consistent user experience required by the Identity Metasystem and is reinforced against tampering and spoofing to protect the user's digital identity.

The P3P provides a standard way for organizations to publish their privacy policies in a machine readable XML format known as *P3P policy* [18]. The P3P policy presents the data-collection practices that contain information about the type of data, the type of usage, the user of data, the purpose of usage, and how long the data will be retained. The users can specify their privacy preferences using APPEL [17].

There are several related work dealing with privacy in iden-

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

978-1-4244-3435-0/09/$25.00 ©2009 IEEE

tity management. [13] discusses three primary ways to address privacy in Identity Metasystem. [9] introduces seven Privacy-embedded laws of Identity indicating Identity Metasystem should be enhanced to provide privacy and data protection features. The AT&T Privacy Bird as a P3P user agent compares P3P polices against a user's privacy preferences [10]. The AT&T Privacy Bird displays the privacy conformance in page level using P3P specification. However, this approach cannot support the privacy conformance in the input field level. To address this issue, Levy introduces the Integrated Privacy View (IPV) system for checking and displaying privacy conformance information at the input field level [16]. They made an extension to P3P that allows a fine-grained linkage of privacy statements to HTML elements.

The multi-level policy approach [22] is a simplified mechanism for handing privacy preference within Liberty Alliance framework. Using a small number of standardized privacy policies, the relying party indicates at least one of the standardized policies for representing its intended usages of the attributes and the users specify which of the standardized policies would accommodate their preferences. Using standardized policies simplifies policy comparison and conflict resolution. Ahn et al. proposed a privacy preference expression language called PREP for storing the user's privacy preferences with Liberty Alliance enabled attribute providers [4]. The PREP language enables the users to tag their attributes with privacy labels and facilitates privacy-enhanced attribute exchange.

III. Privacy in Identity Metasystem

The Identity Metasystem is designed based on the Seven Laws of Identity to provide a security and privacy enhanced interoperable architecture. From the Laws of identity, *Minimal Disclosure for a Constrained Use* and *Consistent Experience Across Contexts* are the two laws for reinforcing the user's privacy in Identity Metasystem [13]. Based on these laws, a relying party should receive only the required information. At the same time, identity selector agents provide a user friendly interface that enables the users to interact with the identity related services.

Several identity selectors such as the Microsoft Windows CardSpace, Higgins identity selector, Ian Brown's Safari identity plug-in selector, and XMLDAP.org identity selector have been recently introduced and tested for the interoperability between different identity providers and relying parities [6]. Although the human readable privacy policy is provided to help build the user's confidence and trustworthiness in the process of personal information disclosure, it does not make it easier for the users to understand privacy policy since the current approach is complex and inappropriate for the diverse users [12].

We adopt a simple business scenario that we utilize to articulate the necessary approaches for dealing with privacy issues in Identity Metasystem. Our scenario includes three entities: *Armageddon.com* is a relying party that sells books online, *MegaBank.com* is an identity provider that provides online credit card services, and *John* is a customer who is going to purchase a book from the online book store. *Armageddon.com* requests various claims including purchase order and shipping record. For fulfilling this requirement, *John* needs to use two information cards as follows:

```
For credit card information
Card Type: Managed
Issued by: MageBank
Attributes: card name, card number, expiration
          date, card security code, holder name
Requested Claims: card name, card number,
          expiration date, card security code,
          holder name

For shipping information
Card Type: Self-issued
Issued by: John Doe
Attributes: first name, last name, email address,
            street address, city, state, postal
            code, country, primary telephone
            number, secondary telephone number,
            mobile phone number, date of birth,
            gender, web page
Requested Claims: first name, last name, street
          address, city, state, postal code
```

In the purchasing process, *John* would provide the shipping information using his self-issued information card and provide the credit card information using his managed card issued by *MegaBank.com*. From the privacy perspective, the credit card claims and shipping claims may have different privacy sensitivity levels. The relying party and a user may also have different privacy preferences for these claims. Current identity selector agents lack appropriate privacy mechanisms that check conflicts in privacy preferences for the requested claims. More systematic privacy solution is needed to allow relying parties and the users to precisely specify their privacy policy for the claims based on their different privacy aspects, respectively. Moreover, if a user can clearly understand the privacy conflicts for the requested claims before releasing them to relying parties, the user can make a more precise decision for disclosing the requested claims to the relying parties. Furthermore, it enhances the privacy and usability features of Identity Metasystem.

To design our solution, we initially considered the P3P and APPEL as a privacy framework. However, there are some drawbacks for adopting the P3P and APPEL approaches. P3P 1.0 specification supports the definition of site's privacy policy but does not support how to link privacy policies to specific data transferring events, and how to make decisions around such events [14]. Moreover, APPEL preference is hard to express [3], [11]. Our approach uses the multi-level privacy policy approach which uses the *P3PLite* and *PREP* languages to implement the privacy labels in federated identity management system [4] . The multi-level approaches show a simplified way of handling privacy preference in federated identity management system using a limited set of privacy policy which can be decided within a CoT. We propose a mechanism to enhance privacy in Identity Metasystem by using the multi-level privacy labels approach

A. Privacy Labels

We observe that people tend to use labels to represent abstract concepts such as seriousness and completeness. One example is that the United States government uses the five hierarchical color code labels such as Red (Severe), Orange (High), Yellow (Elevated), Blue (Guarded), and Green (Low) to represent the national threat level. By using the labels, people can understand the threat level much easier. We applied this label concept to represent the privacy sensitivity of claims.

The privacy labels are similar to security labels in Mandatory Access Control (MAC). In MAC, every resource or object is entitled with a security label repenting the sensitivity of each resource. A subject needs a legitimate security clearance to access resources. Similarly, every claim in Identity Metasystem is tagged with privacy labels based on the privacy sensitivity of each claim. Each privacy label describes different privacy policies using the W3C's P3P elements. The five privacy labels denote the degree of privacy sensitivity for general privacy policies. Therefore, the relying party can represent their privacy practices for claims by assigning a privacy label to the claims and the users can also define their privacy preferences for their own claims using privacy labels. To compare the hierocratical privacy labels, we define the following privacy label comparison rule.

- Privacy Label Comparison Rule
  - If $L(r) \geq L(u)$, $P(r)$ is satisfied with $P(u)$;
  - If $L(r) < L(u)$, $P(r)$ is not satisfied with $P(u)$,

where $L(r)$, $L(u)$, $P(r)$ and $P(u)$ represent the relying party's privacy label for the claim, the user's privacy label for the claim, the relying party's privacy policy for the claim, and the user's privacy preference for the claim, respectively. By comparing relying party's privacy labels and the user's privacy labels for the request claims, the privacy conciliation for the claims are discovered.

B. Applying Privacy labels

We now discuss how we can apply the privacy label approach to the Identity Metasystem from the relying party's perspective. The relying party defines the claims using OBJECT and XHTML tags in the web page to interact with the identity selector. Applying privacy labels to claims is defined similarly. When the relying party defines the claims, it also links the privacy policy to the claims using P3P*Lite* Language. The P3P*Lite* policy defines the privacy labels for the claims.

On the other hand, applying the privacy labels to the claims in the identity selector is not a trivial task since information cards can have duplicated claims and some claims might require different privacy sensitivities. Therefore, we also identify possible ways to apply the privacy label approach to the current identity selector models.

- **Case 1.** *Applying the privacy labels to each claim.* In this case the privacy labels are assigned to each claim in each information card. A user assigns a privacy label to each claim whenever a self-issued information card is created or a managed information card is imported. For example, a user U, has three information cards $\{I_a, I_b, I_c\}$, where each card has the following claims C_i and labels L_j:
  - $I_a = \{C_1(L_{Strict}), C_2(L_{Strict}), C_3(L_{Strict}), C_4(L_{Strict})\}$
  - $I_b = \{C_3(L_{Moderate}), C_4(L_{Moderate}), C_5(L_{Casual}), C_6(L_{Strict})\}$
  - $I_c = \{C_7(L_{Moderate}), C_8(L_{Strict}), C_9(L_{Casual})\}$

From the case 1, there are two issues. First, whenever a user creates or imports the information cards, the user has to assign the privacy labels to each claim. Although this technique provides a fine-grained privacy control over each claim, it is a cumbersome procedure to the user. Second, the user is able to assign different privacy labels to the duplicated claims. For example, the user U assigns the privacy label L_{Strict} to claim C_3 in information card I_a and may assign the privacy label $L_{Moderate}$ to the same claim C_3 in information card I_b. With a large number of information cards, it is difficult for the user to ensure whether the same privacy label is assigned to the duplicated claims.

- **Case 2.** *Applying the privacy label to each information card.* The privacy label is assigned to each information card instead of assigning privacy labels to each claim. This approach is based on the assumption that each information card has the claims that have the same privacy sensitivity. For example, a user U, has three information cards $\{I_a, I_b, I_c\}$, where each card has the following claims C_i and labels L_j:
  - $I_a(L_{Strict}) = \{C_1,\ C_2,\ C_3,\ C_4\}$
  - $I_b(L_{Moderate}) = \{C_3,\ C_4,\ C_5,\ C_6\}$
  - $I_c(L_{Casual}) = \{C_7,\ C_8,\ C_9\}$

By applying the privacy labels to the information card, the number of applied labels are reduced. However, it lacks a fine-grained privacy labeling since all claims in the formation card follow the identical privacy label of the information card. Moreover, the relying party needs to modify the P3P*Lite* for assigning the privacy labels to a group of claims. We propose a category-based privacy preference approach to overcome the issues identified by the above-mentioned cases.

C. Category-based Privacy Preference

In the identity selector, a user can easily have several information cards which have the duplicated claims. For example, a user can have several managed cards issued by different credit card companies. These cards have the duplicate claims such as holder name and holder address. We assume that these duplicated claims should be assigned with the same privacy sensitivity. Our approach is based on classifying the information cards under a specific category. For instance, the finance category contains the information cards that have financial related claims and the health category contains the information cards pertaining health information claims. In addition, our approach enables the generation of a claim list within each category. The claim list contains a refined set of claims that avoid the duplication of claims. The following case summarizes our proposed approach.

- **Case 3.** *Applying the privacy label to each claim in the claim list within a specific category*. For example, a user U has two categories, namely categories CA_a and CA_b. Each category has information cards listed as below:
  - $CA_a = \{I_a, I_b\}$, where $I_a = \{C_1, C_2, C_3, C_4\}$ and $I_b = \{C_3, C_4, C_5, C_6\}$
  - $CA_b = \{I_c\}$, where $I_c = \{C_7, C_8, C_9\}$

  Accordingly, each category has the following unique set of claims and the privacy labels are assigned to each claim in the claim list as follows:
  - $\psi CA_a = \{C_1(L_{Strict}), C_2(L_{Strict}), C_3(L_{Strict}), C_4(L_{Moderate}), C_5(L_{Casual}), C_6(L_{Strict})\}$
  - $\psi CA_b = \{C_7(L_{Moderate}), C_8(L_{Strict}), C_9(L_{Casual})\}$

In the case 3, the information cards I_a and I_b are grouped into category CA_a. The claim list ψCA_a represents the union of all claims included in category CA_a. The privacy labels are assigned to claims in each claim list avoiding duplicated claims and issues of inconsistent privacy labeling in case 1 while at the same time providing a fine-grained privacy labeling for each claim. Hence, the user is able to create categories and assign information cards to different categories. As shown in Figure 1, a user can manage the privacy preference using a category-based privacy preference approach. Under *Personal* category , the user has two information cards A and B which share claims 3 and 4. To remove the duplicated claims in personal category, we implement the claim list component which generates a unique claim list for each category. The user assigns privacy labels to the different claims included in each category. This avoids the duplicated labeling causing less cumbersome to the user.

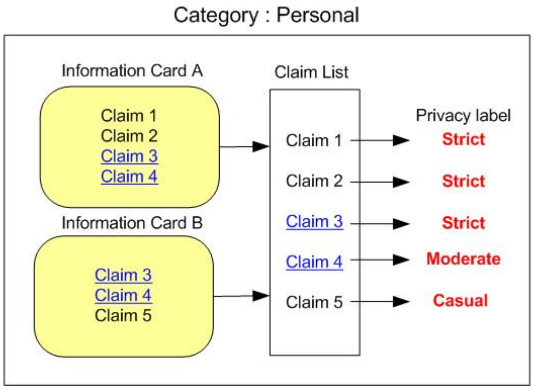

Fig. 1. Category-based Privacy Preference

IV. Implementation Details

We developed a proof-of-concept prototype of our approach in the identity selector, which is a Java-based implementation of Identity Metasystem. In this section we present an overview of our implementation experience and outcomes.

A. Identity Selector

The identity selector is an important component in Identity Metasystem. It helps the users select their digital identity using the visual information cards. Our CardSpace-compatible identity selector provides most functionalities of CardSpace and equivalent user experience of CardSpace. Each information card represents the user's identity in different context and each card contains different subsets of user attributes. Each card mainly includes meta information required to acquire the real attributes from identity providers. The meta information includes the necessary user attribute fields, identity provider contact information, and token information. Our implementation enhances the identity selector with user-specified categories.

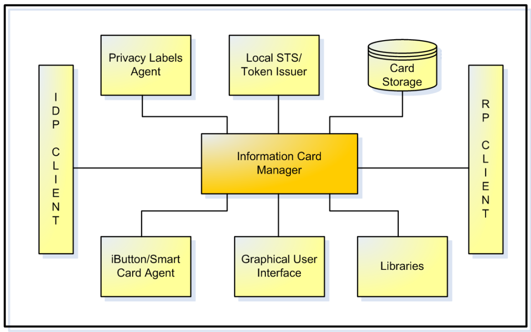

Fig. 2. Components of Java-based Identity Selector

Our identity selector implementation consists of seven components: Information Card Manager, Graphical User Interface, Card Store, iButton/Smartcard Agent, Local STS/Token Issuer, libraries and Privacy Labels Agent as shown in Figure 2. The Information Card Manager handles all events generated by users and systems, and performs the appropriate action. The Information Card Manager helps users manage their information cards and privacy preference under categories. The Graphical User Interface component consists of a set of screens such as the self-issued card screen, category creation screen and so on. The Card Store contains the user's information cards which are stored in XML format. The iButton/Smartcard agent manages the communication between the identity selector and the portable secure devices such as Javacard and Smartcard. The Local STS/Token Issuer generates CardSpace compatible security tokens for self-issued information cards and also transforms the token issued by iButton to the CardSpace compatible security token. Using openSAML 1.1 [2], Bouncy Castle API [1] and our libraries, the local STS/Token Issuer encrypts and signs the XML token. The libraries include the required standard and customized modules that are necessary for supporting the functionalities of identity selector. The Privacy Labels Agent manages privacy preferences and evaluates the user's privacy preferences against the relying party's privacy policy for each requested claim.

B. Privacy Setting

We applied P3P*Lite* language to represent the relying party's privacy policy for the claims in machine readable format. The P3P*Lite* support our privacy label approach using the limited set of P3P elements such as purpose, recipient, retention, access, disputes, and remedies. We locate the P3P*Lite* policy

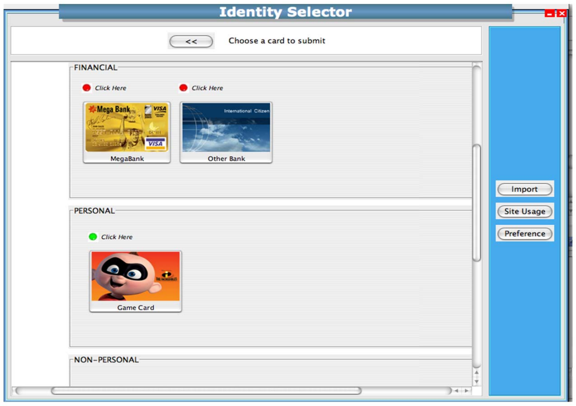

Fig. 3. Privacy enhanced Identity Selector Interface

at the well-known location (/w3c/p3plite.xml) which is similar to the location of the P3P policy. When the identity selector receives a set of requested claims from a relying party it directly retrieves the relying party's privacy policy from a predefined location.

The PREP language is designed to support the attribute provider that collects and stores the user's privacy preferences in federated identity management system. We modified PREP language to support the Identity Metasystem. The identity selector collects the user's privacy preferences when the user creates or imports a new information card into a category. When a new information card is added to a category, the claim list component compares the claims in the new information card against the existing claim list in the category. If the new claims are in the new information card, the user should assign privacy labels to new claims. However, if the claims in the new information card are duplicated with the existing claims in the claim list, the duplicated claims follow the privacy label in the existing claims. The collected user privacy preferences are represented in the PREP format.

The Privacy Preference Engine which is a part of the privacy label agent evaluates the relying party's privacy policy against the user's privacy preference for each claims. It utilizes the decision matrix to expedite the evaluation process. The matrix includes all possible policy labels and prompt options. The Privacy Preference Engine firstly checks whether the incoming policy label matches with the user defined privacy label using the matrix then returns a corresponding decision. The returned decision value eventually displays on user interface using colored icons based on the specified prompt action as illustrated in Figure 3. The red icon over the information card informs the user that there exists a privacy conflict and the green icon indicates no claim conflict. If the user clicks the information card with red icon, the identity selector shows the details of the conflicting claims.

V. Conclusion and Future works

In this paper, we have introduced a category-based privacy preference management for user-centric identity managment. To demonstrate our approach, we have developed a CardSpace compatible identity selector using Java and extended privacy utility functions for P3P*Lite* and PREP languages. We believe our proposed privacy management approach could help users control private attributes before sharing them with relying parties by examining relying party's privacy policy and the user's privacy preferences for the requested claims. Our future work includes a rigorous study to measure effectiveness and usability of our approach. It would help us identify more practical requirements for protecting the user's privacy using an identity selector and investigate a possible enhancement of our interface design.

References

[1] The legion of the bouncy castle. Available at http://www.bouncycastle.org/.

[2] Opensaml - an open source security assertion language toolkit. Available at http://www.opensaml.org/.

[3] R. Agrawal, J. Kiernan, R. Srikant, and Y. Xu. Xpref: a preference language for p3p. *Computer Networks*, 48(5):809–827, 2005.

[4] G.-J. Ahn and J. Lam. Managing privacy preferences for federated identity management. In *Digital Identity Management*, pages 28–36, 2005.

[5] M. Alsaleh and C. Adams. Enhancing consumer privacy in the liberty alliance identity federation and web services frameworks. In *Privacy Enhancing Technologies*, pages 59–77, 2006.

[6] Burtongroup. Recapping the catalyst user-centric interop.

[7] K. Cameron. The laws of identity. Whitepaper, Microsoft, Available at http://msdn.microsoft.com/webservices/, May 2005.

[8] K. Cameron and M. Jones. Design rationale behind the identity metasystem architecture. Whitepaper, Microsoft, Available at http://www.identityblog.com.

[9] A. Cavoukian. The case for privacy-embedded laws of identity in the digital age. Technical report.

[10] L. F. Cranor, M. Arjula, and P. Guduru. Use of a p3p user agent by early adopters. In *WPES*, pages 1–10, 2002.

[11] L. F. Cranor, P. Guduru, and M. Arjula. User interfaces for privacy agents. *ACM Trans. Comput.-Hum. Interact.*, 13(2):135–178, 2006.

[12] M. J. Culnan and G. R. Milne. The culnan-milne survey on consumers and online privacy notices: Summary of responses. Technical report.

[13] T. Daemen and I. Rubinstein. The identity metasystem: Towards a privacy-compliant soultion to the challenges of digital identity. Technical report.

[14] G. Hogben. Suggestions for long term chagnes to p3p. 2003.

[15] M. B. Jones. The identity metasystem: A user-centric, inclusive web authentication solution. Positionpaper, Microsoft, Available at hwww.w3.org/2005/Security/usability-ws/papers/28-jones-id-metasystem/, March 2006.

[16] S. E. Levy and C. Gutwin. Improving understanding of website privacy policies with fine-grained policy anchors. In *WWW*, pages 480–488, 2005.

[17] M. L. Lorrie Cranor and M. Marchiori. A p3p preference exchange language 1.0 (appel1.0). Technical report.

[18] M. M. M. P.-M. Lorrie Cranor, Marc Langheinrich and J. Reagle. The platform for privacy preferences 1.0 (p3p1.0) specification. Technical report.

[19] Microsoft. Windows cardspace.

[20] Microsoft. Microsofts vision for an identity metasystem. Whitepaper, Microsoft, Available at http://msdn.microsoft.com/webservices/understanding/advancedwebservices/default.aspx? pull=/library/en-us/dnwebsrv/html/identitym etasystem.asp, May 2005.

[21] B. Pfitzmann. Privacy in enterprise identity federation - policies for liberty single signon. In *Privacy Enhancing Technologies*, pages 189–204, 2003.

[22] L. A. Project. Liberty architecture framework for supporting privacy preference expression language (ppels). Technical report.

[23] A. C. Squicciarini, A. A. Hintoglu, E. Bertino, and Y. Saygin. A privacy preserving assertion based policy language for federation systems. In *SACMAT '07: Proceedings of the 12th ACM symposium on Access control models and technologies*, pages 51–60, New York, NY, USA, 2007. ACM Press.