Managing Privacy Preferences for Federated Identity Management*

Gail-Joon Ahn and John Lam
University of North Carolina at Charlotte
9201 University City Blvd
Charlotte, NC 28223, USA
http://www.sis.uncc.edu/LIISP/
{gahn,jwlam}@uncc.edu

ABSTRACT

We have witnessed that the Internet is now a prime vehicle for business, community, and personal interactions. The notion of identity is the important component of this vehicle. Identity management has been recently considered to be a viable solution for simplifying user management across enterprise applications. The network identity of each user is the global set of personal credentials and preferences constituting the various accounts. The prevalence of business alliances or coalitions necessitates the further evolution of identity management, named federated identity management (FIM). The main motivation of FIM is to facilitate the federation of identities among business partners emphasizing on ease of user management. In this paper, we investigate privacy issues in FIM, especially focusing on *Liberty Alliance* approach. We attempt to identify practical business scenarios that help us understand privacy issues in FIM. Also, we propose systematic mechanisms to specify privacy preferences in FIM.

Categories and Subject Descriptors

K.4.4 [**Computers and Society**]: Electronic Commerce—*Security*; K.6.5 [**Management of Computing and Information Systems**]: Security and Protection

General Terms

Management, Security, Languages, Verification

Keywords

Privacy, Identity Management, Policy Languages

*This work was supported by the grants from Bank of America through e-Business Technology Institute at the University of North Carolina at Charlotte.

DIM'05, November 11, 2005, Fairfax, Virginia, USA.

1. INTRODUCTION

As enterprises have changed their business operation paradigm from brick-and-mortar to click-and-mortar, they have embraced a variety of enterprise applications for streamlining business operations such as emailing systems, customer relationship management systems, enterprise resource planning systems, supply chain management systems, and so on. However, a non-trivial problem has been compounded by this reinforcing line of enterprise applications, *the problem of managing user profiles.* The addition of such applications has proved to be subject to bringing in a new database for storing user profiles and it was quite costly and complex to manage all those profiles, which were often redundant. Considering business-to-business environments, where a set of users consists of not only their employees or customers but also those of their partners, the abovementioned problem became even worse. As a set of underlying technologies and processes overarching the creation, maintenance, and termination of user identities, identity management (IM) has been recently considered to be a viable solution for resolving such issues.

Furthermore, the prevalence of business alliances or coalitions necessitates the further evolution of IM, so called federated identity management (FIM). The main motivation of FIM is to enhance user convenience and privacy as well as to decentralize user management tasks through the federation of identities among business partners. As a consequence, a cost-effective and interoperable technology is strongly required in the process of federation. Web Services (WS) can be a good candidate for such requirement as it has served to provide the standard way for enabling the communication and composition of various enterprise applications over distributed and heterogeneous networks [1, 2].

Since identity federation is likely to go along with the exchange of sensitive user information in a highly insecure online environment, security and privacy issues associated with such exchanges are key concerns in FIM. The concept of federated identities provides the consumers with a convenient way to create identities and move among various business nexus. Apart from all the simplicity and convenience that it provides the businesses with, the management of these federated identities becomes a crucial task since it needs to take into consideration various threats against the vulnerable and confidential user data. Any identity management framework must adequately protect sensitive user information and must adhere to important elements of privacy pol-

icy. In this paper, we describe business scenarios that help us understand privacy issues in FIM. Also, we propose systematic mechanisms to specify privacy preferences in FIM.

The rest of this paper is organized as follows. Section 2 overviews three approaches involved in IM and discusses the prior research works in IM followed by an overview of FIM models. Section 3 articulates business scenarios for FIM and relevant privacy requirements. Section 4 proposes a privacy preference expression language along with the related works. Section 5 concludes this paper.

2. IDENTITY MANAGEMENT

In this section, we first start with the discussion of IM approaches. We categorize IM approaches into the following three styles: *isolated IM*, *centralized IM*, and *distributed IM*. Thereafter, we discuss the related research works.

The isolated IM model is the most conservative approach of the three models. Each business forms its own identity management domain (IMD) and has its own way of maintaining the identities of users including employees, customers, and partners. Hence, this model is simple to implement and has a tight control over user profiles. However, it is hard to achieve user convenience with this model since different IMDs are likely to have different authentication processes or mechanisms for their users and corresponding authentication policies may vary between players.

The centralized IM model has a single identity provider (IDP) that brokers trust to other participating members or service providers (SP) in a Circle of Trust (CoT). IDP being a sole authenticator has a centralized control over the identity management task, providing easy access to all SP domains with simplicity of management and control. The drawback of this approach is a single point of failure within a CoT infrastructure in case that IDP fails to provide authentication service. User convenience can be also achieved partially in case where the single sign-on (SSO) for users is only effective within SPs which belong to the same CoT.

The distributed IM model provides a frictionless IM solution by forming a federation and making authentication a distributed task. Every member agrees to trust user identities *vouched for* by other members of the federation. This helps users maintain their segregated identities, making them portable across autonomous policy domains. It also facilitates SSO and trust, thereby allowing businesses to share the identity management cost with its partners. Microsoft Passport is based on the centralized IM model, while Liberty Alliance aims to be the distributed IM model.

Earlier works related to user identity management were mostly focused on a user-centric approach [8], where users have control over IM functions. A simple idea of managing user identities is described in [4]. They proposed the use of personal card computers to handle all payments of a user, thereby ensuring the privacy and security of the user's identity on the Web. Hagel and Singer [11] discussed the concept of *infomediaries* where users have to trust and rely on a third party to aggregate their information and perform IM tasks on their behalf while protecting the privacy of their information. The Novell digitalme technology [7] allows users to create various identity cards that can be shared on the Internet according to users' preferences. Users can control both what information is stored in each card and conditions under which it may be shared.

2.1 Federated Identity Management

Federated identity gives the ability to securely recognize and leverage user identities owned by trusted organizations within or across CoTs, and identity federation allows organizations to securely share confidential user identities with trusted ones, without requiring users to re-enter their name and password when they access their network resources. Additionally, identity federation provides the ability to optionally and securely share user information such as their profiles or other data between various trusted applications which is subject to user consent and organizational requirements.

This section overviews two well-known FIM solutions, Liberty Alliance and Microsoft Passport. These solutions have fundamentally the same goal of managing web-based identification and authentication. Both enable organizations to build IM systems that can federate across many disparate sources as shown in Figures 1. Therefore, each user can have a single network identity that provides SSO to the web sites that have implemented either or both of the systems.

2.1.1 Liberty Alliance

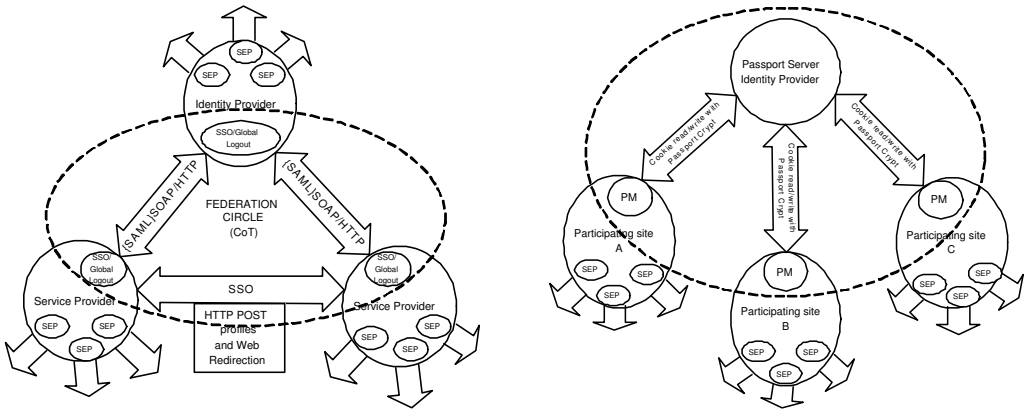
Liberty Alliance is a consortium of more than 150 companies working together towards developing an open, interoperable standard for FIM [12, 20]. It is aimed towards realizing the notion of a cohesive, tangible network identity, which can facilitate SSO and frictionless business operations. It is a distributed IM model, relying on the notion of IDP and SP, as we discussed earlier. IDP is responsible for carrying out identity federation. Authentication messages or authentication requests are passed between IDP and SP. IDP and SP in Liberty Alliance Model actually facilitate WS to discover service locations and handle incoming messages from other IDP and SP.

2.1.2 Microsoft Passport

Microsoft Passport provides authentication services for Passport-enabled sites called participating sites [16]. It was initially released as a service and not an open specification and precedes Liberty Alliance by at least a year. It is the underlying authentication system of Microsoft Hotmail and Microsoft Network, and it is integrated for use in Windows XP. A centralized Passport server is the only IDP in Passport model and contains users' authentication credentials and the associated unique global identifier called Passport Unique Identifier (PUID). Passport is an example of a centralized IM model. Unlike Liberty Alliance, cookies play a major role in Passport architecture where Passport server stores and reads identity information in the form of session and browser cookies stored securely at a client side.

3. PRIVACY CONCERNS IN FIM

Privacy is a growing concern with FIM models due to the voluminous exchange of sensitive information that occurs across enterprises. Securing communication channels and encrypting messages may help preserve the privacy of relevant information only up to some extent. The security concerns that we discussed in [3, 18, 19] are obviously applicable to privacy as well. In WS-enabled FIM where the receiver of a message may not be its ultimate destination, improper security measures may result in unauthorized access to user's personal information which leads to violation of privacy [13].



(a) Liberty Alliance model

(b) Passport model

Figure 1: FIM Models

Protection of user identities and personal information can be achieved by using the principle of pseudonymity. Obfuscating message payloads can also preserve their privacy by making them accessible only through authorized parties having proper credentials or keys [17]. Privacy enhancing technologies like Platform for Privacy Preference (P3P) [6] provide a solution for point-to-point privacy protection based on user preferences. However, such solutions do not scale for a more open, interoperable WS architecture.

Liberty Alliance's SAML implementation uses pseudonyms constructed using pseudo-random values that have no discernable correspondence with users' identifiers at IDP or SP. The pseudonym has a meaning only in the context of the relationship between the two communicating parties. The intent is to create a non-public pseudonym so as to contravene the linkability to users' identities or activities, thereby maintaining the privacy.

Organizations using FIM models are required to follow four key principles of fair information practices which are discussed in [9]:

- *Notice*: Users should receive prior notice of the information practices.
- *Choice*: Users have a choice to specify what information will be used and the purpose for which the information is collected.
- *Access*: Users should be able to access and modify their personal information if necessary and when needed.
- *Security*: Users should be assured that the organizational system is capable of securing their personal information.

Liberty Alliance specifications have recently proposed an approach to sharing user attributes on the basis of user's permission [14, 12]. The specifications also provide a set of guidelines that will help businesses adhere to these principles. Microsoft Passport's approach to online privacy is also based on adherence to these aforementioned principles.

Now we describe business scenarios that we utilize to articulate necessary elements in dealing with privacy issues for federated identity management, focusing on Liberty Alliance specifications. Our scenarios have two hypothetical entities: a financial service institution *Mega Bank* that has online banking services and an online stock trading and brokerage company *Corporate.com*. We assume that *Mega Bank* and other Web Services Consumer (WSC) are Liberty enabled entities and recognize each other as the member of their CoT. Also a WSC has the ability to request one or many attributes which may or may not contain Personally Identifiable Information (PII). In addition, all WSCs and *Mega Bank* have a central policy in the Usage Directives and the user has read and agreed to the posted privacy policies at each service provider before signing up with them. Finally, the user has stored her privacy preferences at the Attribute Provider for some or all of her PII.

We now categorize our scenarios from *Mega Bank* perspectives. *Mega Bank* can act as either attribute provider or attribute requester. Our study was conducted with actual experimentations of each case. As shown in Figure 2, we identify two major cases:

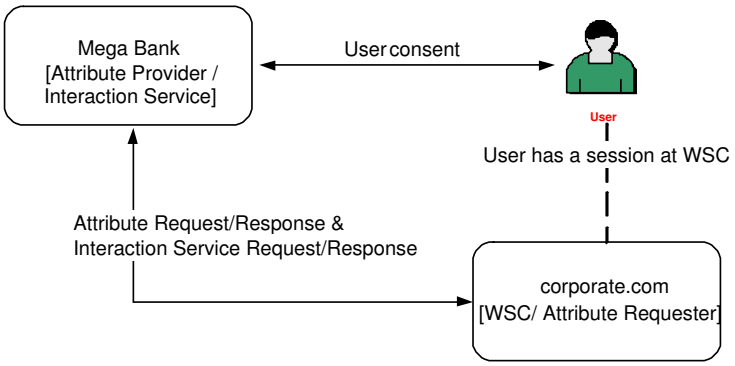
1. *Mega Bank* acting as an Attribute Provider [1]

   Under this scenario, we identify three different cases based on the interaction service (IS) patterns that initiate a communication channel with a user to obtain the user's consent.

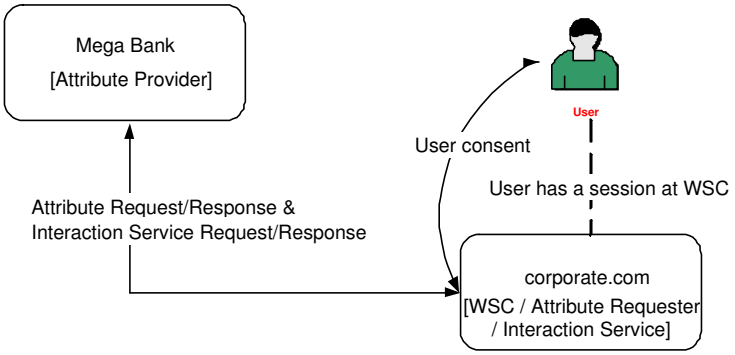
   a. *Direct Interaction with the user for obtaining consent*: Mega Bank can initiate an IS for obtaining user consent before actually releasing the attribute to the WSC. The IS instance is initiated in case of a policy level mismatch between user's stored preferences and the policy level for the intended usage.

   b. *Indirect Interaction through another WSC*: Mega Bank serves only the attribute request without

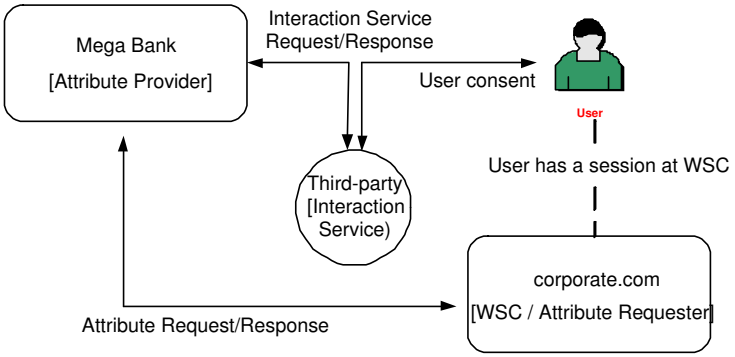
[1] *Mega Bank* can also serve as an IDP or can have another IDP in the CoT. However, since the role of an IDP is limited in our scenarios, we omit such cases in this paper.



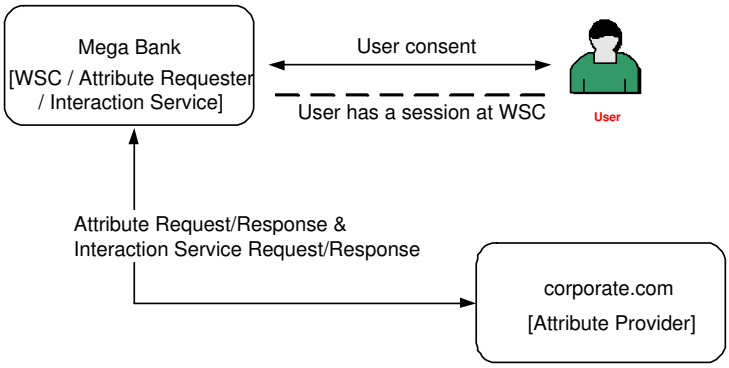
(a) Direct Interaction



(b) Indirect Interaction



(c) Indirect Interaction through a Third-party



(d) *Mega Bank* acting as an Attribute Requester

Figure 2: Practical Scenarios in FIM

invoking an IS by itself. The interaction service is invoked by the same WSC who has requested a user attribute. In this case, Mega Bank (Attribute Provider) does not have a direct interaction with the user.

c. *Indirect Interaction through a third-party IS on behalf of the user*: Mega Bank communicates with a third-party IS for obtaining user consent. The third-party IS for the user is discovered using the ID-WSF Discovery service [14]. Mega Bank and the WSC do not have any direct interaction with the user.

2. *Mega Bank* acting as an Attribute Requester

   As an Attribute Requestor, Mega Bank sends attribute requests to Corporate.com which provides PII as an Attribute Provider. For this case, Mega Bank invokes an IS to establish a direct interaction with the user for obtaining a consent.

4. PREFERENCE EXPRESSION FOR PRIVACY

Our investigation indicated that the current FIM practices lack a well-defined and standardized structure for privacy policies to support identified practical scenarios in Section 3. In addition, there is no systematic protocol for obtaining and storing users' preferences. Another important component to match the privacy policies with users' consent has not been fully discussed in the literature. In this section, we propose a privacy preference expression language called PREP which stands for PReference Expression for Privacy. PREP is a language for storing the user's privacy preference with Liberty enabled attribute providers.

4.1 Motivation and Related Works

Liberty Alliance approach aims to facilitate attribute exchange in the context of user's permission. This means that the user should be put in control of the release and usage of their information stored at the attribute provider. The ID-WSF [14] architecture provides a protocol for transferring the privacy related information in the request and responses. The attribute request and response messages can carry this information in the UasageDirective containers. UsageDirective containers are XML tags that carry such information regarding the usage of the requested user attribute. UsageDirective from the requesting party in an attribute request message specifies the intended usages of the data while the UsageDirective from the Web Services Provider (WSP) in the corresponding response message specifies the user's defined privacy preference or policy for the requested data element. Liberty Alliance has proposed a multi-level policy based approach for addressing this issue. Instead of a large number of varied and personalized privacy policies, there can be a small number of standardized privacy policies to which both attribute requestor and user's or attribute providers acting on behalf of the user can refer. This simplifies the matching of the strictness level of privacy policy in the attribute request message with the user's preferred strictness for attribute release by referring to the UsageDirective element in the request and sending the appropriate response based on the match.[2]

[2]More detailed explanation can be found in [12, 14].

We now introduce an example to elaborate our approach and to highlight the need for PREP:

Consider that a user Cathy has requested a transaction at one of the SPs in a CoT. We assume that Cathy has been authenticated by the WSC at this point. The WSC may require some information regarding Cathy in order to complete the transaction. As a result the WSC makes an attribute request to Cathy's designated WSP which for simplicity in our case would be the IDP. For preserving privacy of the user information, which is the main goal of Liberty Alliance specifications, the IDP should release the requested attribute information with a proper user consent. IDP has already stored the user's preference regarding the release of information based on a multi-level policy approach, meaning that the user has categorized her personal information to be released with different levels of strictness. These strictness levels are directly pointed to the levels of standardized policies defined in the CoT. In such a case, WSPs just need to compare the privacy policy level in the request with the level in the preference and release the information to attribute requester(s) accordingly. In case of a mismatch, WSP can take appropriate actions preferred by the user and already stored in some form at the WSP.

It is obvious that we need to allow SPs and principals to precisely specify the different aspects of their privacy policies, respectively. The various approaches can be considered to support the above scenarios. We may consider P3P [6] as a privacy framework for our scenarios. The major drawback for adopting the P3P based approach is the complexity in determining an intersection of the attribute requestor's privacy policy and the user's privacy preference policy in an automated fashion.

Using a P3P based approach would require a language like APPEL [5] for the WSPs to collect and store the user preferences. APPEL is a privacy preference expression language for P3P but it is very hard to understand and needs a special engine for a browser agent. According to P3P specifications, a single policy can have multiple statements covering different purposes for data collection. In an environment like the one we mentioned in our example, it would be a tough job for WSPs to evaluate all the permutations and combinations between the WSCs policies and the user's set of preferences in APPEL.

There are other related approaches. EPAL [17] is a privacy authorization language that can support authorization and is more stringent for Liberty Alliance requirements. Also, the enhanced SAML [10] can be considered as a way to support user friendly privacy/preference expressions.

Considering all the issues we discussed above, there is a clear need for languages to specify standardized privacy policies and to store the user preferences for corresponding such policies. The multi-level policy approach in Liberty Alliance specifications addresses the purpose of defining a set of standardizes policies for the CoT that both the users and WSCs may refer to. However, it does not propose any specification or rules for storing user preferences in a way that would facilitate the WSPs in matching the privacy policy levels in the attribute request with the levels in the user preferences.

Our work partially adopts P3P to contain various elements that define the web sites privacy policies regarding the purpose of information gathering, release procedures of information and various other factors such as access control

```
<prep: Preference
  xmlns:prep="http://schemas.liisp.net/leplang/pref">
  <prep: Policy ref="http://circle-of-trust.com/policies/strict">
    <prep: DataGroup>
      <pp:Data type="static" xmlns:pp="urn:liberty:idpp:1.0">

        <pp:select>/pp:PP/pp:CommonName</pp:select>
        <prep:prompt action="always"
                 message="Common Name is requested"/>

      </pp:Data>
                ....
                ....
    </prep: DataGroup>
  </prep: Policy>
</prep: Preference>
```

Figure 3: Defining the PREP elements

to the collected data, dispute resolution methods, and so on. We name our privacy policy framework as P3P*Lite*. The discussion of P3P*Lite* is omitted for brevity in this paper. For the privacy preference, we use the preference language called PREP that is much similar to APPEL. PREP shares the same extension formats of APPEL but is more restrictive than APPEL. The important drawback of APPEL in our work is that it cannot be customized to fit our scenarios because it does not support a multi-level policy approach suggested in Liberty Alliance specifications.

4.2 Elements, Operations and Semantics in PREP

PREP is a language used by the attribute provider to collect and store the user's privacy preferences. It further facilitates the decision process in legitimately releasing attributes at the attribute provider by comparing the policy level in the request with the level in PREP at the attribute provider's site. Also, PREP supports multi-level privacy policy approach in Liberty Alliance specifications. The set of standardized privacy policies should be formalized by a mutual agreement between all the entities of the CoT. There are a couple of assumptions that PREP inherits directly from Liberty Alliance specifications [15]:

- The WSP has previously collected a principal's consent, access and privacy preferences/policies for the attributes in question.
- The COT has a web site of its own, or uses an external "Policy Broker" web site, where the privacy policies are available online.
- The SP/WSC sets the Privacy Policy and the principal specifies the Usage Policy (such as preferences).
- The consistent naming is used to indicate *who* decides *which* policy is applied to *what* attributes. In other words, the following hypothetical relationship should be supported: $\text{PrivacyPolicy}_X = \text{UsagePolicy}_X$.

The WSP collects the user's privacy preference at the time of sign-up. Irrespective of the methods used for collecting the user's preferences, the preference should be stored in the format specified in the PREP structure. Just like other standardized protocols proposed by Liberty Alliance, all entities in the COT should be mandated to follow the PREP structure for managing privacy preferences, as each entity acts as an attribute provider.

The PREP contains a set of elements that help the attribute provider store privacy preferences provided by the user into a standardized machine readable XML format. The conversion of the user preferences from a high level to XML is done by the PREP generator. The PREP generator is a program that takes the input from the user and converts it into an XML file satisfying the PREP structure for the user preferences.

The PREP elements are illustrated in Figure 3. We briefly overview the structure of PREP.

<leppl: Preference>

PREP policy must have one Preference element. Preference element signals the start of the PREP policy and can contain various other sub-elements that actually define the user's preferences for the appropriate privacy policies in the relevant CoT. A PREP policy ends with a </LEPPL: Preference>.

<leppl: Policy>

The Policy element signals the start of the user preferences for a particular policy type. There is one Policy element for every policy in the set of standardized policies. The Policy element can contain several sub-elements that encode the user attribute information like the names of the user's attributes and the time they were last updated by the user. This would be helpful in cases where the user wants to update the personal information or the attribute requestor wants to make a decision whether to use the attribute or not based on the freshness of the attribute.

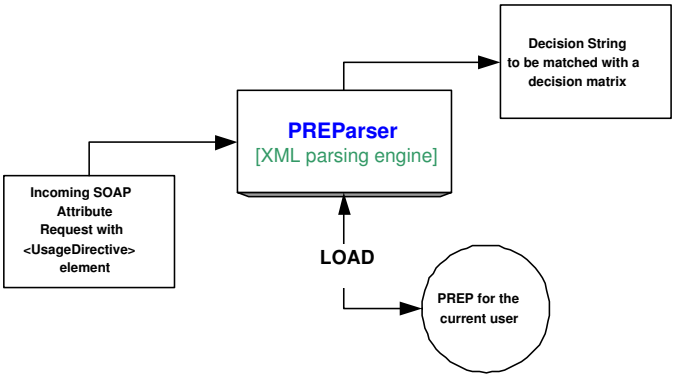


Figure 4: PREParser

Policy Match (True or False)	Prompt Action: Always	Prompt Action: Mismatch	Prompt Action: Never	Return Value
True	True	False	False	1100 (Policy match, Prompt user)
True	False	True	False	1010 (Policy match, Do not prompt)
True	False	False	True	1001 (Policy match, Never prompt)
False	True	False	False	0100 (Policy mismatch, Prompt user)
False	False	True	False	0010 (Policy mismatch, Prompt user)
False	False	False	True	0001 (Policy mismatch, Never prompt)
True	True	True	True	1111 (No Operation)
False	False	False	False	0000 (Missing Attribute)
-	-	-	-	Other cases (Invalid Preference)

Table 1: Decision Matrix

ref

This is an extension to the Policy element and is used for specifying the name/reference of the policy that the Policy element refers to. This element can take values from the policies in the standard set of policies. This is a mandatory extension. The default value can be set to the lowest or the most casual policy.

```
ref = {uri defining the policy location}
```

<leppl: DataGroup>

The Policy element can have a DataGroup element. The DataGroup encapsulates one or more Data elements. There can be only one DataGroup for a policy element. The DataGroup signifies a set of user attributes that the privacy policy in the Policy element deals with.

<leppl: Data>

The Data element contains various extensions and sub-elements that correspond to the user's personal attributes and also some additional information with respect to the user's preferences on notification methods.

type

The type extension specifies the type of data that is represented by the Data element. The type can be either static or updatable.

```
type = {‘"’static‘"’ | ‘"’updatable‘"’}
```

xmlns

The xmlns extension can contain the namespace for the kind of attribute in the Data element.

```
xmlns = {‘"’urn: liberty: idpp: 1.0‘"’ |
         ‘"’urn: liberty: idep: 1.0‘"’}
```

<pp: select>

The Select element forms the sub-element of Data element and represents the actual attribute name that may contain some values provided by the users. Eventually such values make PII. The attribute name should follow the nomenclature as proposed in the Liberty Alliance specifications, particularly in ID-SIS-PP and ID-SIS-EP profiles [12, 20, 14].

```
select = {<user attribute>}
```

<prompt>

The Data element may also contain a Prompt element specifying the user's choice regarding data requests in the corresponding Select element. The Prompt element has two extensions or attributes namely action and message.

```
prompt = { <action> <message>}
```

action

The Action extension specifies the user preferred prompt action. The possible values include always, never, and mismatch. These action values allow us to enforce a multi-level policy approach.

```
action = {‘"’always‘"’ |
          ‘"’never‘"’ |
          ‘"’mismatch‘"’}
```

message

The Message element contains a text message that is displayed in the prompt window. The message is based on the action defined by the user in the Action element.

```
message = {message to the user}
```

Based on the proposed structure, we also developed a mechanism to process user preferences specified in PREP. We named it PREParser. Figure 4 demonstrates the role of PREParser. PREParser is an XML based rule engine for PREP. It evaluates the user preferences upon receiving an attribute request message. We utilize the decision matrix to expedite the evaluation process. The matrix includes all possible policy levels and preference types. PREParser firstly checks whether the incoming policy level matches with the user defined privacy preferences using this matrix then returns a corresponding decision. The matrix scales down the number of expected outputs from the parser as shown in Table 1. The returned decision value eventually triggers the interaction service based on the specified prompt action. PREParser processes the PREP rule set according to the following guidelines:

- A PREP rule set should start with a <leppl: Preference> tag and should contain the `xmlns` extension that specifies the namespace for the XML Schema for PREP. The absence of <leppl:Preference> tag invalidates the rule set and any further processing should be aborted.
- Every PREP rule set can have only one <leppl: Preference> element but can have multiple <leppl: Policy> elements.
- Every <leppl: Policy> element should include mandatory extensions, containing the name of the policy in the CoT. The policy contained in `ref` follows the same nomenclature or is similar to the one that is in the attribute request.
- There can be only one <leppl: DataGroup> element in a single <leppl: Policy> element. A <leppl: DataGroup> should have at least one <pp: Data> element depending upon the type of data contained in it.

5. CONCLUSION AND FUTURE WORKS

Information security and privacy issues are the key concerns in FIM because identity federation requires the exchange of sensitive user information in a highly insecure and open network. In this paper, we have discussed two well-known FIM solutions, Microsoft Passport and Liberty Alliance and addressed privacy issues in FIM through possible business scenarios. In addition, we have proposed a user preference expression language that is crucial to manage users' PII in FIM. We believe our work can be leveraged by the research and industry communities working on privacy issues in identity management.

Our future work will focus on an enhanced privacy attribute management framework within Liberty Alliance which can provide users with a high level of confidence in protecting and controlling their personal data. Developing appropriate information assurance (IA) metrics for FIM is another issue that we intend to work on in the near future. It is generally believed that no single perfect set of IA metrics can be applied to all systems. Thus, we would attempt to investigate IA metrics specifically designed for FIM systems.

Acknowledgements

We are grateful to Todd Inskeep, Sam Phillips, and Larry Hollowood for their support and encouragement in making this work possible. The opinions expressed in this paper are of course our own and should not be taken to represent the views of these individuals.

6. REFERENCES

[1] W3C Note: Simple object access protocol v 1.1. Technical report, Available at www.w3.org, 2000.

[2] W3C note: Web services description language (WSDL) v 1.1. Technical report, Available at www.w3.org/, 2001.

[3] G.-J. Ahn, D. Shin, and S.-P. Hong. Information assurance in federated identity management: Experimentations and issues. In *Proceedings of 5th Web Information Systems Engineering Conference, Lecture Notes in Computer Science (LNCS3306)*, pages 79–90, Brisbane, Australia, November 2004.

[4] D. Chaum. Security without identification: Card computers to make big brother obsolete. *Communications of the ACM*, 28(10):1030–1044, 1985.

[5] L. Cranor, M. Langheinrich, and M. Marchiori. A P3P preference exchange language 1.0 (APPEL1.0). Technical report, Available at www.w3.org, 2002.

[6] L. Cranor, M. Langheinrich, M. Marchiori, M. Presler-Marshall, and J. Reagle. The platform for privacy preferences 1.0 (P3P1.0) specification. Technical report, Available at www.w3.org, 2002.

[7] L. F. Cranor. Agents of choice: Tools that facilitate notice and choice about web site data practices.

[8] H. Damker, U. Pordesch, and M. Reichenbach. Personal reach ability and security management - negotiation of multilateral security. In *Proceedings of Multilateral Security in Communications*, Stuttgart, Germany, 1999.

[9] Federal Trade Commission. Online Profiling - A Report to Congress, part 2. Technical report, 2002.

[10] P. Hallam-Baker and E. Maler. Assertions and protocols for OASIS SAML. Technical report, Available at www.oasis-open.org, 2002.

[11] J. Hegel and M. Singer, editors. *Net Worth: Shaping Market When Customers Make the Rule*. Harvard Business School Press, 1999.

[12] J. Hodges and T. Watson. Liberty architecture overview v 1.2-03. Technical report, Available at www.sourceid.org, 2003.

[13] IBM. Web services security (WSS) specifications 1.0.05. Technical report, Available at www-106.ibm.com, 2002.

[14] Liberty Alliance. ID-WSF security and privacy best practices. Technical report, Available at www.projectliberty.org.

[15] Liberty Alliance. Privacy preference expression languages. White report, Available at www.projectliberty.org.

[16] Mircrosoft Corporations. Microsoft .Net Passport Review Guide. Technical report, Available at www.microsoft.com, 2003.

[17] M. C. Mont, S. Pearson, and P. Bramhall. Towards accountable management of identity and privacy: Sticky policies and enforceable tracing services. Technical report, Available at www.hpl.hp.com, 2003.

[18] P. Shenoy, D. Shin, and G.-J. Ahn. Towards IA-Aware web services for federated identity management. In *Proceedings of IASTED International Conference on Communication, Network, and Information Security*, pages 10–15, New York, USA, December 2003.

[19] D. Shin, G.-J. Ahn, and P. Shenoy. Ensuring information assurance in federated identity management. In *Proc. of the 23rd IEEE International Performance Computing and Communications Conference (IPCCC)*, Phoenix, Arizona, April 2004.

[20] T. Watson. Liberty ID-FF implementation guidlines v 1.2.02. Technical report, Liberty Alliance Project, 2003.