A Delegation Framework for Federated Identity Management

Hidehito Gomi, Makoto Hatakeyama, Shigeru Hosono and Satoru Fujita

NEC Internet Systems Research Laboratories
1753, Shimonumabe, Nakahara-Ku, Kawasaki, Kanagawa 211-8666, JAPAN

{gomi@az, m-hatake@ax, s-hosono@bu, fujita@cd}.jp.nec.com

ABSTRACT

Identity federation is a powerful scheme that links accounts of users maintained distinctly by different business partners. The concept of network identity is a driver for accelerating automation of Web Services on the Internet for users on their behalf while protecting privacy of their personally identifiable information. Although users of Web Services essentially delegate some or all privileges to an entity to perform actions, current identity based systems do not take into sufficient consideration delegation between entities hosting Web Services from a viewpoint of identity and privacy. This paper introduces a delegation model for federated identity management systems and proposes a delegation framework to provide solutions for access control in the context of delegation. The framework has a function of transferring user's privileges across the entities encoded in delegation assertion extending SAML (Security Assertion Markup Language). The framework enables users to manage their own privileges, and service providers to control access of entities based on delegated privileges by the users with assistance of a delegation authority that authorizes delegation of a delegating entity and an authentication authority that authenticates a user and manages user's name identifiers.

Categories and Subject Descriptors

K.6.5 [**Management of Computing and Information System**]: Security and Protection

General Terms

Management, Security

Keywords

Access Control, Delegation, Identity Federation, Privilege, Role

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

1. INTRODUCTION

With the advent of identity management technology, users can perform identity-based Web Services on the Internet securely while protecting their privacy. Identity federation scheme is a viable solution to enhance user's privacy protection and convenience as well as to decentralize user management tasks through the federation of identities among identity and service providers [17]. Since identity federation provides a mechanism to exchange sensitive user information between service providers located in different security domains, users can be provided a variety of identity-based services seamlessly and service providers can control access according to user's attributes. SAML (Security Assertion Markup Language) [12], Liberty ID-FF (Identity Federation Framework) [15] and WS-Federation [4] are emerging technical specifications of distributed federated identity management. These specifications have capabilities to prevent identity tracking and collusion through issuance of an opaque handle for each user.

However, the above specifications do not address identity-based authorization in terms of delegation. SAML and Liberty ID-FF lack delegation capabilities [22] assuming that each provider has a trust relationship in CoT (Circle of Trust). Although WS-Federation on top of WS-Trust [8] supports delegation to issue appropriate security tokens for providers in different security domains, it does not focus on delegation of user privileges or rights constrained by the user and providers in the context of the user identity. Since providers can automatically take actions on behalf of users without involvement of them in Web Services environment, management of privileges in delegation is an inevitable component to control access from providers that are not given appropriate privileges. Few efforts, however, have been put on access control of providers based on user's privileges transferred from the user, because the above specifications are highly dependent on the assumption of trust among providers.

Delegation has been known as an approach that an entity provides all or some of its rights to other entities [3]. Delegation is considered as a useful and effective method to enhance the scalability of a distributed system and decentralize access control tasks. Some papers addressed authorization through delegation in terms of RBAC (Role-Based Access Control) [16], which is a conventional access control mechanism widely accepted. In the RBAC model, permissions are assigned to roles, and not directly users.

Users are assigned appropriate roles according to their tasks, and hence indirectly acquire the permissions associated with those roles. This separation of assignment among users, roles and permissions enables to enhance scalability of access control management by system administrators. Because roles are structured hierarchically, senior roles have junior roles.

There are many variations of RBAC in the literature. Sandhu et al. developed a basic RBAC model, which is called as the RBAC96 [16]. Related to delegation, Sang et al. proposed a role delegation method and protocol [9]. Zang et al. developed a role-based delegation model called RDM200 [24] extending the RBAC96 model. Bhatti et al. developed an XML-based RBPAC policy specification framework for enforcing access control in dynamic XML-based Web Services, which called X-RBAC. Feng et al. proposed a service-oriented role-based access control scheme [7] extending RBAC96 for Web Services. In [6], Chadwick et al. developed a role based access control infrastructure that uses X.509 attribute certificates to store the users' roles.

Delegation has been examined for a wide range of applications. For example, Grid security designs a delegation framework that exchanges proxy certificates based on PKI (Public Key Infrastructure) [2,18,23]. In [23], it is described that proxy certificates allow an entity holding a standard X.509 public key certificate to delegate its privileges to another entity without the assistance of a third party. For another example, in digital rights management, constrained delegation model is introduced to specify how a privilege can be distributed for access control [10]. However, the above previous work does not examine issues about disclosure of privacy information as a result of distribution of certificates between providers in different security domains. In [5], it is described that users' names can be associated with their public keys and the users' organizational structures from certificates.

This paper introduces a delegation framework for Web Services and federated identity management systems on top of RBAC mechanism. The framework has a function of transferring user's privileges across the entities encoded in delegation assertion extending SAML. For users, the framework provides a capability that users can manage privileges related to delegation based on concept of informed consent. For service providers hosting identity based services, the framework provides a capability that they can control access of entities according to the information about authorized privileges by users with assistance of a delegation authority that authorizes delegation of a delegating entity and an authentication authority that authenticates a user and manages user's name identifiers.

The rest of this paper is organized as follows. Section 2 examines a simple scenario requiring role delegation between entities and clarifies several requirements for access control according to delegation information as well as roles. Section 3 introduces a delegation model for an identity based Web Service environments. Section 4 proposes a delegation framework for secure delegation including definition of delegation assertion schema, privilege management through transferring delegation assertions and user identification mechanism by an authority. Section 5 discusses several issues about delegation for this effort and Section 6 addresses an open issue about privacy information sharing among entities. Finally, Section 7 concludes this paper.

2. PRIVACY AND PRIVILEGE ISSUES ON DELEGATION

This section takes a simple delegation scenario exchanging personal information and addresses requirements for identity based authorization about delegation.

Delegation is an approach that an entity provides all or some of its privileges or rights to other entities. This is considered a useful and effective method to enhance the scalability of a distributed system and decentralize access control tasks.

Figure 1 shows a simple scenario requiring a delegation of privileges where entities need to exchange personal information in a distributed environment. In Figure 1, Personal Information Service (PIS) is an Internet-based service that maintains personal information and exposes an interface to obtain the data for authorized entities.

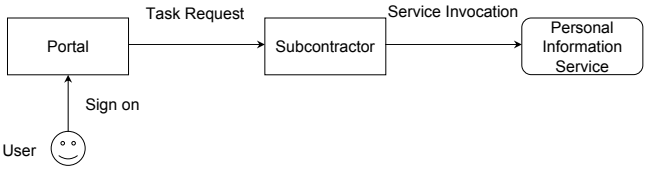

Figure 1: Delegation Scenario of Personal Information Sharing

The followings are some description of the above scenario in order:

1. A user signs on to a site of a portal where the user has an account. Then the user requests some tasks at the site.
2. The portal sends a request of a part of the tasks on behalf of the user to a subcontractor that has a trust relationship with the portal based on business contract between them.
3. The subcontractor might delegate all or a part of its privileges to another subcontractor recursively, if needed. Finally a subcontractor invokes a request for the PIS in terms of the user's privileges.
4. The PIS controls access based on the subcontractor's delegated privileges by the user as well as the roles of the user.
5. The subcontractor obtains personal information from the PIS and returns to the requesting entity the obtained value itself or some added value using the response.
6. The portal finally performs the requested task for the user.

In consideration of the above delegation scenario, the following subsections address several requirements for the delegation.

2.1 Delegation-Based Access Control

In the above scenario, the subcontractor directly makes a request for the PIS on behalf of the user. Even if the user and the portal have an appropriate permission to invoke the PIS, the PIS may deny the request of subcontractor because

the PIS does not have any business relationship with it or because it does not have a permission to invoke the PIS.

In order for the PIS to grant the request from the subcontractor, the PIS needs to be presented with the appropriate information that the portal properly delegates its all or a subset of privileges to the subcontractor. On the contrary, the subcontractor needs to demonstrate valid delegation information to the PIS. In addition, if the subcontractor can do so at runtime when invoking the PIS rather than prior reservation of the information, the system will function more dynamically.

2.2 Privilege Management

In the above scenario, whenever a delegation is occurred between the portal and the subcontractor, some privileges of the user are transferred. From the user's point of view, delegation may cause subcontractors to impersonate taking advantage of the user's privileges improperly. In addition, delegation may cause some risk of giving authorization over a wide range of data and services.

In some cases, a user may hope that she/he gives a permission to do a particular task to another entity only once or may not hope that her/his privileges are transferred to unexpected entities as a result of some steps of delegation. Therefore, the privileges need to be properly managed by each entity. The user should manage subcontractors not to take more actions than the user grants to do for them. The user needs to be able to restrict privileges adding some constraints at each step of delegation rather than to give user's role that has a wide range of privileges.

Moreover, from a viewpoint of privilege management, it is necessary to separate between authorization of accessing the PIS and that of delegating transferring user's privileges. For example, a user can delegate her/his privilege of updating the PIS if she/he has a privilege of doing so and if she/he has a privilege of delegating her/his own privileges to another entity.

2.3 User Identification

To help the PIS control access or to perform delegated task appropriately, a subcontractor may need to identify a user. In this case, some assistance of an authentication authority may be required when opaque name identifiers are used for identity federation as explained in Section 1.

Figure 2 illustrates an example of management of user accounts and corresponding opaque handles for identity federation.

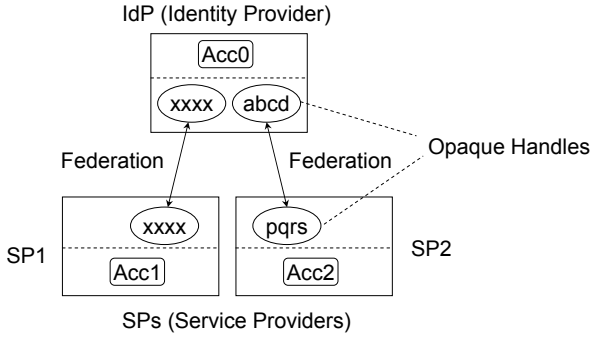

Figure 2: Opaque Handles in Identity Federations

In Figure 2, a user has her/his accounts, *Acc0*, *Acc1* and *Acc2* at IdP(Identity Provider), SP1 (Service Provider 1) and SP2, respectively. The account *Acc0* is federated with *Acc1* and *Acc2*, respectively for simplifying a process of SSO (Single Sign-On). In this case, the identity federation between the IdP and the SP1 is established sharing an opaque handle *xxxx*, not real account names. Because each handle of a federation is private to each pair of the IdP and the SP, this mechanism enables privacy protection from a threat like identity tracking. In order to make collusion among SPs more difficult, different randomized strings can be utilized as shown in a federation between IdP and SP2.

Because of the privacy protection mechanism, service providers cannot exchange identity information with each other in a federated management system. In the above example, if the portal and the subcontractor are SPs and the user has federated accounts at both of the SPs with an authentication authority (IdP), the subcontractor does not have any method to know who the target user is when receiving a task request from the portal.

In this situation, because it is only the IdP that manages the federated accounts and opaque handles, the subcontractor needs some assistance of the IdP to identify who the target user is in order to make a decision to grant the delegation request and perform the delegated task for the user.

2.4 Informed Consent for Delegation

Delegation has a risk that a service subcontractor may impersonate the user without any notice for the user and use the privacy information for improper purpose. In such a case, the user's privacy information may be disclosed in a wide range of environment as unexpected delegations happens. Thus, the user needs to be informed of the delegations and given an opportunity to authorize transfer of the user's privileges from a viewpoint of privacy protection unless the user agrees on the information practices explicitly [13, 14]. In addition, the user needs to know which subcontractor shares the privacy information when the delegation occurs. On the other hand, the subcontractor should contact the user and obtain her/his consent in real time unless it obtains the user's consent prior to delegation.

2.5 Privacy Information Sharing

A subcontractor may obtain the user's privacy information if it is required to carry out some task as a result of delegation. On the other hand, a portal also may obtain the user's privacy information. In some cases the user hopes that her/his privacy information is kept confidential to entities even if she/he consents to the delegation. In such a situation, some existing cryptography technology is available.

When the disclosure of the privacy information is unavoidable, a mechanism to limit information disclosure is needed from a privacy protection point of view. When an entity provides another entity with privacy information, the providing entity needs to confirm that the provided entity agrees upon a privacy policy that it will absolutely keep obtained information confidential and never disclose the information to other entities without any notice of the user. In this case, the provided entity also needs to confirm what kind of information the providing entity will provide in order to avoid managing unnecessary information that may cause some troubles.

3. DELEGATION MODEL

This section presents a delegation model in an identity based Web Service environment. It is noted that this model addresses delegation between entities in different security domains in terms of user's identity. The followings are the basic entities used in this paper.

Principal is a user that has privacy information and stores it to be accessed by authorized entities for a personalized service.

UserAgent is software that interacts with the principal and other entities on behalf of the principal. This software can be controlled by a principal and takes all actions for the principal as she/he instructs.

Resource is an identity-based Web Service that hosts personal information or provides identity related Web Service restricting access based on principal's roles, a delegation assertion presented by an invoking entity and other access control policies.

Authentication Authority (AA) is an authority to authenticate a principal and to issue an authentication assertion indicating that a valid principal signs on.

Delegater is an entity that delegates all of a subset of a principal's privileges to access the resource for the principal.

Delegatee is an entity that is delegated the privileges, and has access to the resource on behalf of the delegater for the principal as a result of the delegation.

Delegation Authority (DA) is an authority to manage roles and privileges of a principal and to issue a delegation assertion stipulating that the authority grants delegation act of transferring the principal's privileges from a delegater to a delegatee.

Additionally, role and privilege are defined as follows:

Role A position or function of an organization that describes the authority and responsibility conferred on a principal assigned to the role.

Privilege A right to carry out a particular permission that is assigned to a role, with some constraints or conditions. A role is associated with multiple privileges.

In this model, it is assumed that a number of entities in different security domains have a AA and a DA in a CoT. Delegation is an act of transferring privileges related to tasks for a principal from a delegater to a delegatee based on business agreement. Moreover, this model focuses on delegation between entities (i.e. machine-to-machine delegation [1,24]), not principal-to-principal delegation. A principal is assigned to her/his own role while a delegater has a special role of *DelegaterRole* that has a permission to act as a delegater and a delegatee has one of *DelegateeRole* that has a permission to act as a delegatee.

Figure 3 illustrates a conceptual model focusing on a relationship among a delegater, a delegatee, a resource and a DA.

The privileges transferred from a delegater (*Entity A* shown in Figure 3) are all or a subset of privileges dependent

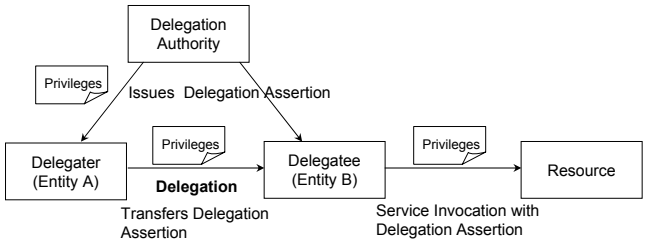

Figure 3: Conceptual Delegation Model

of the roles which are assigned to the delegater originally. The delegater designates transferred privileges to a delegatee (*Entity B* shown in Figure 3) by adding some constraints or conditions such as information about a resource to be invoked and times of invocation in order to avoid unexpected situation caused by the action of the delegatee. These constraints or conditions will be explained in Section 4.1.

In order that the delegation by the delegater is granted by a delegatee, the delegater needs to present to the delegatee an delegation assertion that is an authorization of the delegation issued by a DA. The DA confirms whether the delegation is appropriate in terms of some criteria such as principal's roles and the delegater's role (*DelegaterRole*). Therefore, the delegater needs to obtain the above assertion in prior to the interaction with a delegatee.

When an entity receives a delegation assertion, the entity checks its validity and determines whether it accepts the offer of delegation. If it accepts the offer, it becomes a delegatee and has delegated privileges potentially in addition to its existing permissions. In order to carry out the delegated privileges, the delegatee needs to demonstrate that it is given appropriate privileges by the delegater when it attempts to invoke a resource. Therefore, after the delegatee obtains a delegation assertion specifying the principal's identifiable name for the resource, the delegatee attaches it to the invocation request to the resource.

When the resource receives the service invocation request from the delegatee, it confirms the validity of the above assertion. If the resource discovers that the delegatee has appropriate delegated privileges, the resource makes a decision to grant the access according to the privileges as well as the roles of the principal and the delegatee (*DelegateeRole*). It is noted that delegation assertion is an authorization for delegation, not one for granting access to the resource. Because the resource should have some access control policies, it does not necessarily grant access even if delegation is valid.

Figure 4 illustrates a delegation chain where multiple delegations occur recursively among all the entities explained above.

In Figure 4, there are N entities where N is a natural number ($1 \leq N$). $Entity_0$ is a userAgent, which is the first delegater, not a delegatee. $Entity_N$, which is the last delegatee, not a delegater, invokes the resource directly. $Entity_n$ that accepts a delegation from another entity might invoke the resource as the last delegatee, or delegate all or some delegated privileges to the next entity $Entity_{n+1}$ as a delegater.

$Entity_n$ ($1 \leq n \leq N-1$) acts as both a delegater and a delegatee if $N \geq 2$. After $Entity_{n-1}$, as a delegater, delegates its privileges to $Entity_n$, as a delegatee, then the $Entity_n$, as a delegater, delegates all or some of the del-

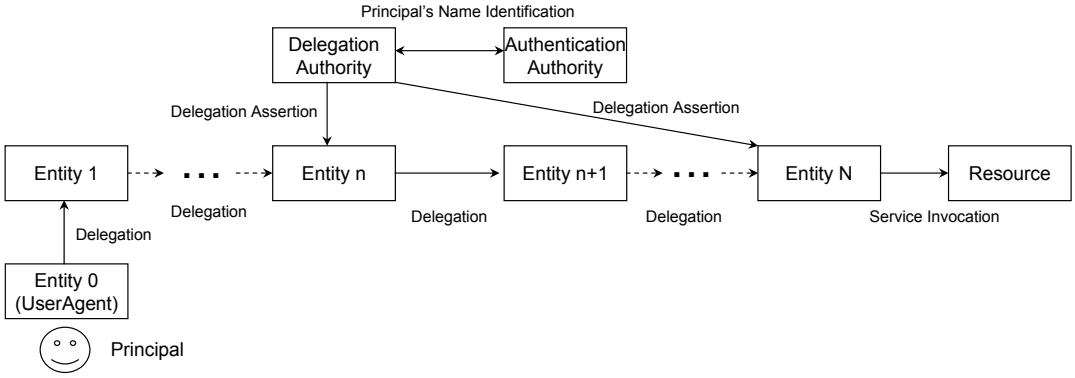

Figure 4: Delegation Chain

egated privileges to $Entity_{n+1}$ which acts as a delegatee ($n \geq 1$). It is assumed that delegation from $Entity_0$ to $Entity_1$ is accepted by the principal in terms of a contract between the principal and $Entity_1$ in this model. Finally, the principal's privileges are transferred from $Entity_0$ to $Entity_N$ in order of N. In this way, as a delegation is occurred between entities, the transition of the principal's privileges are also occurred at the same time.

As stated in Section 2.3, in some cases, a delegater and a delegatee cannot identify a target principal, because they do not share the principal's identity information in federated identity management systems.

This problem can be solved by assistance of an AA. A delegater $Entity_n$ obtains a delegation assertion which is particular for $Entity_{n+1}$ from a DA. When the DA generates the assertion, it collaborates with an AA on resolution of the principal's name identifiers. The AA identifies the target principal and convert her/his opaque handles. The more detail is described in Section 4.2.2.

4. DELEGATION FRAMEWORK

This section proposes a new framework for secure delegation based on the model explained in Section 3 to satisfy requirements presented in Section 2.

4.1 Delegation Assertion

The SAML specification stipulates three different types of assertion statements: *Authentication* (the subject was authenticated by a particular means at a particular time), *Attribute* (the subject is associated with the supplied attributes), and *Authorization Decision* (a request to allow the subject to access the specified resource has been granted or denied). However, schema for delegation is not directly defined by the SAML specifications.

To fulfill the requirements of access control and privilege management in Section 2, delegation assertions exploiting the extensibility of SAML Assertion are defined to encode the information about delegation. The delegation assertion is presented by a delegater to a delegatee to offer the delegation. It is noted that the delegation assertion corresponds to a business contract between the delegater and the delegatee. Based on vocabularies of SAML v2.0, the elements of a delegation assertion are defined as follows:

- *Issuer*: the issuer of the assertion, the DA.
- *Signature*: the digital signature of the delegater. This is utilized to demonstrate the integrity of the assertion.
- *Subject*: principal's information. This contains the principal's name identifier which may be an opaque handle.
- *EncryptedID*: the name identifier of the subject in an encrypted format by means of the AA's key.
- *Conditions*: the valid duration of the assertion.
- *Audience*: the target entity that the assertion is issued to, delegatee.

In addition, SAML *AttributeStatement* is extended to encode delegation information such as user's privileges as her/his attributes. The *AttributeValue* in the extended SAML AttributeStatement stores a *Privilege* element with the following subelements and attributes:

- *Name*: the name of the attribute, indicating "Delegation".
- *Delegatable*: indicates whether the entity that receives this delegation assertion as a delegatee is allowed to delegate the task to another entity as a delegater. If this is "true", recursive delegation is granted.
- *Delegater*: is a delegater which requests to delegate the specified privileges to the designated delegatee.
- *Consent*: is an indicator whether the issuer of the assertion obtains the principal's consent to the delegation. If this is "true", the consent has been obtained.
- *Role*: is a role assigned to the principal and associated with the privileges of the principal.
- *Service*: URI of a resource the delegatee invokes to perform required task as a result of delegation.
- *Description*: the human readable description for the above service.
- *Count*: the maximum number of times the above service is allowed to be invoked.
- *OutputData*: the kind of data which the delegatee obtains as a result of the above service invocation. This corresponds to *output* of *operation* element defined in WSDL [19].

- *InputData*: the kind of data which the delegatee obtains as input required for the service invocation from the delegater. This also corresponds to *input* in WSDL.

- *Encrypted*: indicates to the resource whether the output data is needed to be encrypted by means of the AA's key.

Figure 5 shows an example of a delegation assertion. The assertion contains an *EncryptedID*, therefore the original name identifier of the principal is not disclosed to the delegater.

```
<Assertion ID="abcdefg1000" IssueInstant="2005-07-01T00:20:02Z" Version="2.0">
 <Issuer> http://DelegationAuthority.com</Issuer>
 <ds:Signature> signature by Delegation Authority goes here </ds:Signature>
 <Subject>
    <EncryptedID>
       <xenc:EncryptedData> ... </xenc:EncryptedData>
       <xenc:EncryptedKey>  ... </xenc:EncryptedKey>
    </EncryptedID>
 </Subject>
 <Conditions NotBefore="2005-07-01T00:20:02Z"
      NotOnOrAfter="2005-07-01T00:25:02Z">
  <AudienceRestriction>
    <Audience>http://Delegatee.com</Audience>
  </AudienceRestriction>
 </Conditions>
 <AttributeStatement>
   <Attribute Name="Delegation">
     <AttributeValue>
       <Privilege Delegatable="true">
         <Delegater>http://Delegater.com</Delegater>
         <Consent>true</Consent>
         <Role>Delivery Agent</Role>
         <Service>http://AddressService.com</Service>
         <Description> Address service for principal </Description>
         <Count>1</Count>
         <OutputData Encrypted="true">HomeAddress</OutputData>
       </Privilege>
     </AttributeValue>
   </Attribute>
 </AttributeStatement>
</Assertion>
```

Figure 5: Example of Delegation Assertion

4.2 Delegation Interactions

The delegation interaction mechanism is designed for a delegater and a delegatee to dynamically exchange delegation assertion. Figure 6 depicts interactions including task requests, service invocation and user identification in a simple case of that only one delegation is occurred between a delegater and a delegatee in a series of interactions.

In the following subsections, a delegation protocol, identifier mapping and assertion composition mechanism and task request for delegation are described.

4.2.1 Delegation Protocol

In Figure 6, a delegation protocol corresponds to Step 1 and 4. The delegater sends a request for delegation assertion to the DA (Step 1) and the DA sends the assertion to the delegater (Step 4). This protocol is triggered by a task request from task request for delegation from the userAgent or other entities.

In Step 1, the delegater delivers the principal's authentication assertion or delegation assertion issued for the delegater in SOAP (Simple Object Access Protocol) [21] format to the DA. This assertion is stored in the SOAP header of the message based on a mechanism of WS-Security [11].

The DA makes a decision to authorize the above delegation request from the delegater. At this time, the DA checks the request with its own access control policy and the scope of delegation. In addition, the target principal's consent to delegation is needed in prior to Step 1 in regardless of whether online or off-line.

When receiving an authentication assertion (in Step 3), the DA generates a delegation assertion stipulating that it grants delegation from the delegater. This assertion stores resolved name identifier of the principal for the delegatee and the principal's consent. This is essential for the resource to make a decision to grant access as explained requirements in Section 2. In addition, the assertion is signed by the delegatee's key. Because the name identifier is encrypted by the delegatee's key, it is not disclosed to the delegater.

In Step 4, the DA delivers the generated delegation assertion attached in the response message in SOAP format in the same way of the above request.

The assistance with AA about name identifier resolution of the principal (Step 2, 3, 7 and 8) will be explained in the next section.

4.2.2 Identifier Mapping and Assertion Composition

The previous sections described that there may be a necessity for entities to identify who the principal is and obtain an authentication assertion related to the principal for a particular entity. In order to resolve the above problems, the AA can have a functionality of *identifier mapping* and the DA can have one of *assertion composition*.

As explained user identification issues in Section 2.3, an authentication assertion may include an opaque handle for referring to a principal which is private for each pair of an AA and an entity. Because the AA is the only authority to manage name identifiers of principals, each entity federating with the authority needs some assistance for resolving the principal's name identifier for providing a personalized service for the principal. In addition, the DA is needed to issue a new composite assertion from multiple assertions using a mapped name identifier of the same principal for a different entity. We explain a solution for the above problem according to Step 2, 3, 7 and 8 of Figure 6.

In Step 2 and 7 of Figure 6, the AA receives an authentication assertion or delegation assertion from the DA. Because this assertion includes a particular opaque handle of a target principal for the delegater, the delegatee cannot know who the principal is.

When the AA receives the above query from the DA, it checks the validity of the assertion. If it is a valid assertion, it figures out that the target principal has an account (e.g. *Acc0* shown in Figure 2) from the opaque handle contained in the assertion issued by itself. Then, it also finds that the principal has another opaque handle for the delegatee. The AA issues an authentication assertion that contains the principal's opaque handle to the DA without any disclosure of privacy information (Step 3 and 8).

After the DA receives the authentication assertion from the AA and identifies who the principal is, the DA determines whether it grants the delegation request from the delegater by means of the principal's role and the delegater's role. If the DA grants the delegation, it generates a new composite assertion based on one from the delegater and one from the AA. For example, if the DA receives from an $Entity_n$ a delegation assertion specifying a series of delegations from $Entity_0$ to $Entity_n$ and obtains an authentication assertion for $Entity_{n+1}$ from the AA, the DA generates

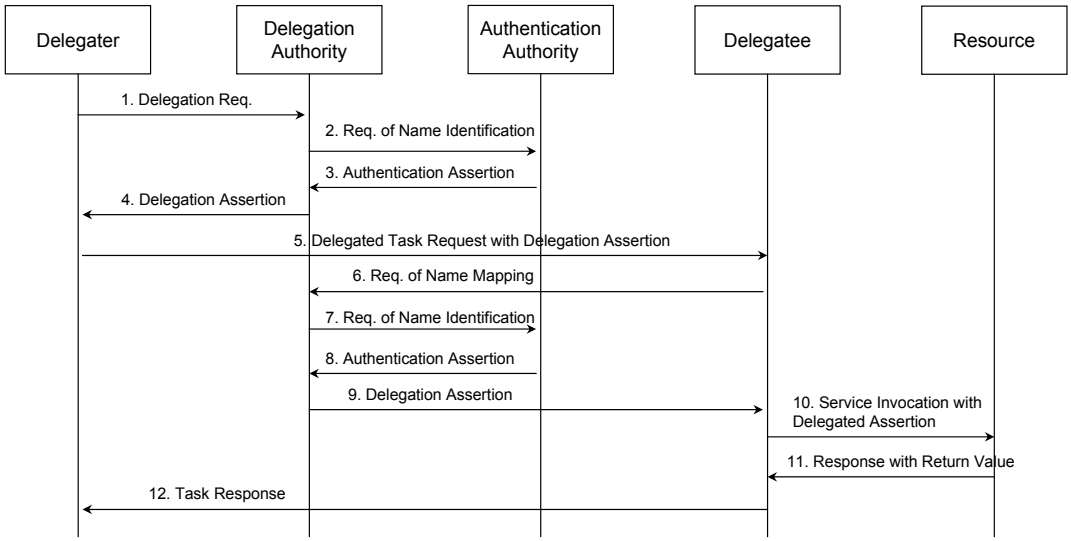

Figure 6: Interactions of Delegation among Entities

a new delegation assertion that grants a series of delegations from $Entity_0$ to $Entity_{n+1}$ containing a name identifier of the target principal for $Entity_{n+1}$.

The identifier mapping functionality is very relevant to Pseudonym Service in WS-Federation and Identity Mapping functionality in Liberty ID-FF in that the authority brokers trust relationships between different entities, and issues the appropriate name identifier unique to the particular entity. On the other hand, the assertion composition functionality is very relevant to STS (Security Token Service) in WS-Trust in that the AA converts an original security token into a required one. Although the delegation framework of this paper does not adopt a specific protocol for the above functionalities, there are several alternatives to implement them.

Although this section presents name identifier resolution scheme, the identifier mapping functionality is not necessary if entities share principal's name identifiers in some environments.

4.2.3 Task Offering and Service Invocation

The task offering interactions correspond to messages shown in Step 5 and 12 in Figure 6. The task request indicates that the request message (Step 5) offers delegation from the delegater to the delegatee, which is the target of the message. The response message (Step 12) has a result of performed task after delegation is completed.

The interactions of service invocation and its response correspond to ones shown in Step 10 and 11. This interaction complies with a protocol for accessing the resource. In Step 10, the delegatee makes a request for service invocation to the resource. In Step 11, the resource makes response after it determines whether it grants the request or not.

The Step 6 and 9 indicate the name identifier mapping request and response, respectively. In Step 6, the delegatee makes a request to the DA for assertion that includes the principal's name identifier particular for the resource. In Step 9, the DA makes a response with a delegation assertion to the delegatee.

The above three interactions are relevant in that the request messages (Step 5, 6 and 10) need to demonstrate a valid assertion particular for the target recipient after the requesting entity presents its own assertion to the DA. For implementation, in the same way as explained in Section 4.2.1, SAML token profile of WS-Security can be utilized to store the assertion in the SOAP header of the above requests.

4.3 Secure Delegation Model

The following is a basic usage of delegation operations showing how protocols work and how messages are exchanged in a delegation chain that includes two delegations among multiple entities. It is considered a situation that principal's account at each entity has already been federated with her/his account at an AA, and that each entity has a trust relationship with each other. In addition, a delegater and a delegatee agree on delegation of privileges for completing their own tasks. Figure 7 shows a message flow based on the above scenario.

1. The UserAgent as an agent of a principal attempts to have access to Provider1 and the Provider1 initiates a session of the principal after receiving an authentication assertion by the AA.

2. The Provider1 explains to the principal that it will delegate some privileges for Provider2 in order to obtain her/his consent to the delegation and redirects to the DA.

3. The UserAgent visits the DA and registers her/his consent about the delegation from the Provider1 to the Provider2. After completion of the consent, the UserAgent returns to the Provider1.

4. In order to act as a delegater, the Provider1 sends to the DA a request of delegation for the Provider2 containing the authentication assertion of the principal.

5. Because the assertion is issued by the AA and includes name identifier of the principal, forwards the assertion to the AA in order to resolve the name identifier of the principal.

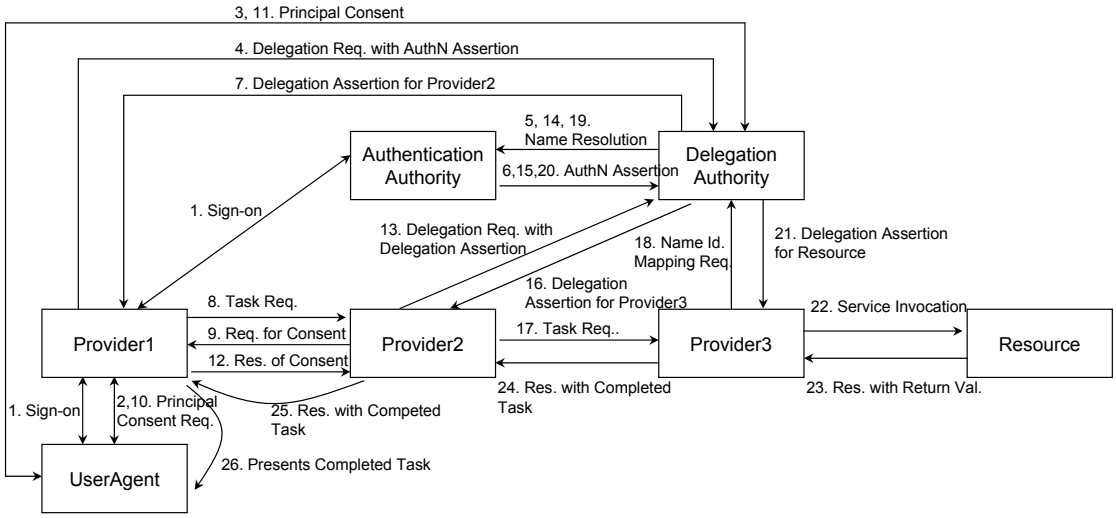

Figure 7: Message Interactions in Secure Delegation Model

6. The AA checks the validity of the received authentication assertion. The AA decrypts the above assertions and identify the target principal.

   The AA generates a new authentication assertion for the Provider2 and provides it with the DA.

7. After the DA determines to grant the requested delegation act by the Provider1, the DA composes a delegation assertion for the Provider2.

   The assertion stipulates that the Provider1 transfers some privileges of the principal to the Provider2 and contains an encrypted name identifier by the Provider2's key. Then the DA returns the generated delegation assertion to the Provider1.

8. The Provider1 as a delegater sends a task request with the above delegation assertion to the Provider2.

9. The Provider2 receives a task delegation request from the Provider1 and makes a decision to accept the request as a delegatee.

   In the process of executing the required task, the Provider2 finds that it needs to delegate some privileges of the principal to the Provider3 which deals with the principal's privacy information.

   The provider2 asks the Provider1 to obtain the principal's consent about delegation to the Provider3.

10. On behalf of the Provider2, the Provider1 explains to the principal that the Provider2 needs to delegate some privileges to the Provider3 for accomplishing its task, and redirects to the DA in the same way as Step 3.

11. The UserAgent visits the DA and registers her/his consent about the delegation from the Provider2 to the Provider3.

12. The Provider1 informs the Provider2 that the consent about the above delegation is obtained.

13. The Provider2 as a delegater sends to the DA a delegation request for the Provider2 with the delegation assertion about the principal obtained from the Provider1.

14. In the same way as Step 5, the DA checks the validity of the received delegation assertion. Because the assertion includes name identifier of the principal, sends a query of the name identification to the AA.

15. In the same way as Step 6, the AA generates a new authentication assertion for the Provider3 and provides it with the DA. After the DA determines to grant the requested delegation act by the Provider2, the DA composes a delegation assertion for the Provider3.

16. The DA returns the generated delegation assertion to the Provider2. This assertion includes the two types of delegations which is delegation from the Provider2 to the Provider3 and the former one from the Provider1 to the Provider2.

17. The Provider2 as a delegater sends a task request with the above delegation assertion to the Provider3.

18. The Provider3 sends a name identifier request to the DA to present a delegation assertion particular for the resource by attaching the above assertion obtained from the Provider2.

19. In the same way as Step 5 and 14, the DA sends a query of the name identification to the AA.

20. In the same way as Step 6 and 15, the AA generates a new authentication assertion for the Resource and provides it with the DA.

21. The DA generates a new delegation assertion based on the above assertion obtained in Step 18. Then the DA makes a response with a new delegation assertion for the Resource to the Provider3.

22. The Provider3 as the last delegatee attempts to invoke the Resource. This invocation request contains the above delegation assertion certifying that privileges are properly delegated from from the Provider1 to the Provider3.

23. The Resource makes a decision to grant access from the Provider3 according to the roles of the principal and delegated privileges of the Provider3. The Resource sends a response message with some return value to the Provider2 as a result of service invocation to the Provider3.

24. The Provider2 responds the value from the Provider1.

25. The Provider1 receives the value from the Provider2.

26. Finally, the Provider1 completes task through the above two delegations of privileges.

The above a series of interactions seem complicated because the scenario includes dynamic acquisition of delegation assertions and user consents. However, some optimization is possible for accomplishing better performance in actual implementation. For example, once the providers obtain a delegation assertion, they do not need to request a new one from the DA as long as the assertion is valid. In addition, message interactions for principal's consent can be omitted if the principal specifies her/his privacy policy about privacy information practices to providers in prior to delegation operations.

5. DISCUSSION

This paper introduces a notion of delegation for access control in Web Services systems from the following viewpoints: 1) The providers essentially have all or some privileges or rights transferred from the principal, because service providers always take actions for Web Services on behalf of the principal. 2) Principal's role is not sufficient for authorization in delegation environment, because providers hosting Web Services do not have a mechanism proving that the principal properly delegates her/his own privileges to the providers invoking the services, and because they should automate service operations if the principal allows them to do so. The adoption of transferring privileges is an essential notion to automate Web Services based on constrained privileges of the principal while she/he is off-line.

In the delegation model of this paper, principal's privileges are transferred from a delegater to delegatee in accordance with the order of delegation assertion flow. In the delegation framework, every transferred privilege is expressed in a delegation assertion and it is signed by the DA which is an issuer of the assertion. In addition, Because the DA has principal's consent to the delegation and the information is stored in the delegation assertion, the principal's privileges are properly transferred from a principal to the resource in delegation chain that consists of multiple delegations among entities. Since this privilege management of the framework assures that the privileges a principal grants to delegate within a subset of her/his own privileges are transferred, service providers cannot take more actions by impersonating of a principal than she/he expected.

In federated identity management systems, a delegater and a delegatee often cannot identify the principal, because the name identifiers of the principal are not shared among them. For this problems, identifier mapping and assertion composition capabilities of the AA and DA can identify the principal and issue a composite assertion for a particular entity while protecting principal's privacy. Therefore, the DA acts as a broker for establishing a delegation chain.

Based on the above privilege management and user identification, the resource can control access in terms of delegated privileges from the last delegatee. As explained in Section 4.3, assertions can be securely exchanged while protecting principal's privacy.

In this delegation model, a single AA in a CoT is assumed. If a CoT has multiple AAs and entities rely on different AAs, name resolution scheme requires interaction between AAs. When an AA receives a name mapping request and cannot resolve the principal's name identifier, the AA can make a request to other AAs to resolve it based on a business contract between the AAs. This scheme requires AAs to identify which other AAs can identify the target principal that is not authenticated.

6. FUTURE WORK

This paper does not sufficiently address privacy information sharing between entities when delegation occurs as explained in Section 2.5. An entity may obtain user's privacy information if it is required for performing required task as a result of delegation. In some cases, when the disclosure of the privacy information is required, a mechanism to limit the disclosure is needed from a privacy protection point of view. When an entity provides another entity with privacy information, the providing entity needs to confirm that the provided entity agrees upon a privacy policy such as the purpose of the information to avoid illegal disclosure of the information. On the other hand, the provided entity needs to confirm that the providing entity has an appropriate privacy policy about the information to be given in order to avoid maintaining unnecessary information that may cause some troubles. Therefore, in order to solve the above problems, a policy negotiation protocol is needed for entities to exchange each privacy policy in the context of user's privacy information. The authors intend to investigate a negotiation protocol extending a SAML protocol with P3P (Platform for Privacy Preferences) policy schema [20] in the future.

7. CONCLUSION

This paper has introduced a delegation framework for transferring of user's privileges across entities encoded in delegation assertion extending SAML assertion schema in federated identity management systems. It has been shown that in a limited delegation environment, the delegation framework provides capabilities that delegation of user privileges is accomplished properly. In this framework, users can have opportunity to manage their own privileges and to give their consent to privacy information sharing between entities as a result of delegation operations. Service providers hosting identity-based Web Services can control access based on privileges delegated by the user and properly transferred by entities in the delegation chain as well as the user's role. When providers cannot identity a target user because of opaque handles in identity federation, they can obtain a delegation assertion containing the user's name private for them without any privacy information disclosure with assistance of an authentication authority that manages the user's name identifier and an delegation authority that authorizes delegation of a delegating entity. Future work includes investigation of privacy policy negotiation protocol for exchanging privacy information sharing between entities that will be on top of the proposed delegation framework.

8. ACKNOWLEDGEMENTS

The authors wish to thank this paper's anonymous reviewers for their comments, which have helped us to improve the paper.

9. REFERENCES

[1] M. Abadi, M. Burrows, B. Lampson, and G. Plotkin. "A Calculus for Access Control in Distributed Systems". *ACM Transactions on Programming Languages and Systems*, 15(4):706–734, 1993.

[2] M. Ahsant, J. Basney, and O. Mulmo. "Grid Delegation Protocol". In *Proceedings of the Workshop on Grid Security Practice and Experience*, July 2004.

[3] O. Bandmann, M. Dam, and B. Firozabadi. "Constrained Delegation". In *Proceedings of the 2002 IEEE Symposium on Security and Privacy (S&P '02)*, pages 131–140, 2002.

[4] BEA, IBM, Microsoft, RSA Security, and VeriSign. "Web Services Federation Language (WS-Federation)". Version 1.0, July 2003.

[5] O. Canovas and A. Gomez. "Delegation in Distributed Systems: Challenges and Open Issues". In *Proceedings of IEEE Internatinal Workshop on Database and Expert Systems Applications (DEXA '03)*, September 2003.

[6] D. Chadwick and A. Otenko. "The PERMIS X.509 Role Based Privilege Management Infrastructure". In *Proceedings of the seventh ACM Symposium on Access Control Models and Technologies (SACMAT '02)*, pages 135–140, 2002.

[7] X. Feng, L. Guoyuan, H. Hao, and X. Li. "Role-Based Access Control System for Web Services". In *Proceedings of the fourth International Conference on Computer and Information Technology (CIT '04)*, pages 357–362, 2004.

[8] IBM, Microsoft, Actional, BEA, Computer Associates, Layer 7, Oblix, OpenNetwork, Ping Identity, Reactivity, and Verisign. "Web Services Trust Language (WS-Trust)", February 2005.

[9] S. Na and S. Cheon. "Role Delegation in Role-Based Access Control". In *Proceedings of the fifth ACM Workshop on Role-Based Access Control*, pages 39–44, 2000.

[10] G. Navarro, B. Fironzabadi, E. Rissanen, and J. Borrell. "Constrained Delegation in XML-based Access Control and Digital Rights Management Standards". In *Proceedings of Communication, Network, and Information Security (CNIS '03)*, 2003.

[11] OASIS. "Web Services Security: SOAP Message Security 1.0". OASIS Standard, March 2004.

[12] OASIS. "Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) V2.0". OASIS Standard, March 2005.

[13] OECD. "OECD Guidelines on the Protection of Privacy and Transborder Flows of Personal Data", 2004. `http://www.oecd.org/document/18/0,2340,en_2649_201185_1815186_1_1_1_1,00.html`.

[14] OPA. "Guidelines for Online Privacy Policies". `http://www.privacyalliance.org/resources/ppguidelines.shtml`.

[15] Liberty Alliance Project. "Liberty ID-FF Protocols and Schema Specification". Version 1.2, November 2003. `http://www.projectliberty.org/specs`.

[16] R. Sandhu, E. Coyne, H. Feinstein, and C. Youman. "Role-Based Access Control Models". *IEEE Computer*, 29(2):38–47, February 1996.

[17] D. Shin, G. Ahn, and P. Shenoy. "Ensuring Information Assurance in Federated Identity Management". In *Proceedings of IEEE Internatinal Performance Computing and Communications Conference (IPCCC '04)*, April 2004.

[18] The Globus Security Team. "Globus Toolkit Version 4 Grid Security Infrastructure: A Standards Perspective". Version 2, December 2004.

[19] W3C. "Web Services Description Language (WSDL) 1.1". W3C Note, March 2001. `http://www.w3.org/TR/wsdl`.

[20] W3C. "The Platform for Privacy Preferences 1.0 (P3P1.0) Specification". W3C Recommendation, April 2002. `http://www.w3.org/TR/P3P/`.

[21] W3C. "SOAP Version 1.2 Part 0: Primer". W3C Recommendation, June 2003. `http://www.w3.org/TR/soap12-part0/`.

[22] J. Wang, D. Vecchio, and M. Humphrey. "Extending the Security Assertion Markup Language to Support Delegation for Web Services and Grid Services". *IEEE International Conference on Web Services (ICWS '05)*, July 2005.

[23] V. Welch, I. Faster, C. Kesselman, O. Mulmo, L. Pearlman, S. Tuecke, J. Gawor, S. Meder, and F. Siebenlist. "X.509 Proxy Certificates for Dynamic Delegation". *3rd Annual PKI R&D Workshop*, 2004.

[24] L. Zhang, G. Ahn, and B. Chu. "A Rule-Based Framework for Role-Based Delegation". In *Proceedings of the sixth ACM Symposium on Access Control Models and Technologies*, pages 153–162, May 2001.