Commitment Issues in Delegation Process

Quan Pham, Jason Reid, Adrian McCullagh and Ed Dawson

Information Security Institute
Queensland University of Technology
126 Margaret Street, Brisbane, QLD 4001, Australia

{q.pham, j.reid, a.mccullagh, e.dawson}@isi.qut.edu.au

Abstract

Delegation is a powerful mechanism to provide flexible and dynamic access control decisions. Delegation is particularly useful in federated environments where multiple systems, with their own security autonomy, are connected under one common federation. Although many delegation schemes have been studied, current models do not seriously take into account the issue of delegation commitment of the involved parties. In order to address this issue, this paper introduces a new mechanism to help parties involved in the delegation process to express commitment constraints, perform the commitments and track the committed actions. This mechanism looks at two different aspects: pre-delegation commitment and post-delegation commitment. In pre-delegation commitment, this mechanism enables the involved parties to express the delegation constraints and address those constraints. The post-delegation commitment phase enables those parties to inform the delegator and service providers how the commitments are conducted. This mechanism utilises a modified SAML assertion structure to support the proposed delegation and constraint approach.

Keywords: Delegation, Commitment, SAML, Access Control, Federated Systems.

1 Introduction

In federated information processing environments which contain multiple component systems and associated users, any entity may be constrained on how it acts upon other entities. In general, one entity has a set of privileges for some services that it can access. For traditional systems, a static set of privileges for each user would be adequate. However, in federated systems this is insufficient, especially in circumstances where it is difficult to anticipate in advance the set of privileges a user will need. In addition to this, there are problems with inconsistency in authentication and authorisation decisions between the member systems and/or between the federation and a local authority. Overcoming these issues while still being able to maintain the autonomy of member systems is a big challenge as it is difficult to make the whole federation understand a consistent set of identities and access control policies. In this context, delegation appears to be a potential solution as it provides a promising means for maintaining consistency of user attributes and authorisation states. This makes delegation, especially user to user delegation (ad hoc delegation) (Section 2), particularly useful in federated environments. Although many delegation schemes have been studied (Gomi et al. 2005; Madsen et al. 2005; Wu et al. 2005; Bhatti et al. 2006; Crampton and Khambhammettu 2006; Fragoso-Rodriguez et al. 2006; Joshi and Bertino 2006; Shen 2006; Wang and Osborn 2006; Zhang et al. 2006), current models do not seriously take into account the issue of delegation commitment of the involved parties. This paper attempts to address this problem.

This paper discusses the concept of delegation commitment and proposes a scheme to monitor and keep track of the commitments of the involved parties when requesting a delegation assertion for a particular task. The proposed mechanism helps the parties involved in the delegation process to express commitment constraints, honour the commitments and track the committed actions. The mechanism looks at two different aspects: pre-delegation commitment and post-delegation commitment. In pre-delegation commitment, the mechanism enables the involved parties to partially express the delegation conditions and constraints. The post-delegation commitment enables those parties to inform the delegator and service providers how the pre-delegation commitments, which include some delegation conditions and constraints, have been conducted. Revocation of delegation and associated issues are out of scope of this paper and are considered as future work.

Section 2 reviews preliminary concepts with respect to delegation and commitment. Section 3 briefly discusses related work. The remainder of this paper will concentrate on the substantial issue of delegation commitment of the involved parties. Section 4 and Section 5 look at the issues of delegation commitment and the use of SAML assertions to express delegation commitment. Section 6 and Section 7 discuss some current unsolved issues and conclude the paper with some potential avenues for future work.

2 Preliminaries

Delegation

Delegation is a mechanism for assigning privileges as well as other attributes to users. The user who performs a delegation is referred to as a "*delegator*" and the user who receives a delegation is referred to as a "*delegatee*". A privilege attribute will be "*delegatable*" if it can be

successfully granted or transferred from one user to another (Sandhu 1998; Sandhu 2005; Crampton and Khambhammettu 2006).

Schaad (Schaad 2003) and Crampton and Khambhammettu (Crampton and Khambhammettu 2006) describe clearly the fundamental concepts of delegation. From the administrative perspective, there are two types of delegation (Crampton and Khambhammettu 2006): *administration (administrative delegation)* and *user delegation* (*ad hoc delegation*). Administration is the basic form of delegation in which a security administrator or authority assigns privilege attributes to users. The ability to assign does not necessarily mean that administrator possesses the capability to use the assigned privileges or attributes. Depending on the access control model, this process may not require great administrative effort and can, itself, be subject to constraints on what can be assigned and to whom. However, this process only meets basic and static requirements of access control. It fails to provide the degree of dynamic flexibility required to support access control decisions in federated systems. On the other hand, user delegation occurs between two or more users who do not necessarily possess any special administrative authority. In this form, rights are not assigned by the administrator or authority but are granted or transferred from one user to another. Specifically, user delegation allows a user to assign the whole or a subset of his/her rights to other users. A user delegation operation requires that the user performing the delegation must possess the capability to use the delegated attributes. It is widely accepted that an administrative delegation operation is often long-lived and more durable (permanent) than a user delegation operation that is short-lived (temporary) and intended for a specific purpose (Schaad 2003; Crampton and Khambhammettu 2006). Some authors argue that both parties in the delegation process may need to meet certain conditions and comply with certain commitments in order to make the delegation happen successfully (Castelfranchi 2004).

From the operational transaction, *direct delegation* is defined as the delegation in which the delegator directly sends the delegation assertion to the delegatee. In contrast, *indirect delegation or multi-step delegation* is performed with the involvement of one or many intermediate parties which can forward the delegation assertion from the delegator to the delegatee.

Delegation may also be classified into two categories: *grant delegation* and *transfer delegation* (Barka and Sandhu 2000; Crampton and Khambhammettu 2006). In grant delegation, a successful delegation operation allows a delegated attribute to be available to both the delegator and delegatee. So after a grant delegation, both delegatee and delegator will share a subset of attributes in common. However, in transfer delegation, following a successful delegation operation, the ability to use delegated attributes is transferred to the delegatee and the delegated attributes are no longer available to the delegator. The grant delegation model makes the availability of attributes increase monotonically with delegations (Crampton and Khambhammettu 2006). The grant delegation model is primarily concerned with allowing the delegatee to use the delegated attributes. On the other hand, in transfer delegation, besides allowing the delegatee to use the delegated attributes, the mechanism must be able to prevent the use of the delegated attributes by the delegator. This requirement makes transfer delegation policy enforcement more difficult (Aura 1999; Schaad 2003; Crampton and Khambhammettu 2006). While some business processes may require grant delegations, it is often desirable that sensitive access rights may not be available to a large number of users (at any given time). Such requirements are usually expressed as cardinality constraints in an access control policy (Sandhu 1990). Transfer delegation policies prove to be more useful when an access control policy specifies cardinality limits on the availability of access rights between users.

Commitment in the Delegation Process

In any delegation process, the delegation transaction is approved or agreed by both parties *only after* both can reach an agreement about the duties or responsibilities of the involved parties. This forms the *delegation commitment* of the involved parties which can be understood as the course of action about what they have to do before and after the delegation takes place to actually complete the delegation process. This forms an important aspect of delegation which is not adequately addressed by many delegation models. Consider the following scenario in which user B on system S(2) wants to access resource R(1) on system S(1) for which B does not have the necessary privileges or attributes. User B (the delegatee) requests user A (the delegator) on system S(1) to delegate the necessary credential. B can stipulate that they only require access to R1 three times for a period of one day. The commitment of the delegatee in this scenario is composed by the following factors: *access to R1, only three times* and *only valid for a period of one day*. The delegator can agree to perform a grant delegation. Then the commitment of delegator in this scenario is *grant delegation for three times and for one day*. The delegation commitment can include some conditions and constraints on the delegation process notably duration and service invocation times. However, commitment is not a condition or constraint with respect to roles/privileges and their conflict resolution; systems constraints such as workload, etc. (Atluri and Warner 2005). Part of the commitment is the trusted responsibility, for example, activities which the delegator believes that the delegatee will perform to effectively comply with the delegation. An example for this type of activity is that after each service invocation, the delegatee has to reduce the allowed number of service invocation by one. This is the delegatee's commitment as the delegator can not monitor how the delegatee controls the times of usage of the delegation assertion.

Optimistic delegation

At the time a delegator receives a delegation request, it does not necessarily know in advance whether a particular set of delegated privileges will be useable by the delegatee, since it may not have a complete understanding of the current security context of the delegatee, the current set of roles and privileges of the

delegatee, the policies of the delegatee's systems, etc. To avoid making a delegation that will not be honoured, the delegator could contact the relevant Authorisation Authorities to ask "if I delegate these privileges to user X from domain Y, will they be honoured?" But asking this question in advance for each delegation transaction is clearly inefficient as the authorisation authority will then need to evaluate the request twice - once for the pre-approval and once for the actual execution by the delegatee.

Therefore, the delegator agrees to conduct the delegation transaction, it does so based on its best knowledge of the constraints and conditions for the delegation transaction, for example, the policies of its systems, the attributes of the privilege attribute itself, etc. It does not guarantee that the delegatee will be able to successfully use this attribute privilege for service invocation. Naturally, this is the best effort delegation of the delegator or in other words, an optimistic delegation.

Thus, one of the advantages of our proposed scheme is that, via the pre- and post-delegation commitment of the involved parties, it supports optimistic delegation wherein the delegator simply assumes that the delegation will succeed - it does not ask the authorisation authority in advance to confirm that the delegation will be effective. If the delegation fails, the delegation commitment framework provides a way of recording, identifying, reporting and correcting the problem. Therefore, it is safe to say that optimistic delegation is more efficient as it does not require pre-approval of the Authorisation Authority.

3 Related Works

Recently, delegation issues have attracted a considerable effort from the research community. Most, if not all, research was conducted based on RBAC. Most of the proposals that study delegation in the context of role-based models employ grant delegation (Barka and Sandhu 2000; Na and Cheon 2000; Zhang et al. 2001; Zhang et al. 2003; Zhang et al. 2003; Tamassia et al. 2004; Wainer and Kumar 2005). Temporal transfer delegation with role hierarchies is also addressed in some papers (Crampton 2003; Crampton and Khambhammettu 2006; Joshi and Bertino 2006).

In 2000, Na and Cheon proposed a basic role delegation method and protocol which can handle simple delegation operations (Na and Cheon 2000). Similarly, Barka and Sandhu also presented a framework for their first notion of delegation called RBDM0 (Barka and Sandhu 2000). This role-based delegation model is based on RBAC96, and provides support for user delegation. RBDM0 is a total delegation model which means the delegator delegates all the permissions, particularly permissions in a role to a delegatee by user to role assignment. Then the original user of the role assigns a delegatee to the role. Revocation is done by a timeout mechanism and by grant-independent revocation. The authors also extend the model to support partial delegation and two-step delegation by defining two different types of permissions in a role: delegatable permissions and non-delegatable permissions. The delegatee can only have delegatable permissions. In their second model – RBDM1, Barka and Sandhu added role hierarchies and source dependent cascading revocation (Barka and Sandhu 2004), which is done automatically along the delegation chain (Wang and Osborn 2006).

In another effort, Zhang et al. extended the RBDM0 to construct a new model called RDM2000 (Zhang et al. 2001). The RDM2000 model supports hierarchical roles and multi-step delegation, which are not supported in the original RBDM0 model. They also specified a rule-based language to describe the policies of RDM2000. Revocation is separated into two categories: revocation by delegation duration restriction which can be considered as a timeout mechanism and explicit user revocation. Recently, Ahn et al. published some papers for access control in a collaborative environment such as health care or law enforcement using this delegation model (Zhang et al. 2002; Zhang et al. 2003; Tolone et al. 2005). The rule-based approach is very powerful for constraint enforcement (Yin et al. 2004; Wang and Osborn 2006). However it only considers the regular user to user delegation.

Zhang, Oh and Sandhu presented a new permission-based delegation model (PBDM) in 2003 (Zhang et al. 2003). This model fully supports partial and multi-step delegation. This model is, later, extended and presented in three variants called PBDM0, PBDM1 and PBDM2. As RBDM0 and RDM2000, all variants are based on the RBAC96 model and use user to role assignment to perform the delegation operations. PBDM2 is designed to support role to role and permission delegation (Zhang et al. 2003). The PBDM family can support multi-step delegation, but they neither support constraints in delegation, nor delegation in distributed environments (Crampton and Khambhammettu 2006).

In 2006, based on the RBDM and PBDM family, Crampton and Khambhammettu proposed an extended scheme which incorporated many features of both families (Crampton and Khambhammettu 2006). This model is argued as more conservative, safer, more fine-grained and more manageable than the two predecessors. This model argued that using only relations such as *can-delegate* and *can-receive* for controlling delegations may not be efficient for implicitly handling updates to various RBAC relations and proposed an alternative way of controlling delegations using the concept of administrative scope (Crampton and Khambhammettu 2006). The administrative scope model is dynamic and implicitly handles any updates to RBAC relations, in particular the role hierarchy relation. In this model, both grant delegation and transfer delegation are supported. In the domain of DRM research, Petkovic and Koster also implemented a framework to grant and transfer user privilege based on constraint delegation (Petkovic and Koster 2005).

In 2005, in an effort to address constraint issues in delegation, Atluri and Warner studied delegation in workflow management and introduced a conditional delegation model (Atluri and Warner 2005). This model introduces several types of constraint (conditions) for the

delegation such as intervals, workload limitations, task attributes, etc. In general, the constraints are also divided into four different types: authorisation constraints, delegation constraints, task dependency requirements and role activation constraints. In the delegation context, there are three kinds of conditions for delegation (Wang and Osborn 2006):

- A temporal delegation condition is a condition on the delegation start time and/or the time interval of the delegation.
- A workload delegation condition is a condition of a specific workload level.
- Value delegation conditions control a delegation by attributes.

Several rules are defined to support conditional delegation. Some constraints can be verified before the execution of workflow and some must be verified and enforced during workflow execution. The authors call this verification delegation consistency; the former is called static consistency and the later is called dynamic consistency (Wang and Osborn 2006).

In a similar approach, Wainer and Kumar considered different constraints that can be applied to RBAC delegation and presented a more fine-grained user delegation model (Wainer and Kumar 2005). Unlike Atluri and Warner's approach, this model distinguishes two types of access rights: object rights and delegation rights with constraints. However, similarly to other models, it uses user to role assignments to perform delegation. An interesting thing about this model is its revocation method. The revocation is source dependent cascading revocation (Wang and Osborn 2006). This method was expressed by the authors as "revocation with downgrade" in which the model tests and updates the depth for cascading revocation. An extension of this model with time-restricted delegation which uses timeout to revoke the delegation is also proposed (Wainer and Kumar 2005). In 2006, Wang and Osborn proposed a hybrid approach. Their model used a combination of user to group assignment to perform partial and role to role delegation, while it employed user-role assignment to do total delegation (Wang and Osborn 2006). Wang and Osborn's model tried to minimize impact on the role hierarchy and overcomes the shortcomings of the user to role assignment approach (Wang et al. 2006).

The most interesting approach is described in the model of Yin et al. (Yin et al. 2004). This model has potential to address some issues of delegation in federated systems. Yin et al. have discussed a decentralized delegation model with constraints resolution, management domain, etc. for distributed systems (Yin et al. 2004). The model divides access control in large, distributed systems into two levels: the management level and the request level. At the management level, the system consists of "multi-centric" management which has its own authorisation management domain (Yin et al. 2004). At the request level, normal users make a cascaded request which requires more than one service to respond to the request. The model classifies delegation into two levels corresponding to the levels mentioned above. At the authorisation management level, the delegation is called delegation of authority. At the request level, the delegation is called delegation of capability which can be effectively considered as grant/transfer access control privilege. As mentioned above, this model, to some extent, can address some similar issues of delegation in federated system. However, it does not fully solve the issues of delegation commitment, inconsistency of privilege attributes or access control policies in delegation operation across component systems in the federation or across federations. So, these issues remain an open research problem.

It is obvious that delegation is a significant problem in managing authentication and authorisation. Except the models of Yin et al. and Wang and Osborn, so far, it is safe to say that most delegation models are centralised and based on user to role assignment. More importantly, with the exception of some delegation models such as Atluri and Warner (Atluri and Warner 2005) or Wainer and Kumar (Wainer and Kumar 2005) which pay some attention to delegation constraints such as time intervals, workload, etc., few models address the issue of commitments, especially tracking the commitment in the delegation process.

4 Delegation and Commitment of Involved Parties

This section explains the conceptual issue of commitment in the delegation process and the role of the involved parties.

In general, the followings are the basic entities involved in the delegation process.

- *Delegator* is the entity which has necessary privilege attributes and is authorised to delegate those privilege attributes to the delegatee (the receiver of delegation assertion).
- *Delegatee* is an entity that is delegated the necessary privilege attributes to access resources controlled by Service Provider on behalf of the delegator.
- *Authorisation Authority* is the entity which is able to verify authorisation decision, regarding access requests from users.
- *Service Provider* is an entity which controls and provides a service to users. The Service Provider provides services based on the authorisation decision of Authorisation Authority. Service Provider and Authorisation Authority can be one entity.

In this paper, it is assumed that each Service Provider and delegators/delegatees in different security domains have an Authorisation Authority. In addition, due to the autonomous nature of federated systems, it is not uncommon that the delegatee will come from a different security domain and/or the delegator may not have had prior contacts with the delegatee. So from the trust perspective, it is reasonable to assume that the delegator trusts the Authorisation Authority and Service Provider more than the delegatee.

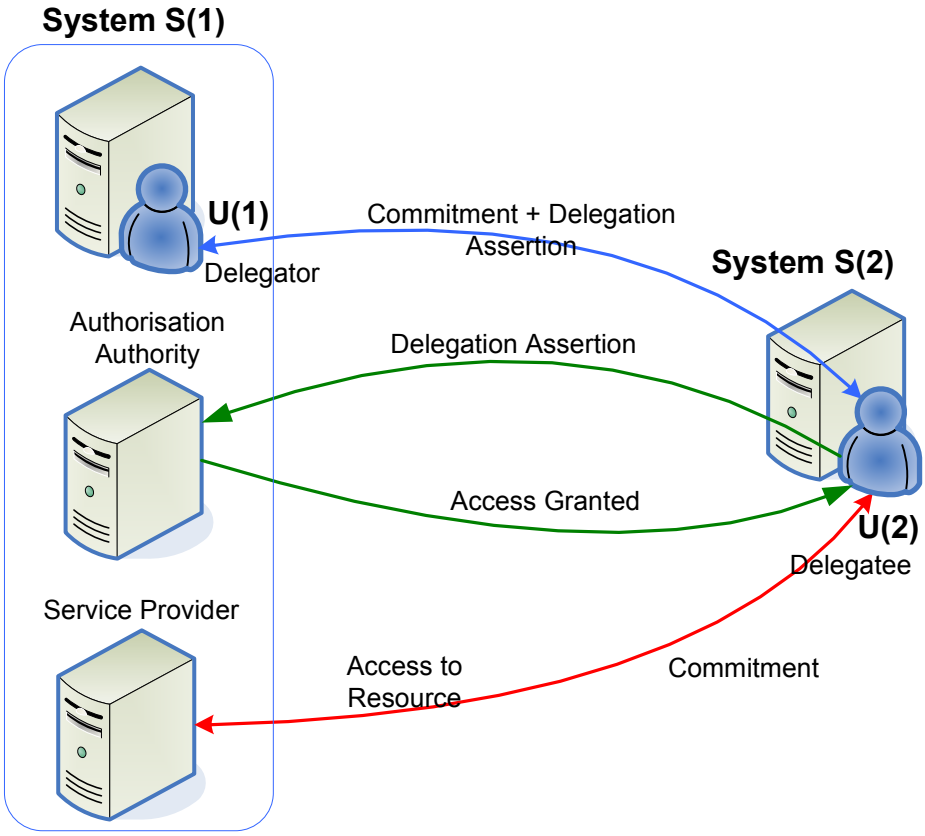

Figure 1: An example of direct delegation transaction

Figure 1 depicts a typical direct delegation with the involvement of two entities in which an initial delegator on system S(1) subsequently grants some necessary privilege attributes to the delegatee on system S(2) to allow access to the necessary service.

In any user delegation model, the delegated privilege attributes are all or a subset of delegator's privilege attributes. The delegator manages to transfer or grant these privilege attributes to the delegatee with some constraints or conditions such as information about the Service Provider, the service to be invoked, times and duration of invocation, etc. As discussed in Section 2, in our model, some of those conditions and constraints form the commitment of the delegator, delegatee and other involved parties such as Authorisation Authority and Service Provider. In our model, making agreement on the commitment and tracking the commitment are the first and the last step of the delegation process.

Pre-delegation Commitment

The pre-delegation commitment phase focuses on the constraints and conditions of the involved parties. For the delegation to happen, the constraints and conditions must be expressed clearly and exchanged to both parties involved. As the delegation process in the paper focuses on user delegation, the delegator will have the authority upon the pre-delegation commitment negotiation with the delegatee.

In the direct delegation process, from the delegator's perspective, the delegator will have to focus on the following tasks:

- Checking the validity of delegation request and making decisions regarding conditions and commitment such as duration, access times, etc.
- Forming delegator's conditions and commitment
- In some cases, the delegator has to negotiate the conditions and commitment with the relevant Service Provider and Authorisation Authority to notify or verify with these parties about the commitment of the delegatee.

From the delegatee's perspective, if the delegation request is accepted by the delegator, the delegatee also has to complete the following duties to make the delegation progress:

- Checking the validity of delegator's assertion.
- Making decision regarding to the conditions and commitment set by delegator. Sometimes, if the original request is changed by the delegator, the delegatee must be aware and repeat the commitment negotiation process.

From the Service Provider and Authorisation Authority's perspective, the pre-delegation commitment is not particularly important as they are not really involved with the negotiation between the delegator and the delegatee.

Post-delegation Commitment

After the delegatee receives the delegation assertion, it must check the validity and determine whether to accept the delegation (after accepting the pre-delegation commitment). Then the delegatee should have the ability

to invoke the necessary services from the Service Provider using the delegated attributes.

When the Service Provider receives requests from the delegatee for a particular service using delegated attributes, the Service Provider can ask or divert the delegatee to the Authorisation Authority to confirm the eligibility of the delegatee for the requested services. If the Authorisation Authority grants the access, the delegatee now can enjoy the service from the Service Provider. It should be noted that the Authorisation Authority of the system, on behalf of the Service Provider, makes the access control decisions. The delegation assertion is an authorisation for delegation, not for granting access to services.

After these interactions, the involved parties need to perform the post-delegation activities which primarily keep track of the activities and conditions set by the involved parties in the pre-delegation commitment negotiation. A typical activity for post-delegation commitment is to update delegation information of the involved parties, especially delegator and delegatee to maintain the consistency of the delegation status. Naturally, the post-delegation commitment is quite simple in comparison to the pre-delegation commitment.

From the Authorisation Authority and Service Provider's perspective, the only task they have to do is to notify the delegator about the request of the delegatee.

The delegatee then also may choose to inform the delegator that the delegation assertion was used so that the delegator can finalise the process of keeping track and monitoring the delegation transaction. The delegator also needs to do some management tasks such as updating information related to the delegation.

Pre- and Post-Delegation Commitment in Indirect Delegation

In an indirect delegation chain, the situation becomes more complicated with the involvement of intermediate entities which act as brokers between the original delegator and final delegatee.

Figure 2 illustrates an indirect delegation where multiple delegations recursively happen to form a chain of delegations from the original delegator to the final delegatee via multiple intermediate entities which, in turn, act in the role of both delegator and delegatee.

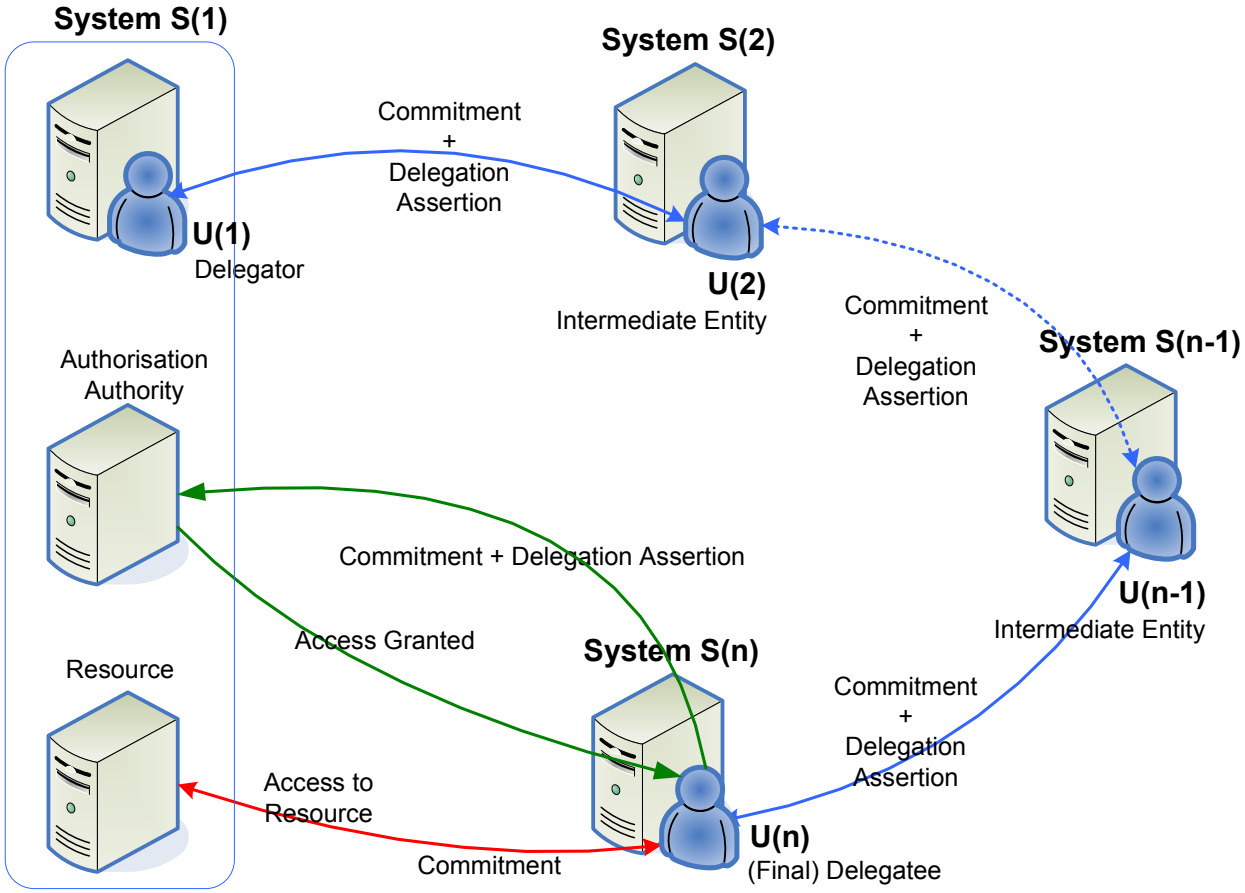

Figure 2: An indirect delegation chain

In Figure 2, there are *n* entities involved in the delegation process. Except the first and the last entities which are respectively the delegator and delegatee, all other entities will act in both roles. For example, in the delegation chain, entity U(k) will accept some delegated privilege attributes from U(k-1) to be a delegatee. Entity U(k), in turn, will transfer those privilege attributes to U(k+1) to effectively be a delegator. The final delegatee U(n) will be the actual entity which asks for the delegation from the beginning.

From the delegation commitment perspective, it is quite difficult to define and keep track of the commitment of the intermediate entities because in some cases, both delegatee and delegator will not be able to get the information of intermediate entities of the delegation chain in advance. So, it is not feasible to define pre-delegation commitment for these entities. Instead, a generic "forward and keep-track" post-delegation commitment should be enforced.

Thus, the intermediate entities have to commit the following tasks:

- Notify the previous delegatee after forwarding the delegation assertion to the next delegatee in the chain.
- Notify the original delegator so that the delegator can keep track of the development of the delegation chain.

5 Commitment Enforcement Framework

This section provides a framework for expressing and enforcing the commitment of the involved parties in the delegation process. For simplicity, only commitment of the involved parties in direct delegation will be discussed. In this section, SAML will be used as the means to carry information in the delegation negotiation process. SAML has been selected due to its expressiveness and the compatibility with various standards and implementations such as Shibboleth and Liberty Alliance.

5.1 Delegation Commitment Assertion

Delegation commitment assertion is the SAML assertion (Cantor et al. 2005) used in the commitment negotiation process. By way of example, the discussion will now centre upon SAML as means to exchange the delegation commitment assertion. However, SAML assertion is not designed to carry delegation commitment assertions directly. So, the assertion is based on a basic set of elements and some extensions as proposed on Gomi et al. and Wang et al. (Gomi et al. 2005; Wang et al. 2005).

Based on vocabularies of SAML 2.0, Gomi et al. (Gomi et al. 2005), Wang et al. (Wang et al. 2005) and Cantor (Cantor 2005) proposed a basic set of elements for a delegation assertion as follows:

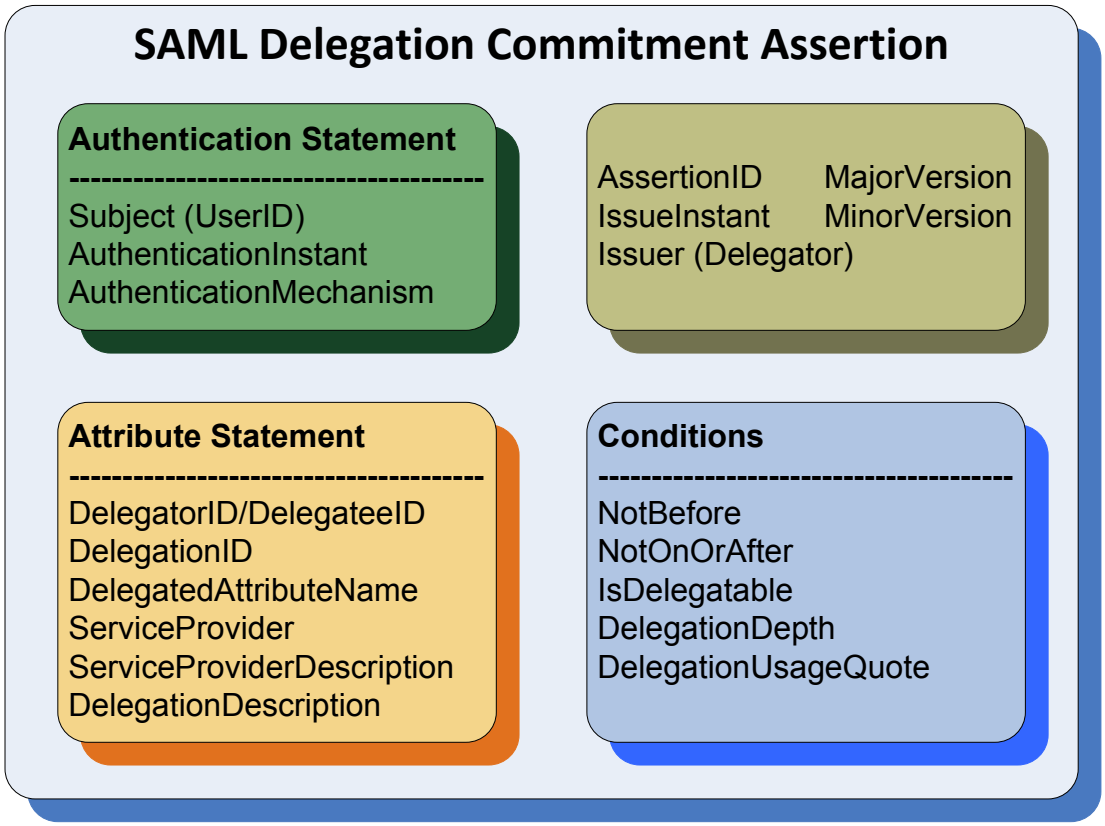

Figure 3: A typical structure of a SAML delegation commitment assertion (extended from Gomi et al. (Gomi et al. 2005) and Wang et al. (Wang et al. 2005))

- *Issuer*: the issuer of the assertion which could be the delegator or delegatee
- *Signature*: the digital signature of the delegator for integrity protection
- *Subject*: principal's information. This contains the principal's name identifier
- *Conditions*: the valid duration of the assertion
- *DelegatorID*: the delegator's identity
- *DelegateeID*: the delegatee's identity
- *DelegationID*: the unique identifier of the delegation transaction. This value will be assigned by the initialised party (can be either delegatee or delegator) and will be maintained uniquely for the whole delegation process
- *IsDelegatable*: indicates whether the delegation assertion can be further delegatable. It will be used for multiple delegations

This is just the basic set of elements. To support delegation, the SAML AttributeStatement field is extended to combine delegation information such as user's privilege attributes (Gomi et al. 2005; Wang et al. 2005).

Based on their designs, we extend and modify some elements to achieve the purpose of supporting commitment negotiation process. Figure 3 depicts a modified SAML assertion to perform delegation. The AttributeStatement and Conditions are extended to contain the following sub-elements and attributes:

- *DelegatedAttributeName*: the name of the privilege attribute subjected to delegation
- *DelegationDepth*: the maximum number of times which the assertion can be delegated (the depth of delegation chain)
- *DelegationUsageQuote*: the maximum number of times which the assertion can be used
- *ServiceProvider*: the resource which the delegatee wants to access
- *ServiceProviderDescription*: the human readable description for the above service
- *DelegationDescription*: general description about the delegation transaction

The Conditions elements will be used to set the validity of the assertion.

5.2 Pre-Delegation Commitment

This section proposes the new basic protocol to achieve the pre-delegation commitment of delegator and delegatee. For the sake of simplicity, it assumes that the authentication and authorisation mechanisms are already in place. Issues such as how delegation requests can be initiated and validated, how identities of delegator and delegatee can be verified, etc. are out of scope.

Assuming that there is a user who wants to access a resource R1 via ServiceProvider SP1 which he/she does not possess necessary privileges (*p1 and p2)*. The user; now is the delegatee; will have to ask another user, delegator, to delegate him/her some certain privilege attributes.

Figure 4 depicts the basic flow diagram for the concept of the pre-delegation commitment.

Delegatee Commitment

From the delegatee's perspective, the delegatee will have to do the following steps:

1. Delegatee starts the delegation process by sending a request for delegation. This is just to inform the delegator that the delegatee is asking for delegation.

2. The delegator will respond and ask for the commitment of delegatee.

3. The delegatee now sends to the delegator an assertion to express the following information:

- The task the delegatee wants to perform
- The involved service provider
- The necessary valid duration (how long for the delegation to last)?
- How many times the delegation assertion can be used?
- The proposed security context and/or access control profile for the delegation transaction. This is to allow the delegatee to suggest a security context it is capable of using.
- If possible, in some cases, the delegatee should suggest the necessary privilege attributes.

These factors form the *pre-delegation commitment of delegatee*. This assertion will be kept by the delegator for tracking purposes later.

Delegator Commitment

The delegator will consequently assess the request. Assume that the delegator agrees to grant the request.

4. The delegator now can issue an assertion which contains the following information:

- Valid duration
- The number of times the delegation assertion can be used. Security context of validation: for example only valid for the service provider on system S(1) or with service providers in the federation F(1).
- Require or not require confirmation upon finishing the use of delegation assertion.

These factors form the *pre-delegation commitment of delegator*. This assertion, a commitment assertion, will be kept by the delegator for tracking purposes later. By issuing this assertion, the delegator is now responsible for any verification requests related to this assertion within the valid duration.

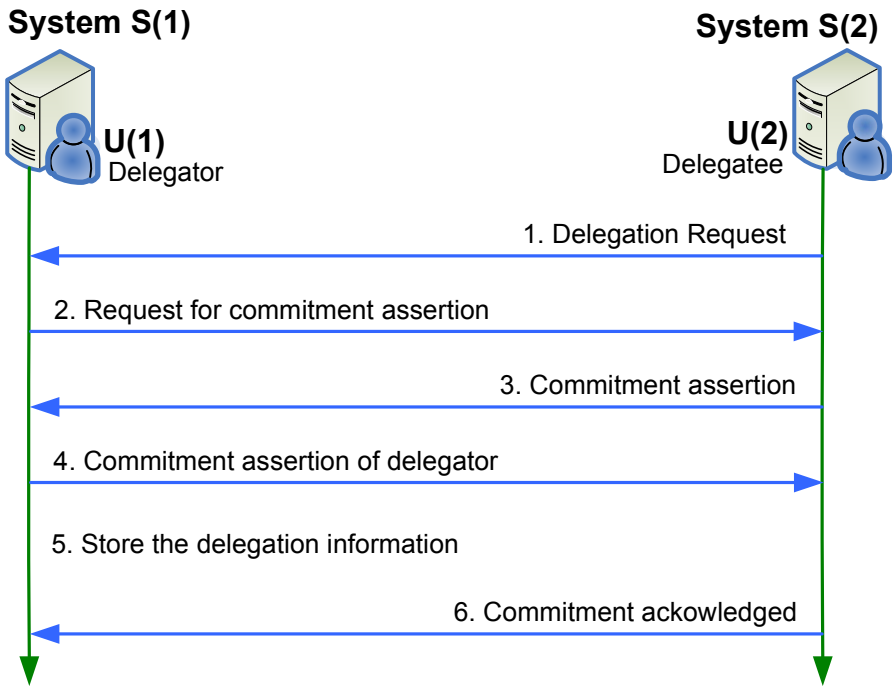

Figure 4: Pre-delegation commitment exchanged between delegator and delegatee

5. Keep track of the use of delegation assertion by putting them into a tracking list. The list is the mapping of delegatee's identity and DelegationID. This list will be stored personally by the delegator. A storage mechanism will be defined by the delegator or the delegator's system authority personally to preserve the autonomy of the federation.

6. The delegatee then has to confirm that it agrees with this arrangement.

5.3 Post-Delegation Commitment

After each service invocation, the involved parties need to complete the commitment by doing the work which is committed. In the post-delegation phase, the commitment is mostly the responsibility of delegatee because the delegator only needs to monitor and keep track of the progress by waiting for the feedback from the delegatee, the Service Provider and the Authorisation Authority.

Figure 5 depicts the basic flow diagram for the concept of the post-delegation commitment.

1. The Authorisation Authority needs to let the delegator know the delegated privilege attributes were used. The Authorisation Authority will send the delegator and the Service Provider the assertion with the following information:

- Request from delegatee
- Timestamp of the request

2. The Service Provider needs to let the delegator know that the delegated privilege attributes were used. The Service Provider will send the delegator the assertion with the following information:

- Request from delegatee
- Timestamp of the request

3. The delegatee needs to conduct the post-delegation commitment. However, due to the trust relationship with delegator, it does not need to report back to the delegator.

- Reduce the DelegationUsageQuote

4. The delegator needs to accept the confirmation from both sides (service provider and authorisation authority) and store the confirmations for tracking purposes.

5. The delegator also needs to reduce the DelegationUsageQuote by one or marks the delegation assertion as expired if the quote reaches zero.

Due to the trust assumption in which delegatee is not trusted by the delegator, there is no need for the delegatee to report back to the delegator. The delegator will rely on the report from Authorisation Authority and Service Provider for the tracking purpose.

6 Discussion and Future Work

In the models of Gomi et al. and Wang et al., delegator's privileges are transferred from a delegator to delegatee in accordance with the order of delegation assertion flow (Gomi et al. 2005; Wang et al. 2005). The main philosophy behind both models is that the design only creates the environment (framework) for users to communicate and perform delegation. The rest (such as how to verify the assertion, how to use the assertion, etc.) is merely up to the involved parties. So, it would be difficult to say whether the model supports grant or transfer delegation.

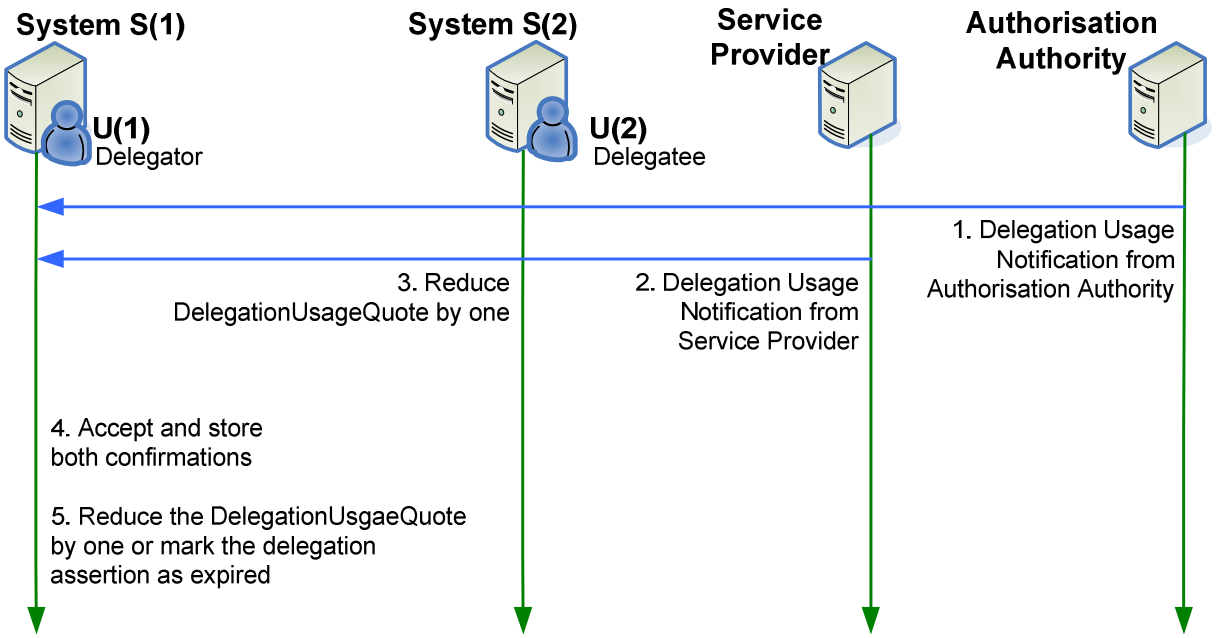

Figure 5: Post-delegation commitment exchanges between involved parties

In general, this is a good delegation mechanism as it is quite clear and simple in terms of administration. It allows the federation to keep track and trace back any delegation transaction and so be able to maintain a precise authorisation state of user at a given time. However, it lacks the capability to check for the constraints and resolve the conflicts between delegated privilege attributes and between the delegated privileges with the involved policies. The models also ignored the delegatee's role constraint and conflict resolution as they do not mention about how the delegated privilege attributes will be fitted into the current privilege attribute set of the delegatee. The model also ignored the issue of commitment of involve parties.

By addressing the commitment issue, the protocol expressed in this paper can be considered as an extension to these models. In fact, the mechanism is a monitoring and checking approach. The mechanism provides means to monitor and keep track the delegation process. The tracking information can be used later for trust assessment or making revocation when necessary. Thus, the mechanism can be considered as a complement module to provide a more conservative protection, more manageable delegation process in federated systems. The mechanism also preserves room for future improvement with the consideration of security context suggested by the delegatee. However, in this model, we have not discussed thoroughly the roles and commitments of Authorisation Authority and Service Provider. The pre-delegation commitment which contains some delegation conditions and constraints of Authorisation Authority and Service Provider is an interesting and important aspect which needs to be addressed. Our mechanism also ignores the issues of delegatee's role constraint and conflict resolution and considers it as part of future work.

In addition, for simplicity, the indirect delegation is not thoroughly analysed. The mechanism detailed in this paper only looks at direct delegation between the involved parties. So, this paper does not sufficiently address the necessary commitment in case of indirect delegation with the involvement of multiple parties in the delegation process. In case of indirect delegation, the commitment of the intermediate parties maybe varied. This makes the task of keeping track of their commitments very complicated. When the number of intermediate parties grows large, the protocol will become too complex with a lot of delegation assertions to be exchanged. So there is also an issue of how to improve the simplicity and clarification of the mechanism. The ability of keeping track of commitment can also lead to the investigation of the issue of trust of access control via delegation. In addition, the future works will also extend the security context and consider the commitment of other involved parties such as delegation authority, identity provider and authentication and authorisation authority. Role constraint and conflict resolution for the delegation process will also form an important part of the future work.

7 Conclusion

This paper discusses the initial concept of delegation commitment and proposes a simple scheme to monitor and keep track of the commitment of the involved parties when requesting delegation assertion for a particular task. This paper introduces a mechanism to help parties involved in the delegation process to express commitment constraints, perform the commitments and track the committed actions. The mechanism looks at two different aspects: pre-delegation commitment and post-delegation commitment. In pre-delegation commitment, the mechanism enables the involved parties to express the delegation constraints and address those constraints. The post-delegation commitment phase enables those parties to inform the delegator and service providers about how the commitments are conducted. The mechanism utilises

modified SAML assertion structures to support the delegation purposes. Future work includes investigation of indirect delegation as well as role and commitments of other involved parties such as Identity Provider, Authentication and Authorisation Authority and Service Provider as well as role constraint and conflict resolution.

8 Acknowledgement

The research is kindly funded by the Smart Services CRC, Australia and the Information Security Institute, Queensland University of Technology, Australia. The research is also supported by Information Queensland, Queensland State Government, Australia.

9 References

Atluri, V. and Warner, J. (2005): Supporting conditional delegation in secure workflow management systems. *Proc. 10th ACM symposium on Access control models and technologies*, Stockholm, Sweden, 49 - 58, ACM Press, New York, NY, USA.

Aura, T. (1999): Distributed access-rights management with delegation certificates. *Lecture Notes in Computer Science (LNCS) - Secure Internet programming: security issues for mobile and distributed objects*, **1603**:211 - 235.

Barka, E. and Sandhu, R. (2000): Framework for role-based delegation models. *Proc. 16th Annual Conference Computer Security Applications (ACSAC '00)*, 168 - 176, IEEE Computer Society, Washington, DC, USA.

Barka, E. and Sandhu, R. (2004): Role-Based Delegation Model/ Hierarchical Roles (RBDM1). *Proc. 20th Annual Computer Security Applications Conference (ACSAC'04)*, 396 - 404, IEEE Computer Society, Washington, DC, USA.

Bhatti, R., et al. (2006): X-FEDERATE: A Policy Engineering Framework for Federated Access Management. *IEEE Transactions on Software Engineering*, **32**(5):330 - 346.

Cantor, S. (2005): SAML 2.0 Single Sign-On with Constrained Delegation, http://shibboleth.internet2.edu/docs/draft-cantor-saml-sso-delegation-01.pdf. Accessed 12th June 2007.

Cantor, S., et al. (2005): Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0, http://www.oasis-open.org/specs/index.php#samlv2.0. Accessed 15th August 2006.

Castelfranchi, C. (2004): Towards a Theory of Delegation for Agent-based Systems, http://www.istc.cnr.it/T3/download/Theory-of-delegation.pdf. Accessed 10th September 2006.

Crampton, J. (2003): On permissions, inheritance and role hierarchies. *Proc. 10th ACM Conference on Computer and Communications Security*, Washington D.C., USA, 85 - 92, ACM Press, New York, NY, USA.

Crampton, J. and Khambhammettu, H. (2006): Delegation in Role-Based Access Control. *Proc. 11th European Symposium On Research In Computer Security (ESORICS 2006)*, Hamburg, Germany, Lecture Notes in Computer Science (LNCS) **4189**:174 – 191, Springer-Verlag, Berlin, Germany.

Fragoso-Rodriguez, U., et al. (2006): Federated Identity Architectures. *Proc. 1st Mexican Conference on Informatics Security 2006 (MCIS'2006)*, Oaxaca, Mexico, IEEE Computer Society, Mexico.

Gomi, H., et al. (2005): A delegation framework for federated identity management. *Proc. ACM Workshop on Digital Identity Management*, 94 - 103, ACM Press, New York, NY, USA.

Joshi, J. B. D. and Bertino, E. (2006): Fine-grained role-based delegation in presence of the hybrid role hierarchy. *Proc. 7th ACM symposium on Access control models and technologies*, Lake Tahoe, California, USA, 81 - 90, ACM Press, New York, NY, USA.

Madsen, P., et al. (2005): Federated identity management for protecting users from ID theft. *Proc. Workshop on Digital Identity Management*, Fairfax, VA, USA, 77 - 83, ACM Press, New York, NY, USA.

Na, S. Y. and Cheon, S. H. (2000): Role delegation in role-based access control. *Proc. 5th ACM workshop on Role-based access control*, Berlin, Germany, 39 - 44, ACM Press, New York, NY, USA.

Petkovic, M. and Koster, R. P. (2005): User-Attributed Rights in DRM. *Proc. 1st International Conference on Digital Right Management: Technologies, Issues, Challenges and Systems (DRMTICS 2005)*, Sydney, Australia, **3919**:75 - 89, Springer-Verlag, Berlin, Germany.

Sandhu, R. (1990): Separation of Duties in Computerised Information Systems. *Proc. IFIG WG11.3 Workshop in Database Security*, Halifax, United Kingdom, 179 - 189.

Sandhu, R. (1998): Role activation hierarchies. *Proc. 3rd ACM Workshop on Role-based Access Control*, Fairfax, VA, United States, 33 - 40, ACM Press, New York, NY, USA.

Sandhu, R. (2005): Role Usage and Activation Hierarchies, http://www.list.gmu.edu/it862/it862s05/Role_Activation_Hierarchies.ppt. Accessed 16th February 2007.

Schaad, A. (2003): A Framework for Organisational Control Principles. PhD Thesis. The University of York.

Shen, H. H. (2006): Access Control for Collaborative Environments. PhD Thesis. Purdue University.

Tamassia, R., et al. (2004): Role-based cascaded delegation. *Proc. 9th ACM symposium on Access control models and technologies*, Yorktown Heights, New York, USA, 146 - 155, ACM Press, New York, NY, USA.

Tolone, W., et al. (2005): Access control in collaborative systems. *ACM Computing Surveys (CSUR)*, **37**(1):29 - 41.

Wainer, J. and Kumar, A. (2005): A Fine-grained, Controllable, User to User Delegation Method in RBAC. *Proc. 10th ACM symposium on Access control*

models and technologies (SACMAT'05), Stockholm, Sweden, 59 - 66, ACM Press, New York, NY, USA.

Wang, H., et al. (2006): A framework for Role Based Group Delegation in Distributed Environments. *Proc. 29th Australasian Computer Science Conference (ACSC2006)*, Hobart, Tasmania, **48**:321 - 328, Australian Computer Society, Inc., Darlinghurst, Australia.

Wang, H. and Osborn, S. (2006): Delegation in the role graph model. *Proc. 11th Symposium on Access Control Models and Technologies*, Lake Tahoe, California, USA, 91 - 100, ACM Press, New York, NY, USA.

Wang, J., et al. (2005): Extending the Security Assertion Markup Language to Support Delegation for Web Services and Grid Services. *Proc. IEEE International Conference on Web Services (ICWS'05)*, Orlando, Florida, USA, 67 - 74, IEEE Computer Society, Washington, DC, USA.

Wu, J., et al. (2005): Delegatable Access Control for Fine-Grained XML. *Proc. 11th International Conference on Parallel and Distributed Systems - Workshops (ICPADS'05)*, 270 - 274, IEEE Computer Society, Washington, DC, USA.

Yin, G., et al. (2004): An Authorization Framework Based on Constrained Delegation. *Proc. Parallel and Distributed Processing and Applications*, Hong Kong, China, Lecture Notes in Computer Science **3358**:845 – 857, Springer-Verlag, Berlin, Germany.

Zhang, L., et al. (2002): A role-based delegation framework for healthcare information systems. *Proc. 7th ACM symposium on Access control models and technologies*, Monterey, CA, USA, 125 - 134, ACM Press, New York, NY, USA.

Zhang, L., et al. (2003): A rule-based framework for role-based delegation and revocation. *ACM Transactions on Information and System Security (TISSEC)*, **6**(3):404 - 441.

Zhang, L., et al. (2001): A rule-based framework for role based delegation. *Proc. 6th ACM symposium on Access control models and technologies*, Chantilly, VA, USA, 153 - 162, ACM Press, New York, NY, USA.

Zhang, X., et al. (2006): A usage-based authorization framework for collaborative computing systems. *Proc. 11th Symposium on Access Control Models and Technologies*, Lake Tahoe, California, USA, 180 - 189, ACM Press, New York, NY, USA.

Zhang, X., et al. (2003): PBDM: a flexible delegation model in RBAC. *Proc. 8th ACM symposium on Access control models and technologies*, Como, Italy, 149 - 157, ACM Press, New York, NY, USA.