Protecting Information Sharing in Distributed Collaborative Environment*

Min Li and Hua Wang

Department of Mathematics & Computing
University of Southern Queensland, Australia
Email: {limin, wang}@usq.edu.au

Abstract. Information sharing on distributed collaboration usually occurs in broad, highly dynamic network-based environments, and formally accessing the resources in a secure manner poses a difficult and vital challenge. Our research is to develop a systematic methodology for information sharing in distributed collaborative environments. It will ensure sensitive information and information assurance requirements, and incorporate new security constrains and policies raised by emerging technologies. We will create a new rule-based framework to identify and address issues of sharing in collaborative environments; and to specify and enforce security rules to support identified issues while minimizing the risks of information sharing through the framework.

1 Aims and background

We aim to develop a policy-based framework for information sharing in distributed collaborative environments with role-based delegation and revocation. The motivation of role-based delegation and revocation are that users themselves may delegate role authorities to others to process some authorized functions and later remove the authorities. Role-based delegation and revocation models will be developed with comparisons to established technical analysis, laboratory experiments, support hierarchical roles and multistep delegation. An innovation policy-based language for specifying and enforcing rules on the models is proposed as the fundamental technique within this framework. The models will be implemented to demonstrate the feasibility of the framework and secure protocols for managing delegations and revocations.

Delegation is the process whereby an active entity in a distributed environment grants access resource permissions to another entity. In today's highly dynamic distributed systems, a user often needs to act on another user's behalf with part of the user's rights. To solve such delegation requirements, ad-hoc mechanisms are used in most systems by

* The research is support by an ARC Discovery Grant DP0663414.

compromising existing disorganized policies or additional components to their applications [26, 16, 18]. The basic idea of delegation is to enable someone to do a job, for example, a secretary. Effective delegation not only makes management systems ultimately more satisfactory, but also frees the delegating users to focus on other important issues. In access control management systems, the delegation arises when users need to act on another user's behalf in accessing resources. The delegation might be for a short time, for example, sharing resources temporarily with others during one week holiday. Otherwise users may perceive security as an obstacle of the resources sharing. With delegation, the delegated user has the privileges to access information without referring back to the delegating user.

Delegation is recognized as vital in a secure distributed computing environment [1, 3, 10]. The most common delegation types include user-to-machine, user-to-user, and machine-to-machine delegation. They all have the same consequence, namely the propagation of access permission. Propagation of access rights in decentralized collaborative systems presents challenges for traditional access mechanisms because authorization decisions are made based on the identity of the resource requester. Unfortunately, access control based on identity may be ineffective when the requester is unknown to the resource owner. Recently some distributed access control mechanisms have been proposed: Lampson *et al.* [12] present an example on how a person can delegate its authority to others; Blaze *et al.* [5, 6],introduced trust management for decentralized authorization; Abadi *et al.* [1] showed an application of express delegation with access control calculus; and Aura [2] described a delegation mechanism to support access management in a distributed computing environment.

The National Institute of Standards and Technology developed role-based access control (*RBAC*) prototype [7] and published a formal model [9]. *RBAC* enables managing and enforcing security in large-scale and enterprise-wide systems. Many enhancements of *RBAC* models have been developed in the past decade. In *RBAC* models, permissions are associated with roles, users are assigned to appropriate roles, and users acquire permissions through roles. Users can be easily reassigned from one role to another. Roles can be granted new permissions and permissions can be easily revoked from roles as needed. Therefore, *RBAC* provides a means for empowering individual users through role-based delegation in distributed collaboration environments.

The importance of delegation has been recognized for a long time, but the concept has not been supported in *RBAC* models [8, 19]. A security

officer has to assign a role to the delegated user if the role is required to be delegated to the user. Such a model significantly increases the management efforts in a decentralized collaboration environments because of the dynamic of delegations and the continuous involvement from security officers. We will provide a bridge of the gap between delegation techniques and *RBAC* models.

2 Significance and innovation

Delegation is an important feature in many collaboration applications. For example, the Immigration Department is developing partnerships between immigration agencies and people in local areas to address possible problems. Immigration officers are able to prevent illegal stay and crime if they efficiently collaborate with the people. The problem-oriented immigrating system (*POIS*) is proposed to improve the service as a part of the Immigration Department's ongoing community efforts including identifying potential problems and resolving them before they become significant. With efficient delegation, officers respond quickly to urgent messages and increase the time spent confronting problems.

In *POIS*, officers might be involved in many concurrent activities such as conducting initial investigations, analyzing and confronting crimes, preparing immigration reports, and assessing projects. In order to achieve this, users may have one or more roles such as lead officer, participant officer, or reporter. In this example, Tony, a director, needs to coordinate analyzing and confronting crimes and assessing projects. Collaboration is necessary for information sharing with members from these two projects. To collaborate closely and make two projects more successful, Tony would like to delegate certain responsibilities to Christine and her staff. The prerequisite conditions are to secure these processes and to monitor the progress of the delegation. Furthermore, Christine may need to delegate the delegated role to her staff as necessary or to delegate a role to all members of another role at the same time. Without delegation skill, security officers have to do excessive work since the involvement of every single collaborative activity. We can find the major requirements of role-based delegation in this example:

1. Group-based delegation means that a delegating user may need to delegate a role to all members of another role at the same time. We introduce a new ability-based delegation model in our recent work [13].

2. Multistep delegation occurs when a delegation can be further delegated. Single-step delegation means that the delegated role cannot be further delegated.
3. Revocation schemes are important characters in collaboration. They take away the delegated permissions. There are different revoking schemes, among them are strong and weak revocations, local and global revocation. We discuss these different revocation with according algorithms in our recent paper [14].
4. Constraints are an important factor in *RBAC* for laying out higher-level organizational policies. It defines whether or not the delegation or revocation process is valid.
5. Partial delegation means only subsets of the permissions are delegated while total delegation means all permissions are delegated. Partial delegation is an important feature because it allows users only to delegate required permissions. The well-known least privilege security principle can be implemented through partial delegation.

Although the concept of delegation is not new in authorizations [2, 3, 5, 10, 21, 25, 16, 17], role-based delegation received attention only recently [15, 27, 28]. Aura [2] introduced key-oriented discretionary access control systems that are based on delegation of access rights with public-key certificates. A certificate has the meaning:

S_K (During the validity period, if I have the rights R, I give them to someone)

S_K denotes a signed message that includes both the signature and the original message. The key that signed the certificate (K) is the issuer and the rights R given by the certificate are the authorization. With the certificate, the issuer delegates the rights R to someone. The systems emphasized decentralization of authority and operations but their approach is a form of discretionary access control. Hence, they can neither express mandatory policies like Bell-LaPadula model [4], nor possible to verify that someone does not have a certificate. Furthermore, some important policies such as separation of duty policies cannot be expressed with only certificates. They need some additional mechanism to maintain the previously granted rights and the histories must be updated in real time when new certificates are issued. Delegation is also applied in decentralized trust management [6, 15, 16]. Blaze *et al.* [6] identified the trust management problem as a distinct and important component of security in network services and Li *et al.* [15, 16] made a logic-based knowledge representation for authorization with tractable trust-management in large-scale, open, distributed systems. Delegation was used to address the trust management

problem including formulating security policies and security credentials, determining whether particular sets of credentials satisfy the relevant policies, and deferring trust to third parties. Other researchers have investigated machine to machine and human to machine delegations [25, 1, 10]. For example, Wang *et al.* [25] proposed a secure, scalable anonymity payment protocol for Internet purchases through an agent which provided a higher anonymous certificate and improved the security of consumers. The agent certified re-encrypted data after verifying the validity of the content from consumers. The agent is a human to machine delegation which can provide new certificates. However, many important role-based concepts, for example, role hierarchies, constraints, revocation were not mentioned.

Wang *et al.* [21] discussed the mobility of user-role relationship in *RBAC* management and provided new authorization allocation algorithms for *RBAC* along with the mobility that are based on relational algebra operations. They are the authorization granting algorithm, weak and strong revocation algorithms. The paper does not use role delegation but instead defines the role mobility, whereby a user with an mobile role may further grant other roles but she/he cannot accept other roles if she/he has an immobile role. The mobility could be viewed as a special case of multistep delegation in their work. But some important delegation features such as partial delegation and delegation revocation have not been considered. Barka and Sandhu [3] proposed a simple model for role-based delegation called *RBDM0* within *RBAC*0, the simplest form of *RBAC*96 [19]. *RBDM0* is a simple delegation model supporting only flat roles and single step delegation. However, they neither gave the definition of role-based delegation relation, which is a critical notion to the delegation model nor discussed the relationships among original user and delegated user. Some important features such as role hierarchies and revocations were not supported in *RBDM0*.

Some researchers have worked on the semantics of authorization, delegation, and revocation. Wang *et al.* [20] described a secure and flexible protocol and its role based access control for M-services. The protocol is based on a Credential Center, a Trusted Center and a ticket based mechanism for service access. It supports efficient authentication of users and service providers over different domains and provides a trusted model for participants. The concepts, protocols, and algorithms for access control in distributed systems from a logical perspective have been studied. However, there is no multistep delegation control mechanism since every delegation can be freely delegated. Hagstrom *et al.* [11] studied various

problems of revoking in an ownership-based framework, but their attempt was still not sufficient to model all the revocations required in role-based delegation, for example, grant-independent and duration-restricted revocations. Zhang *et al.* [27, 28] proposed a rule-based framework for role-based delegation including *RDM2000* model. *RDM2000* model is based on *RBDM0* model with some limitations that we mentioned before. Furthermore, as a delegation model, it does not support group-based delegation. *RDM2000* does not consider administrative role delegation but the deletion of regular roles. The model does neither analyse how do original role assignment changes impact delegations nor implement with XML-based language.

We will focus exclusively on how to specify and enforce policies for authorizing role-based delegation and revocation using a rule-based language. We will continue our previous work and propose delegation frameworks including revocation models, group-based, multistep, and partial delegations. With the revocation models, we will not only consider the deletion of regular roles but also administrative role delegation. Additionally, in order to provide sufficient functions with the framework, we will analyze how do original role assignment changes impact delegations and implement with XML-based language. This kind of language for role-based delegation has not been studied.

3 Approach

Task 1: A role-based delegation framework

This task will develop a delegation framework called *RBDF*. This framework supports role hierarchy and multistep delegation and revocation by introducing the delegation relation, delegation authorization, role-based revocation and revocation authorization.

Two relations are included in role-based access control: user-role assignment (*URA*) and permission-role assignment (*PRA*). *URA* is a many-to-many relation between users and roles and *PRA* is a many-to-many relation between permissions and roles. Users are enabled to use the permissions of roles assigned to them. *RBAC* management systems have many advantages with its flexibility of assigning permissions to roles and users to roles [24]. There are two types of roles associated with user: *Original roles* and *Delegated roles*. The former is a role assigned to the user whilst the latter one is a role delegated to the user.

The same role can be an original role of one user and a delegated role of another user. Role hierarchy is a basic relationship between roles

that specifies which role may inherit all of the permissions of another role. The relationship of *Senior-Junior* shows hierarchies between roles. Senior roles inherit permissions from junior roles. Role hierarchies provide a powerful and convenient means to satisfy the least privilege security principle since only required permissions are assigned to a role. Because of role hierarchies, a role may be an original role and a delegated role of the same user. The original user-role assignment (OUA) is a many-to-many user-role assignment relation between users and original roles. The delegated user-role assignment (DUA) is a many-to-many user assignment relation between users and delegated roles.

Role-Based Delegation Relational database systems will be designed. Database systems have been applied in our previous work to solve consistency problems in user-role assignment and permission-role assignment [23, 22]. A set of relations such as *roles, users, permissions, user-role, role-permission* has been developed [23, 24] for the formal approaches that are based on relational structure and relational algebra operation in database system. There are three major elements in a delegation relation: original user-role assignments (OUA), delegated user-role assignment (DUA), and constraints. Constraints are very important in role-based model [21]. Delegation may associate with zero or more constraints. The delegation relation supports partial delegation in a role hierarchies: a user who is authorized to delegate a role r can also delegate a role that is junior to r.

As we mentioned before, there are various delegations in real-time application: single-step, multistep, group-based, and partial delegations. In single-step delegation the delegated role cannot further delegate. We also can define a maximum number of steps in multistep delegation. The maximum delegation number imposes restriction on the delegation. Single-step delegation is a special case of multistep delegation with maximum delegation number equal to one. We will develop delegation models to support these different delegations.

Delegation Authorization The delegation authorization goal imposes restrictions on which role can be delegated to whom. We partially adopt the notion of prerequisite condition from Wang *et al.* [23] to introduce delegation authorization in the rule-based delegation framework ($RBDF$).

We will develop database systems for $RBDF$ in this task to support group-based, multistep, partial delegations and revocations and analyze what delegation impact will happen if an original role assignment is changed.

Task 2: The rule-based policy specification language

The motivation behind policy-based language are: 1) delegation relations defined in role-based delegation model lead naturally to declarative rules; 2) an individual organization may need local policies to further control delegation and revocation. A policy-based system allows individual organizations to easily incorporate such local policies.

We will show how our construction is used to express delegation and revocation policies.

The Language The rule-based specification language specifies and enforces authorization of delegation and revocation based on the new delegation model. It is entirely declarative so it is easier for security administrators to define policies. The proposed language will be a rule-based language with a clausal logic. A clause, also known as a rule, takes the form: $H \leftarrow B$. where H stands for rule head and B stands for rule body. B is a prerequisite condition of a successful H. If the condition defined in the rule body is true, then it will trigger authorizations. An advantage is that the rule body can include the condition of an authorization policy and the rule head can include the authorization. This provides the mechanism for authorization specification and enforcement.

Rules for Enforcing Policies Basic authorization will specify the policies and facts in the delegation framework. Addition to the basic authorization policies, further derivations are needed for authorization and their enforcement. A derivation rule body describes a semantic logic that consists of basic authorization, conditions and functions. The result can be either authorized or denied.

The language developed in Task 2 will be used in the database systems (Task 1) to process delegation and revocation authorizations.

4 Current progress

1. We develop a flexible ability-based delegation model (ABDM), in which a user can delegate a collection of permissions, named an ability, to another user or all members of a group; we also analyze delegation granting and revocation authorization algorithms in this model [13]. (Part of Task 1)
2. we discuss granting and revocation models related to mobile and immobile memberships between permissions and roles and provide proposed authorization granting algorithm to check conflicts and help allocate the permissions without compromising the security [14]. (Part of Task 1)

3. We specify constraints of Usage Control Model (UCON) with object constraints language (OCL). The specification not only provides a tool to precisely describe constraints for system designers and administrators, but also provides the precise meaning of the new features of UCON, such as the mutability of attributes and the continuity of usage control decisions. This work is under preparation for submitting. (Part of Task 2)

References

1. Abadi, M., Burrows, M., Lampson, B., and Plotkin, G. 1993. A calculus for access control in distributed systems. *ACM Trans. Program. Lang. Syst.* 15, 4(Sept.), 706-734.
2. Aura, T. 1999. Distributed access-rights management with delegation certificates. *Security Internet programming*. J. Vitec and C. Jensen Eds. Springer, Berlin, 211-235.
3. Barka, E. and Sandhu, R. 2000. A role-based delegation model and some extensions. *In Proceeings of 16th Annual Computer Security Application Conference*, Sheraton New Orleans, December, 2000a, 168-177.
4. Bell D.E., La Padula L.J. 1976. Secure Computer System: Unified Exposition and Multics Interpretation, *Technical report ESD-TR-75-306*, The Mitre Corporation, Bedford MA, USA.
5. Blaze, M. Feigenbaum, J. and Lacy, J. 1996. Decentralized trust management. *IEEE Symposium on Security and Privacy*. Oakland, CA, 164-173.
6. Blaze, M. Feigenbaum, J., Ioannidis, J. and Keromytis, A. 1999. *The role of trust management in distributed system security. Security Internet Programming*. J. Vitec and C. Jensen, eds. Springer, Berlin, 185-210.
7. Feinstein, H. L. 1995. Final report: NIST small business innovative research (SBIR) grant: role based access control: phase 1. Technical report. *SETA Corporation*.
8. Ferraiolo, D., Cugini, J., and Kuhn, D.R. 1995. Role-based access control (*RBAC*): features and Motivations. *In Proceedings of 11th Annual Computer Security Application Conference*. New Orleans, LA, December, 241-241.
9. Ferraiolo, D. F. and Kuhn, D. R. 1992. Role based access control. *The proceedings of the 15th National Computer Security Conference*, 554-563.
10. Gladney, H. 1997. Access control for large collections. *ACM Transactions on Information Systems* 15, 2(April), 154-194.
11. Hagstrom, A., Jajodia, S., Presicce, F., and Wijesekera, D. 2001. Revocations-a classification. *In Proceedings of 14th IEEE Computer Security Foundations Workshop*, Nova Scotia, Canada, June, 44-58.
12. Lampson, B. W., Abadi, M., Burrows, M. L., and Wobber, E. 1992. Authentication in distributed systems: theory and practice. *ACM Transactions on Computer Systems* 10 (4), 265-310.
13. Li, M., Wang, H., Plank, A. ABDM: An Extended Flexible Delegation Model in RBAC. *Accepted by IEEE-CIT* 2008.
14. Li, M., Wang, H., Plank, A. Algorithms for advanced permission-role relationship in RBAC. *Submitted to ACISP* 2008.
15. Li, N., Feigenbaum, J., and Grosof, B. N. 1999. A logic-based knowledge representation for authorization with delegation (extended abstract). *In Proceeding 12th intl. IEEE Computer SecurityFoundations Workshop*, Italy, 162-174.

16. Li, N. and Grosof, B. N. 2000. A practically implementation and tractable delegation logic. *IEEE Symposium on Security and Privacy.* May, 27-42.
17. Liebrand, M., Ellis, H., Phillips, C., AND Ting, T. C. 2002. Role delegation for a distributed, unified *RBAC*/MAC. *In Proceedings of Sixteenth Annual IFIP WG 11.3 Working Conference* on Data and Application Security King's College, University of Cambridge, UK July, 87-96.
18. Mcnamara, C. 1997. Basics of delegating. http://www.mapnp.org/library/guiding /delegate/basics.htm.
19. Sandhu, R., Coyne, E., Feinstein, H., and Ouman, C. 1996. Role-based access control model. *IEEEComputer* 29, 2(February). WIELEMAKER, J. SWI-Prolog. http://www.swi.psy.uva.nl/projects/SWI-Prolog/.
20. Wang, H., Cao, J., Zhang, Y., and Varadharajan, V. 2003. Achieving Secure and Flexible M-Services Through Tickets, In B.Benatallah and Z. Maamar, Editor, IEEE Transactions Special issue on M-Services. *IEEE Transactions on Systems, Man, and Cybernetics. Part A*(IEEE03), Vol. 33, Issue: 6, 697- 708.
21. Wang, H., Sun, L., Zhang, Y., and Cao, J. 2005. Authorization Algorithms for the Mobility of User-Role Relationship, *Proceedings of the 28th Australasian Computer Science Conference (ACSC05)*, Australian Computer Society, 167-176.
22. Wang, H., Cao, J., Zhang, Y. 2003. A flexible payment scheme and its permission-role assignment, *Proceedings of the 26th Australasian Computer Science Conference (ACSC03)*, Adelaide, Australia, Vol. 25, No. 1, 189-198.
23. Wang, H., Cao, J., Zhang, Y. 2003. Formal authorization approaches for permission-role assignment using relational algebra operations, *Proceedings of the 14th Australasian Database Conference*(ADC03), Feb. 2-7, 2003, Adelaide, Australia, Vol. 25, No.1, 125-134.
24. Wang, H., Cao, J., Zhang, Y. 2002. Formal Authorization Allocation Approaches for Role-Based Access Control Based on Relational Algebra Operations, *3nd International Conference on Web Information Systems Engineering (WISE02)*, Singapore, 301-312.
25. Wang, H., Cao, J., Zhang, Y. 2001. A Consumer Anonymity Scalable Payment Scheme with Role Based Access Control, *Proceedings of the 2nd International Conference on Web Information Systems Engineering (WISE01)*, Kyoto, Japan, 73-72.
26. Yao, W., Moody, K., and Bacon, J. 2001. A model of OASIS role-based access control and its support for active security. *In Proceedings of ACM Symposium on Access Control Models and Technologies (SACMAT)*, Chantilly, VA, 171-181.
27. Zhang, L., Ahn, G., and Chu, B. 2001.A Rule-based framework for role-based delegation. *In Proceedings of ACM Symposium on Access Control Models and Technologies (SACMAT 2001)*,Chantilly, VA, May 3-4, 153-162.
28. Zhang, L., Ahn, G., and Chu, B. 2002. A role-based delegation framework for healthcare information systems. *In Proceedings of ACM Symposium on Access Control Models and Technologies (SACMAT 2002).* Monterey, CA, June 3-4, 125-134.