A Role-based Access Control Architecture for P2P File-Sharing Systems Using Primary/Backup Strategy

Jianfeng Lu
College of Computer Science and Technology
Huazhong University of Science and Technology
Wuhan 430074, Hubei, P. R. China
lujianfeng @smail.hust.edu.cn

Ruixuan Li
College of Computer Science and Technology,
Huazhong University of Science and Technology,
Wuhan 430074, Hubei, P. R. China
rxli @ hust.edu.cn

Zhengding Lu
College of Computer Science and Technology,
Huazhong University of Science and Technology,
Wuhan 430074, Hubei, P. R. China
zdlu@smail.hust.edu.cn

Xiaopu Ma
College of Computer Science and Technology,
Huazhong University of Science and Technology,
Wuhan 430074, Hubei, P. R. China
xmp@smail.hust.edu.cn

***Abstract*—Nowadays, P2P file-sharing systems have gained a large acceptance among the internet users. However, there has been little relatively work done in access control for P2P networks, where security is a critical requirement for broader applications of the technology not only in the current but also in the future. In this work, a new architecture is presented in this paper, it integrates the aspects of credential, identity and role-based access control policies to provide scalable, efficient and fault-tolerance access control services. It also preserves the decentralized structure of the P2P platform by employing Primary/Backup Strategy, and resolves the two kinds of interoperability conflicts while mapping role from foreign domain to local domain without centralized authority. We believe that the proposed architecture is realistic, secure and preserves P2P decentralized structure.**

Keywords- P2P; Role; Credential; Primary/Backup

I. INTRODUCTION

Nowadays, P2P file-sharing systems have gained a large acceptance among the internet users. A study found that over one million different peers have connected to the network in an 8-day period five years ago [1]. Despite the evolution of P2P to more complicated systems, there has been little relatively work done in access control for P2P networks. It has been suggested that the future development of P2P systems will largely depend on the availability of novel methods for ensuring that peers obtain reliable information on the quality of resources they are receiving [2]. Therefore, it is very necessary to design a comprehensive access control mechanism that is general and flexible enough to reflect and cope with the special access control requirements associated with the P2P file-sharing, especially for specific applications where security is a critical requirement.

As a consequence, a role-based access control architecture for P2P file-sharing systems, it uses Primary/Backup Strategy and integrates the credential, identity and role based access control (RBAC) policies to provide scalable and efficient access control services whilst preserving the decentralized structure of the P2P platform. The primary copy/backup copy (PB) strategy is employed to take the function of fault-tolerant. And the role mapping policy in our architecture is a secure policy to some extent, as it can resolve the two main kinds of interoperability conflicts.

The rest of this paper is organized as follows. Section ⅱ discusses related works. Section ⅲ presents the (primary copy/backup copy super-peer) PBS architecture for P2P file-sharing systems, and discusses why introduce the PB strategy and how to integrate PB strategy into PBS architecture. Section ⅳ introduces PBS architecture how to map roles from foreign domain to local foreign without centralized authority. Section ⅴ gives our concluding remarks.

II. RELATED WORK

The open and unknown characteristics of P2P make it an ideal environment for malicious users to spread unsolicited and harmful content, such as pornography, viruses, or worms. However, as recent experience with P2P networks such as Gnutella [3], Kazaa [4], and Napster [5] show that these systems focus on usability and scalability, rather than security. In this paper, PBS architecture is presented to support a decentralize access control, not only for usability and scalability, but also for security particularly. In the remainder of this section we provide a brief comparison of our scheme with some of the relevant previous works.

Yao and Julita propose a Bayesian network-based trust model in peer-to-peer networks [6], they use Bayesian networks to provide a flexible method to represent differentiated trust and combine different aspects of trust.

Selcuk et al. propose a model that it is architecturally similar to Yao and Julita's one that trust is built on direct experiences and reputation queries [10]. Both of them face similar drawbacks of scalability in that a large database is required for each peer to keep track of all peers it has interacted with these approaches. But this isn't a problem in our approach, as our storage requirements are very much less.

In [7], Marianne Winslett, Charles C. Zhang and Piero A. Bonatti introduced the PeerAccess framework for reasoning about authorization in open distributed systems, and shown how it can be used in reasoning about the behavior of resource owners, their clients, and the Community Authorization Service deployed on supercomputing grids. Zhang and Kindberg [8] introduced an authorization infrastructure in the CoolTown project for flexible and secure access to a group of distributed services in a nomadic computing environment. These models support secure communications and authorization in P2P environments. But their management schemes may not be scalable in large, dynamic P2P environments as they consider identities not only for identification but also access control.

Sandhu and Zhang proposed architecture with trusted computing technology to support peer-to-peer based access control [9]. They also integrate roles into the architecture by using identity and attribute certificates. However, this architecture tightly depended on PKI which makes it too expensive and restrictive for a dynamic, distributed environment. Comparing with it, our PBS architecture is more scalable and supports access control policy in a controlled P2P environment.

III. PBS ARCHITECTURE

A. *overview*

Figure 1 depicts the operational procedures in our primary copy/backup copy super-peer (PBS) architecture (We use dashed lines to denote one-to-one communication relationships, one-directional arrows with plain line to represent message forwarding, The major procedures of the primary copy super-peers are represented in rounded rectangles. two-directional arrows with plain line to connect the PCS with the major procedures). The primary copy super-peer (PCS) takes the function of the server peer, such as the access control decisions, searches for resources and resource management and so on, which can alleviate the overhead of normal peers. The steps of the access control for P2P file-sharing system are described in this Figure.

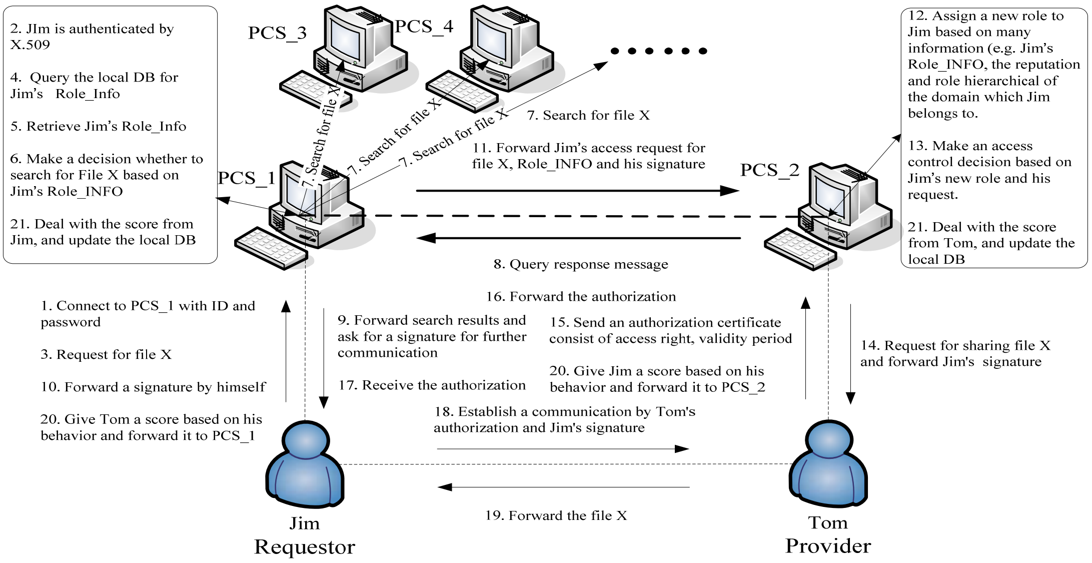
2. JIm is authenticated by X.509
4. Query the local DB for Jim's Role_Info
5. Retrieve Jim's Role_Info
6. Make a decision whether to search for File X based on Jim's Role_INFO
21. Deal with the score from Jim, and update the local DB
PCS_3
PCS_4
PCS_1
PCS_2
7. Search for file X
7. Search for file X
7. Search for file X
7. Search for file X
11. Forward Jim's access request for file X, Role_INFO and his signature
12. Assign a new role to Jim based on many information (e.g. Jim's Role_INFO, the reputation and role hierarchical of the domain which Jim belongs to.
13. Make an access control decision based on Jim's new role and his request.
21. Deal with the score from Tom, and update the local DB
8. Query response message
16. Forward the authorization
1. Connect to PCS_1 with ID and password
3. Request for file X
10. Forward a signature by himself
20. Give Tom a score based on his behavior and forward it to PCS_1
9. Forward search results and ask for a signature for further communication
17. Receive the authorization
15. Send an authorization certificate consist of access right, validity period
20. Give Jim a score based on his behavior and forward it to PCS_2
14. Request for sharing file X and forward Jim's signature
18. Establish a communication by Tom's authorization and Jim's signature
19. Forward the file X
Jim
Requestor
Tom
Provider

Figure 1. The primary copy/backup copy super-peer (PBS) architecture for P2P file-sharing system.

B. *Employing Primary/Backup Strategy into PBS*

The Primary/Backup (PB) strategy has been generally proposed for fault-tolerant dynamic scheduling of tasks in multiprocessor systems [10]. It is a basic strategy that allows multiple copies of a task to be scheduled on different processors. In this section, we will introduce why and how to use PB strategy in our P2P file-sharing architecture.

1) Why introducing the PB strategy and what advantages by introducing PB strategy. In our PBS architecture, the primary copy super-peer takes the function of the server peer, this method simplify the management of leaf-peers. However, it is also even harder to handle security in a dynamic interoperation environment where peers join and leave in an ad-hoc manner. In this case, the PBS architecture uses backup copy super-peer to replace the primary copy super-peer when the latter leaving the network.

2) How to employ PB strategy into PBS architecture. In PBS architecture, the access control decisions, searching for resources and resource management are handled by primary copy super-peers. The normal peers only need to know how to communicate with its super-peer (both PCS and BCS). This case alleviates normal peers' overhead, but increases the overhead of their PCS. Therefore, we choose the PCS and BCS prefer to the peers who have higher process capability, wider bandwidth and more CPU free cycles and so on. Obviously, these should be further research on the policy on how to choose the PCS and BCS in the future. If the PCS fails. The BCS will relay the primary copy super-peer's functions to ensure the system work well. There are three cases may occur as follows:

- The primary copy super-peer fails.
- The backup copy super-peer fails.
- Both the primary and backup copy super-peers fail.

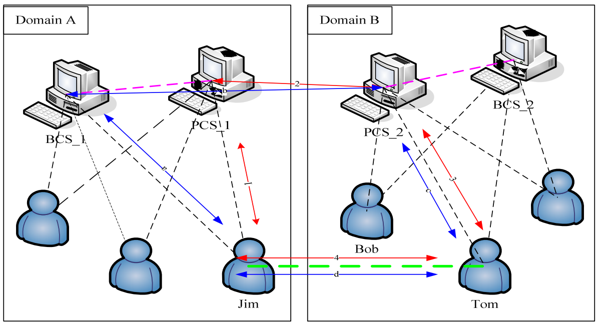

Figure 2. The PB strategy for fault-tolerant in P2P file-sharing system

In Figure 2, we use the plain line to represent one-to-one communicate relationship, two-directional arrows with dashed line to denote one-to-one message transferring, different color of arrows belong to different sessions. For the first case that the PCS (*PCS_1*) fails. Our solution is using BCS (*BCS_1*) replace the PCS's function. *Jim* will establish a communication with *BCS_1* rather than *PCS_1* (step 1). *BCS_1* becomes to be the primary copy super-peer, and it will look for new BCS to keep the system work in normal when *BCS_1* fails. The other two cases is similar to the first one, if the second case occurs that the BCS fails, the PCS will look for another BCS. In the last case, if both the PCS and the BCS in domain A failed. Then the peers in domain A will lost the communication with other peers from foreign domains. Although the general problem of optimal fault-tolerant scheduling of tasks in a multiprocessor system is NP-complete, we also can alleviate the losing by allocating more than a single backup.

IV. Interoperation by Mapping Role without Centralized Authority

In this section we discuss PBS architecture how to map roles from foreign domain to local domain without centralized authority. In PBS architecture, peers will be assigned only one role in local domain based on its attributes (e.g., identity, behavior). When a normal peer wants to access a resource provided by another peer, its PCS will make the access control decision based on the normal peer's role information. We suppose domain *A* holds the *Administrator*, *Author*, *Writer*, *Reader* and *Newcomer* roles, domain *B* holds the *Manager*, *Senior*, *Junior* and *Freshman* roles, and the role hierarchies H_A and H_B for domains *A* and *B* respectively described in Figure 3. One-directional arrow with plain line directed from role *x* to role *y* denotes *x* > *y* which means that *x* is a senior role to *y*. R_A denotes the set of roles in the local domain *A*. In Figure 3, *Administrator* > *Author* > *Writer* > *Reader* > *Newcomer*, and *Manager* > *Senior* > *Junior* > *Freshman*.

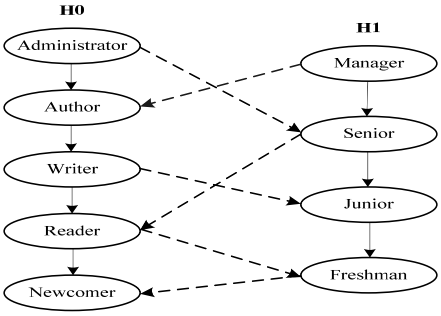

Figure 3. Role hierarchies HA and HB in domain A and B

A. Role mapping polices

Different domains can establish a secure context based on role mapping technique, and applications can make meaningful access control decisions. The associations which from foreign domain to local domain are managed through the primary copy super-peer in local domain, while the other associations from local domain to foreign domain are managed through the PCS in foreign domain. To formalize this, let $Administrator_A$ denotes the role *Administrator* from domain *A*, Let X_B->Y_A implies that the role *X* from the foreign domain *B* will be translated to *Y* in the local domain *A*. E.g., in Figure 3, we have the association from $Administrator_A$ to $Senior_B$. This implies that Administrator from the local domain *A* will be mapped to Senior in the foreign domain *B*.

Consider the scenario that *Jim* is assigned *writer* and wants to access the resource provided by *Tom*, *PCS_1* will forward his request with his role information to *PCS_2*. At first, *PCS_2* will assign a role to *Jim* based on the role association which established by itself based on some information (e.g., the reputation of domain *A*, the contribution by the peers from domain *A*. In this paper, we do not discuss this trust mechanism in that we can use many successful trust mechanisms, such as Gnutella, Kazaa, and Napster. As there is an association of *Writer->Junior*, *Jim* will be assigned *Junior* in foreign domain. Then *Jim*'s access request will be determined based on *Junior*.

B. Security issues

In this section we will resolve the two kinds of conflicts which appear in role mapping policies generally as follows:

1) Cyclic inheritance. It occurs in such interoperation that the cross-domain hierarchy relationship introduce a cycle in the interoperation lattice enabling a subject lower in the access control hierarchy to assume the permissions of a subject higher in the hierarchy. In Figure 4, the NO.1 of one-directional arrow with dashed line denotes $Reader_A$->$Manager_B$ and NO.2 arrow denotes $Junior_B$->$Writer_A$. In this way, $Junior_B$ will be mapped to $Writer_A$, and $Reader_A$ will be mapped to $Manager_B$, this cause a cycle inheritance in the interoperation lattice. Cyclic inheritance conflicts are generally resolved by withdrawing all cross-domain relationships resulting in potential security violation or removing one or more relationships until the violation is corrected. However, there is no centralized security officer in PBS architecture. It's doesn't clear that which cross-domain association should be withdrew. In PBS architecture, the cyclic inheritance was resolved by the communication among domains' administrators. Considering the first association in Figure 4, $Reader_A$->$Manager_B$ and $Writer_A$>$Reader_A$, then $Writer_A$>$Manager_B$ and $Reader_A$>$Junior_B$. Considering the second association, $Junior_B$->$Writer_A$, and $Manager_B$>$Senior_B$>$Junior_B$, and $Writer_A$>$Reader_A$, then $Manager_B$>$Writer_A$ and $Junior_B$>$Reader_A$. In order to resolve these conflicts, the primary copy super-peer in domain *A* will withdraw the first association and establish two new associations that $Writer_A$->$Manager_B$ and $Reader_A$->$Junior_B$, similarly, the PCS in domain B will also withdraw the second association and establish two new associations that $Manager_B$->$Writer_A$ and $Junior_B$->$Reader_A$. The new associations that resolved cyclic inheritance conflicting associations are the NO.3 and NO.4 two-directional arrows with plain line in Figure 4.

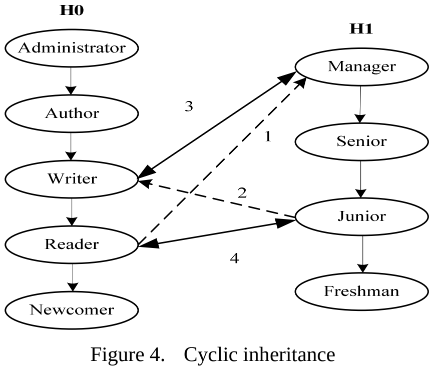

Figure 4. Cyclic inheritance

2) Separation of duties (SoD). SoD prevent two or more subjects from accessing an object that lies within their conflict of interests or disallow a subject from accessing conflicting objects or permissions. There is no SoD constraint in our policy in our PBS architecture. Although there are many roles in a domain, and each role at most has a senior role and a junior role. In this context, the role hierarchies represent the role rank and there is no mutually exclusive role in each domain, therefore, no foreign role can inherit conflict roles while mapping the foreign role into local role. E.g., in Figure 4, $Senior_B$-> $Reader_A$ implies that the Manger from the foreign domain B will be mapped to Author in the local domain *A*. Therefore, $\forall\ r \in R_A$, $Author_A$>r_A, and $Senior_B$>r_A. In this case, $Manager_B$> $Senior_B$> $Reader_A$> $Newcomer_A$. As there is no mutually exclusive role in each domain role hierarchy, a foreign role can't be directly or indirectly mapped to two mutually exclusive roles in local domain. So there is no SoD constraint in our policy.

V. Conclusion

In this paper, we have introduced PBS architecture for P2P file-sharing system, PBS architecture integrates credential-based, identity-based and role-based access control policies not only satisfy the five access control requirements, but also have the merit of fault-tolerance by employing the Primary/Backup strategy in section 4. We employ role mapping technique and resolve the two main types of conflicts which appeared in an interoperation policy generally. In this context, our interoperation policy by mapping role without centralized authority is secure.

Acknowledgment

This research is partially supported by National Natural Science Foundation of China under Grant 60873225, 60773191 and 60403027, National High Technology Research and Development Program of China under Grant 2007AA01Z403. The authors would like to thank the anonymous reviewers for their valuable comments.

References

[1] S. Saroiu, P. Gummadi, and S. Gribbe (2002). A measurement study of peer-to-peer file sharing systems. Technical report UW-CSE-01-06002, University of Washington.

[2] A. Crespo and H. Garcia-Molina. Semantic Overlay Networks. Submitted for publication 2002.

[3] Gnutella, http://www.gnutella.com

[4] Kazaa Media Desktop, http://www.kazaa.com

[5] Napster, http://www.napster.com.

[6] Yao Wang & Julita Vassileva. Bayesian Network Trust Model in Peer-to-Peer Networks, AP2PC 2003, July 2003.

[7] Winslett, M., Zhang, C.C., and Bonatti, P.A.. Access control: PeerAccess: a logic for distributed authorization. Proc. 12th ACM Conf. on Computer and Communications Security, November 2005

[8] Zhang, K., and Kindberg, T.. An authorization infrastructure for nomadic computing. Proc. Seventh ACM Symp. on Access Control Models and Technologies, Monterey, CA, June 2002, pp. 107–113

[9] Sandhu, R., and Zhang, X.. Peer-to-peer access control architecture using trusted computing technology. Proc. 10th ACM Symp. On Access Control Models and Technologies, Stockholm, Sweden, June 2005, pp. 147–158

[10] S. Ghosh, R. Melhem, and D. Mosse. Fault-tolerant scheduling on a hard real-time multiprocessor system. In Proc. International Parallel Processing Symposium, Apr. 1994.