I.J. Information Technology and Computer Science, **2009, 1, 8-15**
Published Online October 2009 in MECS (http://www.mecs-press.org/)

Primary-Backup Access Control Scheme for Securing P2P File-Sharing Systems

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

Ruixuan Li[+], Zhengding Lu, and Xiaopu Ma
College of Computer Science and Technology, Huazhong University of Science and Technology,
Wuhan 430074, Hubei, P. R. China
Email: {rxli, zdlu, xpma}@hust.edu.cn

Abstract—**Peer-to-peer (P2P) file-sharing systems have gained large interests among the internet users. However, wide-scale applications of P2P file-sharing technologies are constrained by the limitations associated with the sophisticated control mechanisms. Moreover, the decentralized and anonymous characteristics of P2P environments make it more difficult to control accesses on the shared resources, especially for using traditional access control methods. To overcome these limitations, we propose a role-based access control architecture for P2P file-sharing systems that supports autonomous decisions and centralized controls. The architecture integrates policies of credential, identity and role-based access control models to provide scalable, efficient and fault-tolerant access control services. Furthermore, we employ the primary-backup (PB) scheme to preserve P2P decentralized structure and peers' autonomy property while enabling collaboration between peers. In particular, we propose a method for setting up interoperating relationships between domains by role mappings and resolve two kinds of interoperability conflicts while mapping roles from foreign domain to local domain without centralized authority. We believe that the proposed architecture is realistic, efficient and can provide controlled communications between peers.**

Index Terms—**P2P, role based access control, credential, primary-backup**

I. Introduction

Peer-to-peer computing (P2P) draws growing interest as a new distributed computing paradigm for its potential to harness "edge" computers (e.g., PCs) and makes their under-utilized resources available to each other. A study found that over one million different peers have connected to the network in an 8-day period in 2002 [1]. In a P2P network, each machine acts both as a client and a server in the network and enables a much more symmetrical and decentralized communications architecture for distributed applications, services and users. This satisfies the requirement that resources should be increasingly made available by being published to other users from a user's machine.

One of the main factors attracting research attention to P2P is the success of P2P file-sharing systems. In the literature, there are many P2P files sharing systems, such as Napster [2], Gnutella [3], KaZaa [4], Chord [5], CAN [6], GridVine [7] and Mercury [8], to name a few. And some systems have been implemented and even widely used. P2P file-sharing systems provide a flexible and universal model for the exchange of information. This has been proven practically by the constantly increasing network traffic volume of P2P systems. Compared with the client-server architecture, P2P-based resource management services have several advantages. For instance, users can manage heterogeneous resources and directly access resources from another's hard discs. With its decentralized nature, A P2P-based resource management architecture can provide higher resource availability. Additionally, P2P technology promises a much better architecture for electronic interactions than the traditional restricted server/client one. In fact, P2P reflects society better than other types of computer architectures [9]. A network traffic measurement at University of Wisconsin has shown that P2P file-sharing system traffic exceeds that of the World Wide Web [10].

Despite the evolution of P2P to more complicated systems, there has been little relatively work done in access control for P2P networks. Wide-scale application of P2P file-sharing is constrained by limitations associated with the especially sophisticated control mechanisms needed between peers. Moreover, the decentralized and anonymous characteristics of P2P environments make the task of controlling access to sharing information more difficult, which cannot be done by traditional access control methods. However, most P2P file-sharing systems give all peers the right to download all files and expect downloading activities to be controlled by human users. They do not provide any mechanisms to defend against malicious users or potential harmful sharing contents. It has been suggested that the future development of P2P systems will largely

Manuscript received January 22, 2009; revised August 25, 2009; accepted September 12, 2009.
[+]Corresponding author: Ruixuan Li.

depend on the availability of novel methods for ensuring that peers obtain reliable information on the quality of resources they are receiving [11]. In this context, the lack of sophisticated access control mechanisms not only in the current but also in the future P2P environment is a serious constraint for broader applications of the technology, especially for specific applications where security is a critical requirement. Consequently, it is very necessary to design a comprehensive access control architecture which is general and flexible enough to reflect and cope with the special access control requirements associated with the P2P file-sharing systems. In this paper, we present an access control architecture for P2P file-sharing systems, which provides P2P users better access control services whilst preserving the decentralized structure of the P2P platform. The main contributions of this paper are as follows:

(1) The architecture is based on role based access control model [12], which also integrates aspects of credential and identity based access control policies to provide scalable, efficient and fault-tolerance access control services. There is no centralized access control authority in our architecture, the proposed architecture can simplify the management process and enhances the secure interoperation. In this context, we imitate the Client/Server access control model where the primary super-peer takes the function of the server peer in our architecture.

(2) We employ the primary-backup scheme to preserve P2P decentralized structure and peers' autonomy property whist enabling collaboration between peers. Considering the dynamism of a large-scale P2P file-sharing systems, where peers from different organizations join and leave P2P community frequently, we employ the PB scheme, even if the primary super-peers or backup super-peers fails, the other peers will take the functions of them. Although the general problem of optimal fault-tolerant scheduling of tasks in a multiprocessor system is NP-complete [13], we also can alleviate the losing by allocating more than a single backup.

(3) We propose a method for setting up interoperating relationships between domains by role mappings and resolve two kinds of interoperability conflicts while mapping role from foreign domain to local domain without centralized authority.

The rest of this paper is organized as follows. Section 2 discusses related works and compares with our architecture to these works. Section 3 identifies five main requirements that the access control architecture for P2P file-sharing systems should support, and proposes a role-based access control architecture for P2P file-sharing systems that supports autonomous decisions and centralized controls. Section 4 employs the PB scheme to preserve P2P decentralized structure and peers' autonomy property in our architecture, includes why introduce the PB scheme and how to integrate PB scheme into our architecture. Section 5 gives the role based interoperation policy without two kinds of interoperability conflicts. Section 6 concludes this paper and presents further directions of research.

II. Related Work

Peer-to-peer file-sharing systems are currently receiving much attention as a means of sharing and distributing information. However, as recent experience with P2P networks such as Gnutella [3], Kazaa [4], and Napster [2] show that these systems focus on usability and scalability, rather than security. However, 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. Therefore, P2P systems focus on usability and scalability is not enough but also need to consider security especially. However, most current peer-to-peer technologies simply focus on sharing services rather than on controls between the peers. Access control is one of the key requirements for the control mechanisms, and this becomes more critical when peers join or leave a community, which represents a set of peers sharing some resources.

In this paper, a role based access control architecture for P2P file-sharing systems is presented to support a decentralize access control, not only for usability and scalability, but also focuses on security particularly. The control of P2P file-sharing systems can be implemented in various ways. The architecture integrates aspects of credential, identity and role-based access control policies to provide scalable, efficient and fault-tolerance access control services. There have been several previous works in each area. In the remainder of this section we provide a brief comparison of our scheme with some of the relevant previous works.

Yao and Julita proposed a Bayesian network-based trust model in peer-to-peer networks [14], they used Bayesian networks to provide a flexible method to represent differentiated trust and combine different aspects of trust. The clients evaluate resource providers' trustworthiness based on its own experiences and recommendations that it queries from other peers in this model, then, a transaction is made with the most trustworthy provider in the list. Selcuk et al. proposed a model that it is architecturally similar to Yao and Julita's one that trust is built on direct experiences and reputation queries [15]. In this model, a peer maintains a binary trust vector for every other peer it has dealt with in the past. 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 is not a problem in our approach, as our storage requirements are very much less.

In [16], Marianne Winslett, Charles C. Zhang and Piero A. Bonatti introduced the Peer Access 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 [17] introduced an authorization infrastructure in the CoolTown project for flexible and secure access to a group of distributed services in a nomadic computing environment. In such systems, users are able to obtain authorization by first receiving a credential from a

'Lobby' service. Their protocol helps maintain the user's privacy. These models support secure communications and authorization in P2P environments. However, identification of each peer (i.e. authentication) is still needed in any system in order to provide the security principle of accountability, which is required in all existing non-public systems. Their management schemes consider identities not only for identification but also access control, thus they may not be scalable in large, dynamic P2P environments as they consider identities not only for identification but also access control.

The RBAC model is extended to support access control in a controlled P2P environment [18]. The environment contains a manager who facilitates provisioning capacities for use of resources and control usage of resource on behalf of a peer. Sandhu and Zhang proposed architecture with trusted computing technology to support peer-to-peer based access control [19]. Their approach considers the integrity and trust of platforms and applications that are used by a user to access an object, which is vulnerable from increasing software-based attacks in client platforms. 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 architecture is more scalable and supports access control policy in a controlled P2P environment.

III. System Design

We first introduce five main requirements that an access control mechanism for P2P file-sharing systems should support, and give our strategies to meet these requirements in our role based access control architecture. Then we propose a role-based access control architecture for P2P file-sharing systems that supports autonomous decisions and centralized controls.

A. Access Control Requirements and Corresponding Strategies

(1) *No centralized access control authority.* Decentralized P2P file-sharing applications lack a centralized authority which access control policies can be stored and evaluated. Therefore, access control mechanisms must take this characteristic into account, the system should be self-policing. That is, the shared ethics of the user population are defined and enforced by the peers themselves but not by some central authority. Our proposed architecture focuses on "pure" P2P architecture. We look for several super-peers to act as "servers" based on peer's information (e.g., capability, credit, bandwidth and so on) as Kazaa system. In this way, this strategy preserves the P2P's decentralized property.

(2) *Differentiate peers.* Generally, there is no true link between a peer and its identity in a public P2P network because of P2P's anonymity property. In this case, an access control mechanism for P2P file-sharing systems must provide a mechanism for a host peer to classify users and assign each user different access rights. Each peer is assigned a role in PBS architecture to differentiate peers based on their behavior in the P2P network. Suppose a peer *A* wants to access a file provided by another peer *B*, *A* should send the resource request to its primary super-peer. If a primary super-peer has the resource, it makes the access control decision based on the requesting peer's privilege. Otherwise, it forwards this resource request to other super-peers which it connected.

(3) *Encourage sharing resources and new peers to join.* The P2P file-sharing system can provide higher resource availability because many peers join the network and share their resource. Access control mechanisms need to give peers the ability to control access to their files it must still encourage them to share their files. In our proposed architecture, each peer will be assigned a role. For example, a *Guest* role will be assigned to a new coming peer. And the role updating are based on peers' behavior, one of the key factor is the contribution that the volume of shared resource measured in megabytes which the peer transfers. In this sense, it resembles a payment scheme (e.g., [20], [21]), where users have to make some contribution in exchange for the benefit they receive from the network.

(4) *Robust to both malicious peer and harmful digital content.* The dynamism and anonymity of a large-scale P2P environment makes it an ideal environment for malicious users cheating and spreading harmful digital content. Access control mechanisms should support mechanisms to limit such malicious spreading, harmful digital content and punish those who are responsible for it. Our proposed architecture can decrease this type building reputation and then milking the system by "slow-to-increase" and "fast-to-decrease" policy. It will need consistent good behavior over a series of sessions.

(5) *Minimize peers' overhead.* In our approach, generic control functions such as searching and indexing are performed by the primary super-peers, while the data path is on the P2P networks. This makes the normal peers' platform thinner. There are many Primary-Backup super-peers join the network to facilitate better scalability and performance.

B. A Role Based Access Control Architecture for P2P File-sharing systems

A P2P computing environment can be an ideal platform for file-sharing services in an organization if it provides trust mechanisms. Technically, we can use the identity-based access control mechanism for this purpose. However, for a large system that supports many peers from different organizations, the identity-based access control mechanism is inefficient and too complicated to manage, since the direct mapping between peers and privileges is transitory. Therefore, we should consider more efficient and secure access control mechanisms for P2P file sharing systems. This becomes more serious when the system is very large. To solve this problem, we separate the mapping between peers and privileges through common job functionalities such as roles. The primary super-peer takes the function of the server peer, such as the access control decisions, searching for resources and resource management and so on, this can alleviate the overhead of normal peers. The remainder of this section discusses the steps involved in a P2P interaction based on our proposed architectures. As we employ the PB scheme in our architecture to preserve P2P decentralized structure and peers' autonomy property whist enabling collaboration between peers. We call the proposed architecture PBS (primary-backup super-peer) architecture for short. 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 super-peers are represented in rounded rectangles. Two-directional arrows with plain line connect the PCS with the major procedures.

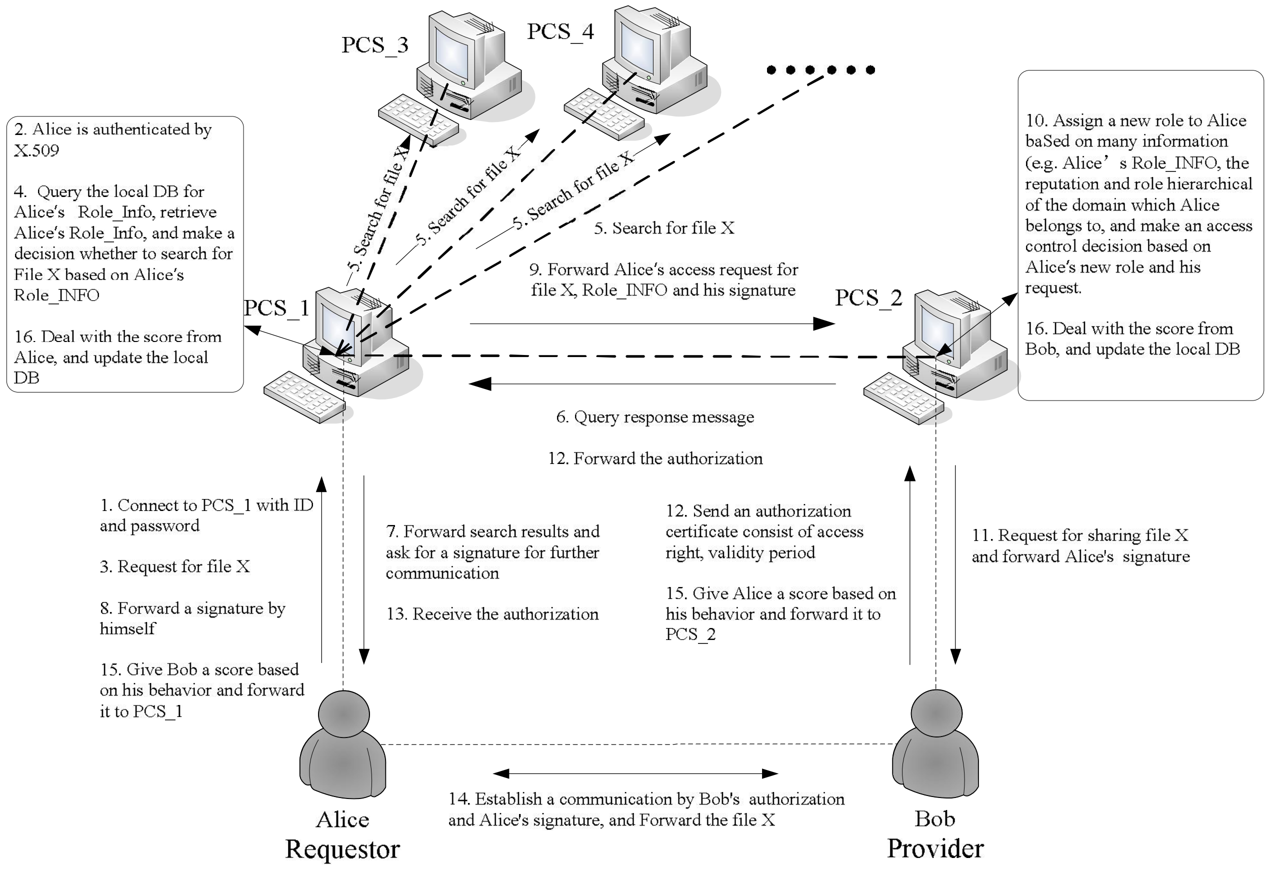
PCS_3
PCS_4
PCS_1
PCS_2
2. Alice is authenticated by X.509
4. Query the local DB for Alice's Role_Info, retrieve Alice's Role_Info, and make a decision whether to search for File X based on Alice's Role_INFO
16. Deal with the score from Alice, and update the local DB
5. Search for file X
5. Search for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

Figure 1. The Primary-Backup Super-peer (PBS) architecture for P2P file-sharing system.

Initially, suppose a resource requestor (Alice) wants to access file X, but does not know initially which peer has this file. Then Alice connects to his current primary super-peer PCS_1 with his ID and password (step 1). The PCS_1 performs authentication via X.509 certificate (step 2). While the authentication is valid, Alice is permitted to request for file X through PCS_1 (step 3). Subsequently, PCS_1 queries the local DB for Alice's Role_Info, retrieves Alice's Role_Info and makes a decision whether allow Alice to search for file X based on Alice's Role_Info (step 4). PCS_1 acts on behalf of its leaf-nodes (Alice), searches for resources in the P2P environment (step 5), receives search queries from PCS_2 who maintains the resource availability tables for their leaf-nodes and forwards the search result to Alice (step 6, 7). The search results show the list of peers who has file X, and optional description about the providers such as policies, reputation, quality of service and so on. Alice considers the resource providers' quality of service and selects one of those resource providers. When Alice requests access to file X from Bob, he forwards his access request to PCS_1 with his signature for further communication (step 8). PCS_1 forwards Alice's access request as well as his ROLE_INFO and signature to PCS_2 (step 9). Then, PCS_2 assigns a new role to Alice based on role mapping which we will discuss in the next section, If Alice's access request is allowed (step 10), PCS_2 requests file X from Bob and forwards Alice's signature to Bob (step 11, 12). If Bob permits Alice to access file X, he will send an authorization certificate consist of access right and validity period (step 12). PCS_1 and PCS_2 forward the authorization to Alice (step 12, 13). After receive the authorization from PCS_1, Alice establishes a communication with Bob by Bob's authorization and Alice's signature, Alice can directly access the file X provided by Bob (step 14). After the interaction, both Alice and Bob give each other a score and forward to their primary super-peer (step 15). Lastly, PCS_1 and PCS_2 deal process these scores and update their database (step 16).

IV . EMPLOYING PRIMARY-BACKUP SCHEME INTO ACCESS CONTROL ARCHITECTURE

The Primary-Backup scheme has been generally proposed for fault-tolerant dynamic scheduling of tasks in multiprocessor systems [22]. 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 employ PB scheme in our P2P file-sharing architecture.

A. Why employing the PB scheme

In traditional C/S models, sensitive objects and policy enforcements can be located on server side, the client generally trusts the server. This gives a central location at which access control policies can be stored and evaluated, obviously simplifies the management. In this paper, we introduce PBS architecture which supports autonomous decisions and centralized controls. The primary super-peer takes the function of the server peer, this method simplify the management of leaf-peers. As is widely known, in a P2P environment, there is no concept of a dedicated centralized server to provide clients with requested resources. Instead, every peer or participant in the system acts as both client and as server, depending on the context. Considering the dynamism of a large-scale P2P environment, where peers from different organizations join and leave P2P communities frequently, the increasing complexity hinders content management. Moreover, security access control does not come easy as it opens the way for several security and privacy breaches. Security is hard to achieve in a “pure” P2P environment, the challenge lies in the fact that the each peer could has heterogeneous security constraints and this adds to the problem. In this way, it is needed a scheme to support fault-tolerant when some peers leave P2P communities. The PBS architecture uses backup super-peer to replace the primary super-peer when the latter leaving the network.

B. How to integrate PB scheme into PBS architecture

In PBS architecture, the access control decisions, searching for resources and resource management are handled by primary super-peers. The normal peers only need to know how to communicate with its current super-peer (both PCS and BCS). This 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, there should be further research on the policy of how to choose the PCS and BCS in the future. If the PCS fails, e.g., it is out of on-line or breaks down. The BCS will replace the primary super-peer’s functions to ensure the system work well.

Considering the dynamic characteristics of P2P networks, peers from different organizations join and leave P2P communication frequently. In this case, we can’t assume the super-peers working forever and never fail, that is irrationality. Fortunately, the failure of normal peers doesn’t affect our system working as usual. We only need to consider the failure of super-peers. There are three cases may occur as follows:

(1) The primary super-peer fails.
(2) The backup super-peer fails.
(3) Both of the primary super-peer and the backup super-peer fail.

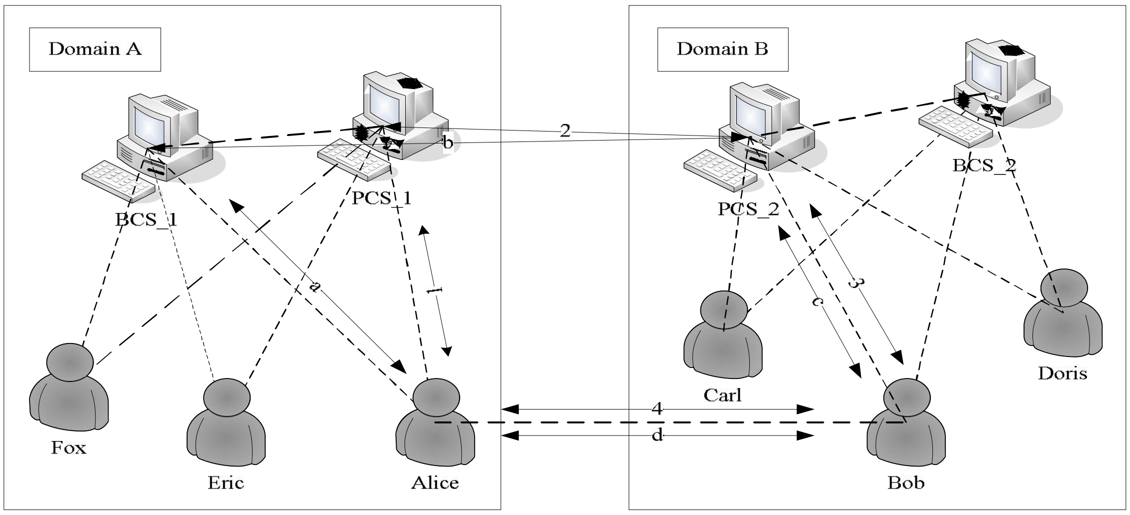

Figure 2. The PB scheme for fault-tolerant in P2P file-sharing system

Let us take the same service that we considered in Section 3. 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, the arrows with different number belong to different sessions. Initially, Alice wants to access a resource provided by another peer (Bob), he connects to his current primary super-peer PCS_1, and establishes a communication with PCS_1 (step 1). After a successful authentication process by PCS_1, Alice sends the resource request information to PCS_1, and PCS_1 forwards this information to PCS_2 which it is currently connected, and establishes a communicate relationship with PCS_2 (step 2). Bob has the required file that Alice looking for, at last Alice establishes a communication with Bob by PCS_1 and PCS_2. In this case, Alice can directly access the required file that provide by Bob (step3, 4).

In the first case, assume that PCS_1 fails, our solution is using BCS (BCS_1) replace the PCS's function. Alice will establish a communication with BCS_1 rather than PCS_1 (step 1). The rest steps are similar to above. BCS_1 becomes to be the primary 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.

V. Interoperation by Mapping Role Without Centralized Authority

Due to the extensive use of the RBAC model and its variants, interoperation based on RBAC is of theoretical and practical importance. A typical method is to create cross-domain role mappings between domains and resolve security breaches arising from the interoperation. Entities in one domain are permitted to access resources of other domains through these mappings. In this section we employ role mapping technique to map roles from foreign domain to local domain without centralized authority in PBS architecture. Apu Kapadia et.al proposed IRBAC 2000 to establish a flexible policy for dynamic role translation [23]. He assumes that there exists a role editor which is used to set up associations between these hierarchies. However, there is no centralized authority in pure P2P environment. Considering the characteristics of P2P file-sharing system, there is no complicated operation besides accessing files if the requisition is permitted. In this way, each peer can be assigned to only one role in local domain, it should be assigned a foreign role in foreign domain when it requests to access the resource provided by another peer in foreign domain. In PBS architecture, peers will be assigned to only one role based on its information (e.g., identity, behavior). When a normal peer requests to access a resource provided by another peer, the PCS makes the access control decision based on the normal peer's role information.

We suppose that 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* f *y* which means that *x* is higher than *y* in the hierarchy, or *x* is the ancestor of *y*. R_A denotes the set of roles in the local domain *A*. In Figure 3, *Administrator* f *Author* f *Writer* f *Reader* f *Newcomer*, and *Manager* f *Senior* f *Junior* f *Freshman*.

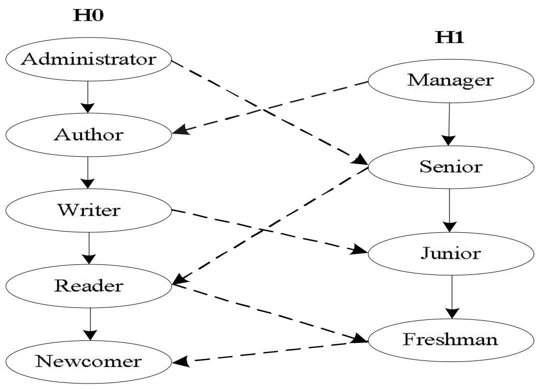

Figure 3. Role hierarchies HA and HB in domain *A* and *B*

A. Role mapping polices

We propose a policy framework that facilitates secure interoperability between foreign domain and local domain. The policy framework works with a set of associations between the local and foreign role hierarchies, in this way, foreign roles can now be translated into local roles, which are understandable to local entities. And applications can make meaningful access control decisions. The associations which from foreign domain to local domain are managed through the primary super-peer in local domain, while the other associations which 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 a 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 Alice is assigned *writer* and wants to access the resource provided by Bob, his current PCS will forward his request with his role information to Bob's current PCS. At first, Bob's PCS will assign a new role to Alice based on the role associations 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 section, we do not present a trust mechanism in that we can use many successful trust mechanisms, such as [2, 3, 4]. As there is an association of *Writer* a *Junior*, Alice will be assigned *Junior* in foreign domain. If the resource provider (Bob) required hat any user must have a role that senior to *Senior* can be permitted to access, Alice's requesting will be denied. Otherwise, the access request will be permitted.

B. Security issues

A key challenge for secure interoperation is the resolution of conflicts that may arise among the RBAC policies of individual domains. In this section, we resolve two kinds of conflicts arising from the interoperation based on RBAC is of theoretical and practical importance.

It is easy to imagine a situation in which a particular association conflicts with another association. Several research efforts give their major emphasis on the detection and resolution of conflict in interoperation policies [24]. Based on their works, Conflicts appearing in an interoperation policy can be divided into two types. We analyze these conflicts and give our mechanisms to resolve them as follows:

(1) *Cyclic inheritance* 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 $Reader_A$ a $Manager_B$ and NO.2 arrow denotes $Junior_B$ a $Writer_A$. In this way, $Junior_B$ will be mapped to $Writer_A$, and $Reader_A$ will be mapped to $Manager_B$. There exists a cycle in the interoperation lattice, it may result in a security hazard in which the security officer may grant a foreign role higher access without meaning to do so. Interoperation 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 is not clear that which cross-domain association should be withdrew. In this way, the cyclic inheritance was resolved by the communication among domains' administrators in PBS architecture. Considering the first association in Figure 4, $Reader_A$ a $Manager_B$ denotes that $Reader_A$ will be mapped to $Manager_B$, as $Writer_A$ f $Reader_A$, then *WriterA* f $Manager_B$ and $Reader_A$ f $Junior_B$. Considering the second association, $Junior_B$ a $Writer_A$ denotes that $Junior_B$ will be translated to $Writer_A$, as $Manager_B$ f $senior_B$ f $junior_B$, and $Writer_A$ f $Reader_A$, thus $Manager_B$ f $Writer_A$ and $Junior_B$ f $Reader_A$. In order to resolve these conflicts, the PCS in domain *A* will withdraw the first association and establish a new association that $Writer_A \leftrightarrow Manager_B$ which has the same meaning as both of $Writer_A$ a $Manager_B$ and $Reader_A$ a $Junior_B$, similarly, the PCS in domain B will also withdraw the second association and establish a new association that $Manager_B \leftrightarrow Writer_A$. 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.

(2) *Separation of duties (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. Considering the characteristics of P2P file-sharing systems, we assume that 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, $Senior_B$ a $Reader_A$ implies that the *Manger* from the foreign domain *B* will be mapped to *Author* in the local domain *A*. Therefore, for any $r \in R_A$, $Author_A$ f r_A, and $Senior_B$ f r_A. In this case, $Manager_B$ f $Senior_B$ f $Reader_A$ f $Newcomer_A$. As there is no mutually exclusive role in each domain role hierarchy, a foreign role can not be directly or indirectly mapped to two mutually exclusive roles in local domain. Therefore, there is no SoD constraint in our policy.

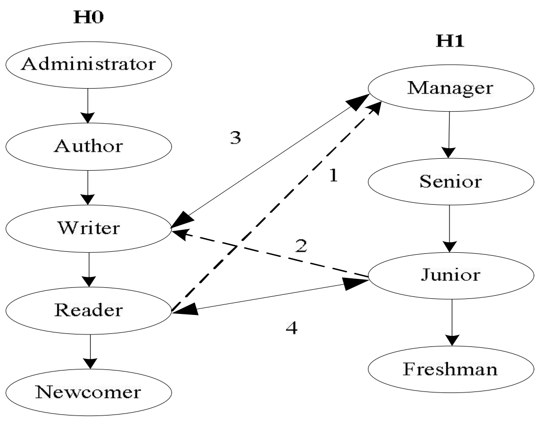

Figure 4. Cyclic inheritance

VI .CONCLUSION AND FUTURE WORK

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 which we identified in Section 3, but also have the merits of fault-tolerance by employing the Primary-Backup scheme in Section 4. Besides, it absorbs many advantages from C/S model for P2P file-sharing systems. We employ role mapping technique to map roles from foreign domain to local domain without centralized authority in PBS architecture, and resolve two kinds of conflicts arising from the interoperation based on RBAC is of theoretical and practical importance. In this context, our role mapping policy is a secure policy to some extent.

Our future work will include refinement and implementation of the proposed architecture in a real P2P file-sharing system. In Section 5, we proposed a role translation policy based on credential which makes up of direct trust, indirect trust, direct contribution, indirect contribution and so on. Following this, we will plan to extend this scheme to develop a comprehensive trust based access approach for P2P file-sharing system.

ACKNOWLEDGMENT

This research is partially supported by National Natural Science Foundation of China under Grant 60873225, 60773191 and 70771043, National High Technology Research and Development Program of China under Grant 2007AA01Z403, and Open Foundation of State Key Laboratory of Software Engineering under Grant SKLSE20080718.

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] Napster, http://www.napster.com.
[3] Gnutella, http://www.gnutella.com
[4] Kazaa Media Desktop, http://www.kazaa.com
[5] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," in ACM SIGCOMM, August 2001, pp. 149–160.
[6] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Schenker, "A scalable content-addressable network," in SIGCOMM. ACM Press, 2001, pp. 161–172.
[7] K. Aberer, P. Cudr´e-Mauroux, M. Hauswirth, and T. V. Pelt, "Gridvine: Building internet-scale semantic overlay networks," in International Semantic Web Conference, 2004, pp. 107–121.
[8] A. R. Bharambe, M. Agrawal, and S. Seshan, "Mercury: supporting scalable multi-attribute range queries," in SIGCOMM, 2004.
[9] D. Clark, "Face-to-Face with Peer-to-Peer networking," IEEE computer, Jan 2001.
[10] Evangelos Markatos, "Tracing a large-scale Peer to Peer System," an hour in the life of Gnutella, CCGrid'2002, May 2002.
[11] A. Crespo and H. Garcia-Molina, "Semantic Overlay Networks," Submitted for publication 2002.
[12] ANSI. American National Standard for Information Technology-Role Based Access Control. ANSI INCITS 359-2004, 2004.
[13] M.R. Garey and D.S. Johnson, "Computers and intractability, a guide to the theory of NP-completeness," W.H. Free-man Company, San Francisco, 1979.
[14] Y. Wang and J. Vassileva, "Bayesian Network Trust Model in Peer-to-Peer Networks," AP2PC 2003, July 2003.
[15] Selcuk, Ali Aydin and Uzun, Ersin and Pariente, Mark R, "A Reputation-Based Trust Management System for P2P Networks," CCGRID2004, April 2004.
[16] 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.
[17] K. Zhang, and T.Kindberg, "An authorization infrastructure for nomadic computing," Proc. Seventh ACM Symp. on Access Control Models and Technologies, Monterey, CA, June 2002, pp. 107–113.
[18] Joon S. Park & Junseok Hwang, "Role-based Access Control for Collaborative Enterprise In P2P Computing Environment," SACMAT, Jun 2003.
[19] R. Sandhu, and X. Zhang, "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
[20] K. Lai, M. Feldman, J. Chuang, and I. Stoica, "Incentives for Cooperation in Peer-to-Peer Networks," Workshop on Economics of Peer-to-Peer Systems, June, 2003.
[21] R. Buyya, D. Abramson, J. Giddy, and H. Stockinger, "Economic Models for Resource Management and Scheduling in Grid Computing," CCPE Journal, May 2002.
[22] 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.
[23] Apu Kapadia, Jalal Al2Muhtadi, R1 Campbell, "IRBAC 2000: Secure interoperability using dynamic role translation," University of Illinois, Technical Report: UIUCDCS-R-2000-2162 , 2000.
[24] E. Lupu and M. Sloman, "Conflicts in Policy-Based Distributed Systems Management," IEEE Trans. Software Eng., vol 25, no. 6, pp. 852-869, Nov. 1999.

Jianfeng Lu received the B.S. degree from College of Computer Science and Technology from Wuhan University of Science and Technology in 2005. He is a PhD candidate in the Intelligent and Distributed Computing Lab, College of Computer Science and Technology, Huazhong University of Science and Technology, and is expected to graduate in June 2009. His research interests include access control, security analysis, multidomain interoperation, and separation of duty.

Ruixuan Li received the B.S., M.S., and Ph.D. degrees from College of Computer Science and Technology from Huazhong University of Science and Technology in 1997, 2000, and 2004, respectively. Since 2004, He has been an Associate Professor of College of Computer Science and Technology from Huazhong University of Science and Technology. His research interests include distributed system security, heterogeneous information integration, peer-to-peer computing, web data management, semantic web and ontology.

Zhengding Lu is currently a professor in College of Computer Science and Technology at Huazhong University of Science and Technology, and is the director of the Intelligent and Distributed Computing Laboratory. His research interests include distributed computing, distributed database, distributed system security, and multimedia information systems.

Xiaopu Ma received his M.S. degree in School of Computer Science and Engineering from University of Electronic Science and Technology of China in 2004. Now he is a Ph.D. candidate in the Intelligent and Distributed Computing Lab, College of Computer Science and Technology, Huazhong University of Science and Technology. His research interests include access control, multidomain interoperation, and distributed system security.