26th IEEE International Symposium on Reliable Distributed Systems

PEACE-VO: A Secure Policy-Enabled Collaboration Framework for Virtual Organizations

Jianxin Li, Jinpeng Huai, Chunming Hu
School of Computer Science & Engineering, Beihang University,Beijing, China
{lijx,huaijp,hucm}@act.buaa.edu.cn

Abstract

The increasing complexity and dynamics of grid environments have posed great challenges for secure and privacy-preserving collaboration in a virtual organization. In this paper, we propose PEACE-VO, a secure policy-enabled collaboration framework for virtual organizations. PEACE-VO employs role mapping to define trust relationships across autonomous domains. Nevertheless, a critical issue emerges when the system applies role mapping, which is potential policy conflict in a local domain. We first develop two concepts to depict such possible conflicts within the collaboration policy. Next, we propose a fully distributed evaluation algorithm to detect potential policy conflicts, which does not require domains to disclose their full local security policies and therefore preserves critical domain privacy. Finally, we design two dedicated protocols for virtual organization management and authorization services, respectively. We have successfully implemented the PEACE-VO framework with two fundamental protocols, i.e., VO management protocol and service authorization protocol, in the CROWN Grid. Comprehensive experimental study shows our approach is scalable and efficient.

1. Introduction

Virtual organizations (VOs) [1] are becoming prevalent today because of increasing demand for distributed resource sharing and collaboration among multiple autonomous domains. A virtual organization is of a set of entities, such as resources, services, and users. These entities may belong to different autonomous domains, which collaborate in order to complete certain tasks. VOs have been adopted in many applications such as dynamic enterprises, on-demand computing, on-demand services providers, outsourcing business processes, business-to-business collaboration, and so on.

However, although the role that virtual organizations play is becoming noticeable, how to establish a secure collaboration environment for virtual organizations has becoming a fundamental and challenging problem. During a collaboration process, new collaborators (i.e., domains) may join and some existing ones may depart dynamically. Furthermore, a collaborator normally does not want to disclose all of its security policies for purposes of privacy protection. A number of approaches [2-5] have been proposed to enforce security management of grid virtual organizations. These approaches fall into two categories, *centralized authority based* and *trust federation based*. The former category introduces a new centralized authority, which assigns new identities or attributes to users or services in a virtual organization. The latter category defines trust relationships through cross-domain identity or attribute mapping. After careful investigation, we find that these approaches suffer from several problems.

Firstly, they do not fully accommodate the natural dynamics of virtual organizations as collaborators join or leave frequently. If entirely new policies are created for every virtual organization, a heavy management burden is introduced. For example, it is overwhelming if the system needs to assign an identity to every potential user or service. It becomes worse when the access control policy for related services need to be updated accordingly to any change of virtual organization policies.

Secondly, there is no mechanism for privacy preservation of autonomous domains, which is particularly important for open distributed environments, whereas few considerations have been made for protecting critical privacy information of local domain polices. As we know, the capability of a common goal-driven virtual organization is provided by all the participating domains, so protecting the privacy information while guaranteeing the sufficient collaboration is an essential requirement.

Thirdly, most importantly, there exist possible conflicts of collaboration policies. But this serious problem is rarely recognized in existing studies on security management of grid virtual organizations. Many solutions take advantage of identity mapping (or delegation) policies to build trust among domains. This precisely leads to a potential security threat to domain policies. For example, a user with lower privileges may acquire a higher privilege through an identity mapping loop. We illustrate this problem in Figure1. In the RBAC (role-based access control) model [6], the role is associated

1060-9857/07 $25.00 © 2007 IEEE
DOI 10.1109/SRDS.2007.12

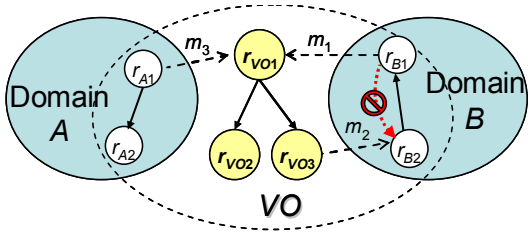

Figure 1. Example of a policy conflict

with permissions. Relation $r_{A2} \preceq r_{A1}$ (role inheritance relation) indicates that if a user *u* is a member of role r_{A1}, then *u* acquires the permissions of role r_{A2}, we also denote this relation as (r_{A1}, r_{A2}) in this paper. As shown in Figure 1, domain *A* and domain *B* form a virtual organization *VO* with role hierarchies $r_{B1} \preceq r_{B2}$ and $r_{VO3} \preceq r_{VO1}$. The administrator of virtual organization defines two task collaboration policies. The first one is $m_1 : (r_{B1}, r_{VO1})$, which means a user of role r_{B1} in domain *B* has the permissions of r_{VO1},. The second is $m_2 : (r_{VO3}, r_{B2})$. Through the two policies we can derive (r_{B1}, r_{B2}). This relation will violate the role hierarchy of domain *B*, and hence such conflict needs to be corrected by *VO*. The policy conflict scenario in this example is simple and not difficult to be found. Nevertheless, as a real grid environment may have a large number of participating domains and role mappings are large, it becomes impossible to manually identify and correct these dangerous risks.

The scale of grids is becoming larger and the complexity of grid applications is also getting higher. Therefore, there is an increasing demand for flexible and secure management of virtual organizations. It has been an urgent yet challenging issue to ensure secure collaboration, while preserving the privacy of domain policies for virtual organizations.

In this paper we propose PEACE-VO (Secure **P**olicy-**E**n**A**bled **C**ollaboration Fram**E**work for Grid **V**irtual **O**rganizations). Our solution considers both security and privacy, which clearly distinguish our solution from existing approaches. We have made the following contributions.

- After identifying the key issues, we propose a novel secure collaboration framework based on security policy. The role mapping mechanism is employed to define trust relationships across domains. We create two concepts, *implicit policy conflict* and *explicit policy conflict* to depict the potential conditions against virtual organization collaboration policies.
- We design a fully distributed algorithm to detect any potential conflict and hence ensure policy coherence in each local domain. The complexity of this algorithm is theoretically analyzed. In addition, our approach is able to protect critical domain privacy, since it does not require domains to disclose their full security policies.
- We have successfully implemented the PEACE-VO framework with two fundamental protocols, i.e., VO management protocol and service authorization protocol in the CROWN Grid [7]. Comprehensive experimental study shows our approach is scalable and efficient.

The rest of this paper is organized as follows. We discuss related work in Section 2. Section 3 elaborates the design of PEACE-VO and the distributed policy evaluation algorithm. We detail the virtual organization management and authorization protocols in Section 4. Section 5 presents the performance evaluation results and their analysis. Finally, we conclude the paper in Section 6.

2. Related work

Security management for grid virtual organizations has been widely studied. Existing systems fall into two categories: *centralized authority based* and *trust federation based*. The former category includes CAS (Community Authorization Service) [2], VOMS (Virtual Organization Management Service) [5] and TrustCoM [8]. The second category includes GridShib [9],CROWN-CredFed [3, 4], Liberty [10] and WS-Federation [11]. In addition, secure interoperation [12-15] is an important concept introduced in the research area of secure collaboration of a multi-domain environment.

2.1. Grid VO security mechanisms

CAS, which is built on the Globus Toolkit middleware for Grid Computing, allows resource providers to delegate their authority to *VO Server* whilst maintaining ultimate control over their resources. Generally, the resource policy is composed by virtual organization policy and domain policy. Thus, the policy is agreed by all the participating domains. VOMS provides many user attributes within the virtual organization: his/her groups, roles and capabilities. CAS and VOMS are two popular representative systems in grids, which are architecturally similar in that both issue attribute assertions to a user. The user presents them to a target service for the purpose of obtaining virtual organization issued permissions. TrustCom and GOLD (Grid-based Information Models to Support the Rapid Innovation of New High Value-Added Chemicals) [16] also develop similar security management mechanisms.

In addition, GridShib is a project that integrates the Shibboleth infrastructure with the Globus Toolkit to provide attribute-based authorization for distributed scientific communities through trust federation. CROWN-CredFed provides identity mapping and credential conversion service to establish trust relationships among

heterogeneous security domains. WS-Federation is a specification that defines mechanisms to allow different security domains to federate, such that authorized access to resources managed in one domain can be provided to entities whose identities and attributes are managed in other domains. This includes mechanisms for brokering of identity, attribute and security assertions between domains [11].

In short, security management of a virtual organization has been widely studied. Especially, these centralized authority based systems have a performance advantage during the service authorization. However, there are still many limitations, primarily reflected in two aspects: dynamism and autonomy. On the one hand, the approaches like CAS and VOMS are not flexible due to the facts of establishing a new centralized authority and assigning new identity for every user. On the other hand, there is lack of consideration on collaboration policy conflicts, and current grid security systems for virtual organizations conern establishment of collaboration policies with little consideration of the policy conflict issue.

2.2. Secure interoperation

Secure interoperation aims to guarantee the security of collaboration through identity mapping in a multi-domain environment [12, 13]. Dawson et al. [14] present a mediator-based approach to provide secure interoperability for heterogeneous databases. This approach assumes a mandatory access control (MAC) policy, such as the Bell LaPadula policy, which is inflexible and not applicable in many commercial applications. Gong et al. [12] characterize the properties that must be satisfied to compose a global secure policy. In all these approaches, a trusted third party that has a global view of the collaboration environment is required to perform the secure policy composition and integration. To handle this problem, Sheha et al. [13] instead propose a distributed secure interoperability framework for mediator-free collaboration environments, which relies on a secure access path to make authorization as well as conflict checking without having a global view of collaboration policies [13]. The primary problem of this approach is the large latency caused by access path construction during requesting services. Furthermore, these studies have some different assumptions from grid virtual organizations. A grid virtual organization often defines some new roles and policies for common tasks, and a centralized *VO Server* may be used for efficient authorization. Secure interoperation is merely for coordinating existing polices among security domains.

3. Design of PEACE-VO

3.1. Basic concepts and framework

In this paper, we choose RBAC model [6] to describe policies in PEACE-VO, i.e., all domains adopt a RBAC to model security policies. Two reasons account for this decision. First, RBAC is policy-oriented but policy-neutral, which make it suitable for specifying security requirements for a wide range of commercial, medical, and government applications. Second, the RBAC viewpoint is that MAC (Mandatory Access Control) and DAC (Discretionary Access Control) are special examples of policies to configure in a RBAC model.

The basic notations used in this paper are as follows:

Domains: We use A, B, and C, sometimes with subscripts, to denote autonomous domains, and use VO to denote a virtual organization.

Roles: A *role* is denoted by r with subscripts or not, e.g., a role in domain A_1 can be denoted as r_{A1}. R represents a set of roles, and the role set of a domain is referred to as R_i. Note that some roles in a domain should be available to the virtual organization to define role mapping, one of which is denoted by r_i^o. In a virtual organization, a set of new roles R_{VO}, named *task roles*, are created for the common tasks. The whole roles in VO is denoted by R'_{VO}.

As shown in Figure 2, the PEACE-VO framework is essentially an overlay of security policy, where the virtual organization G_{VO} consists of several participating domains, and each of which consists of users and resources. The virtual organization policy manager *VO Server* can be established upon agreement by all domains through negotiation, or simply chosen by the virtual organization creator. All trust relationships, users and services in the virtual organization are ultimately defined by virtual organization collaboration policy.

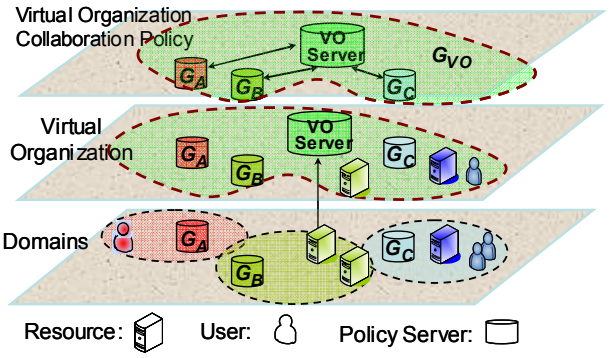

Figure 2. The PEACE-VO framework

In this framework, the centralized *VO Server* makes efficient service authorization for virtual organization users like CAS and VOMS. Next, we formally define several important concepts in PEACE-VO:

Definition1 (Domain Security Policy) [13]**:** The domain security policy is a directed graph $G = \langle R, H \rangle$, where R is a role set of a domain, and H is role hierarchies such that $H \subseteq 2^R$.

Definition2 (Role Mapping Policy): Let R_i and R_j be different role sets of two domains. The role mapping policy is a binary relation M which is a subset of the Cartesian product $R_i \times R_j$, which satisfies $\forall(r_p, r_q) \in M$ $r_p \in R_i$, $r_q \in R_j$ where $i \neq j$. Generally, we denote an element of M as m.

Cross-domain role mapping is a key approach to empowering collaboration among domains [17]. Through role mapping policies, users belonging to a role in one domain can acquire permissions assigned to roles in another domain. In PEACE-VO, we divide role mapping into two categories. One category is *virtual organization role mapping*, which includes every role mapping from a role in a domain to a *task role* in the virtual organization $(r_p, r_{vo}) \in \bigcup_{i=1}^{n} R_i \times R_{VO}$ (the set is denoted as M_{VO}). The other category is *domain role mapping*, which consists of any role mapping from a *task role* in the virtual organization to a role in domains $(r_{VO}, r_q) \in R_{VO} \times R_i$ (the set is denoted as M_i). Furthermore, it is necessary to restrict that some roles in other domains could not be mapped to some appointed roles in a domain. We call such mapping policy as *forbidden role mapping* (denoted as F_j), which satisfies $\forall(r_p, r_q) \in F$ $r_p \in R_i$, $r_q \in R_j$ where $i \neq j$. In several literatures [12, 13], this category of policy is also called *autonomy policy*. In PEACE-VO, if there is a violation of the collaboration policy against F, such a condition is referred to as *explicit policy conflict*. Correspondingly, if there is a violation of the collaboration policy against the role hierarchies H of a domain, such a condition is referred to as *implicit policy conflict*.

It is observed that the expressions of binary relation H and M are the same, but we make two distinct definitions since the former concerns intra-domain collaboration relations, and the latter concerns inter-domain collaboration relations. This is not only much clearer for policy management, but also helpful to design the following algorithm for detecting policy conflicts.

In PEACE-VO, a domain security policy holds the following property.

Property 1: The domain security policy is kept private. This means that most of the local policy G_i, including the *domain role mapping* M_i and *forbidden role mapping* F_i, in a domain need not to be disclosed. Moreover, the authorization policies for domain services are still associated with original domain roles.

3.2. Virtual organization collaboration policy

Definition3 (Virtual Organization Collaboration Policy): Let VO be a virtual organization composed of n domains, the virtual organization collaboration policy is defined as $G_{VO} = \langle R'_{VO}, H'_{VO} \rangle$, where $H'_{VO} = (\bigcup_{i=1}^{n}(H_i \cup M_{Gi}) \cup H_{VO} \cup M_{VO}) - \bigcup_{i=1}^{n} F_i$, R_{VO} is the set of *task roles* of a virtual organization, H_{VO} is the set of role hierarchies on R_{VO}, and M_{VO} is role mapping relation that defined by the virtual organization, i.e., $(r_i, r_{VO}) \subseteq \bigcup_{i=1}^{n} R_i \times R_{VO}$, F_i is the *forbidden role mapping* that every individual G_i defined. The sets of R_{VO}, H_{VO} and M_{VO} together are called the *task policy* of a virtual organization.

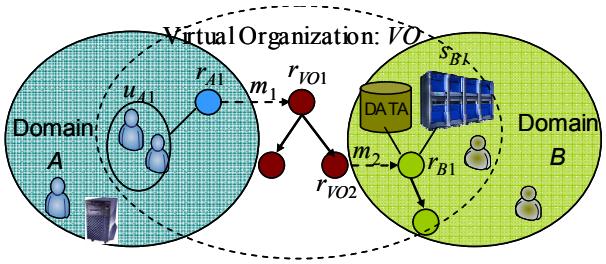

Figure 3. Example of a virtual organization

Subsequently, we give an example to illustrate how the virtual organization collaboration policy is created. As shown in Figure 3, two domains A and B form a virtual organization VO. Domain B wants to share its service s_{B1} with some users from domain A; the related policies involved in the virtual organization are as follows:

1. In domain A, a user u_{A1} is a member of role r_{A1};
2. In VO, role mapping policy set M_{VO} contains a policy m_1:(r_{A1}, r_{VO1}) which represents users belonging to role r_{A1} also acquire the permissions associated with role r_{VO1}, and there is also another role hierarchy $r_{VO2} \preceq r_{VO1}$;
3. In domain B, role mapping policy set M_{GB} contains a policy m_2:(r_{VO2}, r_{B1}) which represents users belonging to role r_{VO2} also acquire the permissions associated with role r_{B1} in domain B, says that role r_{B1} has access permission to service s_{B1}.

As a result, based on these policies in the virtual organization, the user u_{A1} from domain A can access service s_{B1} provided by domain B.

From the above definitions and example, we draw Property 2 and Property 3 that PEACE-VO holds:

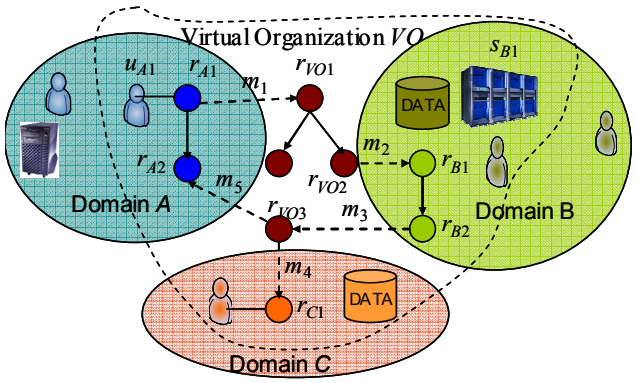

Figure 4. A virtual organization *VO* formed by three domains *A*, *B* and *C*

Property 2: The *task roles* and their *hierarchies* in a virtual organization should be open. Compared with the privacy of a domain security policy, the *task policy* as R_{VO}, H_{VO} and M_{VO} must be open to all the domains. The reasons are two-fold. On the one hand, these sets should be available for domains to create their domain role mapping policy M_{Gi} and to evaluate the security of collaboration policy. On the other hand, these policies are generally created by all participating domains together, and hence their openness is evident.

Property 3: The validity of a role mapping chain is limited. To meet the requirements demonstrated in Property 1, domain security policy may be invisible to other domains. Therefore, role mappings between two domains should not involve roles from the third domain. This means that role mapping policy M_i from *task roles* to domain roles is only valid for service authorization within the local domain. M_i cannot be a part of the role mapping chain for other domains because it may not be completely open.

Figure 4 gives an example of a virtual organization formed by three domains *A*, *B* and *C*. It is easy to observe that the user u_{A1} belonging to role r_{A1} can be mapped to role r_{B1} in domain *B*. Following the relation $r_{B2} \preceq r_{B1}$ and the two mapping policies, $m_3 : (r_{B2}, r_{VO3})$ and $m_4 : (r_{VO3}, r_{C1})$, the role r_{B1} from domain *B* can be mapped to role r_{C1} in domain *C*. In PEACE-VO, however, the user u_{A1} cannot be mapped to role r_{C1} according to the Property 3. Otherwise, this role mapping chain will involve policies from the third domain *B*. Moreover, Property 3 requires that trust relationships between two domains must involve the *task roles* in the virtual organization as a bridge. There are three main aspects to be considered. Firstly, it provides a method to avoid a role mapping loop, which a single domain may not be aware of because the other domains may not disclose enough policies for the purpose of privacy preservation. If Property 3 does not hold, role r_{A1} can be mapped to r_{A2}, i.e., (r_{A1}, r_{A2}), which violates the original role hierarchy $r_{A1} \preceq r_{A2}$. Secondly, the role mapping between domains is simplified, which will benefit the algorithm design for detecting policy conflicts. Finally, the mutual collaboration relationships are defined through the *task roles* in the virtual organization which is additionally an advantage to centralized security audit.

The three properties of PEACE-VO are not only requirements for the security management of virtual organizations but also guidelines for evaluating the security of virtual organization collaboration policy. Based on Property 3, we give the definition of a valid role mapping chain.

Definition 4 (Valid Role Mapping Chain): In PEACE-VO, let $L = (r_0, r_1, ..., r_k)$ be a role mapping chain from role r_0 in domain G_i to role r_k in domain G_j. L is a valid role mapping chain if it satisfies the following conditions:

(C.1) For each $q \in \{0,...,k\}$, **s.t.** $r_k \in R_i$ or $r_k \in R_j$ or $r_k \in R_{VO}$.

(C.2) For every $p, q \in \{1,...,k-1\}$ and $p \le q$, if r_p, $r_q \in R_{VO}$, then each s where $p \le s \le q$ **s.t.** $r_s \in R_{VO}$.

This definition indicates that no additional roles in other domains can be involved, i.e., $(r_0, r_k) \in H_i^+ \circ M_{VO} \circ H_{VO}^+ \circ M_j \circ H_j^+$ (H^+ is a transitive closure set of H).

To ensure that a user accesses a service through a valid role mapping chain, the original domain from which a user originated should be tracked in the authorization protocol. With the support of Property 3, if each $r_0 \in R_i$, $r_k \in R_j$ such that $(r_0, r_k) \notin F_k$ there is no *explicit policy conflict* in G_{VO}, and if each $(r_0, r_k) \in H'_{VO}$, $r_0, r_k \in R_i$ such that $(r_0, r_k) \in H_i$ there is no *implicit policy conflict* in G_{VO}. Therefore, if there is neither explicit policy conflict nor implicit policy conflict, the virtual organization collaboration policy is secure.

Definition 5 (Security of Virtual Organization Collaboration Policy): Let (r_0, r_k) be a role mapping policy through a valid role mapping chain in G_{VO}. The G_{VO} is secure if it satisfies the following conditions:

(C.1) For all $r_0 \in R_i$ and $r_k \in R_j$, there is $(r_0, r_k) \notin F_k$.

(C.2) For every $r_0, r_k \in R_i$, there is $(r_0, r_k) \in H'_{VO}$ if and only if $(r_0, r_k) \in H_i$.

Now, the problem we encounter is how to decide whether a given virtual organization policy is secure. According to Definition 4, we can infer $(\bigcup_{i=1}^{n} H_i)^+ = \bigcup_{i=1}^{n} H_i^+$ since there is no direct role mapping between any two domains. Each role mapping policy (r_0, r_k) where $r_0 \in R_i$, $r_k \in R_j$ in G_{VO} satisfies $(r_0, r_k) \in ((\bigcup_{i=1}^{n} H_i)^+ \circ M_{VO} \circ H_{VO}^+ \circ M_j \circ (\bigcup_{i=1}^{n} H_i)^+)$.

Therefore, if $(\bigcup_{i=1}^{n} H_i^+ \circ M_{VO} \circ H_{VO}^+ \circ \bigcup_{i=1}^{n} M_{Gi} \circ \bigcup_{i=1}^{n} H_i^+) \cap (\bigcup_{i=1}^{n} F_i) = \varnothing$, there is no *explicit policy conflict* in G_{VO}.

Based on this definition and let each role mapping chain L in G_{VO} be a valid chain, we can conclude the virtual organization collaboration policy is secure if and only if the evaluation result of each domain security policy together with task policy is secure. This conclusion plays a key role in secure collaboration because the security of virtual organization collaboration policy is guaranteed while preserving the privacy of domain policies. Most importantly, this conclusion enables the policy evaluation algorithm to be implemented in a completely parallel fashion to reduce the execution time. The detailed algorithm is shown in Figure 5.

Algorithm: *Policy_Evaluation* (G_i, R_{VO}, H_{VO}, M_{VO})
{
Input: A domain security policy G_i, and *task policy* including R_{VO}, H_{VO} and M_{VO} of a VO
Output: The policy evaluation result got by the i^{th} domain (boolean type)

```
1.  H+VO = Warshall(HVO);  //the
    transitive closure of set HVO
2.  H+i = Warshall(Hi);
3.    for (int k=1; k<=n; k++) {
4.      if (k!=i) {
5.        Ho+k = Warshall(Hok); } }
6.      S = (⋃n k=1 Ho+k ∘ MVO ∘ H+VO ∘ Mi ∘ H+i);
7.      if (S ∩ Fi !=∅) {
8.        return false; }
    // explicit policy conflict
9.      if ((r,r)∈S && r ∈Ri)){
10.       return false; }
    // implicit policy conflict
11.   }
12.   return true;
```

Figure 5. Distributed evaluation algorithm for virtual organization collaboration policy

Theorem 1: *Let each role mapping chain L in G_{VO} be a valid chain. The time-complexity of the distributed evaluation algorithm for virtual organization collaboration policy is* $O(\max\{|R_i|^3, |R_{VO}|^3\})$

Proof: The distributed evaluation algorithm for virtual organization collaboration policy can effectively reduce the computing time. As shown in Figure 5, the time-complexity of this algorithm is mainly affected by the computation of transitive closure of a role set, where a *Warshall* algorithm is used to calculate H_i^+ and H_{VO}^+, and its complexity is $O(|R_i|^3)$. Generally, the number of open roles in every domain is few, so we omit their computation costs here. Hence the maximum time-complexity of this phase is $O(\max\{|R_i|^3, |R_{VO}|^3\})$. In addition, due to the size of the set S is $|R_i|^2 \times |R_{VO}|$, the time-complexity of the phase comparing with all elements in S is $O(|R_i|^2 \times |R_{VO}|\})$. Taking both together, the time-complexity remains $O(\max\{|R_i|^3, |R_{VO}|^3\})$. □

Gong, Bertino et al. [12, 15] have proved some valuable results for time-complexity on security interoperation. The mediator-based algorithm needs to evaluate the security of collaboration policies. Gong has proved the time-complexity of this centralized algorithm is $O(|\bigcup_{i=1}^{n} R_i|^3)$. This is because the transitive closure of all role sets need to be computed.

4. PEACE-VO management and authorization protocol

PEACE-VO consists of two protocols: a management and authorization protocol and an authorization protocol for a VO. The management protocol guarantees the security of collaboration policy when a domain joins or leaves. The authorization protocol makes authorization decisions for service requests.

In the two protocols, a VO structure example is depicted in Figure 6, where we use symbol *VOS* for *VO Server*, and symbol *DS* for the *Domain Policy Server*. Unless stated elsewhere, all communications between servers are secured with the security handlers based on standards such as WS-Security, WS-SecureConversation, etc..

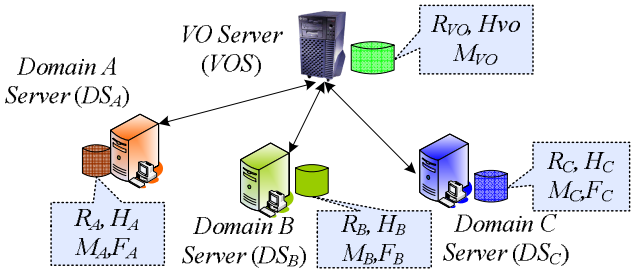

Figure 6. An example of the VO structure

4.1. Management protocol

Throughout the lifetime of a grid-based virtual organization, the collaboration policy often changes dynamically due to the fact that both business mode and participants keep changing.

This paper mainly focuses on how to determine whether a collaboration policy is secure or not. The policy to decide whether a domain is allowed to join the virtual

organization is beyond the scope of this paper, which actually can rely on existing mechanisms such as a pre-assigned policy, a voting mechanism, or manually approved by an administrator. Besides, how to resolve policy conflicts is also a hot topic. Currently, PEACE-VO employs two simple conflict resolution strategies. One is "*domain priority*", i.e., the domain security policy will not be changed. The other is "*collaboration priority*", i.e., the collaboration policy will not be changed.

Table 1. The basic messages in the VO management protocol

Message	Description
JoinReq	a domain wants to join the virtual organization
LeaveReq	a domain wants to leave the virtual organization
VOServerUpdate	the policy in *VO Server* changes
DomainServerUpdate	a domain security policy G_i changes
VOEvaluation	notification to domains to evaluate the collaboration policy
ResponseMsg	message and message identify code

In this paper, we are just concerned with the policy security as a result of the changes of collaboration relationships. Table 1 lists the basic messages within the management protocol. We illustrate its procedure with an example of a new domain requesting to join a VO.

Step 1: The server DS_A on behalf of domain A sends the message *JoinReq* to *VOS*:

$$DS_{\mathrm{A}} \rightarrow VOS : \{JoinReq, G'_{\mathrm{A}}\} .$$

Step 2: *VOS* authenticates the requested domain. If the request is permitted, then *VOS* broadcasts the policy evaluation notification message together with updated M_{VO} and G_{VO} to all domains DS_i except DS_A.

$$VOS \rightarrow DS_i : \{VOEvaluation, M_{VO}, G_{VO}\} , i \in \{1,\ldots,n\}.$$

Step 3: Each domain returns the policy evaluation result, which is true if there is no policy conflict, or else false otherwise. If the conflict resolution strategy is "*collaboration priority*", the domain should revise its local policy until the result is true. If the strategy is "*domain priority*", the domain just needs to report such issues to *VOS*.

$$DS_i \rightarrow VOS : \{ResponseCode\} .$$

Step 4: If an evaluation result is false, and the resolution strategy is "*domain priority*", *VOS* will revise the collaboration policy according to the pre-configuration policy, then repeats Step 2. Once all the results are true, *VOS* will inform the requester of the result.

$$VOS \rightarrow DS_{\mathrm{A}} : \{ResponseCode\} .$$

4.2. Authorization protocol

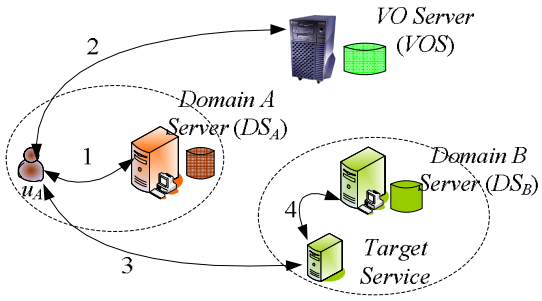

Figure 7. The authorization protocol

During the process of the service authorization, as depicted in Figure 7, there are three key sub-processes, "domain roles assignment", "virtual organization roles assignment" and "target domain roles assignment". The complete authorization protocol is as follows:

Step 1: After a user u sends requests to its local domain DS_A to which open roles he belongs, DS_A then returns a signed credential containing the assigned open roles to *u:*

$$DS_A \rightarrow u : \{\mathbb{R}_A, PK_u, DS_A, \mathbb{Z}\} ,$$

where $\mathbb{R}_A = (r^o_{A1} \circ \ldots \circ r^o_{Ak})$,

$\mathbb{Z} = SIGNATURE_{PrivK\text{-}A}(Hash(\mathbb{R}_A, PK_u, DS_A))$, and DS_A is used to identify to which domain this user originally belongs

Step 2: After user u sends requests to *VOS* to which *task roles* he belongs, *VOS* returns a signed credential containing the assigned roles to *u:*

$$VOS \rightarrow u : \{\mathbb{R}_{\mathbb{VO}}, PK_u, DS_A\},$$

where $\mathbb{R}_{\mathbb{VO}} = (r_{VO1} \circ \ldots \circ r_{VOk})$,

$\mathbb{Z} = SIGNATURE_{PrivK\text{-}VO}(Hash(\mathbb{R}_{\mathbb{VO}}, PK_u, DS_A))$.

Step 3: After user u sends service access requests to the target service in another domain, the service provider will transfer the requests message to its DS_B, DS_B then returns a signed credential containing the roles which u can be mapped to, so as to service provider makes authorization decision according to its original domain security mechanisms.

Generally, the lifetime of roles assignment to a user is limited, so issuing time and validity time periods are also included in the credentials.

5. Experiment and analysis

To effectively evaluate the performance of the policy evaluation algorithm and services in the PEACE-VO system, we conduct comprehensive experiments, and the experimental results show that the PEACE-VO system is scalable and efficient.

5.1. Metrics and environment setup

We design the following metrics to evaluate the performance of our proposed approach.

- *Policy Evaluation Time* (PET) measures the time (excluding phases of policy parsing and evaluation) that the policy evaluator used to decide whether the collaboration policy is secure. In studying the performance of our algorithm, we compare it with the *centralized-like* algorithm as the approach of mediator-based secure interoperation.
- *Evaluation Optimization Ratio* shows the benefit of the distributed evaluation algorithm with comparison to the *centralized-like* algorithm. The optimization ratio of average policy evaluation time is defined as follows:

$$\delta = \frac{PET_{Centralized} - PET_{Distributed}}{PET_{Centralized}}.$$

- *Service Response Time* is the time period from when the domain sends joining requests to *VO Server*, until when the requesting domain gets the response from *VO Server*.

In order to evaluate the efficiency of the policy evaluation algorithm more accurately, we list the main parameters, shown in Table 2, that influence the performance results. We set some default values for similar parameters, and vary the values of the three selected parameters to generate test cases.

Table 2. The parameters in test cases

Parameter	Value
n	
$\eta=\|R_i\|$	the value of η is equal
$\xi=\|H_i\|$	the value of ξ is equal
$\eta_{vo}=\|R_{vo}\|$	default:10
$\xi_{vo}=\|H_{vo}\|$	default: 3
$\theta=\|M_i\|$	the value of θ is equal, default:3
$\partial=\|F_i\|$	the value of ∂ is equal, default: 3
$\theta_{vo}=\|M_{vo}\|$	defualt:10

Our system is implemented and deployed in the CROWN Grid. The services are deployed on a cluster node with Intel Xeon 2.8GHz CPU, 2G RAM, Linux operating systems and 100Mbps Internet connection. We use a notebook with 1.6GHz CPU, 512M RAM, Windows XP operating system and 100Mbps Internet connection as the client. To make sure that the measurements are accurate, no other tasks are running on the cluster node and notebook, except the necessary CROWN middleware. If not explicitly specified otherwise, all the experiments are executed five times to get the average result.

5.2. Experimental results

Experiment Group 1: These experiments study the performance of the distributed collaboration policy evaluation, and study the impact of the parameters n and η on the performance of the two algorithms.

In the first experiment, we generate test cases with n=5, ξ=20, and vary the value of η from 50 to 500 with a step of 50, and compare performance of the two algorithms. The results are presented in Figure 8. Figure 8 shows the policy evaluation time increases roughly linearly with an increasing number of η. With two algorithms, the policy evaluation time is 50ms and 9ms respectively when η=50, it is 122ms and 1392ms when η=500.

In the second experiment, we generate test cases with ξ=20, and vary the value of η from 10 to 150 with a step of 10. The results are presented in Figure 9 and Figure 10. Figure 9 shows that the policy evaluation time, in a centralized manner, increases with the increasing number of both n and η, and becomes faster when the values of n and η are higher. For example, when n=10, η=100, the time is 512ms. Figure 10 shows that the policy evaluation time, in a distributed manner, only increases with an increasing number of η.

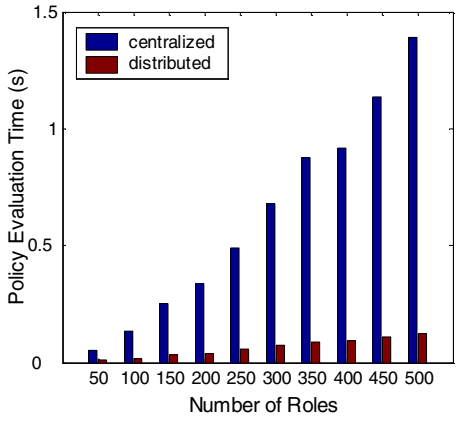

Figure 8. Policy evaluation time v.s. number of roles

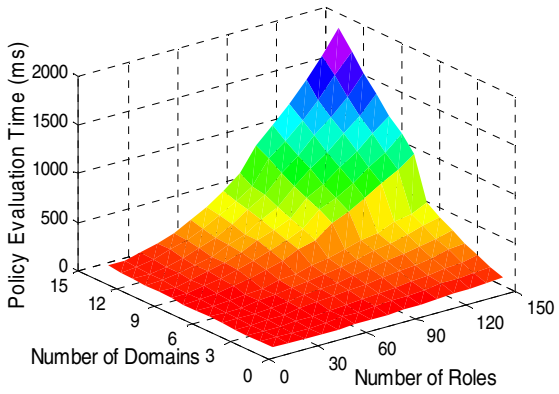

Figure 9. Policy evaluation in a centralized manner

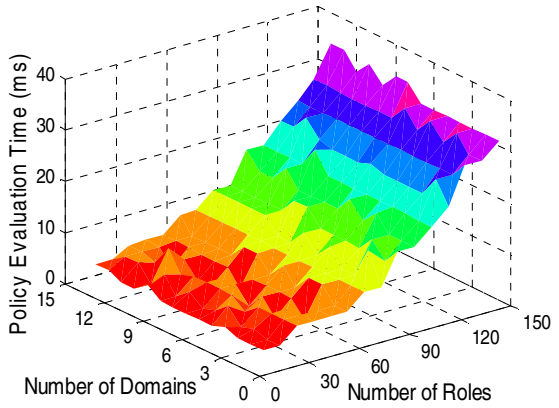

Figure 10. Policy evaluation in a distributed manner

Experiment Group 2: These experiments study the performance of virtual organization collaboration policy evaluation, and study the impact of the parameters n and ξ on the performance of the two algorithms.

In the first experiment, we generate test cases with n=5, η=5, and vary n from 2 to 46 with the step of 4. We then compare the performance of the two algorithms. The results are presented in Figure 11. Figure 11 shows the policy evaluation time has no relation with an increasing number of ξ. With two algorithms, the policy evaluation time is 37ms and 3ms respectively when ξ=10, and it is 34m and 3ms when ξ=30.

In the second experiment, we generate test cases with η=50, and vary n from 2 to 15, ξ from 2 to 48 with a step of 2. The results are presented in Figure12 and Figure 13. Figure 12 shows the policy evaluation time, in a centralized manner, only increases with an increasing number of n. Figure 13 shows the policy evaluation time, in a distributed manner, is between 3ms to 10ms, which almost has not be affected by the values of n and η, only increase with an increasing number of η.

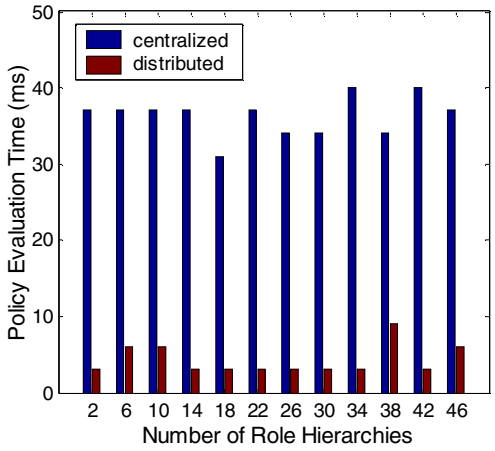

Figure 11. Policy evaluation time v.s. number of role hierarchies

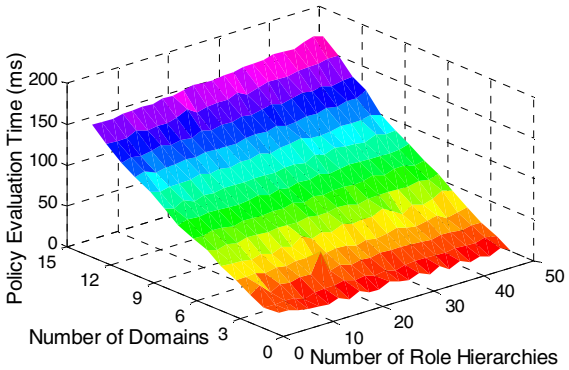

Figure 12. Policy evaluation in a centralized manner

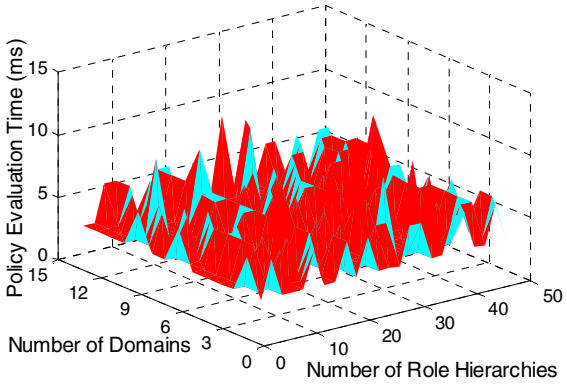

Figure 13. Policy evaluation in a distributed manner

6. Conclusion and future work

In a grid computing environment, building virtual organizations is an effective way to promote resource sharing and collaboration. However, most existing systems for security management of virtual organizations rely on completely new policies, which is not feasible for dynamic and autonomous grid environments. In this paper we have proposed a novel secure policy-enabled collaboration framework, PEACE-VO, in which a fully distributed policy evaluation algorithm is devised to improve evaluation efficiency without disclosing the full domain security policy. In PEACE-VO, we have adopted a centralized server, *VO Server*, for service authorization in a VO with the considerations of performance benefit. It may suffer from the single point of failure problem, so we will propose effective mechanism to enhance its reliability. Moreover, it is very important to resolve possible complex policy conflicts after identifying them. There are many existing studies[15, 18] and we will adopt these ideas in our systems.

Acknowledgement

This work is partially supported by grants from the China National Science Foundation (No.90412011), China 863 High-tech Program (No. 2005AA119010), China 973 Fundamental R&D Program (No. 2005CB321803) and National Natural Science Funds for Distinguished Young Scholar (No. 60525209). We would also like to thank Yanmin Zhu, Li Lin and Paul Townend for their helpful suggestions. We also thank the anonymous reviewers for their valuable comments.

References

[1] I. Foster, C. Kesselman, and S. Tuecke, "The Anatomy of the Grid: Enabling Scalable Virtual Organizations," presented at Proceedings of the 7th International Euro-Par Conference Manchester on Parallel Processing 2001.

[2] L. Pearlman, V. Welch, I. Foster, et al., "A Community Authorization Service for Group Collaboration," presented at the IEEE 3rd International Workshop on Policies for Distributed Systems and Networks, Monteray, California, U.S.A. , 2001.

[3] Q. Li, J. Li, J. Huai, et al., "CROWN-ST:Security and Trustworthiness Architecture for CROWN Grid," presented at IEEE International Conference on e-Science and Grid Computing (eScience2006), Amsterdam, Netherlands, 2006.

[4] J. Li, J. Huai, J. Xu, et al., "TOWER: Practical Trust Negotiation Framework for Grids," presented at IEEE International Conference on e-Science and Grid Computing (eScience2006), Amsterdam, Netherlands, 2006.

[5] R. Alfieri, R. Cecchini, V. Ciaschini, et al., "From gridmap-file to VOMS: managing authorization in a Grid environment," *Future Generation Computer Systems(FGCS), The International Journal of Grid Computing: Theory, Methods and Applications*, vol. 21, pp. 549-558, 2005.

[6] D. F. Ferraiolo, R. Sandhu, S. Gavrila, et al., "Proposed NIST Standard for Role-Based Access Control " *Information and System Security*, vol. 4, pp. 224-274, 2001.

[7] J. Huai, C. Hu, J. Li, et al., "CROWN: A service grid middleware with trust management mechanism," *Science in China Series F: Information Sciences*, vol. 49, pp. 731-758, 2007.

[8] T. Dimitrakos, G. Laria, I. Djordjevic, et al., "Towards a Grid Platform Enabling Dynamic Virtual Organizations for Business Applications," presented at iTrust 2005, Oxford, UK, 2005.

[9] V. Welch, T. Barton, K. Keahey, et al., "Attributes, Anonymity, and Access: Shibboleth and Globus Integration to Facilitate Grid Collaboration," presented at In 4th Annual PKI R&D Workshop, 2005.

[10] "Liberty Alliance Project," http://projectliberty.org/liberty/.

[11] H. Lockhart, S. Andersen, J. Bohren, et al., "Web Services Federation Language (WS-Federation)," 2006.

[12] L. Gong and X. Qian, "Computational Issues in Secure Interoperation," *IEEE Transaction on Software and Engineering*, vol. 22, 1996.

[13] M. Shehab, E. Bertino, and A. Ghafoor, "Secure collaboration in mediator-free environments," presented at Proceedings of the 12th ACM conference on Computer and communications security(CCS05), Alexandria, VA, USA, 2005.
[14] S.Dawson, S.Qian, and P.Samarati, "Providing Security and Interoperation of Heterogeneous Systems," *Distributed Parallel Databases*, vol. 8, pp. 119-145, 2000.

[15] B. Shafiq, J. B. D. Joshi, E. Bertino, et al., "Secure interoperation in a multidomain environment employing RBAC policies," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, pp. 1557-1577, 2005.

[16] P. Periorellis, N. Cook, and H. Hiden, "GOLD Infrastructure for Virtual Organisations," presented at in Proc. 5th UK e-Science All Hands Meeting, Nottingham, UK, 2006.

[17] S. Du and J. B. D. Joshi, "Supporting Authorization Query and Inter-domain Role Mapping in Presence of Hybrid Role Hierarchy," presented at The 11th ACM Symposium on Access Control Models and Technologies (SACMAT2006), 2006.

[18] H. Wang, S. Jha, M. Livny, et al., "Security Policy Reconciliation in Distributed Computing Environments," presented at Fifth IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY'04), 2004.