An Equivalent Access Based Approach for Building Collaboration Model between Distinct Access Control Models

Xiaofeng Xia

An Equivalent Access Based Approach for Building Collaboration Model between Distinct Access Control Models

Xiaofeng Xia

Heidelberg Institute for Theoretical Studies (HITS),
D-69118 Heidelberg, Germany
Xiaofeng.Xia@h-its.org

Abstract. Organizations collaborate with each other for resource sharing and task performing. To protect their resources from unauthorized access the organization domains adopt own access control models. The collaboration thus faces a problem that how a secure collaboration is built between the domains with distinct access control models. Currently there are approaches focusing on role based access control model (RBAC), where role mapping is considered to be the main technique. It assumes that all organizations adopt RBAC model, then builds a global access control policy on role mappings. However if the organization domains, also including collaboration domain, use distinct access control models, role mapping and global policy can not be built on these models. In this paper we propose an equivalent access based approach and introduce a mediator involved collaboration pattern, where access control model entities have corresponding mapping and linking sets on which the equivalent accesses are built. Collaboration also introduces the "Inter Domain Role Mapping" (IDRM) problem and we thus propose new algorithms for IDRM problem based on flat and hierarchical role structures, in addition we also introduce the necessary constraints transforming between organization and collaboration domains. Finally we analyzes our algorithms and present the testing and comparison results with existed approaches.

Keywords: collaboration model; distinct access control models; equivalent access; mapping set; linking set

1 Introduction

When several organizations want to make a collaboration, they could share resources among each other such that some common tasks can be completed. The collaboration pattern discussed in this paper refers to that for the resources shared from participating organization domains, the collaboration domain can have its own access control model. Practical security policy configurations tell us that the security models or policies are not all-purpose. To protect their resources from unauthorized access the organization domains adopt different access control models, e.g. RBAC [10], mandatory access control (MAC)[12], and discretionary access control (DAC)[6]. These models have different model entities related to permissions, which we call core model semantics. For example RBAC model constructs roles, MAC model has security labels. Currently there are approaches, e.g. in [3] and [4], focusing on RBAC, which assume that

all organizations adopt RBAC model, then build a global access control policy on role mappings. A global policy can be generated, because all domains have the same core model semantics, however if the domains use distinct access control models, role mapping and global policy can not be built on these models.
Organizational collaboration also introduces the IDRM problem in [3], which means to find out the minimal role set covering requested permissions from collaboration domain. This problem can then be generalized to distinct models and be defined as finding out an "appropriate" set of core model semantics covering requested permission set.
The third problem for organization collaboration is constraints transforming. As the model entities are mapped between domains, from the perspective of participators there are some constraints that must also be held in collaboration domain, e.g. for RBAC model, the separation of duty constraint(SSD)[9].
Therefore in this paper our contributions are (1)building a collaboration model between distinct access control models; (2)the necessary algorithms of figuring out an appropriate set of core model semantics to requested permission set; (3)constraints transforming between distinct models. The rest sections of this paper are organized as following: section 2 describes related work, in section 3 we present a new collaboration model based on equivalent access and section 4 illustrates the supporting algorithms and methods on building the collaboration model. Our testing and comparison results to algorithms is presented in section 5. Finally we have the conclusion of this paper in section 6.

2 Related Work

The RBAC[10] [11] model provides role-permission management, role hierarchy, and separation of duty constraints. For Lattice Based Access Control(LBAC) or MAC model[8], the information flow is restricted by the constraints on security labels and clearances. DAC[6] model emphasizes owning relationships of resources and permission delegation to be the way of authorization. In the past years RBAC is the most concerned model due to the of its conforming to organization structure.
An context-dependent RBAC model[7] is proposed to enforce access control in web-based collaboration environments. Organization based access control(OrBAC)[1] is constructed from a RBAC model as concrete level, and OrBAC then refers to common organizational contextual entities as abstract level. Based on OrBAC, PolyOrBAC[2] is proposed to implement the collaboration between organizations having OrBAC model in their domains. It takes advantage of abstract organizational entities and Web Services mechanisms, e.g. UDDI, XML, SOAP, to enforce a global framework of collaboration for engaging organization domains.
Role mapping[4] helps one domain obtaining accesses to resources from other domains by role-inheritances across domains. A global access control policy is specified to merge the engaging organization's local policies. This approach also assumes that all domains adopt RBAC model. Due to these contributions on RBAC and collaboration, we start to focus on organization domains with distinct access control models.
The other work on collaboration, or inter-domain operation, refers to IDRM problem. In [3], the proposed greedy-search based algorithm is an approximate solution to IDRM problem, however simple greedy-search has local-maxima problem, therefore a prob-

ability based greedy-search algorithm in this paper is used to avoid local-maxima and get better approximate solution. In section 5 we will discuss the problems of these approaches comparing with ours. To improve the algorithms, [5] presents another idea on greedy-search, they note that the assumption of IDRM problem should be more complex and practical. IDRM problem can be reduced to a weighted-set cover problem instead of minimal set cover problem in [3]. However the algorithm by[5] can not avoid local-maxima either.

3 Equivalent Access and Collaboration Model

3.1 Preliminary definitions

An organization domain or collaboration domain $\mathcal{D}$ should contain part of the following entity sets and relations:

- *User, Resource, and Action*: the sets of system users, resources, and operations on resources;
- T: the set of Tag objects, e.g. roles, security labels;
  As we constructed DAC model with a role based way[6], we view the model semantics as Tag objects, i.e. both role and label object can be instantiated by a Tag class which has at least two attributes: $< type, name >$.
- $Permission \subseteq Resource \times Action$, a set of permissions;

And some predicates and functions:

- $Reslabel(Resource, T)$: the assignment relation between a resource and a security label;
- $mayAccess(User, Resource, Action)$: a common predicate indicating an access request from some user to make an operation on some resource.
- $Usertag : User \times T$, is to indicate the relation that certain Tag objects (roles or labels) are assigned to some user.
- $PS : T \rightarrow Permission$: the permissions assigned or held by a Tag object;
- $PR : Permission \rightarrow T$: the tags holding current permission;

3.2 collaboration model based on equivalent access

Any access request of a user to some resource can be enforced by different access control models. We introduce "equivalent access", which is related to two domain's access control policies. Since in organizational collaborations the preliminary goal is to find appropriate resources, equivalent access refers to that a user's access to some resource under collaboration domain policy has equivalent evaluation results as that under participating domain policy.
Equivalent access should be the preliminary goal of organization collaborations, i.e. the constructing process of collaboration is to find equivalent accesses for the required resources in participating domains.
The collaboration scenario we discussed here refers to a collaboration domain, denoted

as $\mathcal{D}_c$, and a series of original domains, i.e. $(\mathcal{D}_c; \mathcal{D}_1, \ldots, \mathcal{D}_n), n \geq 2$. Each domain applies own access control model and policy. For a collaboration group $(\mathcal{D}_c; \mathcal{D}_1, \ldots, \mathcal{D}_n)$, there exists two sorts of entity relations between collaboration domain ($\mathcal{D}_c$) and other participating domains ($\mathcal{D}_i,\ i \in [1, n]$); one is the entity mapping set, the other is the entity linking set. We denote the former as $\mathcal{Q}$, which maps the entities of $\mathcal{D}_i$ onto those of $\mathcal{D}_c$ simply. The mapping means that for any resource $e_0 \in \mathcal{D}_i$, it has a corresponding virtual resource $e'_0 \in \mathcal{D}_c$. The mappings are classified into "user", "resource" and "action", i.e. $\{\zeta^u,\ \zeta^e,\ \zeta^a\}$.

Another relation is entity linking set, denoted as $\mathcal{L}$, which need to be computed and will be introduced in following parts.

Definition 1 *For a collaboration group $(\mathcal{D}_c; \mathcal{D}_1, \ldots, \mathcal{D}_n)$, considering any participating domain $\mathcal{D}_i, i \in [1, n]$ and its mapping set $\mathcal{Q}$, there are $u \in user_c$ and $e \in resource_c$, as well as $u' \in user_i$ and $e' \in resource_i$, such that $< u, u' > \in \zeta_1^u$ and $< e, e' > \in \zeta^e_{<\mathcal{D}_i, \mathcal{D}_c>}$; we say that the access by u to e is equivalent to that by u' to e' under two policies $\mathcal{P}_c$ and $\mathcal{P}_i$, if for the substitutions $\theta_{\mathcal{D}_c} = \{U_x/u, E_x/e, A_x/read\}$ and $\theta_{\mathcal{D}_i} = \{U_x/u', E_x/e', A_x/read'\}$ and:*

$$\mathcal{P}_c \models mayAccess(U_x, E_x, A_x)[\![\theta_{\mathcal{D}_c}]\!] \wedge \mathcal{P}_i \models mayAccess(U_x, E_x, A_x)[\![\theta_{\mathcal{D}_i}]\!] \quad (1)$$

*Then the **equivalent access** is denoted as:*

$$mayAccess(U_x, E_x, A_x)[\![\theta_{\mathcal{D}_c}, \theta_{\mathcal{D}_i}]\!]|_{<\mathcal{P}_c, \mathcal{P}_i>} \quad (2)$$

Definition 2 *The elements of entity linking set indicate the pairs of related "Tag" objects respectively from collaboration($\mathcal{D}_c$) and original($\mathcal{D}_i$) domains. When two substitutions towards their own policies $\mathcal{P}_c$ and $\mathcal{P}_i$ have equivalent access, a set $S_{\mathcal{D}_c}$ indicates the "Tag" objects which satisfy the request by $\theta_{\mathcal{D}_c}$ and a set $S_{\mathcal{D}_i}$ indicates those by $\theta_{\mathcal{D}_i}$, then the **entity linking set** $\mathcal{L}_{<\mathcal{D}_i, \mathcal{D}_c>}$ is defined as following rule:*

$$\mathcal{L}_{<\mathcal{D}_i, \mathcal{D}_c>} = \{< r, l > \mid < r, l > \in S_{\mathcal{D}_c} \times S_{\mathcal{D}_i}\}. \quad (3)$$

Definition 3 *For a collaboration group $(\mathcal{D}_c; \mathcal{D}_1, \ldots, \mathcal{D}_n)$, where all domain's model has the form of $\{\mathcal{D}_R,\ \mathcal{D}_M,\ \mathcal{D}_S\}$, considering any original domain $\mathcal{D}_i, i \in [1, n]$ and its mapping set $\mathcal{Q}_{<\mathcal{D}_c, \mathcal{D}_i>}$ with $\mathcal{D}_c$, the **collaboration model** Γ of the group will be defined by the above definitions of organization domain as a union of pairs:*

$$\Gamma = \bigcup_{i=1}^{n} < \mathcal{Q}_{<\mathcal{D}_c, \mathcal{D}_i>},\ \mathcal{L}_{<\mathcal{D}_c, \mathcal{D}_i>} > \quad (4)$$

4 Building Collaboration Model Between Distinct Access Control Models

In this section we will analyze the problems of building a collaboration model, then introduce the algorithms we use to build the model, as well as the methods to transfer constraints into collaboration domain. According to the definition of our new collaboration model, there are basically 3 steps to enforce:(1)finding out equivalent accesses;(2)try to minimize the scale of disclosure of the organization's policy information involved into collaboration;(2)domain constraints should be transferred into collaboration domain by configuring them on the policy entities in collaboration domain.

4.1 RBAC as participator's model

4.1.1 Minimal role set covering requested permissions

A greedy-search based algorithm (GSA) is proposed to get a solution to IDRM problem(NP-complete) in [3]. Basically the algorithm handles each candidate role with taking all its permissions that can cover as much as possible target permissions, then put this role into solution set. [3] also provides another probabilistic-greedy-search algorithm (IGSA-PROB) which executes candidate role handling with probability p (near 1). Greedy-search based algorithm however does not guarantee to find the optimal solution R'. It is an H_n-approximation algorithm for IDRM problem. The IDRM approaches proposed in [3] hence has the following problems:(1)the GSA algorithm is non-terminating and will probably not find any solution;(2)the GSA algorithm has local-maxima problem ;(3)the IGSA-PROB algorithm searches with probability p, while the local-maxima problem cannot be effectively avoided;(4)the inheritance hierarchy of roles can be applied to the IDRM problem.
The GSA and IGSA-PROB algorithms select only the roles which have permissions as a subset of required permission set to be candidates. Thus it makes the algorithms non-terminating. We build collaboration model by entity mapping and linking sets. The entity mapping set ensures that only the requests involving mapped entities will be allowed, which means that even if a role r is linked into, but only the mapped permissions will be allowed. This enables our algorithm to terminate. Towards solving IDRM problem we propose three algorithms, the input of them includes RQ, requested permission set; R, set of all roles; P, set of all permissions; R_S, set of initially selected roles; in turn the output has TS, set of candidate roles. They are specified formally in appendix.
I. Improved GSA algorithm (IGSA)

(1) finding out all the roles from R, which have intersected permissions with requested RQ, and put them into R_S.
(2) for a role r in set R_s, if r's permission set covers larger part of RQ than any other roles in R_s, then put r into candidate set TS, and remove r from R_s as well as remove the covered permissions of r from RQ.
(3) if RQ is not empty, then go to step (2).

II. Improved algorithm for local-maxima (IGSAL)

(1) for each permission finding out those which are assigned to a single role r.
(2) for the other roles in R, remove the permissions assigned to them, but also assigned to the role r.
(3) comparing each role r' with all of the other roles, if one of the permissions of r' belongs to another role r^* and r^* has more permissions than r', then remove all of the overlapped permissions from r'.
(4) if the permissions of r' are all removed, then r' should also be removed from R.
(5) performing the steps of algorithm I to compute candidate set TS.

III. Algorithm for hierarchical roles (HCHY)

(1) initially put the roles which have no parent roles, into set S_1, remove them from their child roles' parents list, then make a new set S_2.

(2) for each role r in R, if it has no parent roles and it dose not belong to S_1 and S_2, and if the convergent class set $Converg\_Classes$ is empty, then make a new convergent class set and add r into it; if $Converg\_Classes$ is not empty, then check every convergent class set C in it, if the current role r belongs to the child role set of any role in C, add r into C.
(3) remove r from the parent role set of each child role of r, add r into S_2.
(4) make a new setS_3; for S_3 and each permission p of P, make another new set S_4, thus for each role r' which holds p, if there is a convergent class set C containing r', add r' to S_4.
(5) after checking all of the roles having p, add S_4 into S_3; make new sets S_5 and S_6.
(6) by a recursive process "$recurse$", compute the combinations of sets in S_3 and return the minimal combination results.

4.1.2 Constraints of participating domain

Figuring out the minimal set roles covering requested permissions is the first step to enable the collaboration process, in addition, we must see that some RBAC constraints should also be held in collaboration domain. Here we focus on the static separation of duty constraint(SSD), which is defined as the following statements where "$assigned\_user(r)$" indicates the set of users holding the role "r", and "$assigned\_tag(u)$" indicates the set of roles being assigned to user "u"[9].

- $SSD \subseteq (2^R \times N)$, R is the set of roles and N is the set of natural numbers.
- $\forall < rs, n > \in SSD \forall t \subseteq rs.|t| \geq n \rightarrow \bigcap_{r \in t} assigned\_user(r) = \phi$

Now we know that rs is a set related to SSD, the possible "n-tuple" sets from rs is $C^n_{|rs|}$, which means the possibilities of picking n elements from $|rs|$ elements. For each possibility we define the set s_k of all involved permissions, thus $C^n_{|rs|}$ sets are defined as the following statements, where P_{SSD} indicates the permission sets for each of the SSD constraint elements in participating domain:

$$\forall r_1, r_2, ..., r_n \in rs.\ s_k = \bigcup_{i=1}^{n} PS(r_i)$$
$$P_{SSD} = \{s_1, s_2, ..., s_k\}, k = C^n_{|rs|}.$$

When the participating domain adopts RBAC model, the collaboration domain has also RBAC or DAC model (our DAC model is built by a "role" based way), it is necessary to note that there are 3 new constraints setting for collaboration domain's policy. They refer to in collaboration domain: (1) none of the "Tag" objects can have the whole permissions related to anyone of the SSD elements; (2) no user's permissions can cover the whole permissions related to one SSD element; (3)if the collaboration domain has RBAC model, then configuring new SSD constraints from the role sets which have the requested permissions. The 3 constraints are formally defined as the following statements. When the collaboration domain has MAC model, then only the constraint <1> should be held, since in MAC model each user holds one security label. Each member of P_{SSD} will be mapped to corresponding permission sets $s'_i, i \in [1, k]$ in collaboration domain and the permission sets accordingly to P'_{SSD} in collaboration domain.

<1> $\forall s'_i \in P'_{SSD} \forall t \in T_{\mathcal{D}_c}.\ s'_i \nsubseteq PS(t).$

<2> $\forall s'_i \in P'_{SSD} \forall t \in T_{\mathcal{D}_c} \forall u \in U_{\mathcal{D}_c} \forall l' \in assigned\_tag(u).\ s'_i \nsubseteq \bigcup PS(l')$
where $g = |s'_i| \geq 1, s'_i = \{p_j | j \in [1, g]\}$

<3> $\forall < rs_c, m > \in SSD_{s'_i} \forall t' \in rs_c . |t'| \geq m \rightarrow \bigcap_{l \in t'} assigned\_user(l) = \phi$
where $rs_c \subseteq T_{\mathcal{D}_c} \wedge o_d \subseteq rs_c \wedge m = |o_d| \wedge SSD_{s'_i} = \{o_d | o_d = \{r_s^1, r_s^2, ..., r_s^g\}\}$

4.2 MAC as participator's model

If the participating domain adopts a mandatory access control model, then a resource has exactly one label. When the requested resources and operations are confirmed, these resources can be simply mapped onto different security labels to which they are assigned in participating domain. In this section we discuss on the Bell Lapadula model (BL)[12][6] in collaboration, and the other Biba model is about integrity, which is dual to BL model.
The MAC model assigns for each object exactly one security label and for each user or subject only one security clearance. Comparing with the scenario where RBAC as participator's model, we only need to find out the labels of resources lying in the requested permissions, then these labels can provide equivalent accesses. To prevent disallowed information flow in collaboration domain, additional constraints must be added to collaboration domain policies. Since finding out the labels of resources is trivial, we provide only the definition of newly created constraint in collaboration domain. Assuming that a collaboration model Γ and one of the participating domains $\mathcal{D}_i$ and the collaboration domain $\mathcal{D}_c$ are defined as in section 3.
Single label constraint

<1> $P'_r = \{< e, a > | \forall e' \in Resource_{\mathcal{D}_c} \exists e \in Resource_{\mathcal{D}_i}.r \in T_{\mathcal{D}_c} \wedge < e', a > \in PS(r) \wedge < e', e > \in \zeta^e\}.$
$P'_r \subseteq RQ \wedge |\bigcup \{l | \forall < e, a > \in P'_r \wedge Reslabel(e, l)\}| = 1.$

<2> $\forall u \in U_{\mathcal{D}_c} \forall l, r \in T_{\mathcal{D}_c}.Usertag(u, l) \wedge Usertag(u, r) \rightarrow (l = r)$

<3> $T' = \{l | \forall u \in U_{\mathcal{D}_c}.Usertag(u, l)\}$
$\forall l \in T' \exists t \in T_{\mathcal{D}_i}.P'_l \subseteq RQ \wedge \bigcup \{t | \forall < e, a > \in P'_l \wedge Reslabel(e, t)\} = \{t\}$

In the collaboration domain, the information flow policy of participating domain should be held. Single label constraint will make restrictions on the labels of the resources which are shared in collaboration domain. Each "Tag" object can be assigned with the permissions, whose mapping entities in participating domain have the same security label. Each user or subject in collaboration domain can have either only one "Tag" object or multiple "Tag" objects which are assigned with the permissions related to same security label. Therefore the above constraint is expressed with the following formula: <1> $\wedge$ (<2> $\vee$ <3>).

4.3 DAC as participator's model

In a collaboration process, if the required permissions are provided from a participating domain with DAC model, the delegation of these permissions will not be considered in collaboration domain, since only the access permissions are necessary, while not the delegation permissions.
In our DAC model definitions, resource and different operations construct permissions for which different roles are created. Each resource has an owner, who is assigned "owner role" of the resource. The "owner role" inherits all of the permissions from other relevant roles.
Participating domain only needs to provide the basic roles which are related to the requested permissions. Although our DAC model adopts a "role" based way, in DAC model, there is no high level roles which hold large number of permissions related to different resources. Thus the previous algorithm of finding minimized role set for requested permissions will not be applied in DAC model. In participating domain with DAC model, there are no special constraints to be ensured in collaboration domain.

5 Analysis on Algorithm Properties and Testing Results

We present algorithms IGSA, IGSAL, and HCHY for handling minimal role set problem in section 5. Our collaboration model Γ verifies the entity mapping and linking sets, by which it is helpful to introduce non-required permissions. Only the collaboration relevant permissions, that is, the resources and operations are kept as entity mappings in collaboration model Γ, can be allowed for access.
As discussed in [3], the GSA has local-maxima problem and can be solved by GSA-PROB (probability based greedy search algorithm). By analyzing the problem we found that the permission assignment relationship, i.e. one permission assigned to multiple roles, causes local-maxima problem. Our IGSAL algorithm tries to remove this "multi-inheritance" from the role-permission relation, then the greedy search can be applied to resulted roles and permissions. To describe the complexity characteristics of these

Table 1. Comparison of IGSA and IGSAL on efficiency

Role size	Perm size	Requested perms	Time consuming(IGSA/IGSAL)	Solution size(IGSA/IGSAL)
100	41613	10^3	71 / 5334	80 / 78
100	45807	2×10^3	79 / 14549	90 / 87
100	46055	3×10^3	90 / 23011	91 / 91
100	43696	4×10^3	104 / 31864	93 / 89
100	45252	5×10^3	113 / 43066	96 / 95
100	44701	6×10^3	121 / 54115	98 / 96
100	48191	7×10^3	193 / 81417	99 / 97
100	44323	8×10^3	143 / 84534	99 / 99
100	45879	9×10^3	221 / 109845	98 / 97
100	43841	10^4	164 / 110684	97 / 95
100	47209	11×10^3	243 / 161712	98 / 98
100	45088	12×10^3	266 / 161768	99 / 98
100	46269	13×10^3	269 / 188546	100 / 98
100	44134	14×10^3	300 / 197264	98 / 97
100	44036	15×10^3	299 / 217346	99 / 97

3 algorithms, we assume that the size of requested permissions is N. Comparing with

IGSA and GSA-PROB algorithms, IGSAL spends computation on preprocessing the role-permission relations, then starts a greedy search to obtain solution. However on efficiency of algorithm, IGSAL has a nested loop for checking all of the requested permissions, which makes a $\mathcal{O}(N^2)$ complexity. Since the complexity of greedy search referring to IGSA and GSA-PROB is $\mathcal{O}(lnN)$[3] and the second step of IGSAL is also greedy search, the final complexity of IGSAL is still $\mathcal{O}(N^2)$. By randomly generating permissions and the assignment relationships, a testing for handling 100 roles and 43000 50000 permissions and the size of requested permission ranges from 1000 to 15000. Table 1 shows that IGSAL is less efficient than IGSA, but more precise.

It is mentioned that the role hierarchy can be used to provide minimal role set for requested permissions. The collaboration model can ensure that only mapped and linked entities related permissions can be allowed to access, even if there is a high level role is involved and has more permission than requested. Therefore from one or multiple role hierarchies in an organization domain one can find out the powerful roles to cover as much as possible requested permissions. The hierarchies discussed in section 4 is called convergent classes. The algorithm HCHY computes firstly the convergent classes of roles contained in an access control model, which will make a time consuming with complexity $\mathcal{O}(C_1)$. C_1 indicates that a constant time consuming on convergent classes, since the roles and role hierarchies in a domain has already been determined in advance. It is only necessary to compute it once. The second step of HCHY algorithm is to input the requested permissions, which takes time complexity $\mathcal{O}(N)$. Finally we need to figure out by a recursive process the minimal set of roles covering requested permissions, which is only related to the size of roles, hence the complexity of this process varies by the number of involved role hierarchies, assuming C_2. The total time complexity of HCHY on requested permissions is $\mathcal{O}(N) + C_1 + C_2$. By Table 2 we can see that HCHY is faster than IGSA.
In an organization domain with RBAC model, it adopts flat role structure or hierarchical role structure. our algorithms IGSA, IGSAL, and HCHY can handle and make use of both of these role structures.

Table 2. Performance testing of HCHY

Role size	Perm size	Requested perms	Convergent Classes	Time consuming	Solution
91	66610	10^3	60	29	3
91	66610	2×10^3	60	57	5
91	66610	3×10^3	60	65	7
91	66610	4×10^3	60	70	8
91	66610	5×10^3	60	75	7
91	66610	6×10^3	60	96	8
91	66610	7×10^3	60	86	10
91	66610	8×10^3	60	94	12
91	66610	9×10^3	60	106	15
91	66610	10^4	60	103	14
91	66610	11×10^3	60	119	16
91	66610	12×10^3	60	128	15
91	66610	13×10^3	60	149	21
91	66610	14×10^3	60	156	21
91	66610	15×10^3	60	157	22

6 Conclusion

In this paper we handle 3 problems in organizational collaboration: (1)a secure collaboration is built between the domains with the distinct access control models (2)finding out an ”appropriate” set of core model semantics covering requested permission set (3)constraints transforming between organization and collaboration domains. we present an equivalent access based approach and introduce a mediator involved collaboration pattern for the first problem. New algorithms are in turn proposed for IDRM problem based on flat and hierarchical role structures. Then some new constraints are presented for the third problem. Finally we analyzes our algorithms and present the testing and comparison results with existed approaches.
The collaboration pattern with ”mediator” works for both situations that there is a or there is no domain access control model in collaboration. The access control policies of participating domains are respected. In our future work, we will implement the mediator role, the collaboration model, and transformed constraints in XACML.

References

1. Kalam, A., Benferhat, S., Miege, A. et al: Organization based access control. In: Proceedings of the 4th Workshop on Policies for Distributed Systems and Networks. Page 120. (2003)
2. Kalam, A., Deswarte, Y., Bima, A. et al: Access control for collaborative system: a web services based approach. In Proceedings of International Conference on Web Services. (2007)
3. Du, S., Joshi, J.: Supporting authorization query and inter-domain role mapping in presence of hybrid role hierarchy. In: Proceedings of the eleventh ACM symposium on Access control models and technologies. Pages 228-236. (2006)
4. Joshi, J., Shafiq, B., Bertino, E.: Secure Interoperation in a Multidomain Environment Employing RBAC Policies. In: IEEE Trans. on Knowl. and Data Eng. 17(11):1557–1577. (2005)
5. Chen, L., Crampton, J.: Inter-domain role mapping and least privillege. In: Proceedings of ACM symposium on Access control models and technologies. (2007)
6. Osborn, S., Sandhu, R., Munawer, Q.: Configuring Role-Based Access Control to Enforce Mandatory and Discretionary Access Control Policies. In: ACM Transactions on Information and System Security. Vol.3, No.2. Pages 85-106. (2000)
7. Wolf, R., Schneider, M.: Context-Dependent Access Control for Web-Based Collaboration Environments with Role-Based Approach. In: Computer Network Security, Lecture Notes in Computer Science. Volume 2776, Pages 267-278. (2003)
8. Sandhu, R.: Lattice based access control. In Journal of Computer. Volume 26, Issue 11. Page 9-19. (1993)
9. Incits: ANSI INCITS 359-2004 for information technology role based access control. (2004)
10. Sandhu, R.S., Coyne, E.J., Feinstein, H. L., Youman, C.E.: Role-based access control models. In: IEEE Computer. Volume 29 No.2. (1996)
11. Sandhu, R.S., Bhamidipati, V., Munawer, Q.: The ARBAC97 Model for Role-Based Administration of Roles. In: ACM Transactions on Information and System Security. Special Issue on Role-Based Access Control, Volume 2 Issue 1. (1999)
12. Bell, D.E., LaPadula, L.J.: Secure computer systems: Mathematical foundations and model. MITRE technical report 2547, Volume I. (1973)
13. Nilson, U., Maluszynski, J.: Logic, programming and Prolog. Page 14-16. John Wiley&Sons Ltd. (1995)