A Directed Acyclic Graph based Detection for RBAC Based Secure Interoperation

Xiyuan Chen
College of Computer Science, Zhejiang University
Hangzhou, China
cxyspirit@zju.edu.cn

Miaoliang Zhu
College of Computer Science, Zhejiang University
Hangzhou, China
zhum@zju.edu.cn

***Abstract*—Collaboration enables domains to share resources effectively; however it introduces several security and privacy challenges. To guarantee the secure interoperation in complex distributed environment, a RBAC based secure interoperation model was proposed. Based on the inherent characteristic of the RBAC system, a directed acyclic graph based detection method of security violation was investigated. We also classified the conflicts according to the feature of each four parts of NITS RBAC model: conflicts resulting from unrelated roles, conflicts that arise from related roles and conflicts due to separation of duty. The targeted detection method for different types of conflicts was illustrated systematically. Therefore corresponding detection method can be applied to different types of conflicts according to the actual application environment. Furthermore, we analyzed the algorithmic complexity of the method and demonstrated the application of the directed acyclic graph based detection method with case studies in realistic scenarios.**

Keywords-Directed Acyclic Graph; RBAC; Security Interoperation; Conflict Detection;

I. INTRODUCTION

With phenomenal growth of information interaction and cooperation between distributed systems, security in interoperation becomes a challenging topic in multi-domain application environments. Role-based access control (RBAC) model has become an approval and powerful methodology specifying and applying authorization policies for security management. Compared with traditional access control models, Discretionary Access Control (DAC) and Mandatory Access Control (MAC), RBAC is a model with greater generality and flexibility, and can even express DAC and MAC strategies [1].

To meet the requirement for distributed access control, extension of RBAC model has been further analyzed to ensure secure interoperation in such multi-domain environments[2],[3],[4]. Kapadia et al provided a secure interoperability using dynamic role translation (IRBAC) for implementing access control across domains in the form of role mappings among individual domains[5]; based on IRBAC model in [5], Z. Liao et al further discussed methods to eliminate security violation[6]. XRBAC(XML Role-Based Access Control) was introduced by Joshi et al[7], presenting the description of role mappings and addressing policy specification for access management in multidomain environments. However, none of them presented specific methods for detecting security violation in RBAC based interoperation.

Shafiq and his team proposed a policy integration framework for centralized security violation accounting, whereas the cost of centralized accounting in the large scale distributed systems is too high[8]. A formal definition of secure interoperation in RBAC systems has been introduced by Chen et al[9]; they focused on Core RBAC and Hierarchical RBAC not including Constrained RBAC. Andreas et al demonstrated that, following a rule-based, declarative approach, how conflicts between specified Separation of Duty constraints and delegation activities could be detected[10]; Strembeck presented an implementation of conflict-checking methods for separation of duty constraints in the xoRBAC access control service[11]. Rattikorn et al proposed computational techniques for analyzing SoD by integrating workflows of the enterprise processes into the RBAC framework[12], and later, they proposed an algorithm for constraint checking of simple dynamic SoD at build-time[13]. Yet, there is neither explicit definition of secure interoperation nor systemical detection method for security violation for Constrained RBAC systems.

In this paper, we discuss the secure interoperation in RBAC systems including Constrained RBAC. A directed acyclic graph based security violation detection method is proposed. The Directed Acyclic Graph(DAG) based security violation detection method is much more intuitionistic than the traditional method. Furthermore, we classify the conflicts in the RBAC system. According to the classification of conflicts, corresponding detection method can be applied to exact type of violation under the practical condition.The rest of the paper is organized as follows. Section 2 gives a brief overview of NIST RBAC model. Subsequently, Section 3 describes the secure interoperation in Constrained RBAC systems. Section 4 then classifies the conflicts and introduces the directed acyclic graph based security violation detection method. Section 5 discusses the complexity of this method. We further analyze the directed acyclic graph based security violation detection method with case study in Section 6. In the end, Section 7 concludes this paper and presents the future direction of the research.

II. LEVELS OF RBAC MODEL

The NIST RBAC model is composed of four parts: Core RBAC, Hierarchal RBAC, static separation of duty of Constraint RBAC (SSD) and dynamic separation of duty of Constraint RBAC (DSD) [14]. The latter two are referred as Constrained RBAC generally. Core RBAC embodies the essential aspects of RBAC; Hierarchical RBAC adds a requirement for supporting role hierarchies; Constrained

978-1-4244-5113-5/09/$25.00 ©2009 IEEE

RBAC adds a requirement for enforcing separation of duties which spreads responsibility and authority for an action or task over multiple users. The following definitions (adapted from Ferraiolo [14]) formally define the Core RBAC model, the hierarchy on Roles and Separation of Duty in RBAC.

Definition 1 (Core RBAC)

- Sets Users, Roles, Permissions, and Sessions, representing the set of users, roles, permissions, and sessions, respectively;
- $Permissions = 2^{(Operations \times Objects)}$, representing the set of all permissions;
- $PA \subseteq (Permissions \times Roles)$, the permission assignment function, that assigns the permissions needed to roles to complete their jobs;
- $UA \subseteq (Users \times Roles)$, the user assignment function, that assigns users to roles;
- $user : Sessions \to Users$, that assigns each session to a single user;
- $roles : Sessions \to 2^{Roles}$, that assigns each session to a set of roles.

Definition 2 (Role Hierarchies)

- $RH \subseteq (Roles \times Roles)$, a partially ordered role hierarchy (written $\geq$).
- $roles : Sessions \to 2^{Roles}$ in Role Hierarchies, $roles(s) = \{r \mid (\exists r' \geq r)[(user(s), r') \in UA]\}$

Definition 3 (Separation of Duty in RBAC)

- $assigned\_users : Roles \to 2^{Users}$, the mapping of role r onto a set of users in the presence of a role hierarchy. Formally: $assigned\_users(r) = \{u \in Users \mid (\exists r' \geq r)[(u, r') \in UA]\}$.
- SSD (Static Separation of Duty), $SSD \subseteq 2^{Roles} \times N$ is a collection of pairs (rs,n) in SSD, where each rs is a role set, t a subset of roles in rs, and n is a natural number ≥ 2, with the property that no user is assigned to n or more roles from the set rs in each $(rs, n) \in SSD$. Formally: $\forall n \geq 2, \quad \forall (rs, n) \in SSD, \quad \forall t \subseteq rs$, $|t| > n \Rightarrow \bigcap_{r \in t} assigned\_users(r) = \varnothing$.
- DSD (Dynamic Separation of Duty), $DSD \subseteq 2^{Roles} \times N$ is collection of pairs (rs,n) in DSD, where each rs is a role set and n is a natural number ≥ 2, with the property that no subject may activate n or more roles from the set rs in each $(rs, n) \in DSD$. Formally: $\forall rs \in 2^{Roles}, \forall n \in N$, $(rs, n) \in DSD \Rightarrow n \geq 2 \wedge |rs| \geq n$, and $\forall s \in Sessions, \quad \forall (rs, n) \in DSD,$ $\forall role\_subset \in 2^{Roles}$, $role\_subset \subseteq rs \wedge role\_subset \subseteq roles(s) \Rightarrow |role\_subset| < n$

III. RBAC Based Secure Interoperation

A. RBAC based secure interoperation

One aim of distributed access control is to ensure that no security violations occur during interoperation between distributed systems. In particular, any secure interoperation should enforce the following two basic principles AS (Autonomy and Security) [15]:

Autonomy Principle: Any access permitted within an individual system must also be permitted under secure interoperation.

Security Principle: Any access not permitted within an individual system must also be denied under secure interoperation.

For better illustration of the security violation of interoperation, we firstly introduce the RBAC based secure interoperation [9]. The formal definition for RBAC based secure interoperation is based on the AS principle(Autonomy and Security) [15].

Definition 4 (RBAC based secure interoperation)

- Roles System, $R = \langle Roles, I \rangle$, Roles represents the set of all roles in the system; I represents the role hierarchies on Roles.
- Permitted Access, supposing that R is a directed acyclic graph composed of point set Roles and edge set I, then role hierarchy $r' \geq r$ is permitted in R, if and only if there is a directed path from r' to r, written $(r' \geq r) \propto R$.
- Multiple Individual Roles Systems, $R_i = \langle Roles_i, I_i \rangle, \quad i = 1, 2, ..., n$ and $Roles_i \cap Roles_j = \varnothing, i \neq j$.
- Permitted Access among Individual Roles Systems represents a binary relation on $\bigcup_{i=1}^{n} Roles_i$, mapping relation $RM = \{(r, r') \mid r \in Roles_i \wedge r' \in Roles_j \wedge i \neq j\}$, standing for permitted interoperations among multiple individual systems.
- Global Roles System, $R_0 = \langle Roles_0, RR \rangle, \quad Roles_0 = \bigcup_{i=1}^{n} Roles_i, \quad .$ $RR = \bigcup_{i=1}^{n} I_i \cup RM$
- Secure Interoperation in Global Roles System, if $\forall r, r' \in Roles_i, (r, r') \propto I_i \Leftrightarrow (r, r') \propto RR$, then all interoperations in R_0 are secure.

B. Conflicts in Hierarchical RBAC

Secure interoperations implemented through role mapping between several RBAC systems should also conform to the AS principles. As illustrated in Fig. 1, the role hierarchy is denoted by solid lines, and role mapping between different RBAC systems is denoted by dash lines.

We can draw a conclusion from Fig 1 that in Fig 1(a), interoperations related to links a and b are secure, however in Fig 1(b), interoperation induced by role mappings between two domains violate the basic AS principles.

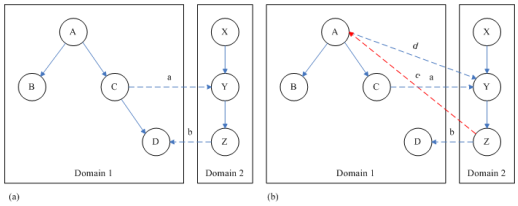

Figure 1. (a) Secure interoperation between RBAC systems (b) Security violation of interoperation between RBAC systems

If we add interdomain links c and d, as Figure 1(b) depicts, users originally authorized for role Z and not for role Y are now authorized for role Y. Because we can get Y from Z through the mapping path c, d from Z to A, and from A to Y. That is in contravention of the role hierarchy between Roles Y and Z.

Because of the inheritance path from C to Y to Z to D, we can infer that there is a role hierarchy between C and D in domain 1 which does not exist in individual domain 1

C. Conflicts in Constrained RBAC

Constrained RBAC adds a requirement for enforcing separation of duties which also brings along specific conflict of interest arising as a result of the simultaneous assignment of two mutual exclusive permissions or roles to the same subject. We take static separation of duty of Constraint RBAC for example.

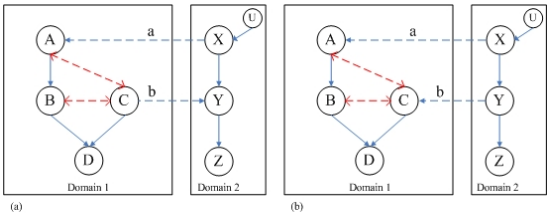

Figure 2. (a) Secure interoperation between Constrained RBAC systems (b) Security violation of interoperation between Constrained RBAC systems

As illustrated in Fig. 2, the role hierarchy is denoted by solid lines, and role mapping between different RBAC systems is denoted by dash lines. Red double-headed arrow between role B and role C denotes that role B and role C are mutually exclusive roles, and it means that if a user is authorized as a member of role B, the user is prohibited from being a member of role C. Meanwhile, due to inheritance between role A and role B and mutual exclusive roles B and C, role A and role C also become exclusive roles. Leaf node U is authorized as a member of role X.

We draw a conclusion from Fig 2 that in Fig 2(a), interoperations related to links a and b are secure, however in Fig 2(b), interoperation induced by role mappings between the two domains violate the rules of separation of duties. User U assigned to role X gets mutually exclusive roles A and C simultaneity via junior role Y and role mappings a and b., which is a SoD conflict.

D. Classification of the Conflicts in RBAC

According to basic principles of secure interoperation and the feature of levels of RBAC models, there are generally three types of violation. First two are conflicts in Hierarchical RBAC and the third is conflict due to violation of Separate of Duty in Constrained RBAC.

- Unrelated Conflict: There exist two roles in one independent role system which do not have access relation but permitted access relation in the global role system, i.e. $\exists r, r', (r, r') \not\propto I_i \wedge (r', r) \not\propto I_i \wedge (r, r') \propto RR$. In Figure 1(b), roles C and D do not have access relation in domain 1, but have access relation in the global role system through the interoperation between domain 1 and domain 2.
- Related Conflict: There exist two roles with access relation in one independent role system, but the one in the lower hierarchy has access to the one in the higher hierarchy in the global role system, i.e. $\exists r, r', (r, r') \not\propto I_i \wedge (r', r) \propto I_i \wedge (r, r') \propto RR$. As roles Y and Z in Figure 1(b).
- SoD Conflict: There exist two exclusive roles in one independent role system which do not have users authorized as a common member of them but users authorized as a common member of them in the global role system. This is a violation of separation of duty, i.e. $\exists r, r' \in rs, ((rs, 2) \in SSD \propto I_i) \wedge ((u, r) \in UA \wedge (u, r') \in UA) \propto RR$. As exclusive roles A and C in Figure 2(b). they can be acquirede by user U at the same time, which is a SoD Conflict.Equations

IV. Directed Acyclic Graph based Detection for RBAC based Secure Interoperation

In a distributed environment which consists of numerous systems, there are vast number of interoperations. Therefore, effective detection method for security violation of interoperation is necessary to assist the administrator to construct the security interoperation among RBAC systems. [8] proposed methods to collect all role information of each distributed application, merge duplicate roles, decompose ambiguous roles, establish new role relations, and compute security violation based on this information. However, it is quite difficult to compute in real distributed environment with such a centralized method. [9] introduced a minimum detection method for Hierarchical RBAC according to the feature of RBAC system in distributed environment. Nevertheless, with the growing amount of roles involved in interoperation, the performance of the minimum detection method is no better than the global detection method. Furthermore, they focused on

Core RBAC and Hierarchical RBAC which did not include Constrained RBAC. Shehab et al proposed a distributed secure interoperability framework for mediator-free collaboration environments[16], and they introduced the idea of secure access paths to avoid the violation in the interoperation between domains. However, their method was based on the assumption that the access history is reliable and trustful meanwhile all users obey the security policy. In fact, there are malicious domains and users all the time.

In this section, the Directed Acyclic Graph based Detection for RBAC based secure interoperation including Core RBAC, Hierarchical RBAC and Constrained RBAC, which is more intuitionistic and flexible, is introduced according to the classification of the conflicts in the interoperation mentioned in section **Error! Reference source not found.**. The exact type of interoperational conflicts can be detected by the use of DAG-based Detection method, which is beneficial for the future conflict resolvation in accordance with the situation. Meanwhile, corresponding detection method can be applied to precise type of violation in light of the practical condition, instead of adopting global detection method every time when any change of roles occurs in the interoperation.

Definition 5 (RBAC Model Expression by Directed Acyclic Graph)

- Role System, directed graph $R = \langle Roles, I \rangle$, Node Set Roles depicts Role Sets in the RBAC systems, directed edge sets I represents hierarchies between roles in the systems. Given $r' \geq r$, $(r', r) \in I$, there is a path from r' to r .
- Accessibility, a path is a sequence of nodes (ri, ri+1, …, rj), $(r_k, r_{k+1}) \in I$, $i \leq k < j$ and $r_i \neq r_j$. If there is a path from ri to rj , then from ri to rj is accessible.

```
int hasConflict()
//return the corresponding code of the tyoe of conflicts
{
   if ( hasUnrelatedConflict(Restructed Access Set) )
 //return 1 when Related Conflict exists
   {  return 1;  }
   if ( hasRelatedConflict() )
 //return 2 when Unrelated Conflict exists
   {  return 2;  }
   if ( hasSODConflict(SOD Set) )
//return 3 when SoD Conflict exists
   {  return 3;  }
   return 0;
}
```

Figure 3. Directed Acyclic Graph based Detection for RBAC based secure interoperationAuthors and Affiliations

Here is the DAG-based Detection for RBAC based secure interoperation. We will discuss the detail in the following section.

A. Detection for Unrelated Conflict

Unrelated Conflict means that two unrelated roles in an individual system become related in the global role system due to the role mappings in the interoperation. For judging the security of the relation between the unrelated roles, we introduce the Restricted Access.

Definition 6(Restricted Access)

- Restricted access $F_i = \{(r, r') \mid r, r' \in Roles_i\}$ is a binary relation on Ri representing that $r' \geq r$ is forbidden in Ri. There is no path from r' to r in Ri.
- Restricted Access among Individual Roles Systems represents a binary relation on $\bigcup_{i=1}^{n} Roles_i$, mapping relation $FM = \{(r, r') \mid r \in Roles_i \wedge r' \in Roles_j \wedge i \neq j\}$, standing for restricted interoperations among multiple individual systems.
- Restricted Access in Global Roles System, $\bigcup_{i=1}^{n} F_i \vee FM$, and $f = \left| \bigcup_{i=1}^{n} F_i \vee FM \right|$.

In the interoperation, there is Unrelated Conflict on condition that Restricted Access occurs, which predicates that there is restricted access between unrelated roles. Given the Restricted Access Set, we check the accessibility of each access in the set with Accessibility Matrix for detection of Unrelated Conflict.

```
boolean hasUnrelatedConflict(Restructed Access Set)
 //return True if there is any UnRelated Conflict
{
   for ( i=0; i<f;i++ )
 //using Accessibility Matrix to check the Accessibility
   {   for each access in the set, check the Accessibility
       if the access is accessible
       { return true; }
   }
   return false;
}
```

Figure 4. Detection for Unrelated Conflict

B. Detection for Related Conflict

```
boolean hasRelatedConflict()
//return True, if  there is any Related Conflict
{
 //using Topological Sort to detect the loop
    once finding  a loop
     { return true; }
    return false;
}
```

Figure 5. Detection for Related Conflict

Related Conflict indicates that two roles have security access relation in one independent role system, but the one in the lower hierarchy has access to the one in the higher hierarchy in the global system, it can be described as $\exists r, r', (r, r') \notin I_i \wedge (r', r) \propto I_i \wedge (r, r') \propto RR$. We can draw a conclusion that there is a circle path from the beginning role to the beginning role due to Related Conflict. We can see in Fig(b), there is a path from Y, Z, A, C, and then to Y, which is obviously a Related Conflict. For detection of Related Conflict, we have to find out whether loop exists in the systems after interoperation.

C. Detection for SoD Conflict

The SoD Conflict could be divided in two types, namely the static and dynamic mutually exclusive roles (SMER) and (DMER) respectively[17]. Mutually exclusive roles are not permitted be assigned to the same user for Constraint RBAC (SSD), while for dynamic separation of duty of Constraint

RBAC (DSD) mutually exclusive roles can be assigned to the same user only can not be activated simultaneity. We take SSD for example. According to the definition of SSD, $\forall n \geq 2, \quad \forall (rs, n) \in SSD, \quad \forall t \subseteq rs$, $|t| > n \Rightarrow \bigcap_{r \in t} assigned\_users(r) = \varnothing$. For detection of SoD Conflict, we have to find out whether there is any leaf node(child node) that is connected to more than n roles in the given exclusive role set.

```
boolean hasSoDConflict(SoD Set)
 //return True if SoD Conflict exists
{
   for each（rs，n）
   check out whether there is any child node that is
   connected to more than n roles in the sets
    //using the method searching for connected nodes
   if exists
    { return true; }
   return false;
}
```

Figure 6. Detection for SoD Conflict

V. Analysis on Computational Complexity

Traditional detection method[15][9] merely takes Core RBAC and Hierarchical RBAC into consider which did not include Constrained RBAC. In Core RBAC and Hierarchical RBAC, conflicts can be classified into Unrelated Conflict and Related Conflict, therefore, in our DAG-based detection method, traditional detection method for conflicts in Core RBAC and Hierarchical RBAC could be substituted by detection for Unrelated Conflict and Related Conflict. Complexity of traditional global detection method[15][9] is $O(n^3)$. For Unrelated Conflict, the complexity of computing accessibility matrix is $O(n^3)$; for Related Conflict, the complexity of topological sorting is $O(n^2)$, hence the complexity of Unrelated Conflict and Related Conflict is $O(n^3)$ which is considerable.

VI. Case Studies

To demonstrate the application of the role based secure interoperation and DAG-based violation detection method, we performed case studies with realistic scenarios in Digital Campus System (DCS) for Zhejiang University. Zhejiang University is a comprehensive national university, with more than 39,000 full time students and 8,400 staff members distributed in 5 campus. DCS is composed of a host of applications such as Student Management System(SMS), Faculty Management System(FMS), Course Management System(CMS), Scientific Research Management(SRM), Student Hostel Management System(SHMS) and etc. With the increase in information and data accessibility, the security becomes more and more important in the interoperation among different domains.

A. Application Environment

University students and staffs are playing many different roles in DCS system for day-to-day affairs such as housing distribution, course management, salary management and so on. Table 1 is a brief list of the roles and responsibilities.

TABLE 1 A Brief List of Roles and Responsibilities in DCS System

Roles	Responsibilities
Senior Human Resource Administrator(SHRA)	Responsible for Recruitment, Employment Management, Employee Relationships, Training & Development, and other personnel assignments in the university.
Junior Human Resource Administrator(JHRA)	To provide the general HR function services in each independent unit, e.g. JHRA for Math College responsible for basic HR service in Math College.
Payroll Administrator(PA)	To perform timely and accurate payroll processing related services for the staffs.
Senior Scientific Research Administrator(SSRA)	To manage the scientific research affairs and provide other related services for all the staff in the university.
Junior Scientific Research Administrator(JSRA)	To provide scientific research related services in each independent unit, e.g. JSRA for Math College responsible for basic scientific research service in Math College.

DCS covers multidomains, such as Student Management System(SMS), Faculty Management System(FMS), Course Management System(CMS) etc and each one has its own environment and applications generally. Each domain maintains its own security settings including user, role and permission etc to protect applications and information. Interoperation often occurs across domains due to information exchange and sharing in the DCS. Roles mapping is a convenient way for interoperation across multi-domains in condition that all the domains share the uniform user identity. Roles mapping can keep logical independency within domain while building links to others. Moreover, compared to the huge amount of users or permissions in the DCS system, roles are more flexible to map.

B. Interoperation Scenarios

Role SHRA, JHRA and PA belong to Faculty Management System(FMS); SSRA and JSRA are roles in Scientific Research Management(SRM). Domain 1 and Domain 2 denote the above applications respectively. In Domain 1, there are two exclusive roles SHRA and PA, since SHRA is in charge of all the information of staff including working effect, technical skill and so on, and PA set the salary according to the information of staff from SHRA.

Scenario One: At the end of the semester, JSRA needs the staff information from JHRA in each unit. JHRA requests the report of scientific research from SSRA. RBAC based interoperation can be built simply as following:

- $Roles_1$ = { JHRA }, $Roles_2$ = { SSRA, JSRA }
- $I_1 = \varnothing$, I_2 = {(SSRA, JSRA)}
- RM = {(JSRA, JHRA), (JHRA, SSRA)}

Interoperation above denotes that:

- Users in domain 2 assigned as the role JSRA can have access permission of the role JHRA in domain 1.
- Users in domain 1 assigned as the role JHRA can have access permission of the role SSRA in domain 2.

Scenario Two: At the beginning of the semester, JSRA requests the information of the certain allowance in their unit from PA. SSRA wants the detail of new teachers of the university this year. RBAC based interoperation can be built simply as following:

- $Roles_1$ = { SHRA, PA }, $Roles_2$ = { SSRA, JSRA }
- $I_1 = \varnothing$, I_2 = {(SSRA, JSRA)}
- rs = {SHRA, PA }, (rs, 2) ∈ SSD
- RM = {(JSRA, PA), (SSRA, SHRA)}

Interoperation above denotes that:

- Users in domain 2 assigned as the role JSRA can have access permission of the role PA in domain 1.
- Users in domain 2 assigned as the role SSRA can have access permission of the role SHRA in domain 1.

C. Conflict Detection Discussion

In scenario one, we don not have Restricted Access and exclusive roles, accordingly we adopt the detection for Related Conflict in DAG-based detection method. Scenario one has such relation JSRA → JHRA → SSRA → JHRA which exits a circle. This security violation is detected by the DAG-based detection method.In scenario two, there are exclusive roles such as PA and SHRA, thus we pay more attention on the SOD Conflict detection. After the detection for SOD Conflict in DAG-based detection method, we find that users who are assigned to SSRA have permission of the role PA through SSRA → JSRA → PA and permission of the role SHRA by SSRA → SHRA . It means that there are users who get the permission of the exclusive roles PA and SHRA simultaneity and this violates the rule separation of duty.Since we are able to apply the exact part of DAG-based detection method according to the actual situation, we can find out the conflict more efficiently.

VII. CONCLUSION

We have discussed the problem of secure interoperation in the distributed environment, classified the conflict occurred due to interoperation between different domains in the RBAC systems including Core RBAC, Hierarchical RBAC and Constrained RBAC. A Directed Acyclic Graph based detection method was proposed. With this method, corresponding detection method can be applied to exact type of violation according to the actual application.Future research directions are outlined: Related conflict solution methods of classified conflicts need to be explored; Provide users with optimum strategy which can not only return the result whether an interoperation is secure or not , but also the conflict solution with less dependentcy on human administrator's participation.

REFERENCES

[1] Sylvia Osborn, Ravi Sandhu, Qamar Munawer, 2000. Configuring Role-Based Access Control to Enforce Mandatory and Discretionary Access Control Policies. ACM Transactions on Information and System Security, 3(2):85-106.

[2] Jianfeng Lu, Ruixuan Li, Zhengding Lu, Bing Li, 2008. Integrating Trust and Role for Secure Interoperation in Multi-Domain Environment. Proceedings of the 2008 International Conference on Information Security and Assurance, 77-82.

[3] Yue Zhang, Joshi, J.B.D., 2007. A request-driven secure interoperation framework in loosely-coupled multi-domain environments employing RBAC policies. Proceedings of the 2007 International Conference on Collaborative Computing: Networking, Applications and Worksharing, 25-32.

[4] Zhuo Tang, Ruixuan Li, Zhengding Lu, 2007. A Request-Driven Role Mapping for Secure Interoperation in Multi-Domain Environment. Proceedings of the 2007 IFIP International Conference on Network and Parallel Computing, 83-90.

[5] Kapadia Apu, Al-Muhtadi Jalal, Campbell R, 2000. Secure Interoperability Using Dynamic Role Translation. Proceedings of the First International Conference on Internet Computing, 231-238.

[6] Z. Liao, H. Jin, W. Qiang, 2005. An improved approach towards the model of IRBAC2000. Journal of Huazhong University of Science and Technology (Nature Science), 33:292–295.

[7] Joshi J.B.D, Bhatti R, Bertino E, Arif Ghafoor, 2004. Access-control language for multidomain environments. IEEE Internet Computing, 8(6): 40-50.

[8] Shafiq B, Joshi J B D, Bertino E, et a1. Secure interoperation in a multidomain environment employing RBAC policies[J]. IEEE Transactions on Knowledge and Data Engineering, 2005, 17(11):1557-1577.

[9] Chen Xiyuan,Wu, Di, Lin Jian, Zhu Miaoliang, A security violation detection method for RBAC based interoperation [C].2006 International Conference on Computational Intelligence and Security, ICCIAS 2006, p 1491-1496.

[10] Andreas Schaad. Detecting conflicts in a role based delegation model[C]. Proceedings of the 17th Annual Computer Security Applications Conference. New Orleans:200l:117-126

[11] Strembeck, M. Conflict checking of separation of duty constraints in RBAC-Implementation experiences[C]. In Proceedings of the Conference on Software Engineering 2004.

[12] Rattikorn Hewett, Phongphun Kijsanayothin, Aashay Thipse, 2008. Analysis of Role-based Separation of Duty with Workflows. Proceedings of the Third International Conference on Availability, Reliability and SecuritySecurity, 765-770.

[13] Thipse A., Hewett R, 2008. Verification of Dynamic Separation of Duty Policy for Role-based Business Processes. Proceedings of the IEEE Region 5 Technical, Professional and Student Conference, 1-6.

[14] Ferraiolo D, Sandhu Ravi, Serban Gavrila. Proposed NIST Standard for Role-Based Access Control [J]. ACM Transaction on Information and System Security, 2001, 4(3): 224-274.

[15] Gong L, Qian X.L. Computational Issues in Secure Interoperation[J]. IEEE Transaction on Software and Engineering, 1996, 22(1): 43-52.

[16] Shehab M, Bertino E, Ghafoor A. Secure collaboration in mediator-free environments. In: Meadows C, Syverson P, eds. Proc. of the 12th ACM Conf. on Computer and Communications Security. Alexandria: ACM Press, 2005. 58−67.

[17] Li N., Tripunitara M., Bizri Z., 2007. On mutually exclusive roles and separation-of-duty. ACM Transactions on Information and System Security, 10(2):1-36.