

Lattice-based Risk Models for Workflows in Multi-domain Environments

Lu Tang*, Zhuo Tang*, Renfa Li*

**School of Computer and Communication, Hunan University, Embedded System & Networking Laboratory, Changsha, China
tanglu1010@sina.com*

Abstract— In multi-domain environments, the risk of access policy for workflows is influenced by many factors, and it is difficult to calculate the accurate value of risk combining these factors. But choosing an access policy does not need to know the specific value of risk, only to know the partial order of risks. This paper introduces the concept of risk, proves that the partial order of risks is a lattice, which can ensure implementation of the multi-level security policy. And the risk of access policy can be compared using the lattice effectively. Then, according to it, this paper proposes an algorithm to determine the access policy. It can reinforce the security of workflow systems in multi-domain environments.

Keywords— risk, security, lattice, access policy, workflow, multi-domain

I. INTRODUCTION

Workflow technology which has been widely used in many fields, is a collaborative modelling method of complex tasks, it consists of a set of collaborative tasks and their dependencies. With the workflow management system (WfMS) [1], tasks can be automatically executed. In order to guarantee tasks executed safely, access control is a very important security mechanism of workflow systems.

In general, an access policy is not absolutely security. Absolute security systems are the systems which prohibit any access. So risk is the basic property of access policies. As long as an access policy exists, there is risk in the system.

As the development of internet technology, more and more applications have developed from centralized to distributed, for this we should make it possible to share information across autonomous domains in heterogeneous environments. A workflow is composed of multiple tasks, different tasks may be in different domains, the subject and the object of a task may be in different domains. And because of the highly dynamic and uncertainty of multi-domain environments, the risk of access policy will be more complex, which will make the access control of workflow systems more difficult.

This paper compares the security between access policies based on their risk level, establishes the partial order between

them based on their security, and proves that the partial order meets reflexive, anti-symmetry and transitivity, it can be seen that policies and partial orders of their risks constitute a poset. Also because there are supremum and infimum between risks of any two policies, in other words, the partial order between policies is a lattice. To adapt multi-domain environments, we introduce the concept of the trust extent between different domains to constitute a 3-dimensional risk lattice model. This lattice model can effectively compare risks between access policies, but it can't accurately calculate the value of risk, in fact it is difficult to calculate it accurately. And we do not need to know the exact value when we choose a policy for each task of workflow. We only need to know the partial order of risks. In addition, this paper presents an algorithm to select access policy based on the risk lattice model in multi-domain environments.

II. RELATED WORK

Access models of workflow systems have been proposed by many researchers. A famous one is TBRC [2], which is a task-based access control model. In TBRC, permissions are assigned to tasks, rather than users, users only can acquire permissions when tasks are executed. Once the user finishes and submits the task, the relevant permissions are released, the user no longer have the permissions. In this way, users can access the permissions dynamically, the system can implement dynamic access control.

As the development of enterprises, business processes frequently change, thereby the complexity of workflow access control has been increased, [3] proposes the idea of separation of access control model and process model in workflow systems, and using it as a starting point, presents a service-oriented workflow access control model, services are the basic units of task implement and access control. Xu Feng et al. propose a service-oriented role-based workflow access control model [4], which implements the access control of workflow systems by introducing concepts of services, actors and authorization migration.

Risk is an important property of access control models, it has been considered in many researches. [5] introduces the concept of risk based on the RBAC model, and reconfigures

the risk-based access control policy to reduce risk, but it doesn't propose specific methods to quantify risks. RARAC model associates risks with benefits, builds transactions graph, when the system's benefit is greater than risk, transactions will be allowed [6]. In [7], McGraw presents a risk-adaptive access control mechanism, which authorizes access permissions based on risks and operational needs, but it doesn't propose the specific methods to quantify them. Cheng et al. present a Fuzzy MLS access control model in [8], this model can quantify the risk value of access, and control risky information flows dynamically.

In multi-domain environments, tasks of workflows are likely to be interoperated cross domains, it will bring greater risk. [9] proposes a risk-based multi-domain security interoperation model, which introduces a risk management mechanism, it quantifies risk by means of sessions and historical records. Tang Zhuo et al. in [10] present a dynamic access control model for inter-operation in multi-domain environment based on risk, it determines the degree of trust between domains by historical records of inter-operations, combined with security level of objects and security of access events, provides risks of access policies in real-time. And it can dynamically adjust the permissions by using the mechanism of revoking high-risk permissions.

III. RISK LATTICE MODEL

Many risk models of access control focus on how to quantify assess risk, but usually the risk of access policy is affected by many factors, it is difficult to accurately estimate it. In fact, choosing an access policy is based on the partial order of risk between them, rather than the specific risk value.

A. Basic concept

This section mainly introduces the formalized description of some basic concepts:

U is the set of users that are executors of access events. We describe all users assigned to the role r as $User(r)$. O is the set of objects which are accessed by users. The set of roles is denoted as R .

In service-oriented workflow access control models, services are permissions [4]. S is the set of services, a service is made of a group of independent basic operations, a basic operation is denoted as a 2-tuple of the form: $\langle op, o \rangle$, where op is an operation, such as reading, writing, execution, etc. o is the system resource, such as documents, database tables, etc.

The actor of a role ($actor$) is a dynamic object which is generated when the role is activated by a user [4], it acts for the user and performs the services of the role. A is the set of actors. We can describe an actor as a 3-tuple of the form: $\langle u, r, lifetime \rangle$, and the actor acts for the user u , $u \in U$; r is the role activated by the user u , $r \in R$; $lifetime$ is the survival time of the actor.

AU is the set of authorization units [4], authorization unit(au) is a basic authorization execution unit, it is an instance of a task which may have more than one authorization units. An authorization unit is made of an actor set(A) and more than one authorization service sets(AS).

Authorization service set(AS) is the set of services which can be executed by members of the actor set(A), the denotation $As(r)$ expresses the set of services which are authorized to the role r . Authorization unit has to finish in the specified time($lifetime$). Authorization unit may experience different states:

- sleep state, which expresses that authorization unit is deactivated;
- ready state, which expresses that run conditions have been met and authorization unit is ready to run;
- activated state, which expresses that authorization unit is in the run state;
- suspended state, which expresses that authorization unit is suspended for some reason.

B. Derivation of risk lattice model

In service-oriented workflow access control models, the execution of a task is eventually broken into executions of several authorization service sets, which are executed by the actor of the authorization unit. This paper formulates an access policy($policy$) as a 2-tuple of the form: $\langle actor, AS \rangle$, and $actor = \langle u, r, lifetime \rangle$, AS is the set of authorization services which $actor$ has. The set of access policies is denoted as $POLICY$.

The theory about poset and lattice can be referenced in [11].

According to the risk of access policies, we define the partial order of access policies:

$$\cong : policy \rightarrow policy$$

$\forall policy_1, policy_2 \in POLICY, policy_i = (actor_i, AS_i) (i=1,2), actor_1, actor_2 \in A, AS_i$ is the set of authorization services which $actor_i$ has ($i=1,2$). Every actor has its own security level expressed as $Safe(actor)$, which varies with the system environment and the user's load.

When considering the partial order, there are three cases. First, if actors of access policies have the same security level and access policies have different sets of authorization services, then the senior set of authorization services, the greater the risk. That is to say if $Safe(actor_1) = Safe(actor_2)$ and $AS_1 \subseteq AS_2$, then $policy_1 \cong policy_2$. Secondly, if access policies have the same set of authorization services and security levels of actors are different, then the lower security level of actor, the greater the risk. That is to say if $Safe(actor_1) \leq Safe(actor_2)$ and $AS_1 = AS_2$, then $policy_2 \cong policy_1$. In addition, another case is that security levels of actors are different and access policies have different sets of authorization services. If $Safe(actor_2) \leq Safe(actor_1)$ and $AS_1 \subseteq AS_2$, then $policy_1 \cong policy_2$, $(u_1, AS_1) \leq (u_2, AS_2)$. This means that if the actor whose security level is lower executes senior set of authorization services, then the risk is greater.

Theorem 1: ($POLICY, \cong$) is a lattice.

Proof: $\forall policy_1, policy_2, policy_3 \in POLICY, policy_i = (actor_i, AS_i) (i=1,2,3), actor_1, actor_2, actor_3 \in A, AS_i$ is the set of authorization services which $actor_i$ has ($i=1,2,3$). Because $Safe(actor_1) \leq Safe(actor_2)$ and $AS_1 \subseteq AS_2$, $policy_1 \cong policy_2$, i.e. $(actor_1, AS_1) \leq (actor_2, AS_2)$. Therefore, the relation $\cong$ is

reflexive. If $(actor_1, AS_1) \equiv (actor_2, AS_2)$ and $(actor_2, AS_2) \equiv (actor_1, AS_1)$, then $(actor_1, AS_1) = (actor_2, AS_2)$. Therefore, the relation $\equiv$ is anti-symmetric. If $(actor_1, AS_1) \equiv (actor_2, AS_2)$ and $(actor_2, AS_2) \equiv (actor_3, AS_3)$, then $(actor_1, AS_1) \equiv (actor_3, AS_3)$, so the relation $\equiv$ is transitive.

It can be seen that $(POLICY, \equiv)$ is a poset. The next part is to prove that there must be supremum and infimum of any two elements of $(POLICY, \equiv)$.

If $policy_1$ and $policy_2$ are comparable, then the supremum of $policy_1$ and $policy_2$ is denoted as: $lub(policy_1, policy_2) = \max(policy_1, policy_2)$, the infimum is denoted as: $glb(policy_1, policy_2) = \min(policy_1, policy_2)$.

If $policy_1$ and $policy_2$ are not comparable, we can construct an element $policy_3 = (actor_3, AS_3)$, making $Safe(actor_3) = \max(Safe(actor_1), Safe(actor_2))$ and $AS_3 = AS_1 \cap AS_2$. Therefore, $Safe(actor_1) \equiv Safe(actor_3)$, $Safe(actor_2) \equiv Safe(actor_3)$, $AS_3 \subseteq AS_1$, $AS_3 \subseteq AS_2$, i.e. $policy_3 \leq policy_1$ and $policy_3 \leq policy_2$. Thus we can see that $policy_3$ is the lower bound of $policy_1$ and $policy_2$.

Assuming that $policy_4 = (actor_4, AS_4)$ is any lower bound of $policy_1$ and $policy_2$, then $Safe(actor_1) \equiv Safe(actor_4)$, $Safe(actor_2) \equiv Safe(actor_4)$, $AS_4 \subseteq AS_1$, $AS_4 \subseteq AS_2$. And $Safe(actor_3) = \max(Safe(actor_1), Safe(actor_2))$, $AS_3 = AS_1 \cap AS_2$, so $Safe(actor_3) \equiv Safe(actor_4)$, $AS_4 \subseteq AS_3$. Then we can know $(actor_4, AS_4) \equiv (actor_3, AS_3)$, i.e. $policy_4 \leq policy_3$. Thereby $policy_3$ is the biggest lower bound of $policy_1$ and $policy_2$.

So there is infimum of any two access policies.

Similarly, we can prove that there is supremum of any two access policies.

Thus, $(POLICY, \equiv)$ is a lattice.

In addition, authorization services have their own risks. High-risk authorization services will bring great risk.

Generally speaking, authorization service set meets $AS \subseteq OP \times O$, that is, safety of authorization service depends on the security level of access object o_s and risk of access operation op_s . Usually, there are several access operations, which are denoted as *read*, *write* and *execute*, the risks of them are in ascending order. The values of o_s and op_s are set according to the specific environment.

Based on the $(POLICY, \equiv)$ lattice, we introduce the risk level of the authorization service set, and it can be used to determine whether the node of the $(POLICY, \equiv)$ lattice is available. If the risk level of the authorization service set exceeds a certain value, then the node is unavailable, in order to avoid high-risk authorization services being executed. Then we define a 3-tuple of the form: $\langle actor, AS, is \rangle$, and is describes whether the risk of AS is beyond the tolerance. If it is beyond the tolerance, then $is=0$, conversely, $is=1$.

Supposing the set of actors $A = \{actor_1, actor_2\}$, the set of services $S = \{s_1, s_2\}$ and $Safe(actor_2) \equiv Safe(actor_1)$. There are $4(4=2^{|S|}=2^2)$ possible sets of authorization services. The risk lattice model is as the Figure 1.

We should choose an access policy with less risk in order to ensure system security. The risk lattice model only allows the

flow of information to flow from high to low in the lattice. Thus it can be used to perform multi-level security policy [12].

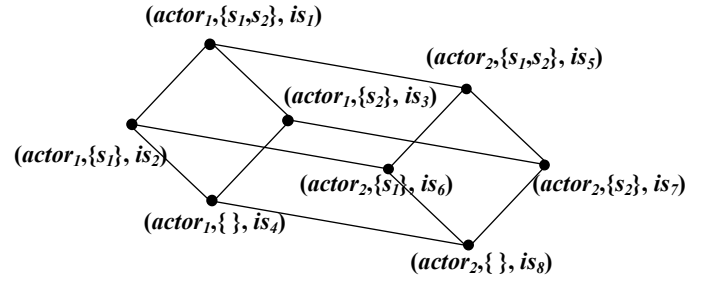


Figure 1. Example of a risk lattice model

IV. RISK LATTICE MODEL IN MULTI-DOMAIN ENVIRONMENTS

In multi-domain environments, not only tasks of workflows may be distributed in different autonomous domains, but also tasks are cross-domain, for example, the executor (user) of a task and the resources(objects) belong to different domains. This is the interoperation in multi-domain environments.

A. Role-based Security Interoperation

If a role mapping mechanism which does not violate the principles of security interoperation can be established, so that all resources and information can be shared by users in multi-domain environments.

About the security interoperations in multi-domain environments, some functions are given in the following:

$M(d_a, d_b)$ returns the role mapping between two autonomous domains d_a, d_b . In figure 2, $M(d_a, d_b) = \{ \langle ra_2, d_a, rb_2, d_b \rangle, \langle ra_3, d_a, rb_1, d_b \rangle, \langle rb_1, d_b, ra_3, d_a \rangle \}$, where, ra_1, ra_2, ra_3 are the roles in autonomous domain d_a , rb_1 and rb_2 are the roles in autonomous domain d_b .

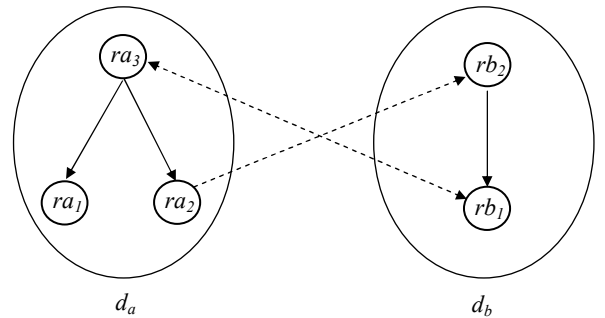


Figure 2. Example of role mapping between two different domains

Let r be a role in autonomous domain d_a , $L(r, d_b)$ is the function between d_a, d_b , which returns all the roles mapped by the role r in autonomous domain d_b . $ASD(r, d_b)$ returns all the authorization services that r can have in d_b . In Figure 2,

$L(ra_3, d_b) = rb_1$; $L(ra_2, d_b) = rb_2$; $L(rb_1, d_a) = ra_3$.

Based on interoperations in multi-domain environments, this paper defines the combination operations between roles as follows:

$$As(r_1 \oplus r_2) = As(r_1) \cup As(r_2); \quad As(r_1 * r_2) = As(r_1) \cap As(r_2);$$

$$As(r_1 \& r_2) = As(r_1) - As(r_2); \quad As(\neg r) = \sim As(r).$$

Operation $\oplus, *, \&, \neg$ are between roles, which return the role whose authorization services equal to the result of $\cup, \cap, -, \sim$ between authorization service sets of the roles. For example, $r_1 \oplus r_2$ returns the role whose authorization services is $As(r_1) \cup As(r_2)$.

According to the definition of algebraic systems, roles of an autonomous domain and operations between them $\oplus, *, \&, \neg$ constitute an algebraic system. Here, two autonomous domains d_a, d_b can be seen as two algebraic systems. L is a mapping function from an autonomous domain d_a to another autonomous domain d_b . It can be proved that function $L(r, d_b)$ is a homomorphism mapping from algebraic system d_a to algebraic system d_b .

Theorem 2: $L(r, d_b)$ is a homomorphism mapping.

Proof: To prove it, we must prove the following equations, supposing r_1, r_2 are the roles of domain d_a :

- $L(r_1 \oplus r_2, d_b) = L(r_1, d_b) \oplus L(r_2, d_b);$
- $L(r_1 * r_2, d_b) = L(r_1, d_b) * L(r_2, d_b);$
- $L(r_1 \& r_2, d_b) = L(r_1, d_b) \& L(r_2, d_b);$
- $L(\neg r_1, d_b) = \neg L(r_1, d_b).$

Firstly, $L(r_1 \oplus r_2, d_b)$ returns a role r in d_b , let's assume that there is a virtual role in d_a , its authorization services are $As(r_1) \cup As(r_2)$, then its mapping role in d_b is r . Now supposing that there are two mappings between these two domains: $\langle r_1, d_a, r_3, d_b \rangle$ and $\langle r_2, d_a, r_4, d_b \rangle$, then $L(r_1, d_b) = r_3, L(r_2, d_b) = r_4$. So $L(r_1 \oplus r_2, d_b) = r_3 \oplus r_4$. $r_3 \oplus r_4$ returns the role whose authorization services are $As(r_3) \cup As(r_4)$, denoting this role as r' . We can easily prove $As(r) \subseteq As(r')$, and $As(r') \subseteq As(r)$, so we get $L(r_1 \oplus r_2, d_b) = L(r_1, d_b) \oplus L(r_2, d_b)$.

Similarly, we can prove $L(r_1 * r_2, d_b) = L(r_1, d_b) * L(r_2, d_b)$, $L(r_1 \& r_2, d_b) = L(r_1, d_b) \& L(r_2, d_b)$, $L(\neg r, d_b) = \neg L(r, d_b)$, so $L(r, d_b)$ is a homomorphism mapping.

With the homomorphism relation, risk assessments in different domains can be mapped to be in an algebraic system.

B. Derivation of risk lattice model in multi-domain environment

The trust degree between domains denotes the trust extent between different domains. The trust is formed by role-mappings. It is denoted as $trust(d_i, d_j)$, i.e. the trust extent between domains d_i and d_j , and its value range is $[0, 1]$. If the user of inter-operation is in domain d_i , and the object is in domain d_j , then the higher trust degree $trust(d_i, d_j)$, the smaller the risk. So the trust degree is an important factor of the risk in multi-domain environments.

Based on the risk lattice model in III, we introduce the trust degree between domains $trust(d_i, d_j)$, define a 4-tuple of the form: $\langle actor, AS, trust(d_{actor}, d_{AS}), is \rangle$, which denotes an access policy of inter-operation ($policy_m$), the set is denoted as $POLICY_m$. The d_{actor} is the domain of $actor$, d_{AS} is the domain of AS . We propose a 3-dimensional lattice model, as Figure 3 shown.

In Figure 3, $trust(d_{actor1}, d_{\{s1, s2\}})$ is the weighted sum of $trust(d_{actor1}, d_{\{s1\}})$ and $trust(d_{actor1}, d_{\{s2\}})$, i.e.:

$$trust(d_{actor1}, d_{\{s1, s2\}}) = \alpha * trust(d_{actor1}, d_{\{s1\}}) + \beta * trust(d_{actor1}, d_{\{s2\}})$$

where, α and β denote the execution probability of s_1 and s_2 respectively, and $\alpha + \beta = 1$.

Similarly, we can get the formula:

$$trust(d_{actor}, d_{\{s1, s2, \dots, sn\}}) = \alpha_1 * trust(d_{actor}, d_{\{s1\}}) + \alpha_2 * trust(d_{actor}, d_{\{s2\}}) + \dots + \alpha_n * trust(d_{actor}, d_{\{sn\}})$$

$\alpha_1, \alpha_2, \dots, \alpha_n$ denote the execution probability of $s_1, s_2, \dots, s_n$ respectively, and $\alpha_1 + \alpha_2 + \dots + \alpha_n = 1$.

In the 3-dimensional lattice model, if the actors of inter-operations are the same and the sets of authorization services are the same, then the higher trust degree $trust(d_i, d_j)$, the smaller the risk. If the trust degrees are the same, then we can get the partial order of risk from the risk lattice model in III. In addition, there are another two cases:

1) Access policies of inter-operations have different actors, trust degrees are different, and the sets of authorization services are the same, then we can set weights for $Safe(actor)$

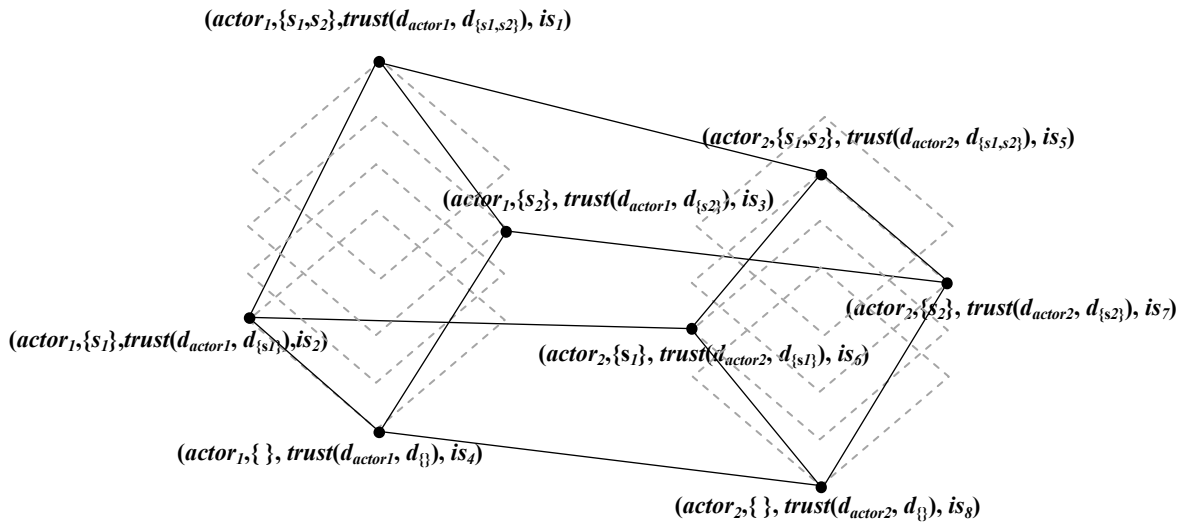


Figure 3. Example of risk lattice model in multi-domain environment

and $trust(d_i, d_j)$ according to the actual need respectively.

2) Sets of authorization services and trust degrees are both different, and access policies have the same actor, then we can set weights for the risk of authorization service sets and $trust(d_i, d_j)$ respectively.

So risks of inter-operations can be compared by this 3-dimension risk lattice model.

Similarly, we should choose the access policy with less risk in order to ensure system security. The risk lattice model can be used to perform multi-level security policy [12].

In addition, it is worth to note that if the set A or the set AS is great, then the 3-dimension risk lattice model may be too large. But in practical applications, only a part of the model is used.

V. RISK-BASED ACCESS POLICY SELECTION ALGORITHM

Workflow is worked by tasks and their dependencies, and the authorization unit is an instance of computer control. From III, we can see that when a task is executed, one or more authorization units will be generated. Each authorization unit contains more than one actors and more than one sets of authorization services. It is a very critical issue that how to choose the optimal access policy.

The selection algorithm is given below, which is based on the 3-dimensional lattice model:

1) Determine the status of authorization unit, if it is the ready state, jump to 2); otherwise, the algorithm ends.

2) An authorization unit usually contains a variety of access policies, and an access policy corresponds to a node of its 3-dimensional lattice model. All nodes corresponding to possible access policies of an authorization unit constitute the risk model. Choose a node ($actor_i, AS_i, trust(d_{actor_i}, d_{AS_i}), is_i$), util is_i is 1, jump to 3); Otherwise the algorithm ends.

3) Select the access policies with the minimum risk from the risk model. The set of them is $POLICY_{min}$, then jump to 4).

4) If $|POLICY_{min}|=1$ ($|POLICY_{min}|$ is the number of access policies in $POLICY_{min}$), then select the access policy in $POLICY_{min}$ as the execution one $policy_{exe}$, then jump to 6); Otherwise jump to 5).

5) If $|POLICY_{min}|>1$, this is due to the introduction of trust degree $trust(d_i, d_j)$, which makes that the risk of different access policies may be equal. In this case, choose the policy of the smallest $trust(d_i, d_j)$ as $policy_{exe}$, then jump to 6).

6) Activate the authorization services set of $policy_{exe}$, denoted as AS_{exe} , $actor_{exe}$ executes AS_{exe} , authorization unit enters the activated state, and timing begins.

7) If the task is finished or exceeds the allowed time *lifetime*, authorization unit enters the suspended status.

In this algorithm, the concept of risk is introduced into workflow access control model in multi-domain environments, when a task is being executed, it generates authorization units. The access policy with the least risk will be executed, so as to ensure the security of access events effectively.

In addition, the algorithm dynamically adjusts the authorization service set of the role. When a task is executed, the set of authorization services that the role required to perform is activated. When the task is finished or terminated,

the role no longer have the set of authorization services, so it can support the least of privilege.

VI. CONCLUSION

In multi-domain environments, the risk of access policy of workflow systems is affected by many factors. It is hard to make quantitative assessment. This paper proves that the risk partial order of access policies is a lattice model in mathematical, with which risks of access policies can be compared effectively, and it can ensure the implementation of multi-level security policy. In workflow systems, the access control of task is broken down into the access control of services. When a task is executed, there are more than one access policies. This paper proposes an algorithm to select the optimal policy with least risk based the risk lattice model, it can ensure the security of workflow systems effectively.

ACKNOWLEDGMENT

This work is supported by the National Natural Science Foundation of China(61103047), National Postdoctor Science Foundation of China(20100480936), Hunan Natural Science Foundation(11JJ4052).

REFERENCES

- [1] HOLLINGSWORTH D, Workflow Management Coalition: the Workflow Reference Model[S], Document Number WPMC-TC00-1003, Brussels, 1994.
- [2] LU Y and ZHANG L, "Types for task based access control in workflow systems," *ET Software*, vol. 2, no. 5, pp. 461-473, 2008.
- [3] Xu Wei and Wei Jun, "A Service-Oriented Workflow Access Control Model," *Journal of Computer Research and Development*, vol. 42, no. 8, pp. 1369-1375, 2005.
- [4] Xu Feng and Lai Haiguang, "Service-Oriented Role-based Access Control," *Chinese Journal of Computers*, vol. 28, no. 4, pp. 686-693, 2005.
- [5] B. Aziz, S. N. Foley, J. Herbert, and G. Swart, "Reconfiguring role based access control policies using risk semantics," *High Speed Networks*, vol. 15, no. 3, pp. 261-273, 2006.
- [6] L. Zhang, A. Brodsky, and S. Jajodia, "Toward information sharing:Benefit and risk access control (barac)," in *Proceedings of the Seventh IEEE International Workshop on Policies for Distributed Systems and Networks*, 2006, p.45-53.
- [7] R. McGraw, "Risk-adaptable access control (RADAC)," in *Privilege (Access) Management Workshop*, NIST-National Institute of Standards and Technology-Information Technology Laboratory, 2009.
- [8] P.-C. Cheng, P. Rohatgi, C. Keser, P. A. Karger, G. M. Wagner, and A. S. Reninger, "Fuzzy multi-level security: An experiment on quantified risk-adaptive access control," in *IEEE Symposium on Security and Privacy*, 2007, p. 222-230.
- [9] Li Ruixuan, Hu Jinwei, Tang Zhuo, and Lu Zhengding, "R2BAC: A risk-based multi-domain secure interoperation model," *Journal on Communications*, vol. 20, no. 10, pp. 58-69, 2008.
- [10] Tang Zhuo, Zhao Lin, Li Kenli, and Li Ruixuan, "A Dynamic Access Control Model for Inter-Operation in Multi-Domain Environment Based on Risk," *Journal of Computer Research and Development*, vol. 43, no. 6, pp. 948-955, 2009.
- [11] Davey B and Priestley H, *Introduction to Lattices and Order*, Cambridge: Cambridge University Press, 2002.
- [12] Denning D E, "A lattice model of secure information flow," *Comm ACM*, vol. 19, no. 5, pp. 236-243, 1976.