Trust and Risk based Access Control and Access Control Constraints

Nurmamat Helil[1, 2], Mucheol Kim[1] and Sangyong Han[1]
[1] School of Computer Science and Engineering, Chung-Ang University
Seoul, Korea
[e-mail: nurmamat@gmail.com, kmooky@ec.cse.cau.ac.kr, hansy@cau.ac.kr]
[2] College of Mathematics and System Sciences, Xinjiang University
Urumqi, China
*Corresponding author: Sangyong Han

Received April 8, 2011; revised September 4, 2011; accepted September 26, 2011; published November 29, 2011

Abstract

Access control in dynamic environments needs the ability to provide more access opportunities of information to users, while also ensuring protection information from malicious users. Trust and risk are essential factors and can be combined together in access control decision-making to meet the above requirement. In this paper, we propose the combination of the trust and risk in access control to balance information accessibility and protection. Access control decision is made on the basis of trustworthiness of users and risk value of permissions. We use potential relations between users and relations between permissions in access control. Our approach not only provides more access opportunities for trustworthy users in accessing permissions, but also enforces traditional access control constraints such as Chinese Wall policy and Separation of Duty (SoD) of Role-Based Access Control (RBAC) model in an effective way.

Keywords: Trust, risk, role-based access control, constraints

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2011-0003235), and also supported by Natural Science Foundation of Xinjiang Uygur Autonomous Region, China (grant number: 2009211B05), China National Natural Science Foundation (grant numbers: 10961024, 61063026, 61063043), and Natural Science Foundation of Xinjiang University, China (grant number: BS090103).

DOI: 10.3837/tiis.2011.11.022

1. Introduction

Access control meets a great challenge in the dynamic, decentralized environments where users are not only foreknown inner users, but also unknown "strangers". In addition, access control in these environments is required to be flexible enough to provide more access opportunities of information to users effectively under the premise of protecting information from malicious users' abuse. However, traditional access control models such as Discretionary Access Control (DAC) [1][2], Mandatory Access Control (MAC) [3][4][5] and Role-Based Access Control (RBAC) [6][7] lack flexibility to meet these above requirements; they are only suitable for addressing access control for predefined known users and access control policy itself is relatively static and rigid.

The concept of trust [8] has been introduced into access control. There are several research works [9][10][11][12][13][14][15][16][17] in access control research area in which access control decisions are made on the basis of access requester's trustworthiness. Unlike traditional access control models, trust-based access control uses quantified trust of users or the communities to which the users belong. A user's trust changes over time according to the user's profile, experience, recommendation, etc [15][16]. The higher trustworthiness the user has, the more resources he can access.

Risk is another basic factor which has been used in access control. Risk is an economic concept which is used in decision-making under uncertain circumstances in economics. In access control, it can be considered as a potential harm that may arise from access to permissions. There are quite a few literatures [18][19][20][21][22] that now exist in the area of access control research which use risk as a fundamental factor in making access control decision. In these works, access control decision can be determined by assessing the risk of user's access to a permission.

Trust and risk are both essentials in access control. Using trust and risk in access control serves the purpose of capturing the features of dynamic multi-centric environments. It is natural to combine trust and risk factors. There are also some works that propose combining the two factors together do determine user's access to permissions [23][14][25].

Access control decisions take both the user and the permission into consideration to determine if a user should be assigned to a permission. However, in trust-based access control, trust mainly concerns the user. User's trustworthiness only describes to what degree the user is reliable, it does not tell us how valuable the permission is, some permissions are much more valuable than others, trust-based access control lacks the ability to quantify the value of permission, whereas risk-based access control mainly concerns the permission; the more valuable the permission is, the more risk exists in accessing the permission. In risk-based access control, user's quantified trustworthiness is somehow neglected. Therefore, connecting quantified trust of user and risk of permission in access control is a non-trivial research issue.

In traditional access control such as RBAC [6][7], constraints are considered to be the principal motivation. Separation of Duty (SoD) is an important constraint in access control in which sensitive combination of permissions is decomposed to different users in order to prevent business fraud. E.g., a *sales order* requires two different persons to create and approve it separately. Chinese Wall policy [26] is a variant of SoD, the main idea of Chinese Wall policy is that a subject is only allowed to access information which does not incur conflict of interest with any other information that the subject already possesses. E.g., a *financial*

consultant can only access information of either *Company_A* or *Company_B*, but not both, if the two companies are competitors. [27] enumerates several kinds of constraints of the RBAC in detail. Access control constraints aim to prevent commercial fraud, i.e. there is a great risk if sensitive combination of permissions is not separated among different users or different communities. However, we find the definitions of these constraints are not flexible enough and are quite rigid. *User-based separation of duty* [27] considers sensitive combination of users via a role, it implies the combination of these users is sensitive with respect to access any permissions. *Role-based separation of duty* [27] considers sensitive combination of roles via a user and implies there is a possibility for a user to get sensitive combination of permissions via combination of roles. *Permission-based separation of duty* and *Object-based separation of duty* considers sensitive combination of permissions or objects respectively, and hypothetically says if the same user or users who have the same role access all of the permissions or objects from that combination, it will bring great risk. They regard any user who tries to access this sensitive combination of permissions as malicious user. In fact, from **Fig. 1**, we can see that in an access control scenario, there exists: (a) an independent user or a group of normal users trying to access to an independent permission, (b) an independent user or a group of normal users trying to access all permissions from risky combination of permissions, (c) a malicious user or users from a malicious user group trying to access all permissions from risky combination of permissions. Therefore, the SoD constraints and Chinese Wall policy are inadequate to flexibly describe all kinds of these varying situations.

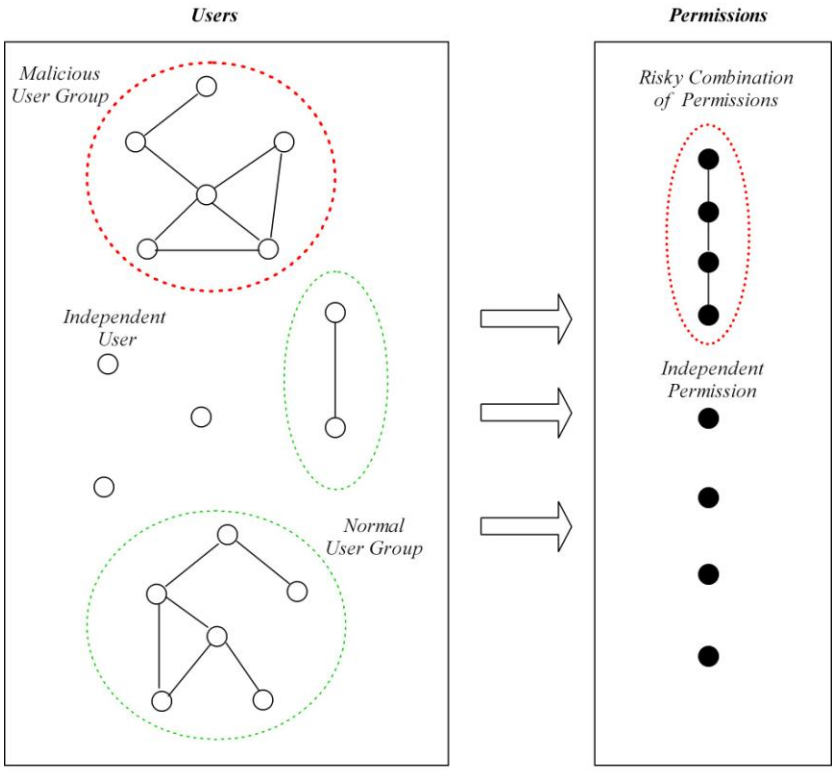

Fig. 1. user relations and permission relations in access control scenario

The inflexibility and rigidness of traditional access control triggers our intention; we first propose combining user trust and permission risk together to provide a flexible access control

for dynamic, distributed environments. In our proposal, we improve the accessibility of information to trusty users, while preventing malicious users' abuse of information. Our proposal not only creates more flexibility for access control, but also simulates traditional access control constraints such as SoD and Chinese wall policy in an effective way.

The rest of the paper is organized into the following sections: Section 2 briefly introduces the evaluation of user trust. Section 3 proposes a method of mining a potential user group. Section 4 gives definition of risky combination of permissions and objects. Section 5 enumerates the relationship between users and the permissions they access. An approach of adjusting user trust and permission risk according the real-time information for access control is developed in Section 6. Section 7 presents how we can simulate traditional access control constraints in an effective way. Section 8 examines other related works, and Section 9 draws conclusions and considers further work.

2. Trust Evaluation

The concept of "trust" can be used in many environments, including access control. Trust builds the basis for allowing a trustee to access resources provided by a truster. i.e. users' access to the resources provided by an application system is on the basis of user trust. There is no unique approach to evaluate user trust. Generally, user trust can be evaluated according to a user's profile, historical behavior and recommendation, and changes overtime because these factors changes overtime [16].

In distributed application systems, resources are accessed by various kinds of users, both from inside foreknown and outside unknown. So, the access control decision is based on user's trustworthiness. The system authenticates a user's personal identities provided by him and also checks user's historical access behavior to calculate the user trust and to decide whether to give him access to the resources. Here we partly follow the general method of trust evaluation, and briefly introduce personal identity trust and historical access trust in order to evaluate total trust of a user.

2.1 Personal Identity Trust

User authentication is the first step of traditional access control models, and users are required to provide some personal identities for authentication. In our proposal, users provide personal identities and exchange it with trust and access resources using trust. Suppose all possible personal identities provided by users and can be authenticated by the system or trusted third party in order to access the resources are listed in the personal identity set below:

$$PID = \{pid_1, pid_2, ..., pid_n\} \tag{1}$$

Different systems require different personal identities for identify of authenticate users. There are some common personal identities, for example, user name, address, phone number, credit card number, age, height, organization, job title, etc. Each identity is associated with a weight, determined by the security policy of the system, that reflects its importance with respect to the trustworthiness of this user. We denote them as $w_1, w_2, ..., w_n$.

A user's personal identities compose a subset of PID. personal identities provided by a user u is denoted as PID_u and we have

$$PID_u = \{pid_{u,1}, pid_{u,2}, ..., pid_{u,k}\} \in 2^{PID} \tag{2}$$

Trust values of identities provided by a user u is calculated by this formula below, if his personal identities are as Eq. 2:

$$PID_{u,t} = \sum_{i=1}^{k} w_{u,i} \tag{3}$$

2.2 Historical Access Trust

User trust evaluation also considers user's previous access history as user's experience. The user earns his historical access trust with "good" behavior and loses it with "bad" behavior. At the start, the default historical access trust of a user is zero, and the user gradually earns his historical access trust. If the user brings benefit to the system after he accesses a permission, then the access control system increases the historical access trust value. $H_{u,t}$ denotes historical access trust of a user u, $c(p)$ denotes the actual consequence of accessing permission p, it might be a benefit or loss to the system, then $H_{u,t}$ can be calculated by this formula below:

$$H_{u,t} = H_{u,t} + \eta c(p)$$

where η is a normalization factor. How to calculate historical access trust will be addressed more in section 6.

2.3 Trust Evaluation

User's entire trust can be calculated according to personal identities, history and recommendation. The system has the trust calculation policy as below:

$$\mathrm{T}_u = W_{PID} \times PID_{u,t} + W_H \times H_{u,t} + W_R \times R_{u,t} \tag{4}$$

In the above equation, $PID_{u,t}$, $H_{u,t}$ and $R_{u,t}$ denotes personal identity trust, historical access trust and recommendation trust respectively. $W_{PID}, W_H, W_R \in [0,1]$; $W_{PID} + W_H + W_R = 1$, W_{PID} is a personal identity factor in calculation of trust value; W_H is historical access factor and W_R is a recommendation factor respectively [16]. The trust in our proposal is mainly involves the personal identity trust and historical access trust. For the sake of independence between application systems, we assume $W_R = 0$ in Eq. 4. So we omit the definition of recommendation trust here.

3. Potential User Group

User's personal identity can tell the access control system many information. Different users may resemble each other in their partial personal identities, moreover these users may also have similar or common interests in accessing the resources. Mining the similarity between personal identities of users and their common interests can help access control system to adjust the access control policy to provide more flexible access control service.

We discuss the similarity between personal identities provided by users first. Identities provided by two users a and b may have similarities:

$$PID_a \cap PID_b = \{pid_1, pid_2, ..., pid_t\}, \quad 0 \le t \le n \tag{5}$$

We first define personal identity similarity.

$$Sim_{pid}(a,b)=\sum_{i=1}^{n}\alpha_i sim(pid_{a,i},pid_{b,i}),\quad \sum_{i=1}^{n}\alpha_i=1,\quad \alpha_i\geq 0 \tag{6}$$

α_i is the importance of similar identity in calculating the personal identity similarities between two users. Similarity of some personal identities is much more important than others in determining the similar user group members. For example, users from same organization might have much similarity than users of same ages in accessing resources.

We denote same personal identities as $\approx$, and value of a personal identity pid as $v(pid)$. Concerning personal identity similarity, we have this rule: if $PID_a \cap PID_b=\varnothing$, then $Sim_{pid}(a,b)=0$.

We have also these rules below for personal identity similarity.

- $sim(pid_{a,i},pid_{b,i})=0$, if $pid_{a,i}\not\approx pid_{b,i}$.
- $sim(pid_{a,i},pid_{b,i})=0$, if $pid_{a,i}\approx pid_{b,i}$, but $v(pid_{a,i})\not\equiv v(pid_{b,i})$.
- $sim(pid_{a,i},pid_{b,i})=1$, if $pid_{a,i}\approx pid_{b,i}$, and $v(pid_{a,i})\equiv v(pid_{b,i})$.

Other than personal identities, users' common access interests also tell the system if these users are similar. Users' common access interests can be another kind of basis for predicting the similar user group members. We can consider common permissions requested (both permitted and denied by the system) and the frequency of these requests of users in measuring user similarity [28], Assume all permissions in a system are $PRMS=\{p_1,p_2,...,p_N\}$, we denote the total number of times that user a request permission p_k as $req\_n(a,p_k)$, then we can define request history similarity between two users a and b as below:

$$Sim_{req}(a,b)=\frac{\sum_{k}^{N}req\_n(a,p_k)\times req\_n(b,p_k)}{\sqrt{\sum_{k}^{N}(req\_n(a,p_k))^2}\times\sqrt{\sum_{k}^{N}(req\_n(b,p_k))^2}} \tag{7}$$

According to the personal identity similarity and request history similarity, we can define user similarity as below:

$$SIM(a,b)=\alpha Sim_{pid}(a,b)+\beta Sim_{req}(a,b),\quad \alpha+\beta=1 \tag{8}$$

Potential User Group: For $\forall u_i\in\{u_1,u_2,...,u_t\}$, $\exists u_j\in\{u_1,u_2,...,u_t\}$ and $SIM(u_i,u_j)\geq SIM_{Max}$, then we say $\{u_1,u_2,...,u_t\}$ is a potential user group and denote it as PUG, SIM_{Max} is a threshold point for user similarity.

Through this method, to some extent, we can mine some potential user groups whose members might have common interest in accessing resources in future. Potential user group describes a group of people who compose a temporal virtual organization according to their similarity in personal identities and request history. There are different criteria for access control system to define a potential user group when considering the personal identities. Sometimes, people who have same email domain might compose a potential user group and sometimes working role.

Mining a potential user group is important in access control. Predicting a group of users who have close resemblance in personal identities and access request helps the system preventing these users from accomplishing a sensitive task.

Access control systems should be highly aware of malicious users. So, we are much more interested in the potentially malicious user group. Potentially malicious user group members have a common goal to achieve: they want to accomplish a sensitive task through distribution of permissions which composes that task to the different group members preventing detection before achieving their goal. If they were to achieve their goal, it would bring a lot of lost for the application system. However, it is difficult to differentiate potentially malicious user group from potentially normal user group, so we are cautious to potential user group which might turns out to be potentially malicious user group.

4. Risky Combination of Permissions

In access control, there are two main components: users and permissions. We define access control components as below:

- $USERS$, $PRMS$, OPS and OBS (users, permissions, operations and objects respectively).
- $PRMS \subseteq OPS \times OBS$, a many-to-many mapping operation-to-object, describing permissions.

Accessing a permission brings the application system benefit or loss. The application system estimates the benefit or loss for each permission. How to estimate is out of our consideration, therefore, we omit it from our paper. We denote corresponding loss and benefit of permission p as $l(p)$ and $b(p)$. The probability of the loss of accessing permission p is denoted as $prob_l(p)$.

Risk: Risk value of a permission p is denoted as $risk(p)$ and we have this formula:

$$risk(p) = prob_l(p) \times l(p) \tag{9}$$

Permission-based separation of duty constraint of RBAC model considers sensitive combination of permissions, i.e. a group of permissions cannot be accessed by the same users or users who have the same role, otherwise there would be conflict of interest, meaning it is very risky for the application system to provide access to the group of permissions for specific group of users. We define our historical risky combination of permissions here.

Historical Risky Combination of Permissions: We denote combined risk of permissions $p_1, p_2, ..., p_n$ as $risk(\{p_1, p_2, ..., p_n\})$. If in a specific time period $[T_a, T_b]$, for any combination of t permissions $\{p_1, p_2, ..., p_t\}$, $t = 2,3,...,n$ from $p_1, p_2, ..., p_n$,

$$risk(\{p_1, p_2, ..., p_t\}) \gg \sum_{i=1}^{t} risk(p_i)$$

then we say combination of permissions $p_1, p_2, ..., p_n$ is a historical risky combination of permissions, and denote it as $HRC\_P = \{p_1, p_2, ..., p_n\}$. "Historical" here refers to the specific time period.

If $HRC\_P = \{p_1, p_2, ..., p_n\}$, and a user or potential user group members invoke all these permissions p_1, p_2, ..., p_n to complete a task (a task execution is equal to successful access of all these permissions), then it will bring huge risk to the application system. Under this circumstance, we define conditional risk, and we have this formula below:

$$risk(p_t \mid p_1 p_2 ... p_{t-1}) = risk(\{p_1, p_2, ..., p_t\}),\ t = 2,3,...,n$$

Definition of HRC_P is significantly different from *permission-based separation of duty* constraint. *Permission-based separation of duty* considers that the potential huge extra risk comes only after all of the permissions from sensitive combination of permissions are successfully accessed by same user, but in our proposal, the potential extra risk except the risk defined in Eq. 9 accumulates gradually when these permissions accessed by similar users successively in a specific time period.

Permission is composed of an operation on an object. Considering the historical risky combination of permissions covers the historical risky combination of objects, we just discuss permissions. We can also denote Historical Risky Combination of Objects as $HRC\_OB = \{ob_1, ob_2, ..., ob_n\}$, and definition of it is similar to HRC_P. We describe users' access to the permissions from the risky combination in a given time period. The time period might be a session, a working day, etc.

5. Relations between Users and Permissions

In previous sections 3 and 4, we discussed potential user group and risky combination of permissions. In this section, we enumerate several scenarios for users' access to permissions.

Independent Users and independent Permissions: a single independent user accesses an independent permission. See part (a) of **Fig. 2**. For this scenario, we have the principle that the user is only responsible for his own behavior and does not influence other users; the predefined risk value of the permission does not have to be adjusted when any other users want to access this permission after this user's access.

Independent Users and Risky Combination of Permissions: Mutually unrelated users access to permissions from a risky combination of permissions separately. See part (b) of **Fig. 2**. For this scenario, we have the assumption that mutually independent users don't have the same goal to achieve through accessing these permissions, so the predefined risk values of these permissions do not have to be adjusted before access from different users. If several users attempt to access a sensitive combination of permissions to achieve a goal, they must have some relations or similarity, otherwise the possibility of having a common goal is very small.

Potential User Group and Risky Combination of Permissions (1): Potential user group is composed of one user; this user tries to access permissions from a risky combination of permissions. See part (c) of **Fig. 2**. Chinese Wall Policy and SoD mainly consider this kind of situation. In this scenario, if this user tries to access all permissions from a risky combination of permissions, we have to adjust the risk values of these permissions before making access control decision. This is different from traditional constraints but is a more flexible approach than them.

Potential User Group and Risky Combination of Permissions (2): This is a general case of previous scenario. Potential user group is composed of more than one user, users from this group try to access permissions from a Risky combination of permissions. See part (d) of **Fig. 2**. In this scenario, if users from this *PUG* try to access all permissions from the risky combination of permissions, we have to adjust risk value of permissions before making access control decisions.

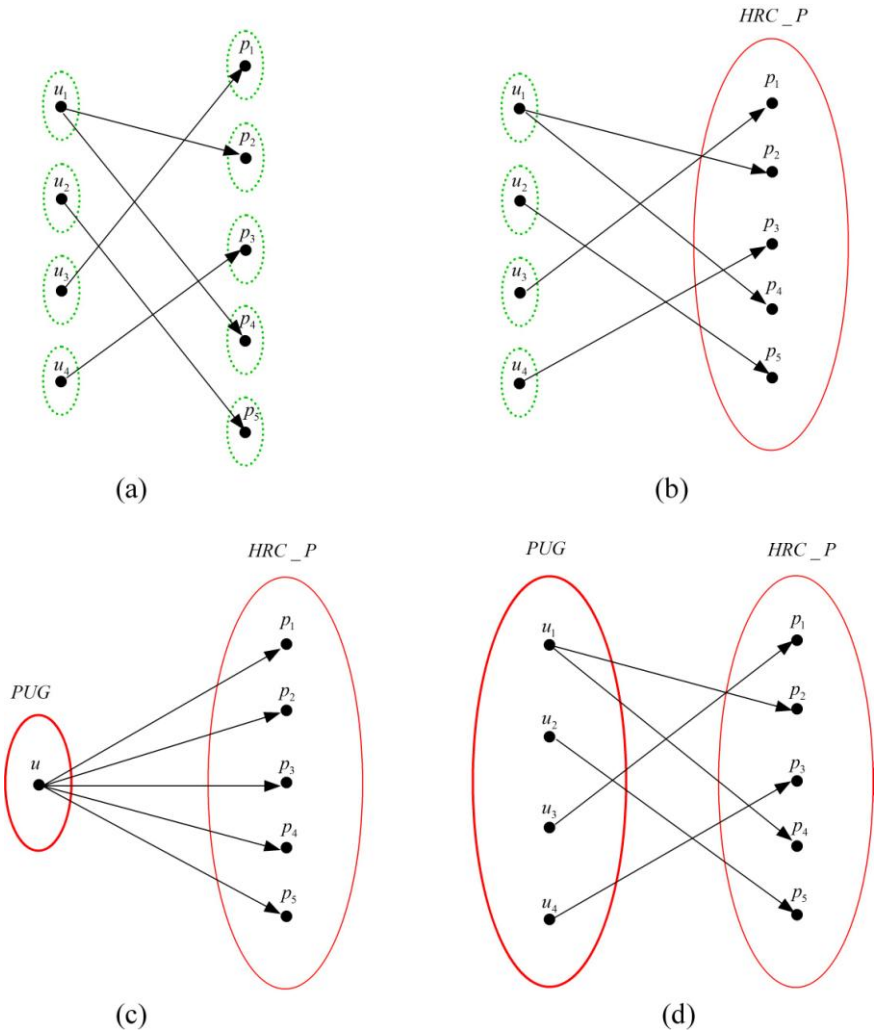

Fig. 2. examples of relations between users and permissions

6. Access Control Based on Trust and Risk

If independent users try to access permissions $p_1, p_2, ..., p_n$ (not necessarily a risky combination), then according to the previous section we have the following:

$$risk(p_2 \mid p_1) = risk(p_2)$$
$$risk(p_3 \mid p_1 p_2) = risk(p_3)$$
$$\ldots$$
$$risk(p_n \mid p_1 p_2 \ldots p_{n-1}) = risk(p_n)$$

6.1 Potential User Group and Risky Combination of Permission

When we say combination of permissions is risky, we cannot neglect the fact that similar users might access these permissions to accomplish a sensitive task to get their goal. Please see Chinese wall policy [26], Static Separation of Duty (SSoD) and Cardinality Constraints in

RBAC [7][27] for detail. So when we discuss these combination, we take potential user group into consideration.

Global Potential Risk: we denote risk difference between combined risk of permissions and the sum of risk of each permission as $risk_{\Delta}$, and name it global potential risk. If $p_1, p_2, ..., p_n$ are risky combination of permissions, then we have this below:

$$risk_{\Delta} = risk(\{p_1, p_2, \ldots, p_n\}) - \sum_{i=1}^{n} risk(p_i) \tag{10}$$

Generally, higher trustworthy users are authorized to access higher risky permissions and vice versa. If we consider the trust value of a user as the user's security label and risk value of permission as the security label of the permission, then as access control connects the user to the permission, we establish the relationship between them. For a user u to access permission p, we have this basic formula:

$$\mathrm{T}_u \geq \theta risk(p) \tag{11}$$

where the θ is the normalization factor, Eq. 11 means if a user u is trustworthy enough, then he can access the permission p. In order to prevent malicious users from accessing permissions from risky combination of permissions, access control system have three options:

- Increase risk value of permissions.
- Reduce trust value of users.
- Increase risk value of permissions and reduce trust value of users.

6.1.1 Increasing Risk Value of Permissions

We first discuss increasing risk value of permission. If same user or users from same potential user group try to access risky combination of permissions, then the access control system tries to make it difficult. We use this sigmoid function below in our proposal for adjusting the risk value of permissions to reflect the global potential risk:

$$y = \varsigma(x) = \frac{1}{1+e^{-\alpha(x-n)}}, \quad x \in N,\ 0 \leq x \leq n,\ \alpha > 0 \tag{12}$$

Assume that $HRC\_P = \{p_1, p_2, ..., p_n\}$ is a risky combination of permissions, and $PUG = \{u_1, u_2, \ldots, u_m\}$ is potential user group. If user u_1 wants to access permission p_1, and $\mathrm{T}_{u_1} \geq \theta risk(p_1)$, then he can obtain the permission. After this, if any user $u_i \in PUG$ wants to access permission $p_j \in HRC\_P$, $p_j \neq p_1$, then we have to adjust the Eq. 11. Because it assumes the combination of the two permissions contain potential risk. Assuming u_2 wants to access permission p_2, then we have this formula

$$\mathrm{T}_{u_2} \geq \theta risk(p_2 \mid p_1) \tag{13}$$

and

$$risk(p_2 \mid p_1) = risk(p_1) + risk(p_2) + 2risk_{\Delta} \frac{1}{1+e^{-\alpha(2-n)}} \tag{14}$$

Similarly, for u_3 and p_3, we have this formula:

$$\mathrm{T}_{u_3} \geq \theta risk(p_3 \mid p_1 p_2) \tag{15}$$

and

$$risk(p_3 \mid p_1 p_2) = risk(p_1) + risk(p_2) + risk(p_3) + 2risk_{\Delta} \frac{1}{1+e^{-\alpha(3-n)}} \tag{16}$$

For the last permission p_n and user u_m, we have this formula:

$$\mathrm{T}_{u_m} \geq \theta risk(p_n \mid p_1 p_2 \ldots p_{n-1}) \tag{17}$$

and

$$\begin{aligned} risk(p_n \mid p_1 p_2 \ldots p_{n-1}) &= risk(p_1) + risk(p_2) + \ldots + risk(p_n) + 2risk_{\Delta} \frac{1}{1+e^{-\alpha(n-n)}} \\ &= risk(p_1) + risk(p_2) + \ldots + risk(p_n) + risk_{\Delta} \\ &= risk(\{p_1, p_2, \ldots, p_{n-1}, p_n\}) \end{aligned} \tag{18}$$

We emphasize that if a user from a specific potential user group wants to access a permission from HRC_P, then the access control system considers conditional risk for his access request. for example, same potential user group members have already accessed permissions $p_1, p_2, \ldots, p_t$ out of risky combination of permissions $HRC\_P = \{p_1, p_2, \ldots, p_t, p_{t+1}, \ldots, p_n\}$, and if user u wants to access permission p_{t+1}, then the access control system should consider $risk(p_{t+1} \mid p_1 p_2 \ldots p_t)$. According to the feature of the sigmoid function, the more permissions accessed by same potential user group members, the faster the conditional risk increases and finally equals to the combined risk $risk(\{p_1, p_2, ..., p_n\})$. However, the system usually is only capable of estimating the combined risk $risk(\{p_1, p_2, ..., p_n\})$, so we use the sigmoid function to distribute the global potential risk $risk_{\Delta}$ to sub-combination of permissions from HRC_P.

6.1.2 Reducing Trust Value of Users

Making access control decision still uses the Eq. 11. According to this equation, under the circumstance that the user brought loss to the application system, we can reduce the user's trust value after his access to a permission, in order to prevent further loss. In Eq.4, there is a historical access trust. At the beginning, the historical access trust of a user is zero, and the user gradually earns his historical access trust. If user brings benefit to the system after he accessed a permission, then the access control system increases the historical access trust value of that user using $H_{u,t} = H_{u,t} + \eta b(p)$ according to the actual benefit the user brought to the system, and in return the total trust value of the user will be increased. The reduced trust value due to user's access to a permission is directly related to the loss the user brought to the system:

$$H_{u,t} = H_{u,t} - \eta l(p) \tag{19}$$

6.1.3 Trust and Risk Based Access Control for Risky Combination of Permissions

We connect user's trustworthiness with the risk value of the permission in the user's access to the permission. User's access to a permission takes user's trust value and risk value of permission, which the user requests to access, into consideration. The access control system tries to find the balance between sharing information and protecting information. If the users are trusty enough, risky permissions should also be accessible. The more risk the permission has, the more trustworthiness is needed from the user to access the permission.

Table 1. access control workflow for risky combination of permissions

1. user u_1 sends an access request of p_1 to the system. System asks the user to provide the minimum necessary personal identities $pid_1, pid_2, \ldots, pid_k$, system calculates T_{u_1} using Eq. 4.
   a) **if** $T_{u_1} \geq \theta risk(p_1)$, then u_1 can access p_1, system puts u_1 into a new PUG, that is $PUG = \{u_1\}$, and puts $< u_1, p_1 >$ as an access list into A_l, that is $A_l = \{< u_1, p_1 >\}$.
   b) **else** system denies the user's access request to p_1, and reset the whole process.
2. user u_2 (can be a user from PUG) sends an access request of p_2 ($p_2 \notin range(A_l)$) to the system. System asks the user to provide $pid_1, pid_2, \ldots, pid_k$, system calculates T_{u_2} using Eq. 4, and $SIM(u_2, u_i)$ for $\forall u_i \in PUG$.
   a) **if** $\exists u_i$, $SIM(u_2, u_i) \geq SIM_{Max}$, then $PUG = PUG \cup \{u_2\}$, system adjusts the risk value of p_2 using Eq.14,
      1) **if** $T_{u_2} \geq \theta risk(p_2 \mid p_1)$, then u_2 can access p_2, $A_l = A_l \cup \{< u_2, p_2 >\}$.
      2) **else** system denies the user's access request to p_2.
   b) **else**
      1) **if** $T_{u_2} \geq \theta risk(p_2)$, then u_2 can access the permission p_2.
      2) **else** system denies the user's access request to p_2.
3. user u_j (can be a user from PUG) sends an access request of permission $p_t \in HRC\_P$ ($p_t \notin range(A_l)$) to the system. System repeats the similar process with 2, when needs adjusting risk value of p_t, use $risk(p_t \mid p_1 p_2 \ldots p_{t-1})$.

Here in this subsection, we present how to establish a potential user group and making access control decision for their access request. At the start, we have no potential user groups; we gradually establish such groups, being cautious in giving access permissions from a risky combination of permissions to them. We try to discover potential user group only when users want to access permissions from a same risky combination of permissions, i.e. our discovery process is in a lazy mode.

User clustering techniques can be used in our proposal. When we think of risky combination of permissions, we are mainly concerned with preventing access to all these permissions from a group of people who have a common goal in accessing these permissions. Therefore, we propose a user clustering method to detect these groups. Some personal identities are strictly required from the user. We should be aware of users who have same attribute value of these required personal identities. For example, for some risky combination of permissions, we should be aware of users from same organizations and have same position at that organization; so we may require users to provide organization id and position at that organization. We describe the workflow of access control based on user's trust value and risk value of permissions from risky combination of permissions.

Assume we have a risky combination of permissions $HRC\_P = \{p_1, p_2, \ldots, p_n\}$, restricted time period is $[T_a, T_b]$, and mandatorily required personal identities are $pid_1, pid_2, \ldots, pid_k$.

Users provide optional personal identities with the purpose of earning trust value. In the case of clustering users with their personal identities, we make the following adjustment to Eq. 6:

$$Sim_{pid}(a,b) = \sum_{i=1}^{n} \alpha_i sim(pid_{a,i}, pid_{b,i}),$$

$$\sum_{i=1}^{n} \alpha_i = 1, \quad \alpha_1 + \alpha_2 + \ldots + \alpha_k \gg \alpha_{k+1} + \alpha_{k+2} + \ldots + \alpha_n \geq 0$$

The reason why these required personal identities so important is that people have similarity in these personal identities have more chance or more motivation of doing a task which is composed of these permissions from HRC_P to reach their goal. **Table 1** describes access control workflow for $HRC\_P = \{p_1, p_2, ..., p_n\}$. The whole access process ends if all of the permissions from the HRC_P accessed from the PUG members or the restricted time period is up.

7. Access Control Constraints in Trust and Risk Based Access Control

Risky or sensitive combination of permissions or objects are important concepts in access control, Chinese Wall policy and SSoD are discussed in different literatures [7][26][27] to present sensitive combination of permissions or objects. Let us justify the effectiveness of our proposal in enforcing these constraints.

7.1 Chinese Wall Policy

Chinese wall policy $(U, \{ob_1, ob_2\}, op)$ claims that user u is permitted to do an operation op only on one of the objects ob_1 and ob_2, but not both. We can enforce this kind of constraint using our trust and risk based access control. Assuming that $HRC\_OB = \{ob_1, ob_2\}$, and a user tries to do same operation on objects ob_1 and ob_2, then combined risk value of these two objects would be $risk(\{ob_1, ob_2\})$. And we also assume that user id uid and user name $uname$ is the personal identities provided by users mandatorily when they try to access the resources. So Eq. 6 is changed to

$$Sim_{pid}(u_1, u_2) = \alpha_1 sim(uid_{u_1}, uid_{u_2}) + \alpha_2 sim(uname_{u_1}, uname_{u_2}),$$

$$\sum_{i=1}^{n} \alpha_i = 1, \quad \alpha_1 + \alpha_2 = 1, \quad \alpha_3, \alpha_4, \ldots, \alpha_n = 0$$

We also adjust Eq.8 as below:

$$SIM(a,b) = \alpha Sim_{pid}(a,b) + \beta Sim_{req}(a,b), \quad \alpha=1, \beta=0 \tag{20}$$

That means $SIM(u_1, u_2) = 1$, if $u_1 = u_2$, and $SIM(u_1, u_2) = 0$, if $u_1 \neq u_2$, so every user construct a potential user group according to our definition (see Section 3). We have these formulas:

$$y = \varsigma(x-2) = \frac{1}{1+e^{-(x-2)}}, \quad x \in N, \ 0 \leq x \leq 2, \ \alpha=1$$

$$risk(\{ob_1, ob_2\}) \gg risk(ob_1) + risk(ob_2)$$

$$risk_{\Delta} = risk(\{ob_1, ob_2\}) - (risk(ob_1) + risk(ob_2))$$

$$\begin{aligned} risk(ob_2 \mid ob_1) &= risk(ob_1) + risk(ob_2) + 2risk_{\Delta} \frac{1}{1+e^{-(2-2)}} \\ &= risk(ob_1) + risk(ob_2) + risk_{\Delta} \\ &= risk(\{ob_1, ob_2\}) \end{aligned}$$

So, theoretically if a user's trust value is high enough, the user can access both objects ob_1 and ob_2, that is:

$$\mathrm{T}_u \geq \theta risk(ob_2 \mid ob_1) = \theta risk(\{ob_1, ob_2\})$$

In our proposal, we compromise on the Chinese Wall policy. If a user who has very high trust value, the Chinese wall policy can be bypassed.

7.2 Static Separation of Duty in Role-Based Access Control Model

In this subsection we discuss static separation of duty constraints that can be enforced in our trust and risk based access control approach. In our proposal, risk is defined in terms of permissions and objects. We discuss two kinds of *permission-based separation of duty* constraints [27]. Since *object-based separation of duty* constraints [27] are similar with *permission-based separation of duty* constraints except for the operation, we omit it here.

$(U, \{p_1, p_2, \ldots, p_n\})$:This constraint is a constraint that denies a user's access to all the permissions $p_1, p_2, \ldots, p_n$ to complete some task (a task execution is equal to successful access of all these permissions). Assume that $HRC\_P = \{p_1, p_2, \ldots, p_n\}$, and same user tries to access permissions $p_1, p_2, \ldots, p_n$ successively, then the combined risk value of these permissions would be $risk(\{p_1, p_2, \ldots, p_n\})$. then we use access control workflow described in **Table 1**.

$(R, \{p_1, p_2, \ldots, p_n\})$:This constraint is a constraint that states no role can access all the permissions $p_1, p_2, \ldots, p_n$ to complete some task. This situation is slightly different from the previous one. Here we consider users who have same role can not access all permissions from $HRC\_P = \{p_1, p_2, \ldots, p_n\}$. The important personal identity in this situation is user's role; so, role related personal identities are critical in determination of potential user group. We assume that role id rid is the only identity that uniquely identifies user's role. Therefore, we have the equation below and Eq.20 for computing the similarity of two users u_1 and u_2.

$$Sim_{pid}(u_1, u_2) = \alpha_1 sim(rid_{u_1}, rid_{u_2}), \quad \sum_{i=1}^{n} \alpha_i = 1, \ \alpha_1 = 1, \ \alpha_2, \alpha_3, \ldots, \alpha_n = 0$$

This means the access control system only cares about the role in detecting similarity between users; role id is the only criteria by which the access control system decide if two users are the same, so users who possess same role are in one potential user group. Making access control decision uses the workflow described in **Table 1**.

8. Related Work

SoD constraints of RBAC [6][7][27] and the Chinese wall policy [26] are for the purpose of preventing conflict of interests or commercial fraud, i.e. similar users or users who possess similar roles should not be assigned to permissions that combination of which is sensitive or risky. These constraints just for users who have similarity in their user id or role, somehow too

rigid. In our proposal, we generalize the similarity of users, so that the access control system chooses different criteria for defining similarity of users and prevent these similar users from accessing a sensitive combination of permissions. User's partial personal identity similarity and similarity in access request are the basis for defining similarity of users. We use educated prediction about users to cluster them.

Traditional access control models such as DAC [1][2], MAC [3][4][5] and RBAC [6][7] describe policies for known users, they imply binary trust (trusted, untrusted). In DAC and RBAC, no risk or value of permission is considered. In MAC, objects have security labels and compare subject security label and object security label for access control decision making, but the security labels are relatively static. Although these models are efficient for predefined known users, they are unable to effectively define access control policy for unknown users. In addition, users' trustworthiness changes all the time even if they are predefined known users in practical environments.

Trust is an important factor in access control upon which access control decision are made under uncertain situations. There is a growing increase of literature on trust based access control [9][10][11][12][13][14][15][16][17]. Trust-based access control improves its flexibility by making access control decisions according to quantified user trust and simulates traditional access control models such as RBAC [14][15][16]. Blaze et al. [9] firstly introduced "Trust Management" and proposed making access control on the basis of trust relation expressed by certificates. In [10][11][12][13][14][15][16][17], trust is considered as a main basis for making access control decision. Ray and et al. integrates trust into RBAC model, in order to make it flexibly describe access control policy [14][15][16]. They propose "user-trust level-role-permission" method. Trust is quantified according to a user's profile, experience and recommendation. User's past behavior influence their current trust in [14]. In [15], Ray et al. connect the trust level to SoD constraints of RBAC, they mentioned that highly trustworthy users can bypass SoD. In our proposal, highly trustworthy users bypass SoD constraints and Chinese Wall policy by keeping original risk value of combined permissions unchanged. The main limitation of trust-based access control is the lack of description of consequence or risk of accessing a permission. In our proposal we not only defined the risk of single permission, but also defined combined risk of mutually related permissions.

Risk has also been introduced into access control research area [18][19][20][21][22]. Application systems try to give more access opportunities to users even if the permission are risky, because they expect some direct or indirect benefits from sharing information to users if the users behave well in their access to the information. In [19], role hierarchy of RBAC is defined according to the security risk ordering relations. The JASON report [18] discusses how the measurable risk plays an important role in risk-based access control. [20] proposes contextual risk-based access control in which the risk value of allowing access to a permission and risk value of denying access to a permission and the two risk values are compared, then choose the less risky action. Their work in fact makes a cost-benefit analysis of accessing a permission. Keser et al. proposes a risk-adaptive access control model [21], in their proposal, Bell-Lapadula model [3] is used, the risk of an object depends on the difference between the users' security label and objects' security label. It is reasonable to say that highly trustworthy users have high security label, lowly trustworthy users have low security label. Bertino proposes risk-based access control in [22] in which risk is also measured according to the difference between subject security label and object security label. Bertino also suggests that providing more access opportunities for users and ask users to perform some obligations after their access to an object. Compared to our work, The shortage of risk-based access control is

that who can access risky permission is not sufficiently addressed, and combined risk of mutually related permissions are not discussed.

Combination of trust and risk in access control also receives attention [23][14][25]. [23] proposes trust and risk in role-based access control policy, principle's trust and cost of permissions are considered together in making access control decision. [24] examines the relationship between risk and trust, risk is about possible outcome of a particular action and trust concerns the transaction history. We connect user trust and permission risk to make access control decision in our proposal. The advantages of our work is that we mine the relationship between users and the relationship between permissions in order to provide more flexible access control service.

9. Conclusion

In this paper, we examined the importance of potential relations between users and relations between permissions in access control environments and propose an novel approach to integrate these two kinds of relations into access control. We connect user trust and permission risk to provide flexible access control for dynamic, distributed environments. Our proposal creates more access opportunities to well behaved users via loosing the access control constraints; meanwhile making it harder for potential user group members to achieve their goal, which might incur great loss to the application system. We generalized the similarity criteria of users according to their partial personal identity similarity and the historical access interest similarity to improves the flexibility of constraints of access control in an effective way.

The evaluation of user trust and estimation of permission or object risk needs more research work. Whether the potential user group is a malicious user group is still under consideration; we did not differentiate them in this proposal. In our future work, we will consider finding more reasonable approach for mining potentially malicious user groups. The applicability of our proposal in a practical environment is also for our further consideration.

References

[1] B.W. Lampson, "Protection," in *Proc. of 5th Annual Princeton Conference on Information Sciences and Systems*, Princeton, pp. 437-443, 1971.

[2] G.S. Graham, P.J. Denning, "Protection - Principles and Practice," in *Proc. of the AFIPS Spring Joint Computer Conference*, Montvale, New Jersey, pp. 417-429, 1972. Article (CrossRef Link)

[3] D. Bell, L. LaPadula, "Secure Computer Systems: Mathematical Foundations and Model," *Technical Report M74-244*, MITRE Corp., Bedford, MA 1973.

[4] K.J. Biba, "Integrity Considerations for Secure Computer Systems," *EST TR-76-372, ESD/AFSC, Hanscom AFB*, Bedford, MA 1977.

[5] R.S. Sandhu, "Lattice-Based Access Control Models," *IEEE Computer*, vol. 26, pp. 9-19, 1993. Article (CrossRef Link)

[6] R.S. Sandhu, E.J. Coynek, H.L. Feinsteink, C.E. Youmank, "Role-Based Access Control Models," *IEEE Computer*, vol. 29, no. 2, pp. 38-47, Feb. 1996. Article (CrossRef Link)

[7] American National Standards Institute, "Role-Based Access Control," ANSI INCITS 359-2004.

[8] T.W.A. Grandison, "Trust management for internet applications," *Ph.D Thesis*, Imperial College of Science, Technology and Medicine, University of London, 2003. Article (CrossRef Link)

[9] M. Blaze, J. Feigenbaum, J. Lacy, "Decentralized Trust Management," in *Proc. of the IEEE Symposium on Security and Privacy Oakland*, pp. 164-173, 1996. Article (CrossRef Link)

[10] L. Kagal, T. Finin, A. Joshi, "Trust-Based Security in Pervasive Computing Environments,"

Computer, vol. 34, pp. 154-157, 2001. Article (CrossRef Link)
[11] R. Bhatti, D. Sanz, E. Bertino, A. Ghafoor, "A Trust-based Context-Aware Access Control Model for Web-Services," in *Proc. of the IEEE International Conference on Web Services*, San Diego, California, USA, pp. 184-191, 2004. Article (CrossRef Link)
[12] R. Sandhu, X. Zhang, "Peer-to-Peer Access Control Architecture using Trusted Computing Technology," in *Proc. of the tenth ACM symposium on Access control models and technologies (SACMAT '05)*, Stockholm, Sweden, pp. 147-158, 2005. Article (CrossRef Link)
[13] G. Ya-Jun, H. Fan, Z. Qing-Guo, L. Rong, "An Access Control Model for Ubiquitous Computing Application," in *Proc. of the Second International Conference on Mobile Technology, Applications and Systems 2005*, Guangzhou, China, pp. 1-6, 2005. Article (CrossRef Link)
[14] S. Chakraborty, I. Ray, "TrustBAC: Integrating Trust Relationships into the RBAC Model for Access Control in Open Systems," in *Proc. of the eleventh ACM symposium on Access control models and technologies*, Lake Tahoe, California, USA, pp. 49-58, 2006. Article (CrossRef Link)
[15] M. Toahchoodee, R. Abdunabi, I. Ray, I. Ray, "A Trust-Based Access Control Model for Pervasive Computing Applications," in *Proc. of the 23rd Annual IFIP WG 11.3 Working Conference on Data and Applications Security XXIII*, Montreal, P.Q., Canada, pp. 307-314, 2009. Article (CrossRef Link)
[16] I. Ray, I. Ray, S. Chakraborty, "An interoperable context sensitive model of trust," *Journal of Intelligent Information Systems*, vol. 32, pp. 75-104, 2009. Article (CrossRef Link)
[17] J. Luo, X. Ni, J. Yong, "A trust degree based access control in grid environments," *Information Sciences*, vol. 179, pp. 2618-2628, 2009. Article (CrossRef Link)
[18] M.C.M.V.J. P. OFFICE, "HORIZONTAL INTEGRATION: Broader Access Models for Realizing Information Dominance," JSR-04-132, December 2004.
[19] N. Nissanke, E.J. Khayat, "Risk Based Security Analysis of Permissions in RBAC," in *Proc. of the 2 nd International Workshop on Security In Information Systems, Security In Information Systems*, porto, Portogal, pp. 332-341, 2004. Article (CrossRef Link)
[20] N.N. Diep, S. Lee, Y.K. Lee, H. Lee, "Contextual Risk-Based Access Control," in *Proc. of the 2007 International Conference on Security Management*, Vegas, Nevada, USA, pp. 406-412, 2007.
[21] P.-C. Cheng, P. Rohatgi, C. Keser, P.A. Karger, G.M. Wagner, A.S. Reninger, "Fuzzy Multi-Level Security: An Experiment on Quantified Risk-Adaptive Access Control," in *Proc. of the 2007 IEEE Symposium on Security and Privacy*, Oakland, California, USA, pp. 222-230, 2007. Article (CrossRef Link)
[22] Q. Ni, E. Bertino, J. Lobo, "Risk-based Access Control Systems Built on Fuzzy Inferences," in *Proc. of the 5th ACM Symposium on Information, Computer and Communications Security (ASIACCS '10)*, Beijing, China, pp. 250-260, 2010. Article (CrossRef Link)
[23] N. Dimmock, A.S. Belokosztolszki, D. Eyers, J. Bacon, K. Moody, "Using Trust and Risk in Role-Based Access Control Policies," in *Proc. of the ninth ACM symposium on Access control models and technologies (SACMAT '04)*, Yorktown Heights, New York, USA, pp. 156-162, 2004. Article (CrossRef Link)
[24] A.J. And, A. Jøsang, S.L. Presti, "Analysing the Relationship between Risk and Trust," in *Proc. of the Second International Conference on Trust Management (iTrust2004)*, pp. 135-145, 2004. Article (CrossRef Link)
[25] Y. Li, H. Sun, Z. Chen, J. Ren, H. Luo, "Using Trust and Risk in Access Control for Grid Environment," in *Proc. of 2008 International Conference on Security Technology*, Hainan, China, pp. 13-16, 2008. Article (CrossRef Link)
[26] D.F.C. Brewer, M.J. Nash, "The Chinese Wall Security Policy," in *Proc. of IEEE Computer Society Symposium on Research in Security and Privacy*, pp. 206-214, 1989. Article (CrossRef Link)
[27] J. Crampton, "Specifying and Enforcing Constraints in Role-based Access Control," in *Proc. of the eighth ACM symposium on Access control models and technologies (SACMAT03)*, Como, Italy, pp. 43-50, 2003. Article (CrossRef Link)
[28] X. Jitian, Z. Yanchun, J. Xiaohua, L. Tianzhu, "Measuring Similarity of Interests for Clustering

Web-Users," in *Proc. of the 12th Australasian database conference Gold Coast*, Queensland, Australia, pp. 107-114, 2001. Article (CrossRef Link)

Nurmamat Helil is an Associate Professor of College of Mathematics and System Sciences, Xinjiang University, Urumqi, China. He received his Bachelor's degree, Master's degree and Ph.D. degree in School of Mathematics, Peking University in 2000, 2003 and 2008, respectively. From 2010 to 2011, he worked as a post-doctor in E-Commerce & Internet Application Laboratory, School of Computer Science and Engineering, Chung-Ang University, Seoul, Korea. His research interests include Information Security, Access Control, Semantic Web, and Social Network.

Mucheol Kim received a Bachelor of Science degree Seoul in 2005 and Master's degree in 2007 in Computer Science and Engineering, Chung-Ang University. And He is Ph.D. Candidate in E-Commerce & Internet Application Laboratory, School of Computer Science and Engineering, Chung-Ang University.. His research interests include Information Retrieval, Social Network, and Sensor Networks.

Sangyong Han is a professor of the school of computer science and engineering, Chung-Ang University, Seoul, Korea. He received Bachelor of Engineering in College of Engineering from Seoul National University in 1975, and the Ph.D. degree in 1984. From 1984 to 1995, he worked at Poughkeepsie Lab. and Watson Research Center in IBM, USA. His research interests include next generation web, information retrieval, and optimization.