Security, Privacy and Interoperability in Heterogeneous Systems

Jian Zhong, Peter Bertok, Zahir Tari

Security, Privacy and Interoperability in Heterogeneous Systems

Jian Zhong[1] , Peter Bertok[1], Zahir Tari[1]
[1] Computer Science and Information Technology, RMIT University, Melbourne, Victoria, Australia
{jian.zhong, peter.bertok, zahir.tari}@rmit.edu.au

Abstract. Partners in VOs can share large amount of data. Sharing of individual data items is straightforward, but sharing components of complex data structures stored in heterogeneous systems is often a challenge. Sharing is typically governed by rules and policies that need to be translated into access right / privilege control and data granularity control. Simultaneous control of privileges and data granularity across different organizations is a difficult task, and most traditional approaches, such role-based access control can become prohibitively complex in such scenarios. We propose a scheme for concurrent control of subject privileges and object granularity. It includes participants, duties and operations, and generates security labels that describe security features. To facilitate interoperability between heterogeneous systems, the labels also carry information about the data model, including dynamic hierarchy description. The model supports subject activity control over objects with variable data access granularity. It encompasses the advantages of existing role based and label based control. First, an abstract subject privilege control model is built, and the mathematical relationships between privileges in the label system are defined. Second, an abstract object dynamic granularity model is produced and the mathematical relationship between granularity levels is established. At last, a pair-wise privacy label system is provided as an integrated information protection mechanism, where relationships between subject activities and privileges are described for actual access control. A formal verification of the proposed method has also been performed.

Keywords: Interoperability, heterogeneous systems, security & privacy.

1 Introduction

Diverse organizations, governments and individuals share vast amount of information. Since information sharing can potentially harm certain parties, it is typically governed by roles and policies that support subject privilege control and data granularity control [1]. Both methods have been well studied on their own, and typically they are enforced by role-based access mechanisms and label based approaches respectively.

The increasing complexity of data, combined with the growing number and diversity of users who must access it, highlighted the limitations of traditional role based privilege control. One of the main problems is that with existing approaches the deployment of data granularity control becomes difficult when the majority of users are unique, i.e. when a role is assigned to a single user only. The solution requires two issues to be addressed simultaneously: (i) when accessing a whole object, which

activities are available for each granular data component, and (ii) with numerous unique users, how to realize dynamic granular data access level control. We look at these issues from three perspectives. Namely, when there are many *unique users*, no advantage can be gained by grouping them into roles or generalizing access rights according to rules [2], because user privileges may vary greatly even within the same group. Secondly, traditional management focuses on subject-based *granular privilege control*; however, for sharing private data, object-based granular privilege control needs to be considered. Thirdly, classical RBAC does not support multi-domain / *cross-domain application* in a straightforward manner.

In this paper, we propose a practical solution to integrate sophisticated access control into a request-based pair-wise privilege control model. The proposed system has three modules: one for parameterized 3D subject granular privilege control, one for object-based dynamic granular data control and one for privilege refinement. The model also utilizes the advantages of existing role-based and label-based control. The major contributions are the following.

- Enhanced subject-based granular privilege control to support unique subjects with diverse privileges as well as reduced privilege assignment (PA) computation complexity and role storage consumption.
- Object-based granular data privilege control that supports dynamic granular data access and object cross-domain sharing. An embedded special request component makes the private and sensitive data sharing more flexible and easier to be customized.
- Cooperation between subject-based and object-based control encompasses the advantages of existing role based and label based control, raises the efficiency of privilege assignment and lowers the cost.

The rest of this paper is organized as follows. First we provide general background about the foundations of cooperation between user privilege granularity control and object dynamic granular data level control. This is followed by details of the proposed method, its implementation and testing. Then a section discusses the advantages and disadvantages of the proposed system, before the paper is concluded.

2 Related Work

A number of papers have looked at unique users with diverse privileges. By involving context tables, one solution supports fine-grained privileges and variable subject requests [3]. Another approach handles complex roles with diverse privileges by using an additional condition list [4]. In a third approach the traditional concept of 'role' is replaced by a new parameterized model, which not only supports unique users better but also reduces storage consumption compared to traditional role-based approaches [5]. These solutions add different modules to extend privileges and support user diversity, but unfortunately also result in decreased efficiency, as the unique users may come and go continuously. In addition, in [5] role models are difficult to build before the users lodge their requests. As opposed to these solutions, the proposed method builds an enhanced model that can directly assign privileges to subjects by request, priority check and privilege refinement. The storage requirements decrease when diverse roles don't need to be stored on the subject server. Moreover, the delicate modules and components offer efficient privilege assignment.

For collaboration control, connecting subject-based and object-based privilege control was considered by some approaches [6, 3]. However, the former solution [6] does not address granular data control, while the latter one, non-independent object based privilege control, limits dual control performance [3]. In the proposed solution, the use of the same hierarchical model for subject-based and object-based privilege control improves the efficiency of granular data control. In addition, an independent object controller offers high performance and supports special-condition control.

For multi-domain applications the use of role-mapping tables was proposed [7]. However, practical issues, such as role switching and data granularity may deteriorate the performance, as the building of mapping tables for all roaming roles and data will not only consume considerable storage space but also lower the efficiency of privilege assignment. Dissimilar to this approach, a dynamic hierarchy component employed by our proposed method caters for data roaming without high management costs.

3 Proposed Method

The proposed pair-wise privilege control scheme contains three parts, namely the privacy-label based subject granular privilege control (PSPC) module, the privacy-label based object granular data control (PODC) module and the collaboration control module as shown in fig. 1. Due to space limitations, only the hierarchical PSPC component, the dynamic hierarchical control component in the PODC module and privilege refinement are described here.

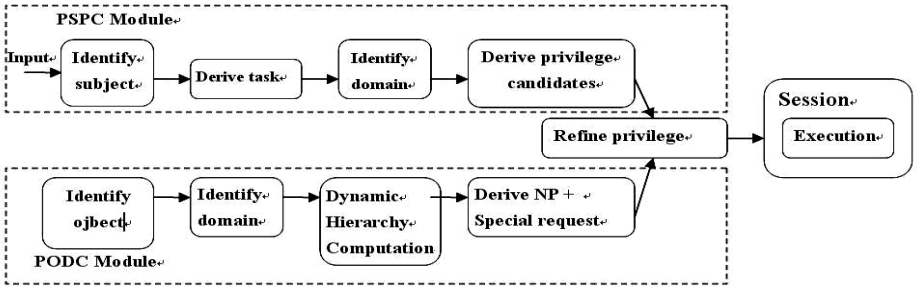

Fig. 1. Pair-wise Privilege Control Model

Basic concepts and definitions are given as follows. Given a set of subjects *S*, a set of objects *O*, each object has a set of granular data $O_x = \{OGD_i \mid i = 1, 2, \cdots, n\}$, where *n* is the number of granular data items. Each subject has a set of activities $SA = \{SA_i \mid i = 1, 2, \cdots, n\}$, where *n* is the number of subject activities. A set of *Subject Activity Level*, represented by $SL = \{\Omega(SA)_i \mid i = 1, 2, \cdots, n\}$ denotes the activity priority on the required object, where *n* is the number of subject activity levels and operation Ω(SA) denotes a subset of SA. Subject Grade denotes the subject's overall privacy priority, and is represented by SG. A set of Subject Sub-grades $SSG_x = \{\Omega(SL)_i \mid i = 1, 2, \cdots, n\}$ denotes the access priority for each granular data item of an object, where *x* denotes a certain sub-grade and n is the number of activity levels included in the sub-grade. The set of Negative Permissions is

$NP = \{NP_i | i = 1, 2, \cdots, n\}$, where n is the number of negative permissions. A permission $P_x = \{(D_i, Boolean)\}$ can be accepted or denied, as indicated by the Boolean.

Hierarchical PSPC defines the control of subject granular privileges.

Rule 1: *Subject Activities* on the same *Subject Activity Level* are mutually exclusive, so only one will be activated in any one session. However, if they are on different levels, these activities can operate in the same session in parallel.

Rule 2: *Subject Activity Level* in a *Subject Sub-Grade* must be different from other activity levels in the same subject grade.

Rule 3: Different *Subject Sub-Grades* are controlled independently.

Basically, after a request has been lodged to a subject server, the subject is to be associated with an overall privacy priority, also called main subject access grade and represented by *Subject Grade* (*SG*). The *SG* usually consists of a set of *SSG*s that are associated with granular data items of the required object. Each *SSG* is composed of a set of *SL*s, which indicates the priority and correlation between different *SA*s. *SA*s included in the *SL* are the activities that can be executed over the object.

The *Dynamic Hierarchy* component has two main functions: hierarchy assignment (HA) and condition assignment (CA). When data is roaming, the original privacy priority control may not be able to work properly. Fig. 3 shows the concept of HA applied mapping.

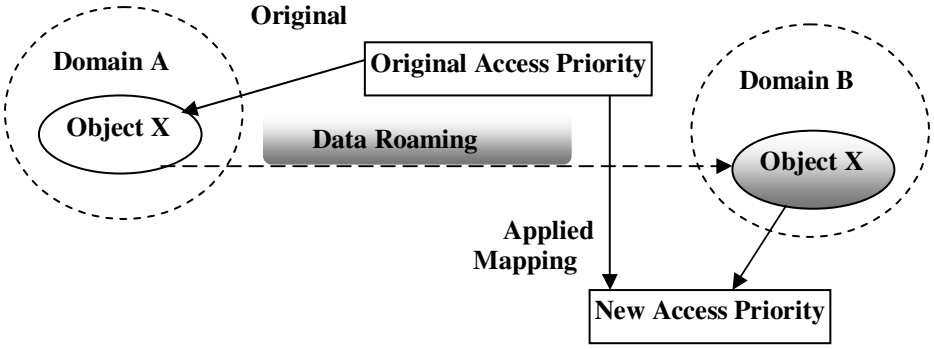

Fig. 2. Hierarchy Assignment (HA)

In fig. 2, Access Priority is represented by the object grade (*OG)* including object subgrades (*OSG*s). When object X (including granular data) is created, it is processed by the original object server in domain A, where the original access priority is a non-absolute classification value. If such data is required for roaming, the original object server will first assign a *basic sub-grade* (*BSG*) to one or more granular data parts. Then, all other data parts can be assigned proper grades by referring to *BSG* at the original object server. When the data is roaming to domain B, the object server in domain B will first map BSG to a basic applied grade (*BAG*) based on the data sharing environment and sharing requirements. Then, the system maps the original granular data sub-grades to applied grades relative to the *BSG*. The PODC control frame merely defines any negative permission (NP) for each granular data where necessary, which states what cannot be done over each granular data. Any special request on granular data is supported by Condition Assignment (CA).

The Privilege Refinement (*RE*) function computes proper permissions for the subject request. The left side of fig. 3 describes the first step of permission computing, which produces the overall permission of an access. If the subject's grade is equal to or higher than the 'object prohibited' grade, authorization will be given, otherwise access will be denied. S_iPL.S_iG denotes the Subject Grade of Subject *i*; OPL.OG denotes object overall grade. The algorithm for sub-grade computing is given in the right side of fig. 3. The process compares all the subject sub-grades to all object granular data grades respectively, where → denotes 'move to the next'. The notations of permission symbols and extended *SL* operations are as follows.

Notion 1: (Positive Permission Operation =Γ) $S_i SG_x = \Gamma^{SA_j} OSG_y$ denotes that subject *i* is allowed to perform activity *j* on the object's granular data *y* in operation *x*.

Notion 2: (Negative Permission Operation !Γ) $S_i SG_x !\Gamma^{SA_j} OSG_y$ denotes that subject *i* is not allowed to perform activity *j* on the object's granular data *y* in operation *x*.

Notion 3: (Activity and Process Level Θ)

$S_i L_x \Theta^{\rightarrow} S_i L_y$ for subject *i* and a granular data of the object, level *y* must be processed after level *x*.

$S_i L_x \Theta^{\leftarrow} S_i L_y$ for subject *i* and a granular data of the object, level *y* must be processed before level *x*.

$S_i L_x \Theta^{\leftrightarrow} S_i L_y$ for subject *i* and a granular data of the object, level *y* and level *x* are mutually exclusive, which means only one of the levels will be processed.

$S_i L_x \Theta^{\updownarrow} S_i L_y$ for subject *i* and a granular data of the object, level *y* and level *x* can be processed simultaneously.

```
function Compare (subject i 's grade, object grade)
        (sg = SiPL.SiG)→(og = OPL.OG)
        if sg ≥ og do
            Si = Γ O
            move Next Process
        else
            Si !Γ O
            end
        end if
```

```
function Compare (all subject subgrades, all object granular data grades)
    for all x such that 1 ≤ x ≤ m do
        if SiSGx ≥ OSGx then
            SiSGx = Γ OSGx
        else
            SiSGx !Γ OSGx
        end if
    end for
    move Next Process
```

Fig. 3. Grade Computing and Sub-grade Computing

The left side of fig. 4 describes the algorithm of computing the relationship between each pair of levels. The computing logic will go through each pair of levels and output results. The right side of fig. 4 shows activity execution based on the results of three algorithms: the two in fig.3 and the one in the left side of fig.4.

The proposed scheme has been verified by Failures-Divergence Refinement (FDR) [8], a model verification tool based on Communicating Sequential Processes (CSP) state machines, and was implemented in JAVA.

4 Example

```
function Compare (SiL)
  for all x such that 1 ≤ x ≤ m do
   for all y such that 1 ≤ y ≤ n do
     if Θ→ then
       SiLy.precondition(SiLx)
       continue
     else if Θ← then
       SiLx.precondition(SiLy)
       continue
     else if Θ↔ then
       select SiLx or SiLy
       selection
       continue
     else if Θ↕ then
       continue
       else continue
     end if
   end for
  end for
```

```
function execution (SiB)
  Compare (subject i 's grade, object grade)
  Compare (all subject sub-grades,
   all object granular data grades)
   Compare (SiL)
   for all x such that 1 ≤ x ≤ m parallel do
     select executing activity
     if selection is in negative permission list
then
          activity denied
          continue
       end if
       if precondition = true then
          select precondition activity or abort
          execute selection
          continue
       end if
       if mutually exclusive = true then
          select one of the mutually exclusive
  activities or abort
          execute selection
          continue
       end if
          execute selection
    end for
```

Fig. 4. Level Computing and Activity Execution

In this paper, we assume that data sharing is taking place without malicious attacks such as identity forgery, object forgery or dishonest subjects. For sake of simplicity we show only one object (O) with 3 pieces of granular data (OSG_x, $1 \le x \le 3$) and three independent subjects (S_A, S_B, S_C). We also assume that there are 8 different subject activities in the subject activity set (SB): *read* (b_r), *edit* (b_e), *add* (b_a), *delete* (b_d), *comment* (b_c), *declare* (b_{dc}), *replicate* (b_{re}) and *manage* (b_m). Also, both subjects S_A and S_B have direct access to the object and S_B has the 'manage' permission. S_C can access the object through S_B. Thus, we have $S_AG \ge OG$, $S_BG \ge OG$, $S_CG < OG$. We also assign the sub-grade sets $S_ASG = \{S_ASG_1, S_ASG_2, S_ASG_3\}$, $S_BSG = \{S_BSG_1, S_BSG_2, S_BSG_3\}$, $S_CSG = \{S_CSG_1, S_CSG_2, S_CSG_3\}$, where $S_ASG_1 \ge OSG_1$, $S_ASG_2 \ge OSG_2$, $S_ASG_3 < OSG_3$, $S_BSG_1 \ge OSG_1$, $S_BSG_2 \ge OSG_2$, $S_BSG_3 \ge OSG_3$. The logical relationship between the subjects and the object are shown in fig. 5.

5 Discussion

In the proposed method, privileges are allocated to a subject request in the automatically executing refinement process, or manually for special requests. The PSPC module supports fine-grained, diverse granular privileges for numerous unique subjects, and generates privacy labels in accordance with the request of the subject and its privacy level. Consequently, it can effectively reduce the consumption of computation and storage resources. The dynamic hierarchy component of the PODC

improves cross-domain data roaming, and can cater for different data sharing environments without redeploying all modules and components. This reduces the cost and processing overhead of subject reorganization.

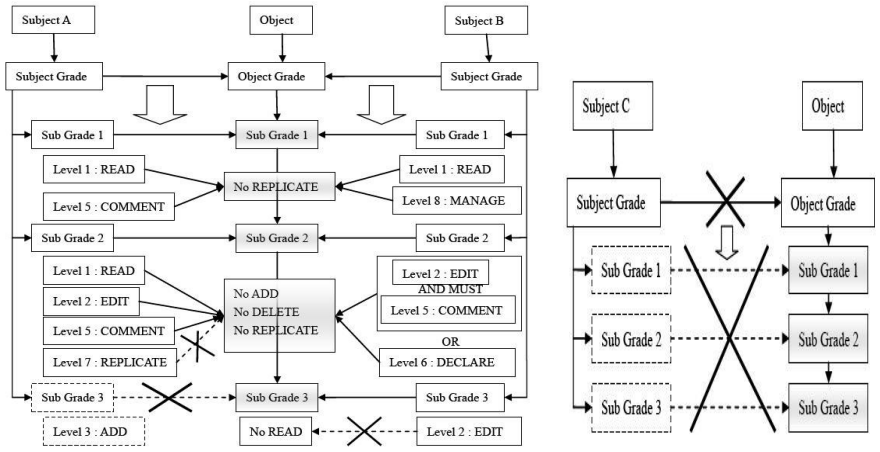

Fig. 5. Instantiation and Scenario

The unique subject issue mentioned in the introduction can be modelled mathematically [2]. Assuming that adding a target system causes 1/10 of the existing roles to be replaced by 10 roles each and the first system has 20 roles, the progressive explosion in roles is shown on the left side of fig. 6. It reveals that 10 systems push the total number of roles up to 6500. In practical deployments, hundreds of systems managed in a traditional role-based manner will aggravate the problem.

By involving granular subject privileges, the improvement is shown in the right side of fig. 6. We assumed that each subject submits two requests that contain three activities for each granular data, at every time unit.

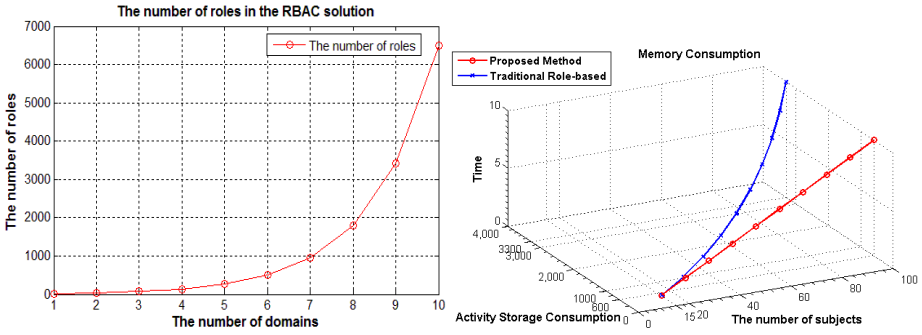

Fig. 6. System-Role Model and Memory Consumption

From the figure it can be seen that with a fixed number of domains, a traditional role-based system consumes around 400% more resources in terms of space than our request-based system. The difference increases with the number of domains. The advantages of our approach are easily noticeable in the case of routine business events, such as changing the responsibilities of a single employee or extending the function of

a department. They require the adjustment of the role model, and the ongoing nature of such events may necessitate the employment of a team of expert staff for role maintenance. Cross-system or cross-field reorganization is also constrained in such systems .Adding and removing subject activities are cumbersome tasks of role model maintenance. In the proposed system, subject activities can be added or removed according to system needs in a fairly straightforward manner.

On the other hand, when many subjects share similar privilege sets, traditional role-based approaches perform better.

6 Conclusion

This paper proposed an approach to combined subject granular privilege control and object dynamic granular access level control. The characteristics of the proposed model are as follows.

- The model can be implemented in various scenarios where private and sensitive data sharing is practical. Its memory requirements are moderate even with a large number of unique subjects when compared to traditional role-based methods. Easy handling of frequently changing privileges along with reduced computation complexity are also advantages of our model over existing solutions.
- Dynamic granular data privilege control is also supported by the proposed model; which allows the handling of object roaming and subject privileges in a compatible manner.
- Using the concept of applied mapping, the method supports data roaming between different domains.

References

1. Zhang, L.,Brodsky, A., Swarup, V., Jajodia, S.: A Framework for Maximizing Utility of Sanitized Documents Based on Meta-labeling. Policies for Distributed Systems and Networks, IEEE International Workshop on, pp. 181-188, 2008 IEEE Workshop on Policies for Distributed Systems and Networks, 2008
2. Hitachi ID Systems, Inc.: Beyond Roles: A Practical Approach to Enterprise User Provisioning. Access in Aug 2009 at http://www.idsynch.com/docs/beyond-roles.html
3. He, Q. and Anton, A.I.: A Framework for Modelling Privacy Requirements in Role Engineering. In Department of Computer Science, North Carolina State University, Raleigh, NC 27695-8207, USA
4. Li, X., Naeem, N.A. and Kemme, B.: Fine-Granularity Access Control in 3-tier Laboratory Information Systems. In the Proceedings of the 9th International Database Engineering & Application Symposium (IDEAS'05), 2005.
5. Abdallah, A.E. and Khayat, E.J.: A Formal Model for Parameterized Role-based Access Control. Research Institute for Computing, London South Bank University, U.K, 2005.
6. Acevedo, M.T., Fillingham,D. and Nicolettos, J.L.: Enterprise Security Application of Partition Rule Based Access Control (PRBAC). In Proceedings of the 6th Workshop on Enabling Technologies on Infrastructure for Collaborative Enterprises, pp: 285 - 292
7. Martino, L.D., Ni, Q., Lin, D. and Bertino, E.: Multi-domain and Privacy-aware Role Based Access Control in eHealth. Computer Science; Purdue University, USA, 2008
8. Goldsmith, M.: FDR2 User's Manual Version 2.82. Formal System (Europe) Ltd.