Secure Interoperation in Multidomain Environments Employing UCON Policies*

Jianfeng Lu[1], Ruixuan Li[1,**], Vijay Varadharajan[2], Zhengding Lu[1], and Xiaopu Ma[1]

[1] Intelligent and Distributed Computing Lab, College of Computer Sci. and Tech. Huazhong University of Sci. and Tech., Wuhan 430074, P.R. China
lujianfeng@smail.hust.edu.cn, rxli@hust.edu.cn, {zdlu,xpma}@smail.hust.edu.cn

[2] Department of Computing, Macquarie University, NSW 2109, Australia
vijay@ics.mq.edu.au

Abstract. Ensuring secure interoperation in multidomain environments based on role based access control (RBAC) has drawn considerable research works in the past. However, RBAC primarily consider static authorization decisions based on subjects' permissions on target objects, and there is no further enforcement during the access. Recently proposed usage control (UCON) can address these requirements of access policy representation for temporal and time-consuming problems. In this paper, we propose a framework to facilitate the establishment of secure interoperability in multidomain environments employing Usage Control (UCON) policies. In particular, we propose an attribute mapping technique to establish secure context in multidomain environments. A key challenge in the establishment of secure interoperability is to guarantee security of individual domains in presence of interoperation. We study how conflicts arise and show that it is efficient to resolve the security violations of cyclic inheritance and separation of duty.

Keywords: Multidomain, interoperation, usage control, cyclic inheritance, separation of duty.

1 Introduction

Ensuring secure interoperation in multidomain environments based on role based access control (RBAC) has drawn considerable research works in the past [1]. Although RBAC [2] has become widely accepted as the principal type of access control model in theory and in practice, it primarily considers static authorization decisions based on subjects' permissions on target objects, and there is no further enforcement during the access. In recent information systems, the interactive and concurrent concepts should be introduced to access control. Obviously, RBAC model and other traditional access control model have difficulties,

* This work is supported by National Natural Science Foundation of China under Grant 60873225, 60773191 and 60403027, National High Technology Research and Development Program of China under Grant 2007AA01Z403.

** Corresponding author.

P. Samarati et al. (Eds.): ISC 2009, LNCS 5735, pp. 395–402, 2009.

or lack the flexibility to specify these requirements. Recently proposed usage control (UCON) [3, 4] offers a promising approach for the next generation of access control, it can address these requirements of access policy representation for temporal and time-consuming problems.

The above observations motivate us to consider new secure Interoperation policy. In this paper, we employ attribute mapping techniques to propose an interoperation policy framework in multidomain environments based on UCON model. In this policy framework, parts of foreign subject attributes will be mapped to local attributes, once these associations are set up, all required foreign attributes are dynamically mapped to local attributes, and the authorization can be made based on these local attributes. A key challenge in the establishment of secure interoperability is to guaran-tee security of individual domains in presence of interoperation. This paper focuses on two types of security violations of cyclic inheritance and separation of duty (SoD). We study how these security violations arise and show that it is efficient to resolve them.

The rest of this paper is organized as follows. Section 2 proposes the attribute mapping technique for secure interoperation framework. Section 3 studies how security violations arise and show that it is efficient to resolve these security violations. Some related work in interoperation are reviewed in Section 4. Finally, Section 5 concludes this paper.

2 Attribute Mapping Technique for Interoperation Policy

Zhang et al. [5] present an example motivating the new features of UCON. As the access control of this motivating example is not a simple action, the authorization decisions are not only based on subjects' permissions on target objects, but also need further enforcement during the access. In this way, traditional access control models lack the flexibility to specify policies in these scenarios, UCON is the preferred policy. However, this example does not fit in with multidomain environments. In multidomain environments, many types of user attributes' semantics cannot be interpreted across multiple domains, the first and foremost problem is to interpret these attributes across multiple domains. We now identify a complete taxonomy of attributes.

Attributes can be classified into different categories based on different items. Firstly, we classify attributes based on available scope as follows.

Localdomain attributes: This type of attributes is defined in a domain whose semantics can be interpreted only within local domain, but has no meaning or visibility in other domains.

Multidomain attributes: Comparing with localdomain attributes, multidomain attributes' semantics can be interpreted across multiple domains.

Secondly, we classify attributes based on liveness as follows.

Temporary attributes: Temporary attributes are created at the time a usage is started and deleted at the end of a single usage.

Persistent attributes: Persistent attributes live longer for multiple usage decisions.

Thirdly, we classify attributes based on whether the attributes can be updated during the usage process as follows.

Mutable attributes: Mutable attributes can be modified by the system automatically and do not require any administrative actions for update.

Immutable attributes: Immutable attributes cannot be changed by the subject's activity. Only administrative actions can change it.

The secure interaction between two or more administrative domains motivates the need for attributes translations that foreign attributes can be interpreted and understandable to local entities. In multidomain interaction scenario, only parts of attributes need to be translated. Firstly, multidomain attributes' semantics can be interpreted across multiple domains. Secondly, temporary attributes are alive only for a single usage. Therefore, we only need to establish a flexible policy for dynamic LPM (*localdomain persistent mutable*) and LPI (*localdomain persistent immutable*) attributes mapping to make interoperation in two domains employing UCON policies, and then the communications between two domains are mainly created by attribute mapping technique. The characterize definition about attribute mapping is as follows.

Definition 1. *Attribute Mapping: The attribute mapping is formalized as a 5-tuple:* $< a_1, D_1, a_2, D_2, m >$, a_1 *is an attribute in domain* D_1, *and* a_2 *is an attribute in domain* D_2 *respectively. In general,* D_1 *is the foreign domain, and* D_2 *is the local domain. The fifth parameter* m *is the mapping modes* $\mapsto_{\text{LPM}}$ *or* $\mapsto_{\text{LPI}}$, *which denotes the association of the two attributes* a_1 *and* a_2. $\mapsto_{\text{LPM}}$ *denotes that LPM attributes from the foreign domain* D_1 *will be translated to local domain* D_2. $\mapsto_{\text{LPI}}$ *implies that LPI attributes from the foreign domain* D_1 *will be mapped to local domain* D_2.

For the $\mapsto_{\text{LPM}}$ mappings, let Γ be a set which includes the LPM attributes mapped from foreign domain to local domain, let Γ' be a set which includes the LPM attributes from local domain associated with Γ, $\mapsto_{\text{LPM}} : \Gamma \to \Gamma'$ is a function from Γ to Γ', then $\mapsto_{\text{LPM}}$ obviously is a monotone increasing function. And for the $\mapsto_{\text{LPI}}$ mappings, let Γ be a set which includes the LPI attributes from two interoperate domains, and $\mapsto_{\text{LPI}}$ be a binary relation on Γ. Obviously, $\mapsto_{\text{LPI}}$ associates the LPI attributes, and these associations form a combined hierarchy that is partially ordered on Γ.

3 Security Issues for Attribute Mappings

A key challenge in the establishment of secure interoperability is to guarantee security of individual domains in presence of interoperation. There are many types of security violations leaded by establishing an interoperation policy among heterogeneous systems. These violations may arise because different domains may adopt different models, semantics, schema format, data labeling schemes,

and constraints for representing their access control policies [6, 7, 8]. This section focuses on two types of security violations: cyclic inheritance, and SoD. We study how these security violations arise and show that it is efficient to resolve them.

3.1 Violations of Cyclic Inheritance

Violations of cyclic inheritance mainly occur in interoperation of systems employing multilevel security policies, such as lattice-based access control (LBAC) and role-based access control (RBAC) [8, 9]. The cross-domain hierarchy relationship may introduce a cycle in the interoperation lattice enabling a subject lower in the access control hierarchy to assume the permissions of a subject higher in the hierarchy.

Definition 2. *A cyclic inheritance violation is expressed as*

$$\exists(a_i, a_j) \in A \times A, (b_k, b_l) \in B \times B\,((a_j \mapsto_{\text{LPI}} b_k) \wedge (b_l \mapsto_{\text{LPI}} a_i)) \Rightarrow (a_j, a_i)$$

where $A = a_1, \ldots, a_m$, $B = b_1, \ldots, b_n$, i, j, k, l, m *and* n *are integers, such that* $1 \leq i \neq j \leq m$, $1 \leq k \neq l \leq n$. *Each* a_i *is an attribute in attribute set* A, *and* b_k *is an attribute in attribute set* B. A *and* B *are two different domains. The notation* (a_i, a_j) *is a two-tuples, which means that the attribute* a_i *is the ancestor of* a_j.

Cyclic inheritance usually arises from the circulation in the $I-hierarchy$. There are four cases of cyclic inheritance: a_i is a direct or indirect ancestor of a_j, and b_k is a direct or indirect ancestor of b_l. It is noted that a_i is an ancestor of itself. Combine with the above cases also can generate other sub cases. A cyclic inheritance causes an attribute to inherit its senior attribute, as get all senior attributes and all junior attributes of a given attribute is tractable.

Theorem 1. *The checking problem for violations of cyclic inheritance is in* $\boldsymbol{P}$.

Proof. One algorithm for detecting problem for cyclic inheritance violations is as follows. For each attribute a in a domain A, one first computes all senior attributes and junior attributes of a, includes the $I-hierarchies$ and LPI attribute mappings. Then compares these two sets of attributes, if the intersection is not empty, there exists at least one cyclic inheritance violations, otherwise not. This algorithm has a time complexity of $O(Na(Na + Nm)$., where Na is the number of LPI attributes with $I-hierarchy$, Nm is the number of LPI attribute mappings. □

3.2 Violations of Separation of Duty

SoD is widely considered to be a fundamental principle in computer security [10]. Violations of SoD constraints may occur in an interoperation policy because of the interplay of various policy constraints across domains. When a sensitive task is comprised of m permissions, an SSoD policy requires the cooperation of at least n (for some $2 \leq\leq m$) different users to complete the task. In other words, there shouldn't exist a set of fewer than n users that together have all the m permissions to complete the sensitive task. We now formally define the SSoD violation.

Definition 3. *An SSoD violation is expressed as*

$$\forall\{u_1,\ldots,u_{n-1}\} \subseteq U \left(\bigcup_{i=1}^{n-1} Auth_P(u_i) \supseteq \{p_1,\ldots,p_m\} \right)$$

where m and n are integers, such that $1 \leq n \leq m$, $\{p_1,\ldots,p_m\}$ is the set of all possible permissions. $Auth_P : U \rightarrow 2^P$ is a function, where U is the user set, and 2^P is the power set of permissions.

In the literature on RBAC, statically mutually exclusive roles (SMER) constraints are used to enforce SSoD policies [2]. In RBAC model, permissions are assigned to roles. But the role is only a special type of subject attribute in UCON model, and there are many types of subject attributes in UCON model as shown in section 2.2, which play a very important role on the authorization based decision, that makes SMER constraints not suit to UCON policy. Consequently, we formally define two types of statically mutually exclusive attributes (SMEA) in UCON.

Definition 4. *An SD-SMEA (single-dimensional statically mutually exclusive attributes) constraint is expressed as*

$$\forall u \in U\,(|ATT(u) \cap AS| < n)$$

where $AS = a_1,\ldots,a_m$ be the set of all mutually exclusive attributes where each a_i is an attribute, m and n are integers, such that $2 \leq n \leq m$, each a_i is an attribute, $ATT : U \rightarrow 2^A$ is a function, where U is the user set, and 2^A is the power set of attributes. SD-SMEA constraint means that no user is a member of n or more attributes in AS.

The SD-SMEA is a general form, SMER is an instance of SD-SMEA where the $ATT(u)$ is a role set that assigned to a user: $ATT(u) = \{r \in R[(u, r_1) \in UA \wedge (r_1, r) \in RH]\}$

For example, as shown in Fig.1, $AS = \{Junior - Member, Rookie\}$, $n = 2$ in (AS, n), from the example,

$ATT(u_1) = \{Administrator, Teacher, Student, Junior - Member, Rookie\}$,
$ATT(u_2) = \{Teacher, Junior - Member\}$,$ATT(u_3) = \{Student, Rookie\}$,
$|ATT(u_1) \cap AS| = 2 = n$, $|ATT(u_2) \cap AS| = 1 < n$, $|ATT(u_3) \cap AS| = 1$

Therefore, this example violates the SD-SMEA constraint as $|ATT(u_1) \cap AS| = 2 = n$ violates Definition 4.

Definition 5. *A MD-SMEA (multi-dimensional statically mutually exclusive attributes) constraint is expressed as*

$$\forall u \in U\{\forall as_i \in AS\,(|ATT(u) \cap as_i| < n_i)\}$$

where $AS = \{as_1,\ldots,as_m\}$ be the set of mutually exclusive attributes sets where each $as_i = \{as_{i1}, as_{i2}, as_{i3}\ldots\}$, each element in as_i is an attribute, the corresponding n_i is integer in the integer array $N = \{n_1,\ldots,n_m\}$, m is an integer,

such that $1 < n_i \leq |AS_i|$, *Each* a_i *is an attribute,* (AS, N) *is a two-tuples, it means that no user is a member of* n_i *or more attributes in* as_i *for every* $1 \leq i \leq m$. *ATT is the same meaning with Definition 4. MD-SMEA also includes mutable attributes. Let* $ATT(u)$ *is the value of mutable attributes,* as_i *is an interval, and* $n_i = 1$*(*n_i *can be any integer which larger than zero).*

Assume that no user can be assigned to both $Junior-Member$ and $Rookie$, and his virtual-money exceeds \$1000. Then we can generate a MD-SMEA constraint to enforce this SoD policy:

$$as_1 = \{Junior - Member, Rookie\},\ as_2 = (1000, +\infty),\ n_1 = 2,\ n_2 = 1.$$

We assume user u_1, u_2 and u_3 be assigned corresponding roles as shown in Fig.1. And the virtual-money of u_1 is \$500, the virtual-money of u_2 is \$900, the virtual-money of u_3 is \$1200. We now use the definition of MD-SMEA to verify whether the above example enforces MD-SMEA policy.

For $u_1 : ATT(u_1) = \{\{Administrator, Teacher, Student, Junior - Member, Rookie\}, \{virtual - money = \$1200\}\}$, $|ATT(u_1) \cap as_1| = |\{Junior - Member, Rookie\}| = 2 = n_1$,$|ATT(u_1) \cap as_2| = 0 < n_2 = 1$; For $u_2 : ATT(u_2) = \{\{Teacher, Junior-Member\}, \{virtual-money = \$1200\}\}$, $|ATT(u_2) \cap as_1| = |\{Junior - Member\}| = 1 < n1 = 2$,$|ATT(u_2) \cap as_2| = 0 < n_2 = 1$; For $u_3 : ATT(u_3) = \{\{Student, Rookie\}, \{virtual - money = \$1200\}\}$, $|ATT(u_3) \cap as_1| = |\{Rookie\}| = 1 < n_1 = 2$, $|ATT(u_3) \cap as_2| = |1200| = 1200 > n_2 = 1$. From the above analysis, the user u_1 and u_3 violate MD-SMEA constraints because they don't satisfy all of the restriction of the MD-SMEA constraints. It is significant that MD-SMEA constraint can't be regarded as the combination of many SD-SMEA constraints since they are different conceptions. When we make the form definition of SD-SMEA and MD-SMEA, the verification problem is urgent: "do the satisfaction checking problems for SD-SMEA and MD-SMEA constraints can be done?" Both SD-SMEA and MD-SMEA constraints restrict the attribute memberships of a single user in order to enforce SSoD policies. Therefore, checking whether an UCON state satisfies a set of SD-SMEA and MD-SMEA constraints is efficient.

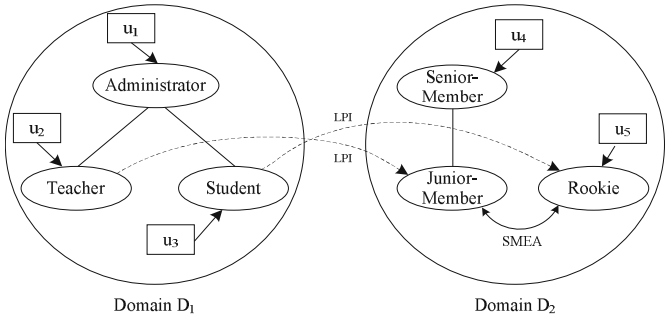

Fig. 1. An example of SD-SMEA constraint

Theorem 2. *The satisfaction checking problem for SD-SMEA constraints is in* $\boldsymbol{P}$.

Proof. One algorithm for solving SD-SMEA constraints is as follows. For each user in U, one first computes the set UA of all attributes in which the user is a member of, and then counts how many attributes in this set also appear in the set of attributes in the SMEA constraint, and finally compares this number with n. This algorithm has a time complexity of $O(Nu \times Na \times M)$, where Nu is the number of users in U, Na is the number of attributes, and M is the number of SD-SMEA constraints. □

Theorem 3. *The satisfaction checking problem for MD-SMEA constraints is in* $\boldsymbol{P}$.

Proof. The proof is essentially the same as that for Theorem 2: The satisfaction checking problem for a MD-SMEA constraint can be regarded as N SD-SMEA constraints, where N is the number of mutually exclusive attributes sets. □

4 Related Work

Ensuring secure interoperation in multidomain environments has drawn considerable research work in the past. Kapadia et al. [1] proposed a secure interoperability using dynamic role translation to implement access control across domains in the form of role mappings among individual domains. In [11], M. Shehab et al. proposed a distributed secure interoperability protocol that ensures secure interoperation of the multiple collaborating domains without compromising the security of collaborating domains. Shafiq et al. [12] extended the IRBAC model by proposing a secure interoperation framework in which all roles in the interacting domains are matched and policies are integrated to form a global RBAC policy.

The first and foremost challenge in establishing secure interoperation is the composition of a consistent and conflict-free interoperation policy. Several research efforts have been devoted to the resolution of the conflicts among role mappings. E. C. Lupu et al. [13] focused on the problems of conflict detection and resolution for policy conflicts, including authorization policies and obligation policies. Cyclic inheritance and separation of duties may appear in an interoperation policy [8]. The resolution of interoperation inconsistencies related to SoD constraint has not been adequately investigated and the existing approaches rely on manual intervention of policy administrators to resolve SoD conflicts [13]. In this paper, we give new definition of the violation of cyclic inheritance and SoD, and show that there exist efficient algorithms to resolve these violations.

However, RBAC primarily consider static authorization decisions based on subjects' permissions on target objects, and there is no further enforcement during the access. Recently proposed usage control [3] models extend traditional access control models for next generation access control by integrating obligations, conditions as well as authorizations, and by including continuity and mutability properties, which make UCON have strong expressive power and policy specification flexibility. Role mappings are the basic approach for the interoperation

among multiple individual domains. The attribute mapping technique can be regarded as the extended of role mapping technique.

5 Conclusion

This paper presents an attribute mapping technique which can establish a secure interoperation in multidomain environments based on usage control policies. In order to ensure the security of individual domains in presence of interoperation, we study how conflicts arise and show that it is efficient to resolve the security violations of cyclic inheritance and SoD.

References

1. Kapadia, A., AlMuhtadi, J., Campbell, R., et al.: IRBAC 2000: Secure Interoperability using Dynamic Role Translation. University of Illinois, Technical Report: UIUCDCS-R-2000-2162 (2000)
2. ANSI. American National Standard for Information Technology-Role Based Access Control. ANSI INCITS 359-2004 (2004)
3. Park, J., Sandhu, R.: The UCONABC Usage Control Model. ACM Transactions on Information and System Security 7(1), 128–174 (2004)
4. Zhang, X., Parisi-Presicce, F., Sandhu, R., Park, J.: Formal Model and Policy Specification of Usage Control. ACM Transactions on Information and Systems Security 8(4), 351–387 (2005)
5. Zhang, X., Park, J., Parisi-Presicce, F., Sandhu, R.: A Logical Specification for Usage Control. In: 9th ACM Symposium on Access Control Models and Technology, pp. 1–10. ACM Press, New York (2004)
6. Bonatti, P., Vimercati, S.D.C., Samarati, P.: An Algebra for Composing Access Control Policies. ACM Transaction on Information and System Security 5(1), 409–422 (2002)
7. Dawson, S., Qian, S., Samarati, P.: Providing Security and Interoperation of Heterogeneous Systems. Distributed and Parallel Databases 8, 119–145 (2000)
8. Gong, L., Qian, X.: Computational Issues in Secure Interoperation. IEEE Transactions on Knowledge and Data Engineering 22(1), 14–23 (1996)
9. Dawson, S., Qian, S., Samarati, P.: Providing Security and Interoperation of Heterogeneous Systems. Distributed and Parallel Databases 8(1), 119–145 (2000)
10. Clark, D., Wilson, D., Kuhn, D.R.: A comparison of Commercial and Military Computer Security Policies. In: IEEE Symposium on Security and Privacy, pp. 184–195. IEEE Press, Los Alamitos (1987)
11. Shehab, M., Bertino, E., Ghafoor, A.: SERAT: Secure Role Mapping Technique for Decentralized Secure Interoperability. In: 10th ACM Symposium on Access Control Models and Technologies, Stockholm, pp. 159–167. ACM Press, Sweden (2005)
12. Shafiq, B., Joshi, J.B.D., Bertino, E.: Secure Interoperation in a Multidomain Environment Employing RBAC Policies. IEEE Transactions on Knowledge and Data Engineering 17(11), 1557–1577 (2005)
13. Lupu, E., Sloman, M.: Conflicts in Policy-Based Distributed Systems Management. IEEE Transactions on Software Engineering 25(6), 852–869 (1999)