SCIS 2006 The 2006 Symposium on
Cryptography and Information Security
Hiroshima, Japan, Jan. 17-20, 2006
The Institute of Electronics,
Information and Communication Engineers

Controlling Access in Databases by Outsourcing Access Control to Trusted Third Parties

Amril Syalim [*] Toshihiro Tabata [†] Kouichi Sakurai [‡]

Abstract— There are situations where users of databases cannot fully trust the administrators of databases where they store their data. In this paper we describe an outsourced access control model for such database. In this model, access control to databases is outsourced to trusted third parties. The tasks of the trusted third parties are: mediate access control to databases and store and manage policies for controlling access to databases.

Keywords: Access Controls, Database Security

1 Introduction

There are situations where users of databases cannot fully trust the administrators of databases where they store their data. An example is in a database service provider (DSP) model [2, 3, 4]. In a DSP model, users of databases outsource management of their data to a not fully trusted provider in the Internet. Another example is in large collaborated organization (enterprise or government) where there is no one trusted authority for managing access to the shared data of the organizations [5].

The problem is what access control model that is suitable with these situations. One solution is by leaving the access control task to every user. However, this solution has the potential problem in security and efficiency because users usually do not have much time and enough expertise in these fields.

Another promising solution is by outsourcing access control to trusted third parties. The trusted third parties are other parties which have capabilities and expertise and can be trusted by users for managing access control to his/her databases. In this paper we describe our model for this solution. The idea of our access control model is by using a multipolicy system to represent the access control authorities in databases that includes the database administrators, users and the trusted third parties. Outsourcing of access control is done between users and the trusted third parties by delegating access control authority from users to the trusted third parties. Data sharing between users is implemented by secure domain interaction using *policy groups* [8] where each user represents an access control domain and access from one user to another is represented by access from one user domain to another user's domain.

Organization of this paper: in section 2 we present the related works that include outsourced database model (ODB), outsourced RBAC and outsourcing paradigm for access control. Section 3 discuses outsourcing paradigm for access control. Section 4 discuses multipolicy system. Section 5 discuses policy group for multipolicy system. In section 6 we discuss our model to outsourcing access control for databases. And section 7 is conclusion.

2 Related Works

Related works that are relevant to be presented here are database service provider model [3, 4], outsourced role-based access control [5] and partial outsourcing paradigm for access control [1]. Research on database service provider model mainly focus to implement cryptographic mechanism (database encryption) and query techniques over encrypted data in an untrusted database server. By encrypting data in database, guarantees the administrator of database can not get information from user stored data, however this technique makes database query more difficult. Our work can be used in complement with this research where database encryption is used at the lowest level of access control enforcement mechanism.

Outsourced role-based access control model (RBAC) [5] is another work we based on. The difference is this work is based on RBAC model and the main method is by outsourcing the RBAC model to one trusted third party. Our work extends this idea by using multipolicy system so that we can outsource the access control to more than one trusted third parties.

Outsourcing paradigm for access controls discussed in [1] presented basic theory in outsourcing access control. The authors of [1] describe their argument for outsourcing access control that is mainly because the increasing complexity of security policy, specification

[*] Graduate School of Information Science and Electrical Engineering, Kyushu University, 6-10-1 Hakozaki, Higashi-ku, Fukuoka 812-8581, Japan (amril@itslab.csce.kyushu-u.ac.jp)

[†] Graduate School of Natural Science and Technology, Okayama University 3-1-1 Tsushima-naka, Okayama 700-8530, Japan (tabata@it.okayama-u.ac.jp)

[‡] Faculty of Information Science and Electrical Engineering, Kyushu University, 6-10-1 Hakozaki, Higashi-ku, Fukuoka 812-8581, Japan (sakurai@csce.kyushu-u.ac.jp)

and the resulting decrease in usability of security mechanisms so that it is more efficient and cost effective to outsource access control management to external expertise, compare with our main reason to outsource access control because the administrator of databases can not be trusted. The author in the paper [1] identified four classes of outsourcing concept: Class α: Single Internal Administration, Class β: Single External Administration, Class γ: Outsourcing via External Security Server and Class δ: Partial Outsourcing Using External Rule Servers.

3 Outsourcing Paradigm for Access Controls

With the increasing complexity of security policy, specification and the resulting decrease in usability of security mechanisms, outsourcing paradigm will become next shift in access control. There are four different classes of outsourcing models for access controls identified in [1]. This section briefly discusses these classes.

3.1 Class α: Single Administration, Internal

This class is the base class, no outsourcing takes place and both administration and database server belong to the same domain. Figure 1 shows this scenario. The circle around the database server symbolises the security domain of a database. Inside this domain all parties are understood to behave with integrity toward the common security goals. Security analysis of this setup is straight forward, because only one point of control exists. Authentication and authorisation of the client are fully handled by the database's own administration server. Advantages are those of central administration known today. Disadvantages include that the full administrational work has to be done by one party and no external expertise can be used. Implementation example: Access control frameworks employed today, e.g. a domain with a central domain controller, in which password is only known by users of databases.

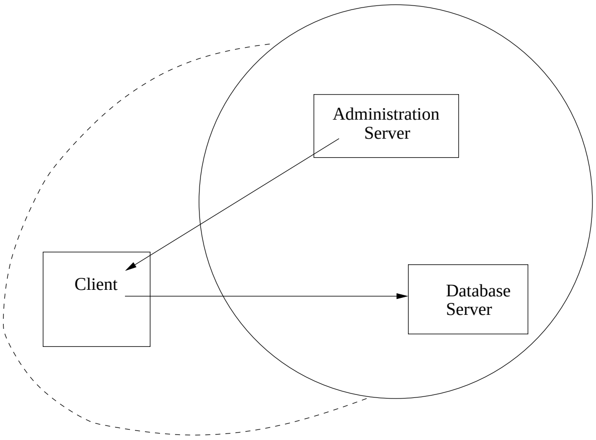

Figure 1: Class α: No Outsourcing, Single Administration

3.2 Class β: Single Administration, External

Figure 2 shows the opposite extreme case: administration is fully outsourced. Again the circle around the database server symbolises the parts fully natively trusted. The external administration server is not included in this domain, because, per definition, the trust in this server is artificial. This time the dotted circle shows the relationship that the client will belong to the domain of the administration server, as it obeys his policies. Finally, the big double circle shows the necessary scope of required trust to make the system work.

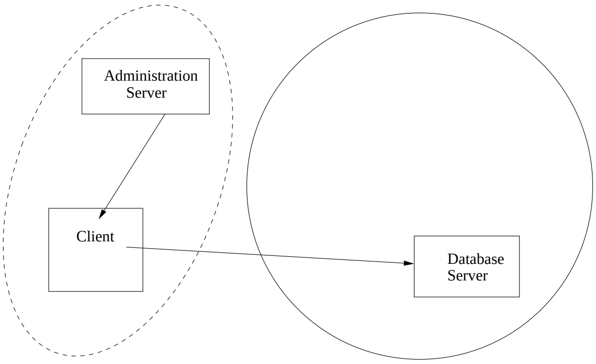

Figure 2: Class β: Fully Administrated by External Administration

3.3 Class γ: Outsourcing via External Security Server

In this class, as shown in Figure 3, administration is done by the database itself. Policies require the client to retrieve a credential of the external security server. This external security server is managed by the external administration. The multiple arrows hint that the same external security server might need to hand out credentials to a large number of different clients. Finally, the two dotted circles show that the client is dependent on the external security server, but also has to play by the rules of the administration server.

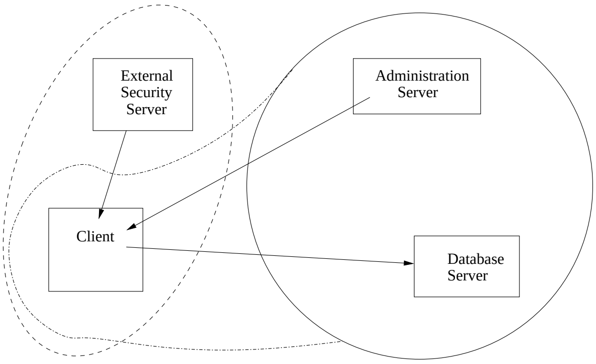

Figure 3: Class γ: Outsourcing Using External Security Server Approach

3.4 Class δ: Partial Outsourcing Using External Rule Servers

Figure 4 shows the final possibility. The external rule server delivers rule implementations to the local administration server. The administration server combines different rule implementations (here, Rule X+Rule Y) to policy definition. The dotted line around the client shows the known trust relationship. The second dotted line around the external rule server and object server indicates a natural trust relationship. This is so, because the external rule implementations are executed by the object server. For security the local administration has the full control over which rule implementations are used. Because the external rule implementations are not the only ones included, the external rule server cannot arbitrarily permit access.

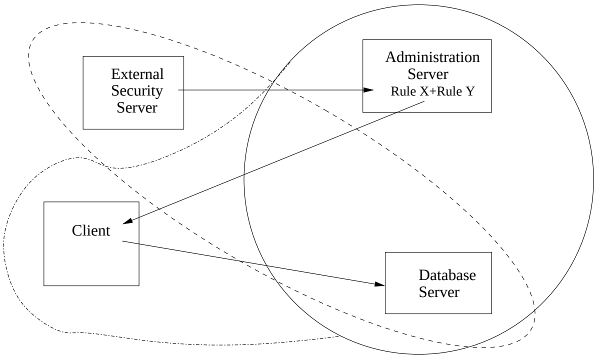

Figure 4: Class δ: Partial Outsourcing Using External Rule Server

4 Multipolicy Paradigm

The Multipolicy Paradigm permits a multilevel secure (MLS) system to enforce multiple, sometimes contradictory, security policies. Metapolicies, policies about policies, coordinate the enforcement of the multiple security policies. Policy domain codes on data indicate which security policies to enforce on the data, and multiple label segments supply the attributes needed for each policy.

The Multipolicy Paradigm permits natural modelling of the multipolicy real world. It permit possibly inconsistent security policies, such as confidentiality and integrity to operate together. It may provide a vehicle for users to add their own security policies to a system without disrupting or invalidating existing evaluated policies. It may ease policy integration problems by preserving the original classification of data when data is passed across policy boundaries. Finally, if implemented in high-speed parallel processing architecture, it may improve trusted system performance.

Multiple security policies may be necessary if:

1. There is more than one security goal, such as privacy, confidentiality and integrity
2. The system serves diverse constituents with individual goals and plans
3. The system is composed of separately evaluated pieces
4. Policies must adapt to changing circumstances

Several components that are required to handle multiple policies [6]:

1. Multiple security policies
2. Multiple security policy enforcers
3. Multiple policy coordinators (metapolicies)
4. Assignments to specify which policies apply to which subjects and objects

5 Policy Groups

A key to multiple security policies is the concept of information domains. The basic idea is an information domain is a set of entities (such as users, processes, files, mailboxes) together with a single encapsulating security policy. Multi domain systems will have as many information domains as there are security policies.

While information domains provide a clear concept to describe the relationship between a security policy and the entities it controls, information domains are a threat to interoperability; they tend to establish autonomous islands that are protected by their security policy, and it is unclear how trips between the islands can be made and managed with respect to security. The major obstacle to interoperability between information domains is the isolation enforced by its security policy. For any well defined domain it is precisely defined whether an entity belongs to the domain or not, and the domain's security policy will contain precisely all the rules that control the interactions of those entities in its domain. Now consider an interaction between entities that belong to different domains. Even in the most simple case when a subject from some domain P accesses an object in another domain Q, at least two policies are involved.

None of the two policies will be able to provide a rule for this access: policy P will not know the object and its security attributes within domain Q, and neither will policy Q know the subject and its security attributes within domain P . In a second example, domain P overlaps with domain Q, and subject and object are located within the overlapping section. In this case, both security policies know both entities and might come up with two and possibly conflicting rules.

A policy group combines a set of security policies with a set of policies that control interdomain actions. It composes a multipolicy system's security policies into a single structure and provides a single point of reference for the discussion of a system's security propertics.

5.1 Classification of Interdomain Actions

There are three classes of interdomain action [7]:

1. **Class 1**: $|\prod_s| = |\prod_o| = 1 \wedge \prod_s = \prod_o$
   Action of class one are characterized by the situation that subject and object are members of the same domain and are not member of any other domain. Actions within this class do not cross domain borders, and consequently, a single policy is both capable and authorized to make the access decision.

2. **Class 2**: $|\prod_s \cap \prod_o| = 0$
   This access class is characterized by the situation that no security policy exists that has both subject and object in its domain. Especially, there is no security policy that is capable of providing a rule for this particular access.

3. **Class 3**: $|\prod_s \cap \prod_o| \geq 1 \wedge \exists e \in \{s, o\} : |\prod_e| > 1$
   This access class is characterized by the situation that on the one hand there is (at least) one policy that might provide a rule for the access; nevertheless, on the other hand (at least) one of the entities is a member of more that one domain

5.2 Policy Group Definition

A policy group is defined as follows [8]:
Let I be a finite index set and $\{P_i\}_{i \in I}$ the set of regular security policies of a given multipolicy system. A *policy group* G is a tuple $G = (\{P_i\}_{i \in I}, T, F, c)$, consisting of [8]:

- a finite set of regular security policies $\{P_i\}_{i \in I}$, implementing the security requirements for class 1 accesses
- a completeness policy T, implementing the security requirements for class 2 accesses
- a conflict mediation policy F, implementing the security requirements for class 3 accesses
- a classification function c that for each access of subject s to object o: $(s, o), s, o \in \bigcup_{i \in I} Dom_{p_i}$, yields the class of (s, o).

6 A Model to Outsourcing Access Control for Databases

The idea is by using a multipolicy system to represent access control authorities in the system (the administrators, users and the trusted third parties). Outsourcing of access control is done between users and the trusted third parties by delegating access control authority from users to the trusted third parties. Data sharing between users is done by secure domain interaction using policy groups where each user represents an access control domain and access from one user to another is represented by access from one user domain to another user's domain.

6.1 Definitions and Basic Model

In this model, we define the independent security domains in databases include:

1. Administrators of databases
   Administrator of databases represent security domain of database administration. The scope of this domain is the system databases that are used for the whole database administration (for example: database of that records the list of users of databases, list of trusted third parties, and other system related databases)

2. Users of databases
   In this model each user represents one security domain. The scope of this domain is the user's databases.

3. Trusted Third Parties
   Each trusted third party is represented by one security domain.

This basic model of the system is illustrated in Figure 5. The outer box represents the whole authority in databases. In the system there are administrator domains, user domains and the trusted third party domains. In user's domains there are user's databases (represented by triangles).

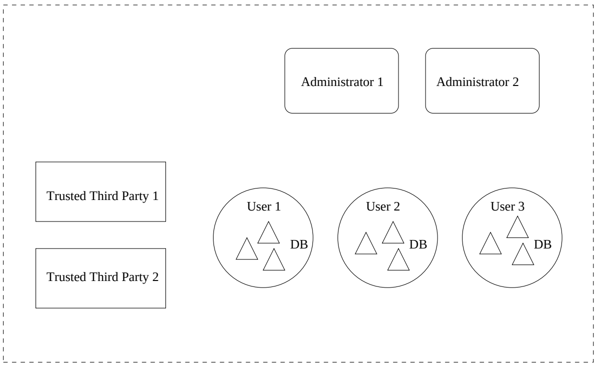

Figure 5: The Basic Model

6.2 Access Control Delegations and Data Sharing

Access control delegation between users and the trusted third parties and data sharing between users are resolved by secure domain interactions using policy groups.

1. Access control delegation

   When a user wants to delegate his authority to the trusted third party, he/she finds and chooses the trusted third party for delegation. The user defines general policy for his/her databases and then the trusted third party, based on general policy given by the user, creates policy group for the user's database. The user records his chosen trusted third party to a public database or a public directory that can be accesses by all database users so other users that want to access the databases can find it. This access control delegation is illustrated in Figure 6.

2. Data Sharing

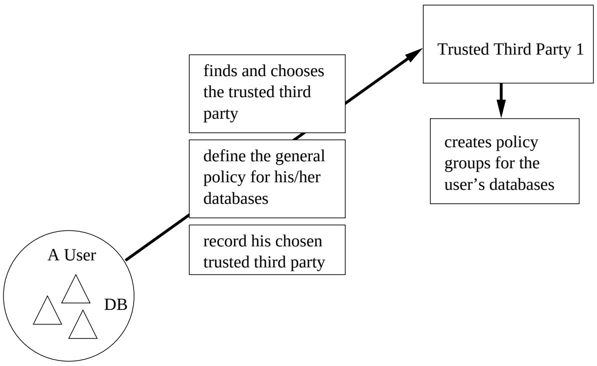
Trusted Third Party 1
finds and chooses the trusted third party
define the general policy for his/her databases
record his chosen trusted third party
creates policy groups for the user's databases
A User
DB

Figure 6: Access Control Delegation

When a user wants to access another user's database, he/she find the trusted third party that mediate access to the target user's databases and send request to the trusted third party to access the databases. By using policy group definition for the target user's databases, the trusted third party decides access control policy that will be enforced to the user when he/she accesses the databases. This data sharing model is illustrated in Figure 7.

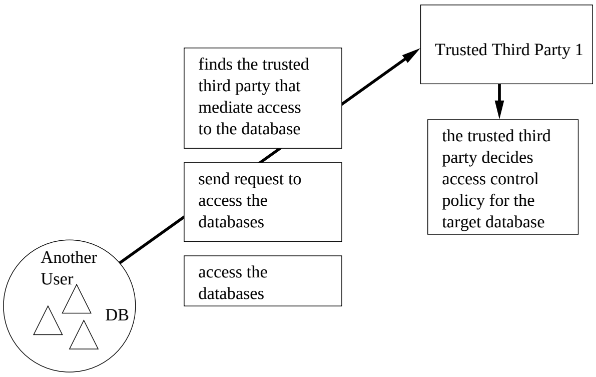
Trusted Third Party 1
finds the trusted third party that mediate access to the database
send request to access the databases
access the databases
the trusted third party decides access control policy for the target database
Another User
DB

Figure 7: Data Sharing Model

7 Conclusion

In this paper we described a model to outsourcing access control for databases. The idea of our model is by using a multipolicy system to divide the access control authority in databases into domains which represent the authorities in the system. We also described access control delegations and data sharing using policy groups.

References

[1] Joerg Abendroth and Christian D. Jensen. *Partial Outsourcing: A New Paradigm for Access Control.* SACMAT'03, 2003.

[2] Luc Bouganim and Philippe Pucheral. *Chip-Secured Data Access: Confidential Data on Untrusted Servers.* Proceedings of the 28th Very Large Data Bases Conference. Hongkong, China, 2002.

[3] Ernesto Damiani, S.De Capitani di Vimercati, Sushil Jajodia, Stefano Paraboschi and Pierangela Samarati. *Balancing Confidentiality and Efficiency in Untrusted Relational DBMS.* CCS'03. Washington, DC, USA, 2003.

[4] Hakan Hacigümüş, Bala Iyer, Chen Li and Sharad Mehrotra. *Executing SQL over Encrypted Data in the Database-Server-Provider Model* Proceeding of ACM SIGMOD Conference, Madison, Wisconsin, USA, 2002.

[5] Thomas Hildmann and Jorg Bartholdt. *Managing Trust between collaborating Companies using outsourced Role Based Access Control.* RBAC'99 Fairfax, VA, USA, 1999.

[6] Hilary H. Hosmer. *The Multipolicy Paradigm for Trusted Systems.* Proceedings on the 1992-1993 workshop on New security paradigms.1993.

[7] Winfried E. Kühnhauser. *A Classification of Interdomain Actions.* Operating Systems Review, 32(4):47-61, October 1998.

[8] Winfried E. Kühnhauser. *Policy Groups.* Computers & Security, Volume 18, Issue 4, 1999, Pages 351-363.