Integrating Different Semantics of Classification Levels in Heterogeneous Distributed Database Systems *

Min-Shiang Hwang† Wei-Pang Yang‡

Department of Information Management †
Chaoyang University of Technology
168, Gifeng E. Rd., Wufeng,
Taichung County, Taiwan 413, R.O.C.
http://www.cyut.edu.tw/~mshwang/
Email: mshwang@mail.cyut.edu.tw
Fax: 886-4-23742337

Department of Computer and Information Science ‡
National Chiao Tung University
Hsinchu, Taiwan 300, R.O.C.

February 27, 2002

*This research was partially supported by the National Science Council, Taiwan, R.O.C., under contract no.: NSC90-2213-E-324-005.

Integrating Different Semantics of Classification Levels in Heterogeneous Distributed Database Systems

Abstract

In this paper, we propose a security control mechanism for distributed database systems. The major contribution of this research is to develop some relationship rules for mapping the security levels among local DBSs in the MDBS. The main advantage is that the participating DBMSs remain autonomous.

Keywords: Distributed Database System, File and Storage System, Security.

1 Introduction

In recent years, distributed database systems have been a very active research area for storing high importance and frequency of transaction data in business and society. Today numerous types of database systems (DBS) are employed in the commercial, industrial, and military sectors. These DBSs are required for the management of different types of large, sensitive data objects.

A heterogeneous distributed database system (HDDS) is a system that handles several different DBSs. A HDDS has the following important features: (1) it supports interoperability among its local DBSs, and (2) each local DBS is autonomously managed by a local database management system (DBMS).

A HDDS can be classified into federated database (FDBS) and non-FDBS systems. The former could be further characterized into loosely and tightly coupled [11, 17]. In general, a HDDS must be capable of integrating local DBSs with different data models (e.g., relational, hierarchy, network, object-oriented data models), different database systems (e.g., Ingres, Sybase, DB2, Orion), and different security policies (e.g., discretionary access control or mandatory access control).

Security is an important issue for almost all kinds of information systems, including distributed, heterogeneous, autonomous multi-database systems, which makes it more difficult to enforce a global security mechanisms [3, 4, 8, 15]. Most commercial DBMSs provide security with discretionary access control. Recently, much research has been published concerning mandatory (or multilevel) access control for various data models, such as the relational model [7, 13, 18], object-oriented model [9, 10, 19], and multi-database model [12]. Although each local DBS may use a different data model or access control method, there are always two terms in the system: subjects and objects. An object is a data file, table of relations, record, or an instance of an object class. A subject could be a user, process, group, role, domain, or a program. Subjects and objects are classified into a number of distinct security classes within multi-level security systems, such as mandatory access control. In the Bell-LaPadula model [1], a subject is allowed a read access to an object only if its security class is greater than or equal to the corresponding security class of the object. A subject is allowed write access to an object only if its security class is less than or equal to that of the object. However, other models may adopt different security policies. It is difficult to integrate differing local security policies in HDDS.

In this paper, we propose a set of mapping rules to facilitate security within autonomous local DBS in a MDBS. We assume that each LDBS has its own data model, DBMS, and security mechanism corresponding to its access control security policy. The security policy of the HDDS depends on that of each local DBS.

This paper is organized as follows. In the next section, we review related works concerning security control mechanisms in MDBM. In Section 3, we present an architecture of the security control system for MDBS. Three modules in the architecture are described in subsections 3.1, 3.2, and 3.3, respectively. Finally, Section 4 presents our conclusions.

2 Related Works

Some of the difficulties in providing security for HDDSs arise from the fact that the local systems are largely independent from one another and autonomously administered. Within the HDDS, they are heterogeneous and are distributed across wide-area networks [14]. Many schemes for achieving partial security in HDDS have been proposed.

Wang and Spooner identified problems that arise when content-dependent access control is enforced in a HDDS [20]. They also proposed a solution that allows content-dependent access control policies to be enforced in each local DBS. Instead of using view replacement, which replaces a view name by its definition in a user query, they preprocess the view to materialize its data and treat the materialized view as a base object for processing the rest of the user query.

Goyal and Singh proposed an access control mechanism which supports content-dependent and functional access control policies [5]. They also implemented these policies by defining views and by controlling user access to these views.

Lu et al. proposed a multilevel security control scheme for MDBSs [12]. They built a front-end security management system in each LDBS and used a query modification approach to enforce multilevel security control. They also developed access rules for relational data models. In Lu et al.'s scheme, however, different data models use different access rules.

Thuraisingham and Rubinovitz identified the impact of multilevel security on the function of a HDDS [16]. They presented a multilevel security architecture for HDDS and discussed issues on query processing and transaction management.

3 The System Architecture

In this section, we present a multi-database security management system (MSMS), shown in Figure 1. This system consists of three major modules: the subject relationship definition module, the grant/revoke privileges module, and the security enforcement module, shown in Figure 2. The subject relationship definition module provides the global database administrator (DBA) with facilities that can define the security level for global users among the LDBSs. It also checks the completeness and consistency of the definitions. The grant/revoke privileges module ensures that those privileges propagated by the system can be revoked by the system security manager. The security enforcement module uses a security level mapping approach to enforce the security policy of each LDBS.

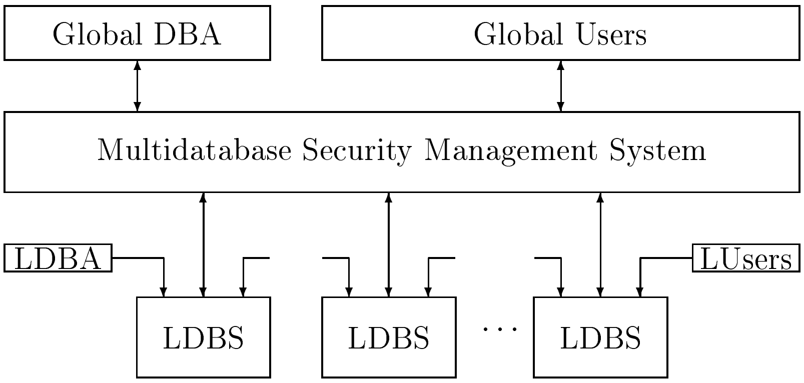

Figure 1: The architecture of the security control system for MDBS

We describe those modules in the following three subsections.

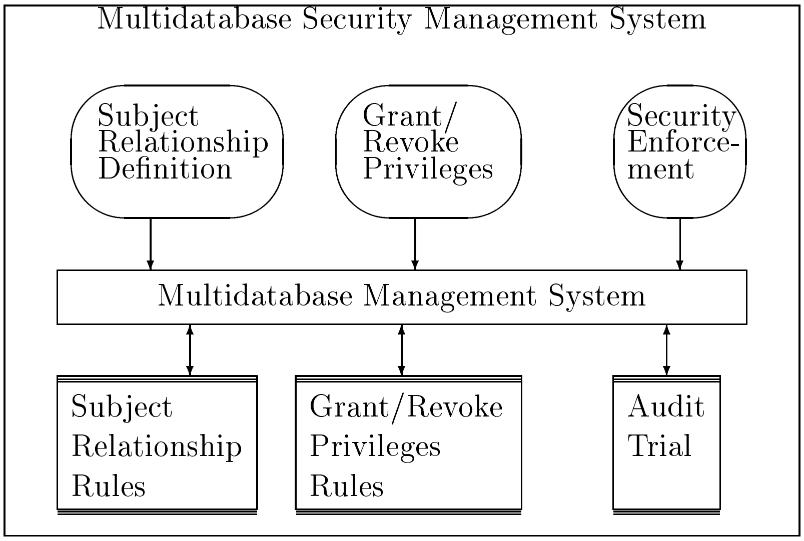

Figure 2: The architecture of the multidatabase security management system

3.1 Subject Relationship Definition

The subject (say, a user) may be a local user or a global user in our system. If a user is in the ith DBS (DBS_i), the user is called a local user in the DBS_i and is called a global user in the jth DBS (DBS_j, $j \neq i$). Every local user is assigned a security level when the account is registered in a local DBS. A user need not register in a local DBS. Its account can also be registered in the MDBS as a global user to every local DBS.

The security levels among the local DBSs are interrelated. This means that a subject who has the security level "Top-secret" in DBS_i may be denied access to data objects at the "Secret" level in DBS_j, because the "Secret" level in DBS_j may be higher than the "Top-secret" level in DBS_i in the global view of the MDBS. These relationships in the MDBS may form a hierarchy or a partial structure by means of the relationship rules. A relationship rule specifies which security level should be placed on a global user when the user attempts to access data objects in a LDBS.

Next, we present several properties of the relationships between DBSs. Let sl be a security level and sl_i be a subject in the ith DBS (DBS_i).

Property 1: (The lowest level) Any security level of subjects in local DBS_i, which enforces the security policy by discretionary access control, is assigned to the lowest security level. That is, $sl_i \leq sl_j$, for all DBS_j, $j \neq i$.

Property 2: (Equality) Two or more subjects in different DBSs with the same security level form a cycle structure. That is, $sl_i \leq sl_j$ and $sl_j \leq sl_i$.

Property 3: (Total hierarchy) Assume that the subject SU_i has the lowest security level in DBS_i and another subject SU_j has the highest security level in DBS_j. If $sl_i > sl_j$, their relationship forms a total hierarchical structure.

Property 4: (Tree hierarchy) Two subjects SU_i, SU_j have the highest level in DBS_i and DBS_j, respectively. If $sl_i = sl_j$, their relationship forms a tree hierarchy.

Property 5: (Poset hierarchy) Any two subjects SU_i, SU_j can have any security level in DBS_i and DBS_j, respectively. If $sl_i > sl_j$, $sl_i = sl_j$, or $sl_i < sl_j$, the relationship forms a partially ordered (poset) hierarchy.

Property 6: (Isolated structure) If two subjects SU_i in the DBS_i and SU_j in DBS_j do not satisfy property 5, their relationship forms an isolated structure.

Although subjects and objects are classified into a number of security levels only within multi-level security systems, like mandatory access control (MAC) but not within discretionary (DAC) or role-based security systems. However, each security level is defined by two components: a classification and a set of categories in the MAC system. The classification of subjects and objects is an element of a set composed of security class hierarchies, such as top secret, secret, confidential, and unclassified. The set of categories is a subset of a non-hierarchical set of elements, such as the access rights which allow subjects to read, write, own, delete, etc. objects. These access rights are the same as that of a DAC. Therefore, the mapping of DAC to MAC systems is based on the property that subjects of DAC systems are classified at the lowest possible security level in Property 1.

An example of the different relationship hierarchy structures formed by the above six properties is shown in Figure 3. Here, U_i denotes that the security level of the subject SU_i in DBS_i is unclassified; S_i and TS_i denote that the security level of the subject SU_i in DBS_i is Secret and Top-secret, respectively.

Assume that there are m LDBSs in the MDBS. The basic format of the relationship rules is as follows:

$$(sl_i, sl_{i1}, sl_{i2}, \cdots, sl_{im}) :: -DBS_i, DBS_1, DBS_2, \cdots, DBS_m$$

where sl_i is a security level in a local DBS_i and sl_{ij} is the security level in a local DBS_j for which a security level from sl_i maps to sl_j.

The relationship assignment is also closely related to the completeness and consistency of the relationship rules. The completeness ensures that each subject in the MDBS is assigned to a security level within the relationship hierarchy. The consistency ensures that no subject is assigned to more than one security level with different relationship rules in the relationship hierarchy. In our system, consistency is guaranteed because the relationship rule is used in only two cases. One is when the system is started up or initiated. The other is when a new LDBS is added to the MDBS. Since the security level of each LDBS is unique and identical, consistency is guaranteed. Completeness is also guaranteed, because a subject with any security level is obeyed among the properties from Property 1 to Property 5. We give the subject a default situation the same as that of Property 6 so as to form an isolated structure.

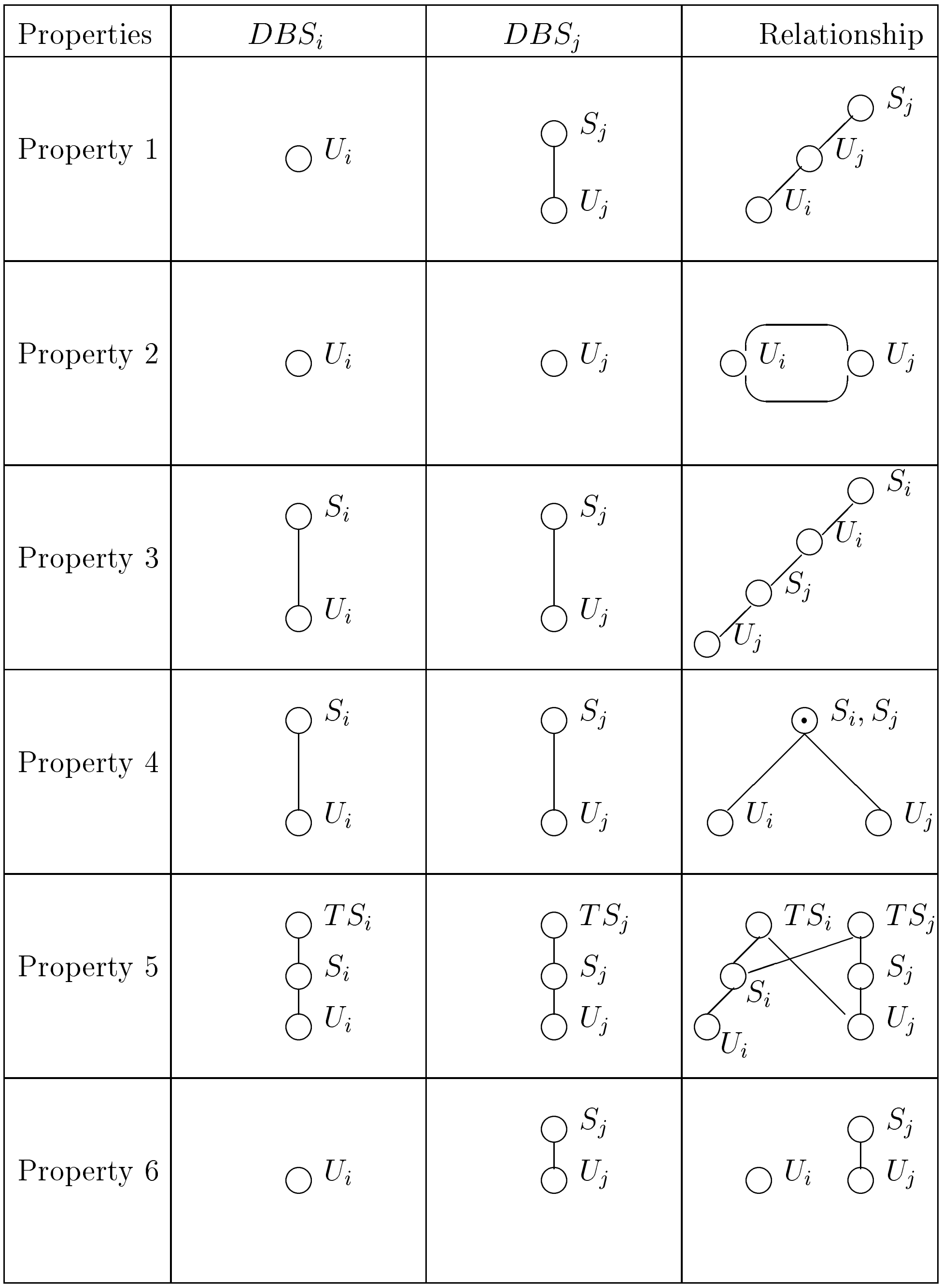

Figure 3: An example of the subject relationship hierarchy

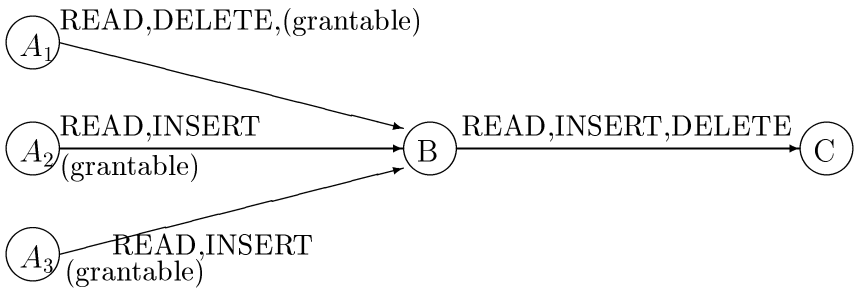

Figure 4: An Example of grant privileges

3.2 Grant/Revoke Privileges Rules

A subject is given privileges to select, create, update, and drop data objects using a GRANT statement. A subject may also be given the privilege to propagate his privileges to other subjects. The privileges granted a subject can be revoked by the security manager in the LDBS using a REVOKE statement [6].

A so-called information flow control problem [2] is raised when a subject is given the privilege to propagate his privileges to other subjects in DBSs. We explain the problem as follows. We assume that A_1, A_2, and A_3 grant certain privileges on the ORDERS table to B, who in turn grants them to C, as shown in Figure 4.

Now, we suppose that A_2 issues the REVOKE command to revoke all rights on ORDERS from B. Are B's grants of ORDERS also revoked? It is difficult for the system to make this decision because the system does not know who grants the READ and INSERT privileges via B to C. As a solution for this problem, Griffiths and Wade use a time-stamp concept [6]. When a subject issues a GRANT command. The system records the time-stamp indicating the relative time of any grant. Since the time-stamp is monotonically increases and no two GRANT commands are tagged with the same time-stamp, the system can revoke the privilege by comparing the time-stamps when the REVOKE command is issued.

Although Griffiths and Wades [6] use of the time-stamp concept can solve the information flow control problem, their scheme cannot be used in heterogeneous distributed database systems. Because the time-stamp for each LDBS is unique and every LDBS has its own time-stamp facilities, which are independent of those of other LDBSs, the time-stamps between the individual LDBS units may be inconsistent [14]. It is difficult for the individual LDBS to synchronize clocks. Therefore, an identical and unique time-stamp facility (hereafter referred to as a global time-stamp) is required in the MDBS. The global time-stamp is tagged by grant/revoke privileges rules. A grant/revoke rule specifies which privilege on the object is granted the other subject and has the following basic format:

$$(SU_i, privileges, object, SU_j, timestamp) \tag{1}$$

where SU_i is a grantor and SU_j a grantee. The meaning of (1) is that SU_i grants the privilege on the data object to SU_j at the time stamp.

When a subject SU_i in a local DBS_i wants to revoke the privileges he has granted to other subjects, the subject uses a REVOKE statement through the MSMS. The MSMS retrieves the grant/revoke privileges rules using the index of the grantor, privilege, and object, as in the following format:

$$(SU_i, privileges, object, SU_j, ts_1). \tag{2}$$

In security enforcement, the time-stamp ts_2, is used when a user invokes the REVOKE statement, and is compared with the ts_1 of (2). If $ts_1 \leq ts_2$, then the rule is deleted.

3.3 Security Enforcement

When a subject in local DBS attempts to manipulate data objects in MDBMS, the module derives which security level should be placed on the global user by the relationship rules and grant/revoke privileges rules. There are two cases which decide access rights. If the security level of the subject is higher than or equal to that of the objects in MDBMS, the module allows this access. Otherwise, the module rejects this access.

When a subject SU_i in a local DBS wants to access data objects in the same DBS, the local DBS allows the subject to access the data objects according to the local DBS's security policy. When the policy is used in the Bell-LaPadula model [1], the subject is allowed a read access to the object if its security class is greater than or equal to the corresponding security class of the object. The subject is allowed a write access to the object if its security class is less than or equal to that of the data object.

When a subject, SU_i in DBS_i, wants to access data objects in DBS_j, the multi-database security management system (MSMS) allows the subject to give queries that map his security level in DBS_i to that in DBS_j. The mapping process or discovery of the relationship from the semantics of the integrated database systems simply and efficiently retrieves the relationship rules using the index of the security level using the following format:

$$(sl_i, sl_{i1}, sl_{i2}, \cdots, sl_{im}) := DBS_i, DBS_1, DBS_2, \cdots, DBS_m. \tag{3}$$

The MSMS only maps the security level sl_i to an sl_{ij} that is a security level $s1_i$ in the local DBS_i which corresponds to that in the local DBS_j.

4 Conclusions

The proposed security control scheme for MDBSs has the following advantages.

1. It does not require that the data type and security policy in the existing LDBSs be modified. Thus, our scheme allows the LDBSs to remain autonomous.

2. Under the security control, each subject efficiently accesses data objects in other LDBSs with a proper security level. Thus interoperability among the LDBSs is supported.

3. Regardless of their data type and security policy, new LDBSs can easily be added to the system without affecting our scheme.

References

[1] D. Bell and L. LaPadula. Secure computer systems: Unified exposition and multics interpretation. Technical Report ESD-TR-75-306, Hanscom Air Force Base Technical Report, Beford, MA. (1975).

[2] Dorothy E. R. Denning. *Cryptography and Data Security.* Addison-Wesley, Massachusetts (1982).

[3] W. Essmayr, F. Kastner, G. Pernul, S. Preishuber, and A.M. Tjoa. Access control for federated database environmrnts. In *Proc. Joint IFIP TC6 and TC11 Working Conf. on Communications and Multimedia Security*, Graz, Austria (1995).

[4] W. Essmayr, F. Kastner, G. Pernul, S. Preishuber, and A.M. Tjoa. Authorization and access control in IRO-DB. In *Proc. of the 12th Int. Conf. on Data Engineering*, New-Orleans, Louisiana, USA. (1996).

[5] M.L. Goyal and G.V. Singh. Access control in distributed heterogeneous database management systems. *Computers & Security*, 10(7), 661–669 (1991).

[6] P. P. Griffiths and B. W. Wade. An authorization mechanism for a relational database system. *ACM Transactions on Database Systems*, 1(3), 242–255 (1976).

[7] J. He and M. Wang. Cryptography and relational database management systems. *2001 International Symposium on Database Engineering & Applications*, 273–284 (2001).

[8] M. S. Hwang and C. H. Lee. Secure access schemes in mobile database systems. *European Transactions on Telecommunications*, 12(4), 303–310 (2001).

[9] T. F. Keefe, W. T. Tsai, and M. B. Thuraisingham. Soda: A secure object-oriented database system. *Computers & Security*, 8(6), 517–533 (1989).

[10] S. Lewis and S. Wiseman. Securing an object relational database. *Proceedings of 13th Annual Computer Security Applications Conference*, 59–68 (1997).

[11] Witoldd Litwin, Leo Mark, and Nick Roussopoulos. Interoperability of multiple autonomous databases. *ACM Computing Surveys*, 22(3), 267–293 (1990).

[12] Hongjun Lu, Beng-Chin Ooi, and Hwee Hwa Pang. Multilevel security control in multidatabase systems. In *IEEE Conf. on Interoperability MDBS*, 359–363 (1990).

[13] Tereas F. Lunt, Dorothy E. Denning, Roger R. Schell, Mark Heckman, and Willian R. Shochley. The seaview security model. *IEEE Transactions on Software Engineering*, 16(6), 593–607 (1990).

[14] Matthew Morgenstern, Teresa F. Lunt, Bhavani Thuraisingham, and David L. Spooner. Security issues in federated database systems: panel contributions. In *Fifth IFIP WG 11.3 Workshop on Database Security*, Shepherdstown, West Virginia, U.S.A. 131–148 (1992).

[15] M. G. Piattini and E. Fernandez-Medina. Secure databases: state of the art. *Proceedings of IEEE 34th Annual 2000 International Carnahan Conference on Security Technology,* 228–237 (2000).

[16] Harvey Rubinovitz and Bhavani Thuraisingham. Design and implementation of a query processor for a trusted distributed data base management system. *Journal of Systems Software*, 21, 49-69 (1993).

[17] Amit P. Sheth and James A. Larson. Federated database systems for managing distributed, heterogeneous, and autonomous databases. *ACM Computing Surveys*, 22(3), 183–236 (1990).

[18] Paul D. Stachour and Bhavani Thuraisingham. Design of ldv: A multilevel secure relational database management system. *IEEE Trans. on Knowledge and Data Engineering*, 2(3), 190–209 (1990).

[19] B. M. Thuraisingham. Security in object-orientedd database systems. *Journal of Object-Oriented Programming*, 2(6) (1990).

[20] Ching-Yi Wang and David L. Spooner. Access control in a heterogeneous distributed database management systems. In *IEEE Proc. of the 6th Sympo. on Reliability in Distributed Software*, 84–92 (1987).