Multilevel Databases

Alban Gabillon
IUT de Mont de Marsan, France

INTRODUCTION

In the context of multilevel security, every piece of information is associated with a classification level, and every user is associated with a clearance level. The classification and clearance levels are taken from the same set of security levels. This set is totally or partially ordered and forms a lattice. The ordering relation is called the dominance relation and is denoted by $\geq$. An example of a totally ordered set is \{Unclassified, Confidential, Secret\} with Secret $\geq$ Confidential $\geq$ Unclassified. An example for a partially ordered set is \{low, (Secret, NATO), (Secret, Defence), high\} with (Secret, NATO) $\geq$ low, (Secret, Defence) $\geq$ high, (Secret, Defence), (Secret, NATO) and (Secret, Defence) are incomparable.

In multilevel security, the security policy (also called the confidentiality policy) states that a user has the permission to know a given piece of information only if the clearance level of that user dominates (i.e., is higher than or equal to) the classification level associated with the piece of information. Multilevel security is traditionally opposed to discretionary security. In discretionary security, the security rules (i.e., permissions or prohibitions) explicitly refer to users’ identities.

BACKGROUND

Most of the existing multilevel security models for databases are based on the following properties (Bell & LaPadula, 1975):

- **No read up:** This rule states that a subject at level X cannot read information at level Y if Y strictly dominates X.
- **No write down:** This rule states that a subject at level X cannot write information at level Y if X strictly dominates Y.

In these rules, subject refers to a user or a process. The process level is the working level of the user, on behalf of which that process executes. The level at which the user decides to work can be his or her clearance level or any level that is dominated by the user’s clearance level. The no write down restriction is necessary to prevent high level processes from illegally disclosing sensitive data. Consider a program that contains a Trojan horse, and assume a user with a high clearance decides to work at a high security level and run that program without knowing that it is infected. Without the no write down restriction, the Trojan horse would be capable of writing high classified data into a low level information container, making the high level data available to users with low clearance.

The Bell and LaPadula (1975) rules do not disallow write up. However, in multilevel databases, write up is very often prohibited because of the integrity problems arising from its blind nature (Thomas & Sandhu, 1993). The Bell and LaPadula properties are necessary to enforce the confidentiality policy, but they are not sufficient. Indeed, it is possible to illegally transmit data by other means than simple read and write operations. In the literature, such unauthorized communication paths are referred to as covert channels. Covert channels can be of several types: timing channel, inference channel, and signaling channel (National Computer Security Center, 1993).

Multilevel security is for applications requiring a high confidentiality level. It is mainly used by military organizations, and it is sometimes called military security. Many multilevel security models have been proposed for multilevel relational databases (see Denning, Lunt, Schell, Shockley, & Heckman, 1988; Haig, O’Brien, Stachour, & Toups, 1990; Jadodia & Sandhu, 1991b; Jukic, Vrbsky, Parrish, Dixon, & Jukic, 1999; Qian & Lunt, 1996; Sandhu & Chen, 1998; Smith & Winslett, 1992) and object-oriented databases (see Jadodia & Kogan, 1990a; Keefe, Tsai, & Thuraisingham, 1989; Lunt, 1990; Millen & Lunt, 1992; Cuppens, Gabillon & Yardanian, 1993, 1997, 1999). Experience has shown that when designing a multilevel security model for a database, the following issues must be addressed:

- Granularity of the data must be defined. This means the data structures that will later be classified must be identified. For example, in the context of relational databases, should security levels be assigned to tables? To rows? To attributes? To attribute values? Once the information granularity is defined, the semantics of the association between a security level and a granule must be specified.
Inference channels must be prevented. When labeling the data with security levels, one should make sure that no sensitive data can be derived from low level data.

Multilevel database must be decomposed into a collection of single-level views. Indeed, each user has to be provided with a consistent and complete view of the multilevel database that is compatible with his or her security level.

Operational semantics for the different update operations must be defined.

Throughout this paper I shall illustrate these points with a small multilevel relational database reduced to a single relation and shall consider the following set of security levels: \{Unclassified (U), Confidential (C), Secret (S)\}.

**MULTILEVEL DATABASE DESIGN**

**Information Granularity**

When designing a multilevel security model for a database, the first problem is to define the information granularity. The finer the information granularity, the better the expressive power of the model. For example, a model for relational databases in which only tables or rows can be classified would be a model with a poor expressive power, whereas a model in which attribute values (i.e., intersection between a row and a column) can be classified would be a model with a great expressive power.

In fact, defining the information granularity depends on the application needs. To illustrate my point, consider the Wing relation (see Table 1). Primary key of this relation is \{Name\}.

This relation represents an imaginary French American wing ready for attack. The military security administrator of this database knows the various sensitivities of the data contained in this table:

- The existence of the wing is unclassified.
- The existence of each plane except Firefox is unclassified.
- The existence of Firefox is confidential.

Analyzing these facts suggests a model in which tables, attributes, and attribute values can be classified. The security level assigned to a table protects the existence of the table. The security level assigned to a primary key value protects the value itself, and therefore the existence of the entity that is identified by the value. The security level assigned to a nonkey attribute value protects the value itself.

 knowing this, data contained in the wing relation are classified, as shown in Table 2.

From this simple example is one important lesson: Designing a multilevel security model with a fine granularity is not an insuperable problem, provided that the semantics of the association between a granule of information and a security level is precisely defined. This approach was not always followed. For example, Sea View (Denning et al., 1988), which is one of the first multilevel security models for relational databases, assigns security levels to rows as well as to primary key values. Unfortunately, the semantics of such associations is not clearly defined.

**Inference Control**

Again, the Bell and LaPadula (1975) rules are necessary to enforce the confidentiality policy, but they are not sufficient. Covert channels can be used to disclose sensitive data. Many types of covert channels cannot be represented in the security model because many of them are due