The instance-based multilevel security model (IB MSM)

Jianmin Su
Memorial University of Newfoundland
St. John’s NL Canada A1B 1P4
709-737-2369
jianmin@cs.mun.ca

ABSTRACT
The basic motivation of multilevel database systems is to share data from different security levels. However, current models have to rapidly store some information in different security levels to protect signal channel. Leaks could also occur when current models deal with the null values. In this research we propose a novel security model, the instance-based multilevel security model to solve the current problems. The proposed work includes defining the model, extending the SQL operation to the model and comparing the model with the current models in three areas, complexity, efficiency and reliability, and to point out the advantages of the proposed model in each area.

Categories and Subject Descriptors
H.2.0 [Database Management]: General – Security, integrity, and protection.

General Terms: Security

Keywords
Access control, confidentiality, instance-based security model, security, multilevel security, polyinstantiation

1. INTRODUCTION
A multilevel database is intended to satisfy the security needs for database systems that contain data at a variety of classifications and serve a set of users having different clearances. The basic motivation of multilevel database systems is to share data from different security levels. Many multilevel security database models have been proposed. Different models have advantages in different applications. For example, the Bell-LaPadula model points out the basic needs of multilevel security systems. The model is expressed using terms of objects and subjects. An object is used to express a passive entity such as a record, or a field within a record. A subject is used to express an active process that can request access to objects. Every object is assigned a classification and every subject a clearance. Classifications and clearances are expressed as labels. A label expresses sensitivity of information. Labels are assigned in hierarchical sensitivity order. For example, a military organization might define levels from Top Secret, Secret, Confidential to Unclassified as sensitive labels TS, S, C, and U (here we have the hierarchical TS>S>C>U).

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Top Secret, Secret, Confidential to Unclassified as sensitive labels TS, S, C, and U (here we have the hierarchical TS>S>C>U).

Given two labels S1 and S2, S1 >= S2, the Bell-LaPadula model proposed the following two restrictions on all data accesses:

1) No Read Up: A subject is allowed a read access to an object if and only if the subject’s label is higher than or equal to the level of the object’s label.

2) No Write Down: A subject is allowed a write access to an object if and only if the object’s label is higher than or equal to the level of the subject’s label.

2. PROBLEM STATEMENT
The purpose of the two restrictions above is to avoid any upward signal channel between each security levels. SeaView [1] and the following multilevel security model define the polyinstantiation integrity rule to achieve the goal. However, polyinstantiation integrity rule is not a data sharing rule.

Another problem is the inference when we are dealing with sensitive data in multilevel security models. An inference channel in a database is a means by which one can infer data classified at a high level from data classified at a low level. The inference problem is the problem of detecting and removing inference channels. An inference of sensitive data from nonsensitive data can only be represented within a database if the nonsensitive data itself is stored in the database.

The difficulties with multilevel security models arise from the basic concepts of class-based models, rather than multilevel security models themselves. In a class-based database, a schema is the global view of the data in the database. This schema forms a closed world for any data in the database and also the users. Because of the existence of such schema, any data (and/or users) have to be assigned to a certain (global) security level; this is where problems arise.

3. IB MSM
In this research we propose a new multilevel security data model, the instance-based multilevel security model (IB MSM), to solve problems. IB MSM is based on the instance-based database model [2][3]. It consists of two levels: The instance level and the class level. The definition of the two layers as following:

Definition 1: An instance is denoted by i (\{P_i, \{P_{1.C_j}\}\}) | P_i \in \text{P} \text{ and } C_j \in \text{C}, where i is an instance identifier, P_i is a property over the domain \text{P}, P_{1.C_j} indicates that property P_i belongs to security level C_j, which C_j is a classification level over the classification domain \text{C}.
Definition 2: A class is denoted by Class_ID (\{P_i\}, \{U_i\}), where Class_ID is a class identifier, \{P_i\} is a subset of properties over domain P, and \{U_i\} is a subset of user identifiers over all the system.

As with most multilevel security models, there is a rule in the instance layer: A certain level security subject can read objects consist of two parts; the objects owned by the subjects of the security level and the objects owned by lower-level security subjects. The latter can be changed by the lower-level subjects who own them. A subject can only update the objects at the security level that is the same as the subject, it cannot update objects in other security levels even it can read them.

3.1 Data interpretation in IBMSM
For all instance i \{(P_i,\{P_i,C_i\})\} \in P and \{U_i\} and for all \{P_i\} \in i, we define data are interpreted as follows:

3.1.1 Property P, and its security level P_{i,C}

a. An instance possessing a property P_i and its security level C_j are denoted as a pair of \(\{(P_i,P_{i,C})\}\). However, since an instance may possess the property at more than one security level, we use \(\{(P_i,P_{i,C})\}\) to denote more than one pairs of \((P_i,P_{i,C})\), like \((P_i,P_{i,C1}), (P_i,P_{i,C2}), \ldots, (P_i,P_{i,C10}), \ldots\). For the same reason, if an instance possesses more than one property that belongs to one security level C, we denote them as \(\{(P_i,C)\}\) as short.

b. An instance possesses a pair of a property at a security level \((P_i,P_{i,C})\) means a C_j level subject create a property P_i of the instance. This property can only be changed by C_j level subjects.

c. A subject at security level C can create a property P_i, that is \((P_i,C)\), of an instance if and only if the instance does not have property P_i at all (at any security level) or the instance does not have property P_i at the security level C.

d. A subject at security level C can read a property P_i of an instance if and only if \(C \geq P_{i,C}(=C_j)\).

3.1.2 An instance and its security levels

a. An instance identifier identifies an instance in the database. \(i(C)\) means that is an instance which possesses some properties at security levels C. to express an instance possesses properties belong more than one security levels we use \(i(\{C_j\})\), which means there is an instance which possesses some properties at security levels \(C \in \{C_j\}\).

b. A user at security level C can read the instance identifier of an instance \(i(\{C_j\})\) if and only if \(C \geq C'\), which C' is the lower bound of \(\{C_j\}\).  

3.2 Rules

Rule 1. Instances are identical at a certain security level. Instance \(p\) and instance \(q\) are identical at a security level C, if and only if for any property \(P_i\in P\) if \(P_{i,C} = C\), then there always are \(\{P_i, P_{i,C}\} \in p\) and \(\{P_i, P_{i,C}\} \in q\).

Rule 2. Instances are identical in the database. Instance \(p\) and instance \(q\) are identical in the database, if and only if for any property \(P_i\in P\) there always are \(\{P_i, P_{i,C}\} \in p\) and \(\{P_i, P_{i,C}\} \in q\).

Rule 3 (instance polyinstantiation) Two instance \(i\) and \(j\) satisfies instance polyinstantiation if and only if for any \(i, j \in I\) and \(P_i \in P\):

(1). \(\{(P_i, C)\} \in i, j\) which C \in C and

(2). \(\{(P_i, C)\} \in i, j\) which C \in C

Rule 4. A subject, \(S_i\), can access data through a class, \(CL(\{P_i\}, \{U_i\})\) if and only if \(S_i \in \{U_i\}\).

Rule 5. A subject, \(S_i\), can access data through a class, \(CL(\{P_i\}, \{U_i\})\), if and only if \(S_i \in \{U_i\}\).

3.3 Two-layer access control

The two-layer security approach is different from the original multilevel security model in class-based models. There are advantages associated with the two-layered approach. The greatest advantage is that users (subjects) with the same security level may be authorized to access different data (objects).

4. METHODOLOGY OVERVIEW

The research of the IBMSM in this project will include four parts:

First, we propose a multilevel security model followed by Bunge’s ontology and IBDM. There are three main parts in the proposed model: model definition, data interpretation and the definition of integrity rules. A new control model (two-layer control model) is also proposed for the model.

Second, we point out the manipulation of the proposed model. The manipulation includes the traditional SQL statements such as insert, delete, update, etc. and some extensions of the instance-based model. Examples and formal syntax and semantics for these statements will be presented in this part.

Third, a comparison of IBMSM and the traditional class-based multilevel security models will be presented. The comparison will focus on the ability, the adaptability, and the complexity. This part will evaluate the works done above but focus on the two-layered access control method. There is also a gap between costs and the benefits of a security technology. The two-layered access control will reduce this gap. The evaluation will be largely based on the comparison of the two-layered access control with the multilevel relational data model in multilevel applications.

Finally, after the theoretical comparison, a practical comparison will be presented based on a prototype of the two-layer access control.

5. REFERENCES

