Proceedings of the 7th International Conference on Applied Informatics
Eger, Hungary, January 28–31, 2007. Vol. 1. pp. 353–359.

Developing an MLR security layer for relational databases

László Kovács[a b], Zoltán Irházi[b]

[a]ME University of Miskolc, Department of Information Technology
e-mail: kovacs@iit.uni-miskolc.hu

[b]KEFO College Kecskemét, Institute of Information Technology
e-mail: irhazi.zoltan@gamf.kefo.hu

Abstract

A key factor in data management is the security of the databases. The security systems for databases with higher security demand are usually based on the DAC and MLR models. In the existing MLR models, either the operations are too complex to manage or the granularity of the security systems is not appropriate. In our investigation, a model was developed that merges some of the good properties of the basic methods. The main goal of the proposed model is to provide a transparent MLR layer for the database users.

1. Introduction

A key factor of the data management is the security of the databases. One of the main components in the security field is the management of access control. The drawback of the liberal models, like the DAC model, is that the information can flow uncontrolled within the database. A more restricted model is the MAC (mandatory access control) model where the database administrator can determine a security grid. In the basic MAC model for relational databases, the security objects are the tables. There are some proposals in the recent years to provide a more flexible and controlled security model at a finer granularity level.

In the MLR (multi-level relation) models, the basic security unit is the attribute-level. An α-level cut of a relation R is a subrelation R' that contains those elements from R where the security is dominated by the α-level. As a modification of an object is visible only for users of higher levels, the same object may exist at different security levels with different instance values. Thus, the MLR model should support the polyinstantiation of the database objects.

Analyzing the existing MLR models, all the models have some drawbacks. Either the operations are too complex to manage or the granularity of the security systems is not appropriate. In our investigation, we have developed a model that

merges some of the good properties of the basic methods. The paper describes the core properties of the proposal and compares it with the traditional models. In the second phase of the investigation, the model was implemented in a test system. As an implementation platform, the VFP desktop database environment was selected. The paper demonstrates our first experiences with the test system. The main goal of the implementation is to provide a test environment for educational purposes.

2. The MLR security model

If the different attribute values within the same relation can belong to different security levels, the relation should contain not only the normal user-data but the security information too. This leads to the concept of multilevel or extended relation.

Definition 2.1. A multilevel relation schema is represented with $R(C_0, A_1, C_1, \ldots, A_M, C_M)$ where

- A_i : a normal user-data attribute;
- C_0 : the tuple-level access class of tuple;
- $C_i, i > 0$: the access class of an attribute.

The interpretation of the access class depends on the concrete MLR model. Usually, the instances of the access classes constitute a lattice or sometime a chain.

In the MLR model, the access class structure is based on the MAC security model. In the Sea View / MAC model [4], an object or subject is descripted with an access class consisting of two components: a secrecy class and an integrity class.

Definition 2.2. A $C(L, I)$ access class is a tuple of secrecy and integrity classes. The $L(S_L, A_L)$ secrecy class determines the security level and the security domain. The $S_L \in (S, \leqslant)$ security level has a value from a chain of level values. In the basic version $S = \{'U', 'C', 'S', 'TS'\}$. The $I(S_I, A_I)$ integrity class determines the reliability level and the reliability domain. The $S_I \in (I, \leqslant)$ integrity level has a value from a chain of level values. In the basic version $I = \{'U', 'I', 'VI', 'C'\}$

The decision what kind of access can have a subject to an object, depends on the corresponding access classes. The access rules of the model are based on the Bell-LaPadula [1] and on the Biba [2] models. The relationship between the access classes is given with the concept of dominancy. The core elements of the access rules can be summarized as follows.

- A subject can read an object only if its access class dominates the access class of the object.
- A subject can insert new data into an object only if its access class is dominated by the access class of the object.

The main purpose of the rules is that users can not read data on higher levels and can not transfer data to lower levels. The different MAC versions add some refinements to this basic model.

The main focus of our investigation was on the viewpoint of database users and of DBMS implementation. In our opinion, the following problems may arise during the usage of the classical MLR models:

- The α-level cuts contain access-class descriptors. This means that the users are permanently confronted with existence of other security layers. Our opinion is that the existence of other layers should be hidden from the users. The users should have the feeling that what they see is the whole and true database. The suggested models can not provide consistent α-level cuts without access class descriptors.

- Because of the access class descriptors in the schema, the existing interfaces to the relational models can not be used for these models. Thus, this model is not open for the standardized database connections. It would be desirable to have such views for the different levels that have the same relational structure as the normal relational model has.

- In the Jajoda and Sandhu model, there is a high level of storage redundancy, as a data item is copied into every instances that dominates the user. The required space cost is significally higher compared with the normal relational database. In the Smith and Winslett model, this redundancy is lower but still present.

- The storage redundancy causes some drawbacks in the database efficiency. A DML command can cause modification in several physical tables that means extra execution costs.

- The models include some additional integrity rules to preserve the consistency between the different α-level cuts. These rules require the invocation of additional time-consuming procedures during the data manipulation operations. These also decrease the DBMS efficiency.

3. Development of a hybrid model

Based on the analysis of existing models, a modified version is suggested to cope with the mentioned drawbacks. The main characteristics of the proposed model can be summarized in the following list:

- it meets the Bell-LaPadula requirements;

- both the records and the attributes have security labels;

- it has only logical polyinstantiation, no physical polyinstantiation at the record level;

- three-value state variables to manage the security assignments;
- efficient data manipulation algorithms.

3.1. Structure part of the hybrid model

The main purpuse of the modifications is to provide a natural view to the users, to hide the existence of other security layers. In this model, the α-level cut borrows data from lower levels only if the current level has no information on the data value. The α-level cuts do not contain access information, the structure of the cut is a normal relational structure.

In this approach, the identification of the objects is global. Thus taking a person for example, not every layer may have information about the existence of the person, but the layers that have information about it, use the same identification key value.

The proposed model uses only one instance for the MLR relation. This instance stores all the normal data and access information of the different access levels. This solution avoids the storage redundancy. The α-level cuts are derived from this instance table.

Due to the redundancy elimination, the operation on the α-level cuts are more easier. The model includes only few simple integrity rules similar of the rules in the relational model.

The structure of the proposed model can be given on the following way.

Definition 3.1. An MLR relation is a triplet $r(R, i, v)$, where

- R is an MLR schema of form $R(C_1, \ldots, C_N, A_1, A_2^1, A_2^2, \ldots, A_2^N, A_2^1, \ldots, A_2^N, \ldots, A_M^N)$. The Nsymbol denotes the number of different access classes. The symbol M is the number of attributes. The A_1 denotes the key attribue.
- i is the relation instance of schema R.
- v is the set of α-level cuts of i. Every α-level cut belongs to the schema $V(A_1, \ldots, A_M)$.

The C_i tuple level markers may have three values: 'Y','N','U'. These markers denote the belief of the users about the existence of the tuple. The meaining of the values:

- Y : the user is sure that the tuple exists
- N : the user is sure that the tuple does not exist
- U : the user has no knowledge about the existence of the tuple

If the C_i marker denotes an unknown state, the largest lower bound level with 'Y' or 'N' value is considered for decision making. If there exists no lower bound, then the 'N' value is assumed. The value generated on this way is called default value

of C_i. The default value of an attribute in a tuple is the value at the largest lower bound level having no NULL value of the same attribute. If there is no such largest lower bound exists, the default value is NULL. For example in

A_U	A_C	A_S	A_{TS}
NULL	10	NULL	12

the deafult A value for the TS level is 12, for the S and C levels is 10 and for the U level is NULL.

The derivation of the α-level cuts r_α is performed on the following transformation rule:

- A $t \in r$ is present in r_α if and only if the default value of C_α is 'Y'.
- The value of attribute in r_α is equal to its default value related to the α level.

The scope of the primary integrity rule is the whole r instance table. In this model, a user may use a lower security level for the SELECT command to browse the believes of the lower layers. For this purpose, the BELIVED BY L tag ([6]) can be used where L denotes the required access level.

3.2. Operational part of the hybrid model

In the operational part, the basic data manipulation operations are discussed. Based on the simple representation schema, also the operational rules can be described with simple formulas.

- INSERT rule: the tuple can be inserted into r if there exists no tuple with the same key or there exists such tuple but the default C_α value at the given level is 'N'. In the new or old tuple, C_α is set 'Y'. The A_i^α values are set to the given attribute values.
- DELETE rule: the tuple can be deleted if there exists a tuple with the same key and the default value of C_α is 'Y'. After the delete oparation the C_α is set 'N'. The A_i^α values are set to NULL values. The tuple can be removed from r, if there is no C_i with a 'Y' value.
- UPDATE rule: the tuple can be updated if there exists a tuple with the same key and the default value of C_α is 'Y'. After the update oparation the C_α is set 'Y' and the corresponding A_i^α values are set to the given attribute values.

In order to see the differences between this model and the traditional models, the same operations are executed on the $R(K, A, B)$ schema.

- INSERT INTO R VALUES (1, 10,10) [level TS]
- INSERT INTO R VALUES (1, 11,11) [level S]
- INSERT INTO R VALUES (1, 12,12) [level C]

- INSERT INTO R VALUES (1, 13,13) [level U]
- UPDATE R SET A=14 WHERE B=12 [level S]

Having an empty initial state, the resulted instances are the following in the proposed model:

C_U	C_C	C_S	C_{TS}	K	A^U	A^C	A^S	A^{TS}	B^U	B^C	B^S	B^{TS}
Y	Y	Y	Y	1	13	12	11	10	13	12	11	10

This instance is a very compact form of the MLR relation. In the example, the UPDATE command will modify no rows as there is no record with B=12 value at the S-level.

4. The structure of the MLR module

To demonstrate the funcionality of the model, a small test system was developed. The test system implements a MLR modul layer for a relational database. The MLR module is a simple middleware between the application and the database. The modul is implemented as a set of functions, providing a simple API for the VFP applications. The API consists of three interface functions:

- MLR_CONNECT : it identifies and connects a user to the module.
- MLR_DISCONNECT : it dettaches a user
- MLR_EXECUTE : it executes a normal SQL command on the MLR relations

The management of the MLR relations requires some administration tables. The key tables are in the system

- USERS : the account data of the users
- MLRTABLES : the description of the MLR tables

The conversion module receives a normal SQL command. The module generates temporary tables to perform the required operations. The operations are executed on the temporary tables. After finishing the operation, the base MLR table is synchronized using the temporal tables.

The following example shows a simple application routine using the MLR module.

```
PROCEDURE TEST1
  LOCAL st_ST, st_VA, nu_VA,CI,RE

  st_VA = INPUTBOX("‘MAX. VALUE"’)
  nu_VA = VAL(st_VA)
```

```
st_ST =  "‘SELECT COUNT(*) FROM BOOKS WHERE PRICE < "‘
     + ALLT(STR(nu_VA))

CI = MLR_CONNECT(UP)
  IF (NOT EMPTY(CI)) THEN
     RE = MLR_EXECUTE(CI, st_ST)
  ENDIF
  MLR_DISCONNECT(CI)
```

5. Conclusion

The proposed model can provide a high level of transparency of the additional security layers. The schema is consistent with the traditional DAC relational security model. The key features are, that the α-level cut borrows data from lower levels only if the current level has no information on the data value and it does not contain access information, the structure of the cut is a normal relational structure. The identification of the objects is global. The proposed model can be implemented on a cost effective way. The developed system is used in the education for demonstrating the MLR security model.

References

[1] Bell, D., LaPadula, L., Secure computer systems: mathematical foundations, ESD-TR-73-278, (1973).

[2] Biba, K., Integrity considerations for secure computer systems, ESD-TR76-372, (1976).

[3] Castano, S., Fugini, M., Mertella, G., Samarati, P., Database Security, Addison-Wesley, ISBN 0201593750, (1994).

[4] Denning, D., Secure distributed data views: the Sea View formal security model, *Technical report A003 SRI International,* (1987).

[5] Sandhu, R., Jajodia, S., Honest databases that can keep secrest, *Proc. 14th NIST-NCSC National Computer Security Conference,* (1991).

[6] Smith, K., Winslett, M., Entity modelling in the MLS relational model, *Proc. VLDB Conf.*, (1992).