Asserting Beliefs in MLS Relational Models

Nenad A. Jukic and Susan V. Vrbsky
njukic@cs.ua.edu , vrbsky@cs.ua.edu
Department of Computer Science
The University of Alabama
Tuscaloosa, Alabama 35487-0290

Abstract: Multilevel relations, based on the current multilevel secure (MLS) relational data models, can present a user with information that is difficult to interpret and may display an inconsistent outlook about the views of other users. Such ambiguity is due to the lack of a comprehensive method for asserting and interpreting beliefs about lower level information. In this paper we identify different beliefs that can be held by higher level users about lower level information, and we introduce the new concept of a mirage tuple. We present a mechanism for asserting beliefs about all accessible tuples, including lower level tuples. This mechanism provides every user of an MLS database with an unambiguous interpretation of all viewable information and presents a consistent account of the views at all levels below the user's level.

1 Introduction

In many business, military, medical, and other environments, where multiple users share the same database, some of the users have restricted access to information from the database. For such databases, security subsystems based on database security models should be provided as a part of the DBMS [2]. The multilevel secure (MLS) relational data model is one such model. MLS databases can be used in many environments that exhibit hierarchical propagation of information. MLS models, such as Sea View [3], Jajodia-Sandhu [6, 7], LDV [5], Smith-Winslett [12, 13], and MLR [10], are based on the classification of the system elements, where classifications are expressed by security levels. This concept was first introduced by the Bell-LaPadula model [1]. In an MLS database users and information are classified into one of the *security levels* from the partially ordered set of security levels H. A security level $l_1 \in$ H dominates security level $l_2 \in$ H (stated as $l_1 \geq l_2$), if l_1 is higher than or on the same level as l_2. A subject (user) can read a certain object (data) only if the subject's clearance level dominates the object's security level. This property is called Bell-LaPadula *simple* property (read from the same level or down, no read-up). Another restriction on multilevel secure databases is the *star* property (write to the same level or up, no write-down) which states that a subject can write to an object if the subject's clearance level is dominated by the object's security level.

Different models define multilevel relations in different ways. In one of the more typical approaches adopted by many models [3, 6, 10], a multilevel relation is defined by the multilevel relation schema R(A_1, C_1, ..., A_n, C_n, TC) where each A_i is a data attribute defined over domain D_i, each A_i has security classification attribute C_i defined over the set H, and TC is the tuple-classification attribute also defined over the set H. There is a set of relation instances R_l(A_1, C_1, ... , A_n, C_n, TC), with one instance for each security level $l \in$ H. Each instance is a set of distinct tuples of the form (a_1, c_1,...,a_n, c_n, tc) where tc is the least upper bound of the classifications of the attributes in the tuple.

This paper presents a new approach to the interpretation of MLS databases. We believe that within an MLS database, every user should not only be able to interpret the information on their own level, but they should also be able to interpret the information on lower levels. Consequently, we also believe that a higher level user should have insight into how a user on a lower level interprets information on levels dominated by both users, since they can have totally different beliefs about the same information

The goal of this work is to provide the user with a mechanism for asserting beliefs about lower level information, which enables an unambiguous interpretation of all information accessible to the user and provides a consistent account of the views at all levels dominated by the user. None of the existing MLS relational data models has a mechanism for asserting beliefs in place, and none of them provides a user with the beliefs of the lower level users. As a result, they can allow information that is both visible yet ambiguous to certain users. This problem is solved by the approach presented in this paper.

2. Motivation

Although substantial progress has been made in making the MLS model complete, secure and semantically sound, there is still ambiguity present in the existing systems. Within any of the current MLS models, it is not possible for a user to interpret all the visible tuples. Some visible tuples are fully interpretable only when compared to other tuples, and some visible tuples simply can not be adequately interpreted. Consequently, it is even harder for a user to understand the lower level user's interpretation of the tuples. The following example illustrates the problem.

We assume the environment in which security level S dominates security level C, which dominates security level U (S ≥ C ≥ U). Consider the MLR model [10] MLS relation:

Starships(Vessel_Name, Objective, Destination)

shown in Figure 1. Attribute Vessel_Name is the primary key. Security levels are assigned to all the value attributes, and to the tuple as a whole (TC). A tuple is accessible to a user only if the user's security level *l* dominates TC (*l* ≥ TC).

In Figure 1, ti indicates the i-th tuple. For example t1 indicates the first tuple, and t5 indicates the fifth tuple. In this example S level users can see all tuples t1 - t8. C level users can see tuples t4 - t8, and U level users can see tuples t5 - t8. In particular, U and C level users can not access the information about the starship Avenger. The information about Voyager on the U level is not completely accurate. Instead, U level users are given a *cover story* [6, 11, 13] that hides the real objective of the Voyager's mission. A false objective "Training" is given instead of the real objective "Spying".

Figure 1: MLS relation Starships.

	Vessel_ Name (K)		Objective		Destination		TC
t1	Avenger	S	Shipping	S	Pluto	S	S
t2	Atlantis	U	Diplomacy	U	Vulcan	U	S
t3	Voyager	U	Spying	S	Mars	U	S
t4	Atlantis	U	Diplomacy	U	Vulcan	U	C
t5	Atlantis	U	Diplomacy	U	Vulcan	U	U
t6	Voyager	U	Training	U	Mars	U	U
t7	Falcon	U	Exploration	U	Venus	U	U
t8	Eagle	U	Patrolling	U	Degoba	U	U

"Read" access of the MLS relation in Figure 1 is secure at any level, since the Bell-LaPadula simple property is satisfied. However, there is a certain level of ambiguity associated with the interpretation of the information contained within the relation. To illustrate, suppose an S level user is aware of the fact that starship Falcon does not exist. In that case, the insertion of t7 by a U level user could have been caused by a number of reasons. For example, an innocent error by a U level user, a malicious attempt by a U level user in order to provoke a reaction from a higher level, or a deliberate intention of making U level users believe that the number of starships in the starfleet is higher than the actual number (perhaps to boost the morale of unclassified users). Also suppose that S level users do not need to have any belief about starship Eagle represented by t8.

Given this situation, the interpretation of the relation in Figure 1 is somewhat ambiguous. The interpretation from the S point of view is as follows. The interpretation of t1, t2, and t3 is straightforward. They correctly represent the objectives and destinations of starships Avenger, Atlantis, and Voyager. Tuples t4 and t5 are interpreted simply as lower level tuples. Only if compared to t2 can t4 and t5 be interpreted as a necessary duplication on the lower levels of the true information contained in t2 (later we show that such duplication, which would occur in any of the existing models, is actually not necessary, and how it can be avoided). Tuple t6 is also interpreted simply as a lower level tuple. Only if it is compared to t3 can t6 be interpreted as a cover story tuple on the lower level, that contains false information about the objective of the starship it describes. Tuples t7 and t8 are again interpreted simply as lower level tuples. Even though t7 does not correspond to any existing starship, an S level user can not (in any of the existing models) assert such a belief about t7. Therefore, by viewing this relation, such tuples can not be properly interpreted by S level users. Moreover, it is also ambiguous how a C level user should interpret most of this relation. And even when a C level user decides how to interpret this relation, most of that interpretation is not available to an S level user. In simpler terms, an S level user does not know "*what a C level user thinks*". The interpretation of tuples in Figure 1 is summarized by the Table 1.

Table 1: Interpretation of tuples from Figure 1.

tuple	U level interpretation	C level interpretation	S level interpretation
t1	-	-	true
t2	-	-	true
t3	-	-	true
t4	-	true	true*
t5	true	true *	true*
t6	true	?	cover story*
t7	true	?	?
t8	true	?	?

"-" indicates tuples that are not visible
"?" indicates tuples whose interpretation is ambiguous
(*interpretation possible only if compared to other tuples)

In this paper, we present an approach that provides an unambiguous interpretation of MLS relations. Our goal is to enable any user to interpret *unequivocally* every visible tuple, without having to compare it to any other tuple, and to provide a user with insight into the beliefs of all lower level users about every tuple visible to them.

3. Interpreting Accessible Information

In order to unambiguously interpret MLS databases, we first need to identify different beliefs that can be held about information stored in MLS databases. The Smith-Winslett model [12] is based on a *"believing in your own level, seeing the levels below"* semantics, which is also known as belief based semantics. A finer approach to this semantics, presented here, gives a substantially greater amount of knowledge to the users of the MLS relational database, and provides a mechanism for unambiguous interpretation. In order to expand the semantics of Smith-Winslett, we include the concepts of *true* and *false* to define the interpretation of a lower level tuple from a higher level.

In a belief based MLS relational database, where TC indicates tuple classification and l indicates a user's security clearance, the interpretation of the *same level* tuple (where TC = l) is straightforward. Every user with a security clearance l has the following belief about every tuple classified on the security level l (TC = l): "*Every attribute value of a tuple from this level, correctly represents a property of a real-world entity described by this tuple*".

The belief based MLS models (and all other existing MLS models) do not consider the user's interpretation of *all of the lower level tuples* (where TC < l). We maintain that every lower level tuple should and can be interpreted [8]. In our model every lower level tuple can be interpreted by every higher level user as one and only one of the following:

1) A lower level tuple can be interpreted by a higher level user as a ***true lower level tuple***. A user on the higher level has the following belief about such a tuple: "*Every attribute value of this lower level tuple correctly represents a property of a real-world entity described by this tuple*". In the existing models, due to the lack of a mechanism for asserting beliefs about lower level tuples, such a tuple is duplicated on the higher level.

2) A lower level tuple can be interpreted by a higher level user as a ***false lower level tuple.*** There are two kinds of false lower level tuples:

(a) If a lower level tuple (TC< l) represents the same entity as some other higher level tuple (TC = l), the lower level tuple is interpreted by a higher level user as a false tuple that represents a ***cover story*** [6, 11, 13] for the entity represented by the higher level tuple as well. Every user on the higher level l has the following belief about the cover story lower level tuple: "*Some attribute values of this lower level tuple incorrectly represent a property of a real-world entity described by this tuple*". Cover stories can exist in most of the existing models but they are hard to distinguish as such, unless compared by the user to a higher level tuple.

(b) A lower level tuple can be interpreted by a higher level user as false, even if it is not a cover story tuple. Such a false tuple does not correspond to any real world entity in the belief of the higher level subject, and hence it is called a ***mirage tuple***. Every user on the higher level has the following belief about the false mirage lower level tuple: "*This tuple represents an entity that does not exist in the real-world*". The concept of a mirage tuple is introduced here for the first time. None of the existing models offers such interpretation to the users, and yet they can all contain such tuples.

3) A lower level tuple can be interpreted by a higher level user as an ***irrelevant lower level tuple***. Every user on the higher level has the following belief about the irrelevant lower level tuple: "*This tuple represents either an existing or a nonexistent entity, and this tuple is not relevant for my understanding of the real-world*". Even though we assume that a higher level user can recognize any lower tuple as a true tuple or false (cover story or mirage) tuple, the higher level users do not have to do so with every lower level tuple. Some information may simply be extraneous for higher level subjects.

3.1 Richer Set of Security Labels

In order to interpret all of the lower level tuples we expand the set of security labels associated with the attributes and tuples in the MLS database. We propose a *richer set of labels* which utilizes the idea that security labels can contain a list of levels [4]. The purpose of this expanded set of security labels is to provide a user with a comprehensive interpretation of every part of every visible tuple. When the labels from the richer set are used, a user is able to deduce his own beliefs as well as beliefs of all lower level users about every visible tuple. Such interpretation is not possible with the regular (simple) set of security labels.

We assume totally ordered security levels, since the additional implementation details that would be necessary in order to accommodate partially ordered security levels would not contribute to the understanding of the new concepts introduced here. However, the approach we describe here, can be easily expanded to include partially-ordered security levels.

We can describe the richer set of labels (rsl) with the following regular expression:

$$\mathbf{rsl := (l)^+ \ (minus \ (l)^+ \ plus \ (l)^+)^* \ (minus \ (l)^+ \ )^?}$$

where * means 0 or more, + means 1 or more, ? means 0 or 1, **l** indicates a simple label (single security level), **minus** indicates "**-**" symbol, **plus** indicates "+" symbol, and every simple label **l** on the left is strictly dominated by every simple label **l** on the right.

The number of possible labels increases with the number of levels according to the following formula:

P_n - number of possible richer labels for n linearly (totally) ordered security levels

if n = 1 $\quad \mathbf{P_n} = 1$

if n > 1 $\quad \mathbf{P_n} = 3 * (\mathbf{P_{n-1}}) + 1$

therefore: $\quad \mathbf{P_n} = \dfrac{3^{\mathbf{n}} - 1}{2}$

In our previous example for three totally ordered security levels, c_1 = U, c_2 = C and c_3 = S, existing MLS models allow for only three security labels: U, C and S. For the same environment, our richer set of labels necessitates 13 labels shown in Figure 2.

Figure 2: Richer set of security labels:

Label No.	Label L	U view of the label	C view of the label	S view of the label
1	**U**	**U**	**U**	**U**
2	**U**S	**U**	**U**	**U**S
3	**U**-S	**U**	**U**	**U**-S
4	**U**C	**U**	**U**C	**U**C
5	**U**CS	**U**	**U**C	**U**CS
6	**U**C-S	**U**	**U**C	**U**C-S
7	**U**-C	**U**	**U**-C	**U**-C
8	**U**-CS	**U**	**U**-C	**U**-CS
9	**U**-C+S	**U**	**U**-C	**U**-C+S
10	**C**		**C**	**C**
11	**C**S		**C**	**C**S
12	**C**-S		**C**	**C**-S
13	**S**			**S**

The description and the meaning of the labels shown in Figure 2 are as follows. The first letter (denoted in boldface) indicates the security level at which the tuple or attribute value was inserted. Such a level is called the *primary level* of that tuple or attribute. Information is implicitly believed to be true by the primary level users. We define a function *pl(L)* that extracts the primary level of a richer security label L. For example pl(UC-S) = U, or pl(CS) = C.

While the letters that follow the first letter of the label indicate the security levels where the labeled information was not created, users on those levels do have a belief about labeled information. Such levels are called *secondary levels*. Letters that are not preceded by the "-" symbol indicate the secondary levels where the information is believed to be true. The letters following the "-" symbol indicate the secondary levels where the information is believed to be false. The letters that follow the "+" symbol indicate the levels where the information is believed to be true, and the same information is believed to be false at some lower secondary level. If security levels, higher than a primary level, are not represented in a label, subjects at those levels have no belief about the labeled information (the information is irrelevant on those levels). The levels below the primary level have no knowledge, and therefore no belief, about the labeled information.

In order to extract a belief held by an *l* level user about information labeled by a richer label L, we define a function *level's belief*: *lb(l, L)*. For example: lb(S, UCS) = -S indicates false information according to an S users' belief; lb(C, US) = ∅ indicates irrelevant information for C users; and lb(C, UC-S) = C indicates true information according to a C users' belief.

A user on a specific level can see only the components of the label that describe the levels dominated by the user's level. Hence, instead of seeing the entire label, a user is able to see only his level's view of a label. For example: if a tuple is labeled U-S, U is the primary level of that label and a U user sees the label as U. A C user sees the label as U as well. S is a secondary level of that label and an S user sees the label as U-S. If the primary level of a label is higher than the user's level, the labeled information is not visible to that user. That fact is illustrated by empty entries in Figure 2.

The labels provide the user with expanded information about the data at lower levels and the beliefs of users at lower levels. For example, label 2 in Figure 2 indicates unclassified information that is believed true on the secret level, while the classified level views it as irrelevant information. Label 6 indicates unclassified information that is believed true on the classified level and believed false on the secret level. Label 9 indicates unclassified information that is believed false on the classified level and believed true on the secret level.

4. Example

The example in Figure 3 illustrates how tuples from the MLS relation Starships extended by the richer set of labels are interpreted by users from different security levels. The interpretation of any tuple on any level (illustrated by Table 2) is straightforward and it does not require comparison with other tuples. If all the security labels within a tuple are true on a certain level, the tuple is interpreted as true on that level. If all the classification labels within a tuple are false on a certain level, the tuple is interpreted as a mirage tuple on that level. If, within a tuple, TC is labeled false and KC is labeled true on a certain level (while other attributes are labeled either true or false), the tuple is interpreted as a cover story on that level. And finally, if no attributes are labeled on a certain level, that tuple is interpreted as irrelevant on that level. In [9] we describe why no other combination of labels is possible.

Figure 3: MLS relation Starships with expanded labels

	Vessel_ Name (K)		Objective		Destination		TC
t1	Avenger	S	Shipping	S	Pluto	S	S
t2	Atlantis	UCS	Diplomacy	UCS	Vulcan	UCS	UCS
t3	Voyager	US	Spying	S	Mars	US	S
t4	Voyager	US	Training	U-S	Mars	US	U-S
t5	Falcon	U-S	Exploration	U-S	Venus	U-S	U-S
t6	Eagle	U	Patrolling	U	Degoba	U	U

Tuples that are true on more than one level do not have to be duplicated, as is the case in Figure 1. In addition, users are aware of lower level interpretations as well. For example, an S level user interprets t5 as a mirage tuple, and he also knows that a C level user interprets the same tuple as irrelevant and a U level user interprets the same tuple as true.

Table 2: Interpretation of tuples from Figure 3

tuple	U level interpretation	C level interpretation	S level interpretation
t1	-	-	true
t2*	true	true	true
t3	-	-	true
t4	true	irrelevant	cover story
t6	true	irrelevant	mirage
t5	true	irrelevant	irrelevant

"-" indicates tuples that are not visible

(*t2 did not have to be triplicated as in Figure 1)

5 Asserting Beliefs

A tuple is always believed to be true by the users on the primary level of the tuple when l = pl(TC), where l is the user's security level. However, a mechanism for asserting beliefs within a multilevel relation R is necessary in order to document the *beliefs of the higher level users* about *lower level tuples* ($\forall$t $\in$ R, where pl(t[TC]) $<$ l) as either true or false. We call the process of asserting beliefs about lower level tuples: the *verification process*. The verification process can be either *explicit* (user asserted beliefs) or *automatic* (indirectly asserted beliefs). The user may explicitly verify a tuple by using the VERIFY command. The automatic verification process is a consequence of update, delete, and explicit verification in certain cases.

The VERIFY command is used for the explicit verification of a lower level tuple by a higher level user and has the following form:

VERIFY (TRUE | FALSE) *R*
WHERE *P*

where R is an MLS relation, and P is a select condition.

Explicit verification occurs when a higher level user recognizes a mirage tuple (VERIFY FALSE) or a true lower level tuple (VERIFY TRUE). The following restriction is placed on explicit verification.

Explicit Verification Constraint:

$\forall$ t $\in$ R, VERIFY (TRUE | FALSE) t is accepted iff: pl(t[TC]) $<$ l and lb(l, t[TC]) = $\emptyset$. Otherwise the operation is rejected.

In other words, a user on a specific level, can not explicitly verify (as true or false) tuples that are already verified on the user's level, and can not verify the higher level tuples that are not visible to the user. Therefore, only non-verified lower level tuples that satisfy the WHERE clause will be explicitly verified by the VERIFY command. For every tuple that was explicitly verified by a user, all other lower level tuples that represent the same entity must be verified as well. The verification of those tuples is *automatic*, as a consequence of explicit verification.

When a user at level l recognizes a non-verified lower level tuple as a mirage tuple, all other lower level tuples that represent the same (non-existing) entity must be recognized mirage tuples as well since they represent the same entity. For each non-verified lower level tuple that satisfies the WHERE clause of the VERIFY FALSE command, all classification attributes are labeled as false on level l by this command. In addition to explicit verification, automatic FALSE verification at level l of all other non-verified lower level tuples which represent the same (non-existing) entity is also triggered.

When a user at level l recognizes a non-verified lower level tuple t_T as a true tuple, all other lower level tuples that represent the same entity are recognized as cover story tuples for that entity. For each tuple that is being explicitly verified by the VERIFY TRUE command, this command causes all the classification attributes to be labeled true at the user's level l. In addition to this explicit verification, automatic verification of all other non-verified lower level tuples, that represent the same entity as the newly explicitly verified tuple, is triggered. The TC of those tuples will be labeled as false at level l. The classifications of the attributes that have the same values as the corresponding attributes of t_T will be labeled as true at level *l*. The classification of those attributes that have different values from t_T will be labeled false.

Verify Example

Suppose an S user recognizes t2 from the following relation *R* as a true tuple.

	Vessel_ Name (K)	KC	Objective	Destination	TC
t1	Excelsior	UCS	Exploration U-C	Degoba UC	U-C
t2	Excelsior	UCS	Spying C	Degoba UC	C

The S user issues the following command:

VERIFY TRUE *R*
WHERE *Vessel_Name* = *"Excelsior"*
AND *Objective* = *"Spying"*

The relation *R* changes as follows:

	Vessel_ Name (K)	KC	Objective	Destination	TC
t1	Excelsior	UCS	Exploration U-CS	Degoba UCS	U-CS
t2	Excelsior	UCS	Spying CS	Degoba UCS	CS

All the classification attributes of t2 are labeled true on the S level, due to explicit verification. In t1 the Vessel_Name and Destination are labeled as true, and TC and Objective are labeled as false due to the automatic verification process that was triggered by the explicit verification of t2.

The update and delete operations must be modified in order to enable automatic verification. The complete formal definitions and details of the verification process, modified update and delete operations, as well as completely redefined relational algebra for the extended MLS model are given in [9].

6 Conclusion and Future Research

In this paper we have presented a mechanism for asserting beliefs and interpreting all lower level tuples by the higher level users. We have shown how the usage of this assertion and interpretation mechanism results in an MLS database that enables an unambiguous interpretation of all visible information and gives users access to the beliefs of lower level users, none of which is possible in any of the existing MLS models. We distinguished between the previously defined type of a false tuple: cover story, and a newly defined type of a false tuple: mirage tuple. We developed the expanded set of labels to represent the true or false beliefs at each level. In our future work, we will extend this model in order to include a concept of non-key related cover stories. We will also examine the possibility of enforcing a hierarchy of non-decreasing truth across totally ordered security levels.

REFERENCES

[1] Bell D.E, and LaPadula L. J. (1974), "Secure Computer Systems: Mathematical Foundations and Model", *Technical Report*, MITRE Corporation.

[2] Castano S., Fugini M., Martella G., and Samarati P. (1994), *Database Security*, Addison-Wesley

[3] Denning D. E. (1988), "The Sea View Security Model", *Proceedings: IEEE Symposium on Security and Privacy*, Oakland, Ca, April, 218-233.

[4] Gong L., and Qian X., (1995), "Enriching the Expressive Power of Security Labels", *IEEE Transactions on Knowledge and Data Engineering*, Vol. 7, No. 5., October, 839-841.

[5] Haig J.T., O'Brien J.T., and Thomasen D. J., (1991), "The LDV Secure Relational DBMS Model", *Database Security, IV: Status and Prospects*, Elsevier Science Publishers B. V. (North-Holland), IFIP, 265-278.

[6] Jajodia S., and Sandhu R. (1990), "Polyinstantiation Integrity in Multilevel Relations", *Proceedings: IEEE Symposium on Security and Privacy*, Oakland, Ca, May, 104-115.

[7] Jajodia S. and Sandhu R. (1991), "Toward a Multilevel Secure Relational Data Model", *Proceedings: ACM SIGMOD,* Denver, Colorado, May, ACM, New York, 50-59.

[8] Jukic N., Vrbsky S.V. (1996) "Subjects Interpretation of Objects on Lower Security Levels", *A supplement to the Proceedings: IEEE Symposium on Security and Privacy*, Oakland, Ca, May,

[9] Jukic N. (1997) "A Multilevel Secure Relational Model Based on the Semantics of All Visible Information", *Ph.D. Dissertation,* The University of Alabama, Department of Computer Science.

[10] Sandhu R., and Chen F., (1995), "The Semantics and Expressive Power of the MLR Data Model", *Proceedings: IEEE Symposium on Security and Privacy*, Oakland, Ca, May, 128-142.

[11] Sandhu R. S. and Jajodia S. (1992) "Polyinstantiation for cover stories", *Proceedings European Symposium on Research in Computer Security*, Toulouse, France, Springer-Verlag LNCS 648.

[12] Smith K., and Winslett M. (1992), "Entity Modeling in the MLS Relational Model", *Proceedings: 18th VLDB Conference*, Vancouver, British Columbia, Canada, August, 199-210.

[13] Winslett M., Smith K., and Qian X. (1994), "Formal Query Languages for Secure Relational Databases", *ACM Transactions on Database Systems*, Vol. 19, No 4, December, 626-662.