Bell and LaPadula Axioms: A "New" Paradigm for an "Old" Model

T. Y. Lin
Department of Mathematics and Computer Science
San Jose State University
San Jose, California 95192

Abstract

Ideally secure systems must be provable secure, so they are all defined by mathematical models. Most of current systems are based on the Bell and LaPadula Model (BLM), however, many usages are not logically sound. In this paper, a 'new' paradigm is proposed to reinterpret the BLM. BLM is treated as axioms to define the multilevel security, in the same spirit as Hilbert axioms to the Euclidean geometry. Absolutely no violations are tolerated. So many usual trusted subjects are **no longer admissible** in this 'new' BLM. Three layer architecture is proposed to accommodate such requirements.

1 Introduction

"...[T]he system must not only be secure, but must be demonstrably so ..." [Land81]. So a secure system should be defined by a sound mathematical model. However, in current practices, there are some 'flaws.'

1.1 Tolerance of Inconsistency

The logical system adopted by natural science and engineering has very low tolerance on inconsistency. Let us consider the following sentence:

$$S1 : If(x \neq x), then(anyconclusionistrue). \quad (1)$$

In the traditional logic system, this sentence is a valid statement. However, the conclusion is not necessary a true statement until one can established the condition is a true statement (modus ponens). In mathematical modeling, the underlying hypotheses is that the model axioms are true statements, and its general goal is to infer more true statements. If there is any inconsistency in the axioms of mathematical model, we can never be sure that any conclusion is valid. The choice of such a logical system is, of course, a philosophical issue; we could adopt other systems. However, if we do decide not to use the traditional logic, then we have to redevelop "mathematics" and "science" based on the new logical system(at least the portion that are used in our secure system). Obviously this is not feasible. So we should stick to the traditional logical system.

In this paper, we offer a 'new' paradigm for the well known BLM with mathematical precision, and three layer architecture to support all the essential capability of secure systems. We believe the axiomatic multilevel data model is a very healthy system.

1.2 Some 'Flaws'

Almost all secure systems appeal to Bell-LaPadula Model (BLM) for their notion of security. In many of these systems, trusted subject are used or abused [Tay84]. The trusted subjects include the downgrading operation, information flowing downward, which is inconsistent with the constraints of the *-property. Consequently the validity of the security policy of the system is rather obscure (see 1.1). Some current systems essentially include human judgements in their models. These judgements are based on the semantics of data. But, in the model level, semantics of data are not yet available. So the validity of their approach are questionable.

In multilevel data models, security labels of relational operations are often unspecified. Some default assumptions on the label of the join (in the same model) vary from the least upper bound (of the labels of individual factors) to the high water mark (the highest class that the relevant data are touched by the systems). Unfortunately, these two assumptions have rather different implications. The first one requires

*This work is supported by research grant MDA904-91-C-7048.

that all implementations should obey its specification, whereas the second model is up to the implementation to complete its model specification. High water mark model is well commented in [Denn76]. Here, we just pointed out that each high-water-mark-label that is not the least upper bound, then there is an (algebraic) aggregation or inference [Lin89a], [Lin90b], [Lin92f,g]. For the least upper bound case, we have shown that all relational operators carry information downward or horizontally [Lin91a], [Lin92f]. So relational operators are all trusted subjects; this is unacceptable.

1.3 Axiomatic Approach

To present the model with mathematical precision, we apply the axiomatic method. The notion of security is defined axiomatically. The usual two properties (simple security and *-property) of BLM are the axioms of security. We absolutely disallow any violation. Many usual trusted subjects are no longer admissible. Downgrading is not acceptable. Handling the trusted subjects is critical in multilevel relational model, because all relational operators carry information downward or horizontally (trusted subject?). We handle the problem by storage organization—called clustering. Data clustering is extremely important, so we keep it as part of the data model.

We would like to take this opportunity to thank the participants at this workshop for general support, especially to Lapadula for explicit support of this axiomatic method toward the security. This paper is one of our rigorous comprehensive study of multilevel world and multilevel data model.

2 Mathematical Models and Axiomatic Approach

Mathematical models have been utilized by scientists, engineers and mathematicians in two ways:

1. *Informal approach*: In this approach, mathematical model is used as a vehicle to capture the essential feature of phenomena or processes. The model is essentially a mathematical description of a piece of a real world (or perceived world). It is never meant to be a definition of that phenomena. Any new phenomena will trigger a change in the specification of the model. Newton's classical mechanic (CM) is a mathematical model for forces. However, CM was updated when relativity was discovered. Mathematical models in natural science are of this approach. Models are mathematical description of a slice of universe. The description are never meant to be complete. They change with new discoveries.

2. *Formal approach*: In this approach, a mathematical model is used as definition of a system. For example, Hilbert axioms are used as a definition of Euclidean geometry. One is not allowed to add new assumptions while he is trying to deduce a theorem in the Euclidean geometry. Properties of Euclidean geometry have to be derived from Hilbert axioms.

In the first phase of computer security research, Bell LaPadula Model (BLM) is used as a guide for designers to capture some important features of secure systems. So BLM is used as an informal model. At that stage, the notion of computer security is not well defined yet. The notion is in each researcher's intuitive mind, BLM is merely served as one of their convenient tools.

However, the role of BLM changes, if we want to implement a system based on BLM. Then BLM becomes the definition of the system. As the computer security move from paper research to prototype/commercial system building, the BLM moves from the first phase (informal approach) to the second phase (formal approach). In the second phase, BLM defines the secure system. The situation is just like Euclidean geometry; Hilbert axioms define the geometry. So our attitude toward BLM has to be much more serious and rigorous. We propose an axiomatic Bell and LaPadula Model in this paper.

3 Axiomatic Bell LaPadula Model

In BLM, there are two parts. One is the two properties (simple security and *-property), the other is the trusted subjects:

1. Two Properties:
   (a) *Simple security property*: A subject can read only data whose access class is dominated by the subject's access class.
   (b) **-property*: A subject can write only data whose access class dominates the subject's access class.

2. (2) Trusted subjects: "A trusted subject, to define them, is something that is allowed to violate

the *-property provided it is proved that the security compromise that the property is designed to guard against does not happen" [Taylor84].

In an informal approach, a trusted subject can be anything which violates *-property but does not violate designer's intuitive security. In this approach the security is not well defined. BLM does not capture the intrinsic notion of security, it is an intuitive idea in the designer's mind. We decide not to accept this practice.

In a formal approach, BLM defines the security. The security is defined by the two properties. Thus trusted subject can be anything which obeys the two properties and does not violate *-property. It seems contradictory:

- What are the operations that do and do not violates *-property? or
- What are the operations that apparently do and in reality do not violates *-property?

Filtering! A manager goes to the secret safe to pick up (read) confidential documents and gives (writes) to a confidential person.

Axiomatic Bell and LaPadula Model

Let SC be a partial ordered set of security classes (the partial ordered is denoted by $\leq$). Let SO be the set of all subjects and objects. Let TS be a special subset of subjects, called trusted subjects. The set of all objects OB is a partial ordered set with partial order—contains. If A contains B, then A is called a container of B. The partial ordering, contains, is essentially the set theoretical inclusion.

Axiom 1 *Any object or subject is assigned a security class (label).*

That is, there is a map

$$[] : SO \longrightarrow SC \qquad (2)$$

Moreover, [] is monotonic on OB, i.e., [B] $\leq$ [A] if A contains B.

Axiom 2 *Simple security property*

Axiom 3 **-property*

Axiom 4 *Filtering (Trusted subjects). Let $L \leq M \leq H$ be three security classes in SC. A trusted subject of class H is allowed to read an object of class L from a "container" of class M and write to a container of any class lying between L and M inclusive.*

In the axiomatic Bell LaPadula Model, the trusted subjects only allowed to do filtering, the usual downgrading is not permissible.

In the sequel, we introduce 'downgrading'. Effectively, it is equivalent to the usual downgrading, however in different setting. So there is no real loss in Axiomatic BLM. Roughly, the usual downgrading is a subject (process) which reads a named high (sensitive) object and writes to a low object with the same name. In our formulation, downgrading is not a primitive, it is an operation that is definable by read and write.

4 Tuple Labeling—Incomplete Modeling

The so called tuple level classification models are often incomplete in their model specification. These models often do not specify the rules of security classification on the relational operations.

4.1 Label of a Joined Tuple?

The follow example is taken from Hinke's work. The schema is Hinke's data are manufactured by us. We will be responsible for the interpretation or misinterpretation.

In Hinke's paper, there are no rules on how the tuples in the joined relation NEW should be classified.

NEW = VISITOR-LOG * MEETING * CONTRACTS.

This tuple is called second path by Hinke. The two elements which are in bold print is a tuple in the CONTRACTORS:

This tuple in CONTRACTOR is called direct path by Hinke.

In his paper, there is no explicit spec on what should be the label for a joined tuple, so the TupleClass of second path is ?, unknown.

1. If ? equal S, then there is no inference.
2. If ? less than S, then there is inference.

Implicitly Hinke assumes that the label of a joined tuple is the least upper bound of the labels of its factors. So ?= U, and hence, by (2), there is an inference from second path to direct path.

4.2 High Water Mark Policy and Inferences

As we have pointed out earlier that there are two possible default assumptions. One is the High-Water-Mark policy, another is the lattice model. These two

Table 1: Visitor log relation

Visitor-name	Visitor-company	Contact	Tuple-Class
Peterson	Hughes	John	U

Table 2: Meetings relation

Room	Time	Project-number	Contact	Tuple-Class
MH123	13:00	SP92745	John	U

assumptions appear to be similar, but actually they are fundamentally different. In both policies, the security labels of primitive data are assigned by security officer. In High-Water-Mark policy, the label of derived data are assigned by the implementation. In lattice model, the label of derived data are assigned by algebraic rules. The implementation has to confirm the model requirements. Lattice model will be discuss in Section 5.

Let us comment about a consequence of "bad" of High-Water-Mark policy. In High-Water-Mark policy, the label of the derived data is the highest labels the system touched. In such High-Water-Mark, the model may have many inference channels.

Let A, B, C, D, E be relations. Let B be stored in file B-file, C be stored in C-file, and D , E be stored in E-file. B-file and C-file are Unclassified files, and E-file is a Secret file.

Suppose we have

$$A = f(B, C, E) \tag{3}$$

where f, for example, is the union of three relations. Suppose the query optimizer has to touch Secret file, because E is there. Then the system (High-Water-Mark) will assign the derived data A the following label:

$$[A] = S. \tag{4}$$

Instead of using this system, an U-user can take the available data B, C and E. to another machine which is unclassified. Use this machine to perform the relational algebraic operation f, we still can get A. Then the U-user has an inference, because he can infer from U-data which are B, C, and E , an S-data which is A.

This seems silly, however, it is a consequence of High-Water-Mark, it could occur in many systems. Some models do not require (except SeaView) that data of different levels should be stored in different physical files. So E-file is permissible in these system (but not SeaView and our model).

The real story of this example is that the inference is not real. The real problem is that the data A got over-classified by High-Water-Mark policy. The inference will disappear when the security officer down grades the data A. Such systems are obviously unhealthy.

We will defer the discussion of lattice model to next section. Our conclusions here are as follows:

1. the specification of a data model should be complete; unfortunately many models are not, and
2. let the implementation follow the specification of a model but do not let the implementation define the model.

We urge the researchers to be precise and rigorous. All the tuple level labeling systems have not given a complete description about their systems. If they do, they will find that they have to do element labeling too. For example, the tuple-level-labeling model has to specify the policy of labeling the new relation, one-column-relation, which is a projection of whole relation; its tuples are elements.

5 Semantics and Structures of Security Classes

There are many misconception about the meaning of a security label. In this section, we discuss the meaning and structure of security labels. In particular, the lattice model.

5.1 Semantics of Security Classes

A data model is a mathematical representation of a slice of the real world. Each mathematical notion in the data model represents certain portion of the real world. A primitive data is a representation of a primitive fact. A complex data represents a complex fact. In a secure world, all facts, simple or complex, are classified. So in a secure data model, all data,

Table 3: Contracts relation

Project-number	Classification	Tuple-class
SP92745	Secret	U

Table 4: Contractors relation

Project-number	Company	Tuple-class
SP92745	Hughes	S

simple or complex, are classified according to their real meaning. We would like to stress that **classification of data is a reflection of the classification of facts of the real world**.

In the relational model, the element is the primitive data. We will examine the meaning of its label. In the following relation, the **first 50,000** represents one aspect of the entity Mr. Smith. So if we do label the element 50,000, we are **not** labeling 50,000 *per se*, it is Smith's 50,000. In general a tuple represents an entity in the real world, and an element represents a property of the entity. The label of the element should be so interpreted.

Example 5.1.

There are two labels for 50,000 in this relation.This does not mean that the security labeling is inconsistent. It merely means that the raw data 50,000 was used twice, first is to represent Smith's salary, sc, second is Jone's salary. Element labeling is never meant to be the labeling of raw data. G. Smith suggested that there are several meanings to the element labeling; we disagree. In database, one element in a tuple has only one meaning in the real world, so the **security label of an element represents the labeling of the unique real world meaning**.

5.2 The Structure of Security Classes

A relation scheme is defined by attribute names. The security class of relation scheme or its name can then be derived from the security classes of attribute names. A relation instance can be generated from elements (see [Lin92f] on set representation). The security classes of intensional and extensional objects can be derived from the label of its primitive data.

5.2.1 Security Classes of Intensional Objects

In this subsection we will discuss the security classes of intensional objects. To simplify our exposition, we use Name(X) to denote the name of attribute X. Attributes are usually the name of the domain in consideration, to avoid confusion, we use Name(attribute) to emphasize that we are talking about names, and Domain(attribute) about the data.

- The security classes of primitive data : The primitive data in the intensional world is the Name(attribute). These security class has to be assigned by security officer, and dominated by all the data in its active Domain [Maie83].
- The security classes of derived data: Intensional derived data are view schema, which includes relation schema.

A relation schema which is an organized collection of Name(attribute)s. So [Name(relation)] should be dominated by [Name(attribute)]'s ([x] denotes the security class of x).

Same comments can be applied to views (View is an "virtual" relation). However, a view is normally not defined by its attributes, but by a query statement. Attributes are the consequence (or the output) of the query. We treat the "output attributes" as the canonical definition of the view (it will be treated as schema of view). We will use this canonical definition of view to derive the security label of a view. The query statements or the schema are the "alias" of the view. So their labels are all equal.

View defines a collection of data. In general, the bigger view (the collection) the lower the security class of its name. This agrees with our intuition of secrecy semantics. The name of a collection will be used by every member of the collection, so the [Name(collection)] should be dominated by [Name(member)]s. The security algebra of intensional objects is an "upside down" poset.

There are many query expressions. By BLM's requirements, they all have to have security labels. If there are no rules to "automate" such labeling, then we need a security officer to label every query written by users—an impossible task. Besides, there are relationships among these expressions. Any careless

Table 5: New relation

Visitor name	Visitor company	Contact	Room	Time	Project number	Classification	Tuple class
Perterson	Hughes	John	MH123	13:00	SP92745	Secret	?

Table 6: New relation

Visitor-company	Project-number	Tuple-class
Hughes	SP92745	S

assignment would result in inferences. Some systematic way of assigning the security classes is necessary.

We propose the following scheme for classifying all the derived intensional objects.

$$[Name(relation)] = g.l.b\{[Name(A)] : A \textit{ are the defining attributes of the relation scheme}\}$$

Or more generally,

$$[Name(view)] = g.l.b\{[Name(A)] : A \textit{ are the defining attributes of the view scheme}\}$$

Example 5.2.

Let us consider the following query against the relation RECORD in the example above. For convenience, we will name the view defined by the following query

```
URECORD:
Q1:  SELECT RECORD.NAME,
  RECORD.OCCUPATION
FROM RECORD
WHERE RECORD.SALARY ≤ 55,000
AND RECORD.CL2 ≤ U
```

The "output attributes" are RECORD.NAME, RECORD.OCCUPATION, so
[URECORD] = g.l.b. { [RECORD.NAME], [RECORD.OCCUPATION] }

Here we should point out that the security label of a view is independent of its defining syntax, but depended on its semantics. Let us consider the following query Q2, which is the same as Q1, except there are some added "nonsense"

```
Q2:  SELECT RECORD.NAME,
  RECORD.OCCUPATION
FROM RECORD
WHERE RECORD.SALARY ≤ 55,000
AND RECORD.CL2 ≤ U
AND PETERSON.SALARY = 65,000
AND PETERSON.OCCUPATION =
  NUCLEAR ENGINEER
```

The last two conditions use sensitive attributes, so syntactically Q2 is, however, the high attributes have no real contributions to the semantics, so Q2 and Q1 should receive the same label. Labeling is labeling the real world entity, not the "character string" of the statement. Intuitively, our label of a view is the lowest labels among all possible syntactically different but semantically equivalent expressions.

5.2.2 Security Classes of Extensional Objects

In this section, we discuss the security classes of extensional objects.

1. The security label of primitive extensional object: The primitive data is the elements, and their security classes are assigned by security officer.

2. The security class of derived data: View instances are the derived data. A tuple is a view with single row. Relation is a view with physical meaning. All derived data are sets of elements (see next section).

The security class of a view is dominated by the l.u.b of the security classes of its elements. Given a database with n primitive data, there are potentially 2**n view instances, they all have to be classified. This is an exponential problem [Lin90a]. Some systematic way of assigning security is needed. In many existing models, this was totally disregarded. Their meaning of security is really questionable; see the critique below.

The simplest suggestion is to use Denning's lattice model. In a lattice model, the security class of a view instance is the least upper bound of the labels of all the

Table 7: Relation 'B'

Name	Salary	Telephone	Occupation	Tuple-class
Jones	50,000	123-654-0987	Accounting	U
Johnson	75,000	231-544-6890	Manager	U

Table 8: Relation 'C'

Name	Salary	Telephone	Occupation	Tuple-class
Smith	50,000	123-456-7890	Engineer	U
Peterson	65,000	321-654-0987	Nuclear Engineer	S
Thompson	50,000	123-654-0987	Accounting	U
Tamale	60,000	231-545-7890	Security Expert	S

elements. Under this labeling scheme, the extensional security algebra is a lattice.

5.2.3 Relational Algebra and Semantically Admissible Operations

Cartesian product, join, union, intersection, difference and divideby can all be carried to multilevel data model. In last two subsections, we gave the syntactical rules in assigning labels for derived data (the data derived from relational operations). We should stress again, the labels of derived data (or complex data) is intended for the corresponding real world objects. Although the labeling rules will work for any operation, a label is meaningful only if there are real world entity there. Let us consider the following sequence of operations. Project the relation RECORD to each column, and then take the Cartesian product. The resulting relation will have the same security label as RECORD, however, there is no real object corresponding to the relation. Such sequence of operations is not a semantically admissible operations; it should be avoid.

6 Bell and LaPadula Data Model (BLDM)—The Data Model As A Bell and LaPadula Model

6.1 Security Objects

In Bell Lapadula Model (BLM), every object or subject is assigned a security class. Now if we apply BLM to database systems, then BLM requires that every object processed by database systems should have security classification. What are the objects processed by databases?

1. Intentional Objects: They are objects in Data Dictionary, such as, names of attributes, relation schema, query statements, and constraints.
2. Extensional Objects: They are the elements, tuples, relations, view instances, and relational algebraic expressions.

6.2 Data Clustering and Data Flows

In [Lin91a], [Lin92f], we have shown that all relational operator carries information downward or horizontally. Without proper storage structure all relational operators are trusted subjects; this is unacceptable. In order to satisfy the BLM axioms, we need the notion of data clustering. Please referred to [Lin92b] for more general discussion.

Definition 1 (Data clustering) *Primitive data of different security labels are stored in different physical volumes.*

If the relational operator is applied to View A and produces View B, it may appear as if the data is flowing downward. In fact, both views are drawing their respective data from their own clusters. So there is no actual data movement. Therefore with data clustering, all the relational operations satisfy the BLM axiom.

6.3 The Axioms of Bell and LaPadula Model

Revised Axiom 1 *Subjects are classified: Users and the processes initiated by them are the subjects.*

Objects are classified: All intensional and extensional objects are classified. All primitive and complex objects are classified.

Remark: Some multilevel data models only assign security classes to their primitive data (elements, tuples, or etc), and there are no further specification on

Table 9: Relation 'D'

Name	Salary	Telephone	Occupation	Tuple-class
Tamale	60,000	231-545-7890	Security Expert	U
Johnson	75,000	231-544-6890	Manager	U
Phillips	110,000	231-346-7891	Top Agent	S
Barn	200,000	231-346-7891	Top Agent	S

Table 10: Relation 'E'

Name	Salary	Telephone	Occupation	Tuple-class
Smith	50,000	123-456-7890	Engineer	U
Pace	65,000	321-654-0987	Engineer	U
Thompson	50,000	123-654-0987	Engineer	U

how the complex objects (tuples, relations, and views) are classified. These models are incomplete BLMs.

Revised Axiom 2 *Simple security property*

Revised Axiom 3 **-property*

Revised Axiom 4 *not needed*

Axiom 5 *Data clustering. Each primitive data belongs to one and only one cluster, and each cluster has to be stored physically together. Different classes of data are stored in different volumes.*

7 Three Layer Architecture

In our formulation, the traditional trusted subjects are disallowed. In a real system, we definitely need to do some downgrading, and so on. In [Lin90a], we include the human (SSO) into the model to execute the trusted subjects; almost all current secure systems implicitly assume some humans are in the models. Such model is mathematically sound, however, realistically, we never really know what the security means. The meaning of security is reshaped by each SSO's decisions. Remember that all SSO's decisions are based on the semantics of data, which are not available in mathematical model. In fact this is the fundamental reason that mathematical model can not characterize the trusted subjects.

Current approach is better, within the secure system, trusted subjects are completely disallowed. However, this does not mean that we close our eyes on the trusted subjects in reality. Insertion and deletion are always needed in any system, They are performed by authorized users, thus we can think of **downgrading as two operations, deleting the high data and then reinserting into the system as low data.** In practice, we certainly should have software to assist authorized users to perform these two operations. We may have software to help him to keep a copy of the deleted data in his own workspace, also to assist him to insert the data from his own workspace into the secure system. These software systems are his tools, but not part of the secure system. Using these tools authorized users (trusted persons) can downgrading (deleting and inserting) the data in the system. We could call this particular software as downgrading operator.

Based on such view, we propose that every system must have some specially selected persons (trusted subjects) who have certain special software tools to perform the "trusted operations." These software tool box are not part of the data model, but are part of global picture of secure systems. So we have a three layer architecture.

1. Reference Monitor which enforces BLM axiom.
2. The Database layer which is relied on the monitor to enforce the security. The data is clustered so all relational operations can be perform without any additional security components.
3. Software Tool Box are available for different trust-level-users. Some trusted subjects (users) can use highly reliable software tools (trusted software) to perform the trusted operations (trusted process) such as downgrading.

There are several advantages to this approach:

1. The whole system is clearly modularized.
2. Each trusted operation by trusted subjects is forced to be examined and executed by trusted

Table 11: Relation illustrating labeling of elements

Name (CL1)	Salary (CL2	Telephone (CL3)	Occupation (CL4)
Smith (S)	50,000 (S)	123-456-7890 (S)	Physicist (S)
Peterson (S)	65,000 (S)	321-654-0987 (S)	Nuclear Engineer (S)
Jones (U)	50,000 (U)	123-654-0987 (U)	Accounting (U)

persons. So there are no unexpected compound effects of trusted operations.

3. Certified by component is workable.

Security is a very complex notion, there are conflicting requirements (the best examples are the traditional trusted subjects). The whole system cannot be expressed by one mathematical system. Mathematics cannot tolerate inconsistency (see Section 1.1), so common approaches are unacceptable. However, we do want mathematics to assure us the consistency on each piece between "conflicts." We believe we have built such a mathematically sound secure data model and architecture.

8 Conclusion

A critical examination of current secure systems, one will find that they are not really "secure" in the sense they claim. We hope this report will generate a serious effort in clear and rigorous development of secure systems.

References

[Date81,86,90] C. Date, *Introduction to Database Management Systems*, Addison-Wesley, 1981,86,90.

[Denn76] D. E. Denning. "A Lattice Model of Secure Information Flow," *Communications of the ACM*, Vol. 19, No. 5, May 1976, pp. 236–243.

[Denn86a] D. E. Denning. The Inference Problem in Multilevel Database systems, In the *Proceeding of the National Computer Security Center Invitational Workshop on Database Management Security*, June 1986.

[Denn86b] D.E. Denning, S.G. Akl, M. Heckman. T.F. Lunt, M. Morgenstern, P.G. Neumann, and R.R. Schell. View for Multilevel Database Security, *Proc. IEEE Symposium on Security and Privacy*, 1986.

[Denn87] D.E. Denning, T.F. Lunt, R.R. Schell, M. Heckman, and W.R. Shockley, "A Multilevel Relational Data Model," *Proceedings of 1987 IEEE Symposium on Security and Privacy*, 1987.

[Denn87b] D.E. Denning, T.F. Lunt, R.R. Schell, M. Heckman and W.R. Shockley, "The SeaView Formal Security Policy Model," Computer Science Laboratory, SRI International, July, 1987.

[Denn88a] D.E. Denning, T.F. Lunt, R.R. Schell, W.R. Shockley, M. Heckman, The SeaView Security Model, *Proc. 1988 IEEE Symposium on Security and Privacy*, 1988.

[Denn88b] D.E. Denning, T.F. Lunt, P.G. Neumann, R.R. Schell, M.Heckman and W.R. Shockley, "Security Policy and Policy Interpretation for a Class A1 Multilevel Secure Relational Database System," Computer Science Laboratory, SRI International, Aug. 1988.

[Frost86] Richard Frost, *Introduction to Knowledge Base Systems*, Macmillan, 1986.

[GaMiNi84] H. Gallaire, J. Mikner, and J. Nicolas. Logic and Databases: a deductive approach. *ACM Computing Surveys* 16(2), 1984, 153–185.

[Hsiao71] D.K. Hsiao, D. K., and F. Harary, A Formal System for Information Retrieval from Files, *Communications of the ACM* Vol. 13, No 2, (February 1970).

[Jajo90a] S. Jajodia and R. Sandhu, "Polyinstantiation Integrity in Multilevel Relations," *IEEE Symposium on Security and Privacy*, Oakland, California, 1990.

[Jajo90b] S. Jajodia and R. Sandhu, "Polyinstantiation Integrity in Multilevel Relations Revisited," *Fourth IFIP11.3 Database Security Workshop*, Hailfax, England, Sept 18-21, 1990.

[Jajo90c] S. Jajodia and R. Sandhu, "Update Semantics for Multilevel Relations," *Sixth Computer Security Applications Conferences*, Dec 3-7, 1990.

[Land82] Carl Landwehr, Formal Model fro Computer Security, *ACM Computing Surveys*, September, 1981, pp.247-278.

[Lin92a] T.Y. Lin. "Aggregation and Fuzzy Sets," *Fifth Rome Laboratory Database Security Workshop*, October 4–7, 1992.

[Lin92b] T. Y. Lin. Concurrent Automata, Database Machine and Security, New Security Paradigms Workshop, September 22-24, 1992.

[Lin92c] T. Y. Lin, "Attribute Based Data Model and Polyinstantiation," Sept 7-11, Madrid, Spain, IFIP Congress, 1992.

[Lin92d] T. Y. Lin, "Rough Patterns in Data—Rough Sets and Intrusion Detection Systems," *First invitational Workshop on Rough Sets*, Poland, University of Warsaw, September 2-4, 1992.

[Lin92e] T. Y. Lin, "Rough Sets in AI as Clustering in Databases," *First Invitational Workshop on Rough Sets*, Poland, University of Warsaw, September 2-4, 1992. (Coauthor David Hsaio)

[Lin92f] T.Y. Lin, "Inference Secure Multilevel Databases," *Proceeding of IFIP WG11.3 Workshop on Database Security*, August 18-22, 1992.

[Lin92g] T. Y. Lin, The World Model and Polyinstantiation, TR December 1992

[Lin92h] T. Y. Lin, Inferences And Multlelvel Databases, TR December 1992

[Lin91a] T.Y. Lin, "'Inference' Free Multilevel Database System," *Proceeding of the Fourth RADC Database Security Workshop*, Providence, Little Compton, RI, April, 1991.

[Lin91b] T.Y. Lin, Entropy, "Ordering and Aggregation," *Proceeding of the Fourth RADC Database Security Workshop*, Little Compton, RI, April, 1991.

[Lin91c] T.Y. Lin, "Message and Noncommutative Aggregation," *Third RADC Workshop on Database Security*, January 1991. (Coauthor: Al-Eifan. Final Revision of Message and Inference Aggregation, Third RADC Workshop on Database Security, June 1990.)

[Lin91d] T.Y. Lin, "Multilevel Database and Universal Security Algebra," *Third RADC Workshop on Database Security*, January 1991. (Final Revision of Aggregation and Guidelines for SSO, Third RADC Workshop on Database Security, June 1990.)

[Lin90a] T.Y. Lin, "Probabilistic Measure on Aggregation," *Proceeding of 6th Annual Computer Security Application Conference*, December, 1990.

[Lin90b] T.Y. Lin, "Multilevel Database and Aggregated Security Algebra," *Database Security, IV: Status and Prospects*, edited by S. Jajodia and C. E. Landwehr, North Holland, 1991 (Final Revision of Database, Aggregation and Security Algebra (Lattice), IFIP WG11.3 Workshop on Database Security, Sept. 1990.

[Lin90c] T.Y. Lin, "Rough Sets, Neighborhood Systems and Approximation," *Fifth International Symposium on Methodologies of Intelligent Systems, Selected Papers*, Oct. 1990 (Coauthors: Q.Liu and K. J. Huang).

[Lin90d] T.Y. Lin, "A Model of Topological Reasoning Expert System with Application to an Expert System for Computer-Aided Diagnosis and Treatment in Acupuncture and Moxibustion," *'International Symposium on Expert Systems and Neural Network Theory and Application*, August 1990. (Coauthors: Qing Liu and K.J. Huang)

[Lin89a] T.Y. Lin, "Commutative Security Algebra and Aggregation, Research Direction in Database Security, II," *Proceedings of the Second RADC Workshop on Database Security*, December 22, 1989.

[Lin89b] T.Y. Lin, "Some Remarks On Inference Controller, Research Direction in Database Security, II," *Proceedings of the Second RADC Workshop on Database Security*, December 22, 1989.

[Lin89c] T.Y. Lin,16. Security Algebra and Formal Models, *Proceedings of IFIP WG11.3 Workshop on Database Security*, September 5–7, 1989 (with L. Kerschberg and R. Trueblood, and final revison appear at *Database Security, III: Status and Prospects*, edited by D. Spooner and C. E. Landwehr, North Holland, 1990.

[Lin89d] T.Y. Lin, A Generalized Information Flow Model and Role of System Security Officer, *Database Security, II: Status and Prospects*, edited by C. E. Landwehr, North Holland, 1989.

[Lunt91] T.F.Lunt "A Poyinstantiaton, An Inevitable Part of Multilevel World," Franconia, June, *Fourth Foundation Workshop for Computer Security*, June, 1991.

[Lunt90a] T. F. Lunt and Donovan Hsieh, "Update Semantics for a Multilevel Relational Database System," *Fourth IFIP11.3 Database Security Workshop*, Halifax, England, September 18-21, 1990.

[Lunt90b] T. F. Lunt and Donovan Hsieh, "SeaView Secure Database System, A Progress Report," *Proc. of European Symposium on Research on Computer Security*, France, October, 1990.

[Lunt89] T. F. Lunt, "The True Meaning of Polyinstantiation," *Proceedings of The Third RADC Database Security Workshop*, June 5-7, 1990, pp.26–36.

[Mai83] D. Maier, *The Theory of Relational Databases*, Computer Science Press, 1983.

[Smith92] K. Smith and M. Winslett, "The Importance of Declarative Semantics for MLS Relational Databases," RADC Workshop 1992.

[Stan77] D. Stanat and D. McAllister, *Discrete Mathematics in Computer Science*, Prentice Hall, Englewood Cliffs, N.J., 1972.

[Qian92] Xiolei Qian, "Integrity, Secrecy, and Inference Channels," RADC Workshop, 1992.

[TsLo82] D. C. Tsichritzis and F. H. Lochovsky, *Data Models*, Prentice-Hall, Englewood Cliffs, N.J., 1982.

[WoCh71] Wong, E. and Chiang, T. C., "Canonical Structure in Attribute Based File Organization," *Communications of the ACM* Vol. 14, No. 9, September 1971.

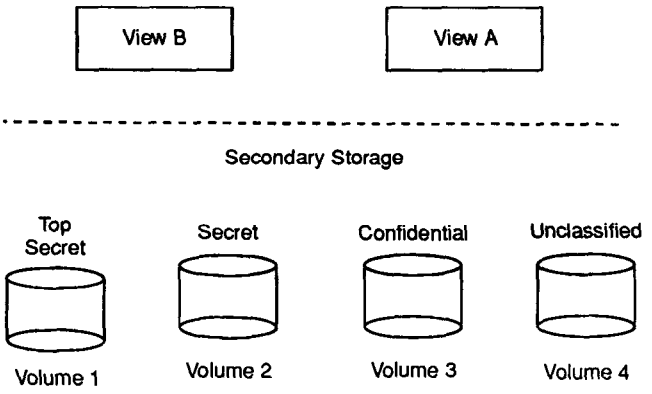

Figure 1: An example of data clustering

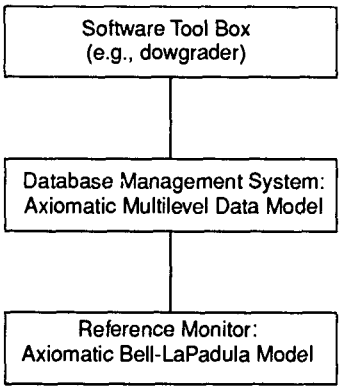

Figure 2: Three layer architecture