A NEW DECOMPOSITION TECHNIQUE FOR DECOMPOSEING A MULTILEVEL SECURE RELATION INTO SINGLE-LEVEL RELATIONS

BY

MAHER ABDIN

Bachelor of science
Oklahoma State University
Stillwater, Oklahoma
1991

Submitted to the Faculty of the
Graduate College of the
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
December, 1998

A NEW DECOMPOSITION TECHNIQUE FOR DECOMPOSEING A MULTILEVEL SECURE RELATION INTO SINGLE-LEVEL RELATIONS

Thesis Approved:

H. Lu
Thesis Adviser

R. E. Hen

Wayne B. Powell
Dean of the Graduate College

TABLE OF CONTENTS

Chapter Page

Chapter Page

LIST OF TABLES

LIST OF FIGURES

Chapter I

Introduction

Database security is a major concern for the database designer, the database should be protected against improper disclosure of sensitive information and unauthorized users. Mandatory security and disrectionary security are the two categories of database security. Mandatory security restricts access to classified information to cleared personnel, and remains static during database operations; the database that supports mandatory security is called multilevel secure database. On the other hand, disrectionary security controls access to data on the basis of identity of users, type of access, and the specific object being accessed.

Database is a vital commodity in both government and industry. The parallel evolution of multilevel security (MLS) and database management systems (DBMS) has revolutionized the way information is stored, protected, and accessed. Starting in the past decade, a major research effort was undertaken with the goal of merging and integrating multilevel security with database technology. This new area of research

generated much progress in the following years in developing mechanisms to ensure greater security in these systems.

The multilevel secure relational database model introduces the concepts of security clearance levels for both data and users. Data stored in the database are tagged with different security classifications level based upon the sensitivity of data. Security clearance levels are also issued to database users, and users may not access data unless their security level exceed the security level associated with the data.

Multilevel secure (MLS) relations exist only at the logical level, in reality, MLS relations are decomposed into a collection of single-level base relations which are then physically stored in the database. The MLS relations are reconstructed from these base relations on user demand. There are several advantages of decomposing MLS relation into a collection of single-level base relations. The primary advantage is to enforce mandatory controls with respect to single-level base relations. The Trusted Computing Base (TCB) is responsible for enforcing mandatory controls with respect to single-level base relation , TCB is a small part of the operating system that must always be invoked, and must be shown to perform only its intended functions.

There have been several efforts to build a multilevel secure (MLS) relational database models. A major issues is how to decompose a multilevel secure relation into single level secure relation that will be stored physically in the database. The Proposals have ranged from decomposing a MLS relation based on attribute classification [Lunt, 90], decomposing a MLS relation based on tuples classification [Jajo,91], to decomposing a MLS relation based on views and fragments [Pern, 91].

SeaView model [Lunt, 90] decomposes a MLS relation into a collection of single level secure base relations using horizontal and vertical fragmentation. The reconstruction of MLS relation requires a repeated expensive join operations. Novel [Jajo, 91a] decomposes MLS relation using horizontal fragmentation into single level relations. The base relations are large which requires large I/O that reduce the performance, Furthermore, Novel model, has a large storage overhead during updating operations. A new decomposition technique is required that minimize number of join operations and create a base relation that are relatively small compares to Novel model.

The goal of this research is to reconcile and unify the differences among several decomposition techniques, and to develop a new decomposition technique to decompose a MLS relation into single level secure relations.

The thesis is organized as follows:

* Chapter 2 presents a review of related research which has influence the thesis research, this chapter discusses several security models in detail and their advantages and disadvantages, it also discusses integrity and polyinstantion issues in detail.

* Chapter 3 provides the New decomposition and Recovery algorithm. Moreover, chapter 3 demonstrates how the New decomposition and Recovery algorithm works with an example and discusses the result. Finally this chapter perform an analytical comparison among several decomposition techniques based on Space and Time complexity.

* Chapter 4 summarize the thesis and presents area requiring further research.

Chapter II

Literature Review

The significant development of information technology in recent decades has lead to the widespread use of computer systems in virtually all private as well as public organizations such as banks, universities, libraries, utility companies, military, etc. The increasing availability of the computer systems at lower costs, coupled with the more reliable software and hardware technologies, have all encouraged the widespread use of computing services. As a result more data than ever before is now stored and managed by computer systems [Kang, 95].

A database is a collection of permanent data, managed by the DataBase Management System (DBMS). DBMSs were designed to organize and maintain large amount of information in an efficient and effective manner so that the data can be made available as quickly as possible when requested [Mark, 96].

Although the widespread use of database has proved necessary to support business function, it has also posed serious problems of data security. As a result while DBMSs were developing and matching, a parallel development occurred in the maturing of

computer security policy. Computer security policy deals with those aspects of information security policy that applied or contained within the computer system. Traditionally the following broad objectives applied to computer system: availability, integrity, and security.

AVAILABILITY is concerned with denial of service. It means preventing / detecting /deterring improper denial of access to services provided by the system [Hamm,93]. For example, in military environment, when proper command is issued, the missile should fire. Analogously, in commercial environment, payment orders regarding taxes should be made on time as fixed by law.

INTEGRITY is concerned with preventing the unauthorized modification or destruction of information or data. There are three types of integrity: object, access, and data. Object integrity deals with network transmissions and program libraries; Access integrity is the counterpart of security as it is concerned with which users are authorized to change information in the system; and data integrity deals with the correctness of information in the database context [Date, 90]. Since the integrity concept is crucial for any database security model, a complete section will be devoted to this topic in the literature review.

SECURITY is a system of safeguards designed to protect data and information from intentional or accidental access by unauthorized person. Before the widespread use of computer systems, security of information was provided through a set of rules and procedures for marking and controlling information in the form of printed materials. These rules and procedures were include labeling documents in accordance with the

security required for their handling. A simple security hierarchy was developed with four classification of increasing level of security. The level from lowest to highest are : Unclassified (U), Confidential (C), Secret (S), and Top Secret (TS). A document is given an overall classification, such as Top Secret and individual are also given classification that can not be higher than overall classification [Thur, 95].

This simple security hierarchy has been adopted to computer systems with many of its original concept intact. Information in a computer system is required to be labeled with its security classification. Users of the information are then required to have the proper clearance level to access the data in the system. Then the secure computer system compares the clearance of the users to the classification of the information and mediate the access of the users to the data in accordance with the security rules. For example, a basic security procedure is that no user can read any information with a greater classification than the user has, a confidential user can not have access to secret and top secret information [Hwan, 97] .

Many computer systems run at only one security level. System may run as top secret, meaning that all users are cleared at the same level of top secret, even though the information may be of any level (U, C, S). The information are treated as top secret, and users are trusted to change the level of output information to the appropriate level such as unclassified, etc. This approach to security is not only inefficient, but also is very costly. To apply such a system, all users must be cleared to the highest level of clearance, and all information are entered into the system as if they were at the highest level. Theoretically, a computer can be programmed to distinguish among users and

information in such a manner that one can provide for a range of needs. System which provides this procedure, that is, have more than one security level of classification is called Multilevel Security System . Multilevel Security System deals with the computer system that contains information with a variety of classification and has some users who are not cleared for the highest classification of data that contained in the system [Qian, 97].

Multilevel secure database is intended to provide the security needed for database systems that contain data at a variety of classification, and serves a set of users having different clearance [Pesa, 97]. As a result, there have been several efforts to design multilevel secure relational database models. The following chapter discusses several security models such as Bell- LaPadule model, SeaView model , LDV model, MLR model , Pernul model, Chinese Wall Lattice model.

The Bell-LaPadule model (BLP)

Bell and LaPadule model [Bell, 75] is a formal mathematical model which specifies the requirements for a secure computer system. The model defines three entities, users (human being and is recognized by a unique identity), subject (program in execution that is associated with one user), and object (files, data, etc.). the model provides a simple mechanism for mediating the access of subjects to objects. A security violation occurs when an object is returned to a subject who either has a clearance lower than object in question, or the subject does not have the need-to-know assigned to object [Bell, 75]

Bell-LaPadule model enlarges disrectionary access control with mandatory access control to enforce information flow. Both discretionary and mandatory access must

authorize the operation before the operation is carried out. Discretionary Access Control (DAC) is expressed by a discretionary access matrix M, whose contents can be modified by the subject. On the other hand, Mandatory Access Control (MAC) is expressed by security labels for the objects and security clearance for the subjects, the user has no control over the mandatory access rules. More specifically, for security to be maintained, two properties must be hold [Lin, 93].

1- Simple-security property: NO READ UP !

Subject ***s*** can read object ***o*** only if λ (s) $\geq$ λ (o).

2- * property: NO WRITE DOWN !

Subject ***s*** can write object ***o*** only if λ (s) $\leq$ λ (o).

λ signifying the security label of the indicated subject or object.

The first rule prohibits a program running at an confidential level from reading information from a secret object and the second rule prohibits a program running at a secret level from writing to confidential objects even it is permitted to do so by the discretionary access controls. Information is allowed to flow from bottom to top. BLP prevents information flow between security classes, but it does not prevent information flow between similar security classes[Lin, 93].

A curious aspect of * property is that an unclassified subject can write a secret file. This means that secret files can be destroyed by unclassified subjects. To prevent this problem, a modified * property is sometime used that requires λ (s) = λ (o) that is subjects can write at their own level but can not write up[Lin, 93].

The advantage of BLP model is that it formalize the concept of mandatory access controls and discretionary access controls to enforce information flow policies.

The disadvantage of BLP model is that, the Mandatory Access Controls only prevent information flows between security classes and do not prevent information flows between same security classes. Mandatory controls do not solve the Trojan horse problems . Trojan horse software performs normal functions expected by its user, but also engages in unauthorized activities to undermine security .

SeaView Model

The SeaView model [Lunt, 90] was developed as a prototype of a Multilevel Secure Database Management System based on view mechanism. The model governs access to the data stored in the database on the basis of mandatory as well as discretionary policies. The model is formulated in two layered : Mandatory Access Control model (MAC) and the Trusted computing Base (TCB).

The Mandatory Access Control model defines the mandatory security policy which states that users may not have access to data unless their security levels exceed the security levels associated with the data. In other words, the users have the requisite secrecy and integrity authorization for the information, based on information classification [Lunt, 93].

The MAC model assigns two access classes to each subject S: read-class(S) and write-class (S), where read-class(S) >= write-class (S). The access requirements are formalized by the following rules [Lunt, 88]:

a) A subject S can read data of access class c only if read-class (S) >= c, and

b) A subject S can write data of access class c only if write-class (S) <= c.

The Trusted Computed Base defines the discretionary access control policy and the supporting policies; it specifies the component of multilevel secure relational database system, including multilevel secure relations, snapshots, views, relational expression, integrity constraints, and discretionary authorization. TCB model also defines the multilevel secure relations and formalizes policies for labeling new and derived data, transaction consistency, and discretionary security [Denn, 88].

The SeaView Decomposition Technique was initiate by SeaView model in 1988. SeaView decomposes a relation into partitions using vertical and horizontal fragmentation. The decomposition is based on classification of attributes of the relation, all attributes with the same classification are grouped together to form a single level relation, each relation is labeled with a single classification and then physically stored in the database. The underlying trusted computed base enforces the mandatory controls with respect to the single level base relation. The multilevel secure relation is reconstructed by performing a repeated join operation. [Lunt, 90].

SeaView model has produced the first design for a multilevel secure database system that allow users to obtain data they are cleared for from systems that also contain data clarified higher than their clearance. Furthermore, SeaView has provided the first interpretation of database integrity in the context of multilevel security, by requiring that database integrity hold with respect to the subset of the database visible at any security level. SeaView also introduce the notion of polyinstantiation, which prevent low level users from inferring the existence of high data objects [Jajo, 90].

However, the reconstruction of a multilevel secure relation required a number of repeated joins operations. The join operations are very expensive operations and the reconstruction would lead to a bad performance if a large number of joins is involved. Moreover, the SeaView decomposition and reconstruction leads to a set of Spurious tuples (tuples exist after the reconstruction but didn't exist before the decomposition); however, SeaView method provides an algorithm to remove the spurious tuples [Mukk, 94].

The SeaView Decomposition Algorithm decomposes a multilevel secure relation into single level base relations [Lunt, 88].

The following **Notations** will be used for all algorithms :

R: multilevel secure relation, n : number of attributes in relation R.

A1 : primary key, C1 : classification of primary key.

Ai : data attribute, i={2,3,...,n}, Ci: classification attribute for Ai.

TC: Tuple-level classification, TC = Least upper bound { Ci, i= 1, 2, 3, ..., n).

Least upper bound: Let a and b two elements in a partially ordered set (A, ≥). An element c is said to be an upper bound of a and b if a ≤ c and b ≤ c. An element c is said to be a least upper bound of a and b if c is an upper bound of a and b, and if there is no other upper bound d of a and b such that d ≤ c [Liu, 87].

A binary relation from set A to A is said to be a binary relation on A.

A partial ordering relation M is a binary relation that is reflexive, antisymmetric, and transitive. A partially ordered set is a partial ordering relation together with set A.

A binary relation M on A is said to be a reflexive relation if (a,a) is in M for every a in A.

A binary relation M on A is said to be an antisymmetric relation if (a, b) is in M implies that (b, a) is not in M unless a = b.

A binary relation M on A is said to be a transitive relation if (a, c) is in M whenever both (a, b) and (b, c) are in M.

L1 : lowest level in the security classification , H1 : highest level.

R1,c and R2,c : base relations with classification c, c ∈ { U, C, S, TS }.L1 : lowest level in the security classification , H1 : highest level.

R1,c : Primary group base relations with classification c.

Ri,c : Attribute group base relations with classification c.

Input: A multilevel secure relation.

Output: A collection of single level base relations.

1) Primary Key Group Relations :

. ∀ c ∈ { U, C, S, TS] create : R1,c (A1) with class c.

2) Attribute Group Relations :

For i = 2,..., n

∀ c ∈ { U, C, S, TS }

create : Ri,c (A1, C1, Ai) with class c.

End

The SeaView Recovery Algorithm reconstructs a multilevel secure relation from its single-level base relations[Lunt, 88] :

Notations :

P1,c : represent the derived relation (derived from (R1,c))

Pi,c : represent the derived relation (derived from (Ri,c))

Input: A collection of single level base relations

Output A multilevel secure relation.

P1,c = (A1,C1=c) $\forall$ c $\in$ (U, C, S, TS }

For i = 2 to n

$\forall$ c $\in$ (U, C, S, TS }

Pi,c = (K,C1,Ai, c)

End

$Pi = P_{i,U} \cup P_{i,C} \cup P_{i,S} \cup P_{i,TS}$

End

R = (P1 JOIN P2 JOIN P3,... JOIN Pn) [Lunt, 88].

The following example shows how the decomposition and the recovery techniques works in SeaView. The multilevel secure relation in table 1 will be decomposed into single base relations which are stored physically in the database. Table 1 will be used in the SeaView, Novel, and New decomposition techniques.

EMPLOYEE NUMBER C1		NAME C2		JOB C3		BDATE C4		SALARY C5		TC
555	S	DAVID	S	MANAGER	S	02-10-67	S	$65,000	TS	TS
333	S	OMER	S	SPY	TS	12-19-55	S	$69,000	TS	TS
333	S	OMER	S	JANITOR	S	12-19-55	S	$20,000	S	S
666	TS	MIKE	TS	PRESIDENT	TS	10-28-45	TS	$99,000	TS	TS
666	S	SONIA	S	SECRETARY	S	05-05-48	S	$28,000	S	S
444	S	ALI	S	SPY	TS	02-19-65	TS	$75,000	TS	TS
444	S	ALI	S	SALESMAN	S	01-20-60	S	$35,000	S	S

Table 1· MULTILEVEL SECURE RELATION (R)

The SeaView decomposition example, the algorithm decomposes the MLS relation in Table 1 into the following single level base relations. The following example shows how the decomposition techniques works in SeaView. The multilevel secure relation in table 1 will be decomposed into single base relations which are stored physically in the database.

555
333
666
444

R1, S

666

R1, TS

555	DAVID
333	OMER
666	SONIA
444	ALI

R2, S

555	S	MANAGER
333	S	JANITOR
666	S	SECRETARY
444	S	SALESMAN

R3, S

555	S	2-10-67
333	S	12-19-55
666	S	05-05-48
444	S	01-20-60

R4, S

333	S	20,000
666	S	28,000
444	S	35,000

R5, s

666 TS	MIKE

R2, TS

333	S	SPY
666	TS	PRESIDENT
444	S	SPY

R3,TS

666 TS	10-28-45
444 S	02-19-65

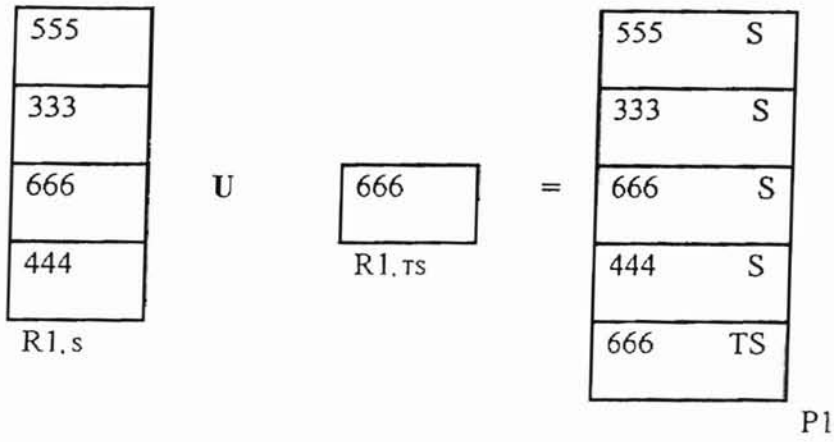
R4,TS

555	S	65000
333	S	69000
666	TS	99000
444	S	75000

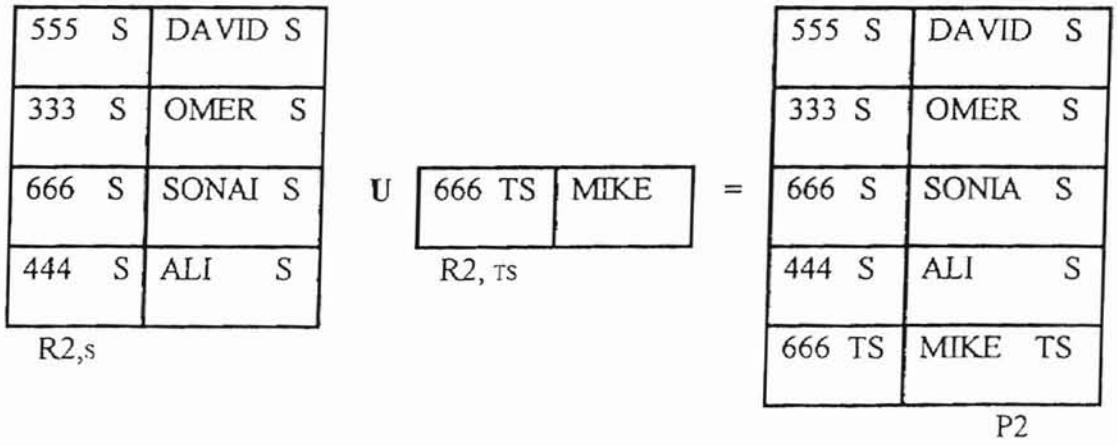
R5, TS

The SeaView Recovery example, the algorithm constructs the multilevel secure relation R from its single level secure base relations.

$P1 = R1,s \cup R1,TS$



$P2 = R2,s \cup R2,TS$



$P3 = R3,_S \cup R3,_{TS}$

555	S	MANAGER	S
333	S	JANITOR	S
666	S	SECRETARY	S
444	S	SALESMAN	S

R3, S

U

333	S	SPY	TS
666	TS	PRESIDENT	TS
444	S	SPY	TS

R3,TS

=

555	S	MANAGER	S
333	S	JANITOR	S
666	S	SECRETARY	S
444	S	SALESMAN	S
333	S	SPY	TS
666	TS	PRESIDENT	TS
444	S	SPY	TS

P3

$P4 = R4,_S \cup R4,_{TS}$

555	S	2-10-67	S
333	S	12-19-55	S
666	S	05-05- 48	S
444	S	01-20-60	S

R4, S

U

666	TS	10-28-45	TS
444	S	02-19-65	TS

R4,TS

=

555	S	2-10-67	S
333	S	12-19-55	S
666	S	05-05- 48	S
444	S	01-20-60	S
666	TS	10-28-45	TS
444	S	02-19-65	TS

P4

$P5 = R5,S \cup R5,TS$

333	S	20,000	S		555	S	65,000	TS		333	S	20,000	S
666	S	28,000	S	U	333	S	69,000	TS	=	666	S	28,000	S
444	S	35,000	S		666	TS	99,000	TS		444	S	35,000	S
					444	S	75,000	TS		555	S	65,000	TS
										333	S	69,000	TS
										666	TS	99,000	TS
										444	S	75,000	TS

R5,S R5,TS P5

$R = P1$ JOIN $P2$ JOIN $P3$ JOIN $P4$ JOIN $P5$

The result is the original multilevel secure relation R (table 1).

LOCK Data Views

LOCK Data Views [Stac, 90] is a multilevel secure relational database model that is builds on the security policies for operating system. LOCK Data View (LDV) is designed to run on SCTC's (LOCK) Trusted Computing Base (TCB). SCTC's LOCK has been designed to provide control in terms of abstract entities and operations which reflect an operating system policy. Access to data is controlled by LOCK and information in the database are stored in single level file. LOCK ensures that these database files may be opened for read/write operations only by subjects executing at the appropriate levels [Stac, 90].

LDV Security Policy consists of a mandatory security policy and a discretionary security policy. The mandatory security provides a multilevel control policy which addresses both access to data and the flow of information in the system, while the discretionary security policy enforces the need to know structure [Haig, 91].

LDV allows individuals posses a rang of clearance to create, share, and manipulate relational databases, that is a subject is allowed to access an object only if the subject security level is more than or equal the object security level. There are three principal entities in the LDV security policy, subject, object, and Effective Access Matrix (EAM). Subjects are the active process-like entities, the objects are the passive file-like entities, and the EAM defines the permissible flows of information within the system [Haig, 91].

The LDV enforces the security policies **by three assured pipelines**. The pipelines are a set of communication processes each of which could be verified separately. The three pipelines are 1) the response pipeline 2) the update pipeline, and 3) , the metadata pipeline. The response pipeline maps a query from the application domains to the database, processes the query to produce a result relation, and export it to the user domain. The update pipeline allows subjects executing in special data input domain to prepare records for input to the DBMS, identify records to delete, and transforms them into a data type readable by the DBMS domain. The metadata pipeline provides the mechanisms for defining a database structure, specifying relations, views, attributes,

classifications, and would normally be restricted to access by a database administrator [Haig, 91] .

LDV Data Distribution: The multilevel secure relation is distributed across LOCK data files by assigning a set of files per security level, there is no replication of data across files. The Update Pipeline determines the appropriate assignment of data to files by examining several classification constraints (Context constraints are rules that refer to combination of data items). Each security level of the MLS relation is stored in a separate file, each file contains the partial relation visible at the level of that file or higher. Each attribute involved in by context constraints is placed in a separate file. The partial relation for the view at any given level is computed from the data stored at that level and from lower level data using the MERGE [Haig, 91].

Reconstruction of a Multilevel secure relation: The query processor (The Response pipeline is the query processor for LDV) reconstructs a partial relation representing a user view from the data distributed across files. There is one partial relation corresponding to each base relation in the user's query. The remaining query processing such as JOIN is performed using these partial relations[Stac, 90].

The query processor merge tuples from different files with the same primary key to reconstruct a partial relation at a particular level. The reconstruction is performed in two steps. 1) the partial tuples are retrieved, based upon the query, the level of the user, and the attribute classification constraints, 2) an operator called MERGE is used, MERGE works with the knowledge of the properties of the tuples in the different partitions of a relation [Stac,90].

The major advantage of LDV is that LOCK enforces its security policy on data stored in operating system files which reduce the amount of trusted code in the database systems that are responsible for enforcing security. However, the major disadvantage of LDV data distribution is the performance penalty for retrieval requests for the recovery procedure. The reconstruction process required costly merge and join operations which slow down the performance. The LOCK security policy is incomplete in dealing with DBMS security because of its operating system orientation [Silv, 95].

MLR Data Model

MLR [Chen, 95] is a multilevel secure relational data model that supports classification labels for both elements and tuples. It has five operation statements and five integrity properties for manipulating MLS relations. It combines several ideas from other security models such as SeaView and LOCK Data View models. Chen and Sandhu [Chen, 95] have redefined many concepts and introduced several new ones. The major difference is the requirement that there can be at most one tuple in each access class for a given entity; this gives the simplicity for converting the MLR model to tuple-labeling data model .

MLR model took Entity Integrity and Foreign Key Integrity from original SeaView model [lunt,88] and redefined Referential and Polyinstantiation Integrity; moreover, it introduced Data-Borrow integrity. Polyinstantiation Integrity in MLR model is more general than SeaView model, that is , it takes care of both element polyinstantiation and entity polyinstantiation while SeaView model only applied entity polyinstantiation [Chen, 95].

Polyinstantiation integrity constraints (PI) have been required by several models; The following model is similar to the one proposed by SeaView except for the first requirement. Chen and Sandhu [Chen,95] defined PI as : An instance Rc of a multilevel secure relation R satisfies polyinstantiation integrity if and only if for $1 \leq i \leq n$

a) A1, TC $\rightarrow$ Ci

b) A1, C1, Ci $\rightarrow$ Ai

A1, Ai : attributes in relation R , C1, Ci, : classification level for A1, Ai.

TC : Tuple level classification , TC = Lub { C1, C2,..., Cn}.

The second requirement is the same as the polyinstantiation integrity constraints in SeaView model, while the first requirement says that every entity in a relation can have at most one tuple for every access class [Chen, 95].

Data-Borrow Integrity Constraints (DBI) is a major constraints in MLR because it ensures that changes to data at a lower levels can be automatically propagated to higher levels [Chen, 95]. The Data-Borrow integrity constraints is: An instance Rc of a multilevel secure relation R satisfies Data-Borrow integrity:

if and only if for all $t \in Rc$ and

$1 \leq i \leq n$, if $t[Ai] \neq null \wedge t[Ci] < t[TC]$, there exists $t' \in Rc$ such that

$t'[A1,C1] = [A1,C1] \wedge t'[TC] = t'[Ci] = t[Ci] \wedge t'[Ai] = t[Ai]$ [Chen, 95].

t, t' : tuples in relation R.

The following example is borrowed from Pernul model to explain the Data-Borrow Integrity [Chen, 95] :

Let R1 and R2 be two relation instances :

Ship Name C1	Objective C2	Destination C3	TC
M1 C	Discovery C	Jerusalem S	S
M1 C	Discovery C	Amman C	C

R1 : relation instance

Ship Name C1	Objective C2	Destination C3	TC
M1 C	Discovery C	Jerusalem S	S

R2 : relation instance

R1 satisfies Data-Borrow integrity while R2 dose not satisfies DBI. Here in R1 DBI requires that C-tuple must exist. This is because absence of C-tuple means that to C-subject the entity M1 dose not exist; which implies that C-subject can not access C-tuple [Chen, 95].

The Referential Integrity in MLR model is similar to referential integrity in SeaView model except for the third condition in page 24 which states that in SeaView $t1[FKc] \geq t2[PKc]$ while in MLR $t1[FKc] = t2[PKc]$.

SeaView allows downward references that is the foreign key calls all tuples in referencing relation which their security level greater than security level of the primary key of referenced relation. On the other hand, MLR does not allow downward references [Chen, 95].

MLR model has five manipulation statement. Four of them are the conventional SQL statements of INSERT, SELECT, UPDATE, and DELETE. Chen and Sandhu [Chen,95] redefines the semantics of the four standard SQL statements and introduced the fifth statement which is UPLEVEL to create a tuple whose contents are borrowed from other tuples and insert it in the relation.

Decomposition Technique in MLR: A multilevel secure relation R (A1,C1,A2,C2,...,An,Cn,TC) can be decomposed to several tuple-level labeling relations [Chen, 95]:

1) R1 (A1,Eid,C1)

2) R2 (Eid,El2,...,Eln,TC)

3) $R3_k$ (El_k,Eid,A_k,C_k)

R1, R2, $R3_k$ are single level base relations that store the decomposed MLS relation. Eid is an entity identification, El1,...,El_k are element identifications, A_k is attribute value and C_k is classification of A_k [Chen, 95]. The multilevel secure relation is reconstructed by performing several join operations.

MLR Model combines several ideas from SeaView and LDV models, redefines others, and finally introduced new ones that simplify the Database security model. MLR improve the Polyinstantiation and referential integrity which removes the ambiguity from the database security model. However, the disadvantage of MLR is that the reconstruction of a multilevel secure relation requires several join operations, and the join operations are expensive and lead to bad performance.

Pernul Model

Pernul model [Pern, 91] is a multilevel secure relational data model that is not fully based on Bell-LaPadula model but is fully based on views. Pernul model supports mandatory access control; that is restricts access to classified information to cleared personnel. Fragments and views are the granularity of data to which we provide automated security labeling. Views are the only user interface to the users and users access only data that is contained in the user's view [Pern, 91].

Decomposition Technique: Decomposition in Pernul model is based on the definition of isolated and overlapping views. Pernul [Pern,91] defines several notations that forms the basis of the model and those notations are necessary to understand the model:

View : A view is a virtual table that is derived from other tables, the view forms the area of the database the user is allowed to access, it has been considered as a object for access control [Qian, 96a].

RS (A1, A2,..., An) : A relation schema RS is a set of attributes {A1,...,An}, each attribute has a domain. The relation schema is used to describe a relation R.

R : A relation R of relation schema RS is a set of distinct tuples {t1,...,tn} of the form <a1,...,an> where ai is a within the domain of Ai.

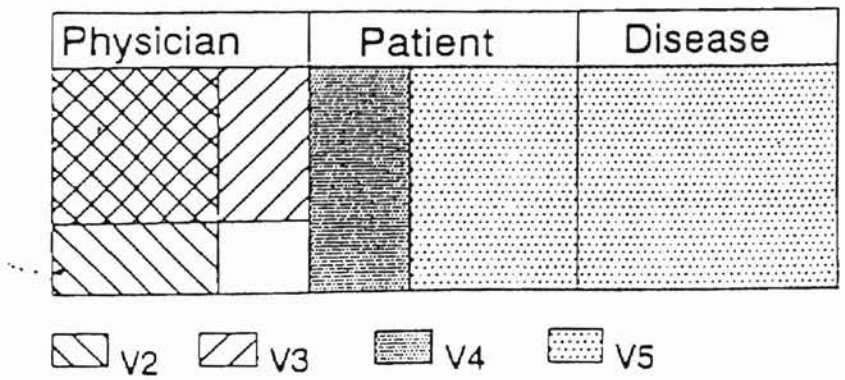
Vertical fragmentation: is the projection of a relational schema RS into subsets of its attributes called fragments.

Horizontal fragmentation: is partitioning a relational schema RS into disjoint fragments based on a predicate defined on RS. The predicate is a Boolean combination of terms, each term is a comparison that can be true or false.

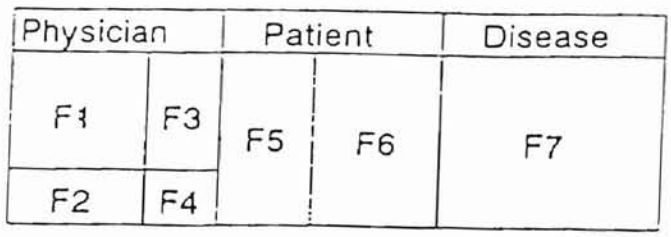
Derived horizontal fragmentation: is partitioning a relational schema RSi by applying to it the same partitioning criterion as applied to RSj

Pernul [Pern, 91] decomposes a relation R into a set of disjoint fragments by performing the following procedure. First finds all the overlapping and isolated views, second, for each pair of the overlapping views performs a vertical, horizontal, and derived horizontal decomposition, and finally performs a vertical, horizontal, and derived horizontal decomposition for each pair of the isolate views. It is recommended that the vertical decomposition should be performed before the horizontal decomposition to minimize the number of fragments [Pern, 91].

The following example is borrowed from Pernul model [Pern, 91] to show the relation between views and fragments. Figure 1 shows the original relation with five views and figure 2 shows the decomposed relation with seven fragments. For example, view V3 is stored as fragment F1 and F3, V5 is stored as F6 and F7.



"Figure 1: Start Schema of the Hospital Database" [Pern, 91]



" Figure 2 : Decomposition of the hospital database" [Pern, 91]

The multilevel secure relation is reconstructed by appending horizontal fragments to create larger horizontal fragment and concatenating the vertical fragments. The join operations are only performed if the view combine two relations[Pern, 91].

Pernul model provides protection to database fragments and user views through security labels, it also provides a methodology for automated labeling of security objects and subject, and finally it supports reassignment of clearance levels to users[Pern, 91].

Fragments are the security objects of the database and represent the granularity level of data to which the classification labels are assigned. Views are the only interface of the users to the database and are the security subject to which different levels of clearances are assigned [Pern, 91]

Pernul uses the cardinality functions card(a:F→V) and card (d: F→V) as a way of labeling which relate each fragment to the set of views accessing that fragment, and different fragments to which views V have access. With regard to the example, a(F1) = { V1,V2, V3} , a (F4) = { V1}. The card (a(F1)) = 3 ,and the card (a(F4)) = 1. That is fragment F1 is accessed by views V1, V2, V3 while fragment F4 is accessed by only view V1. At the other side, d(V1) = { F1, F2, F3, F4, F5, F6, F7} and d (V4) = {F5}, that is view V1 can access {F1, F2,...,F7}, while view V4 can access only F5. The

card (d (V1)) = 7 and card (d(V4)) = 1. [Pern, 95]

The ordering of fragments represents the sensitivity of information contained in the fragments. Fragment F4 is accessed by one view only , it contains the most sensitive information and the highest level of classification has to be assigned to this fragment. On the other hand, view V1 accesses the most fragments and should therefore have a level of clearance that enables the users to access the corresponding fragments[Pern, 91].

The reconstruction of a multilevel secure relation requires mostly concatenating and appending operations and performed join operations only if view combines two relations. The decomposition is lossless (no spurious tuples) and avoid using repeated join operations which are expensive. The major disadvantage is the limitation of the model for specific organizations such as hospitals; in addition, Pernul model does not support the discretionary access control which most security models support both mandatory access control and discretionary access control[Pern, 91].

The Chinese Wall Lattice Model

The Chinese Wall policy that Brewer [Brew, 89] identifies arises in the segment of the commercial sector that provides consulting services to other companies. The Chinese lattice model [Sand, 93] design a lattice based access model for enforcing that policy. The objective of this policy is to prevent information flows that result in a conflict of interest for individual consultants.

Information flow is a key component of lattice models , A partially order set is said to be a lattice if every two elements in the set have a unique least upper bound and a unique greatest lower bound. A binary relation is said to be a partial ordering relation if

it is reflexive, transitive, and antisymmetric. Information flow means flow of information from one security level to another. The information is controlled by assigning a security level for every object in the system. For example , A --> B means information can flow from A to B; while A ≠> B means information can not flow from A to B [Fole, 96].

The Chinese Wall lattice Model [Sand, 93] prevents information flows that result in a conflict of interest for individual consultants. Company information is categorized in mutually disjoint conflict-of-interest classes, each company belongs to one conflict of interest class, and the Chinese wall policy requires that a consultant not be able to read information for more than one company in any given conflict-of-interest class.

The Chinese Wall Lattice Security Policy: The clearance of a user is a high-water mark that can float up the lattice but not down. A newly enrolled user in the system is assigned [⊥,⊥] (clean state), Now, by reading information about company 1 in conflict of interest class 1, the user clearance is modified to [1,⊥], reading information about company 2 in conflict-of -interest class 2, the user clearance is modified to [1,2]. This floating up of user's clearance is allowed as long as the clearance does not float up to Syshigh. The floating up of a user's clearance corresponds with the ability to create subjects with new labels for that user[Brew, 89].

This model is very useful when consultants deal with confidential company information for their clients, and the consultant should not have access to information about, say, two banks because such information creates a conflict of interest in the

consultant's analysis. Insider information about two similar types of companies also presents the potential for consultants to use such knowledge for personal profit. However, the Major disadvantage is the limitation of that model, that it is only useful for specific organizations and can not be modified to suite other security requirement for different applications.

Novel Decomposition Technique

Jajodia and Sandhu [Jajo, 91a] designs a decomposition technique that decompose a Multilevel Secure relation into single- level base relations using just horizontal fragmentation. This is in contrast to the SeaView decomposition which is based on vertical and horizontal fragmentation. The decomposition is based on the security level of the tuple of the relation. The tuple security-level is calculated as the least upper bound of the classification of attributes.

All tuples with the same security class are grouped together to form a single relation and then physically stored in the database The multilevel secure relation is reconstructed by union operations. Since the decomposition doesn't require any vertical fragmentation, it is possible to reconstruct the multilevel secure relation from the underlying single level base relations without having to perform any join operations; only union operations are required to be taken [Jajo, 91].

The major disadvantage of Novel technique is that the partition is large which requires large I/O that degrade the performance. Another disadvantage of Novel is that during update operations, Novel replicate the entire tuple when one attribute or more is updated which increases the storage overhead.

The Novel Decomposition Algorithm decomposes a multilevel secure relation into single level base relations [Jajo, 91] :

The notations have been defined in page 11 in chapter 2.

Input : R (A1, C1, A2, C2,..., An, Cn, TC) .

Output : A collection of single-level secure base relations :

Rc (A1, C1, A2, C2, A3, C3,.., An, Cn)

$\forall c \in \{ U, C, S, TS \}$

Create R1,c (A1, C1, A2, C2, ..., An, Cn)

END

The Novel Recovery Algorithm reconstructs a multilevel secure relation from its single level base relations [Jajo, 91].

Input : A collection of single-level secure base relations

Output : A multilevel secure relation R where

R (A1, C1, A2, C2,..., An, Cn, TC) .

Output: A multilevel secure relation R, R (A1, C1, A2, C2,..., An, Cn, TC)

. $R = R_{1,U} \cup R_{1,C} \cup R_{1,S} \cup R_{1,TS}$

The Novel decomposition algorithm decomposes the MLS relation in table 1, page 13, into the following single level base relations.

EMPLOYEE NUMBER C1		NAME C2		JOB C3		BDATE C4		SALARY C5		TC
333	S	OMER	S	JANITOR	S	12-19-55	S	$20,000	S	S
666	S	SONIA	S	SECRETA	S	05-05-48	S	$28,000	S	S
444	S	ALI	S	SALESMAN	S	02-19-65	S	$35,000	S	S

Table 2 R1, S

EMPLOYEE NUMBER	C1	NAME	C2	JOB	C3	BDATE	C4	SALARY	C5	TC
555	S	DAVID	S	MANAGER	S	02-10-67	S	$65,000	TS	TS
333	S	OMER	S	SPY	TS	12-19-55	S	$69,000	TS	TS
666	TS	MIKE	TS	PRESIDENT	TS	10-28-45	TS	$99,000	TS	TS
444	S	ALI	S	SPY	TS	02-19-65	TS	$75,000	TS	TS

Table 3 $R2_{,TS}$

The Novel recovery example, the Novel Recovery algorithm reconstructs MLS relation R from its single level secure base relations by performing union operation.

$$R = R1_{,S} \cup R2_{,TS}$$

The result will be the original relation R in table 1.

Integrity In Multilevel Secure Relational Database

Integrity and security are two of the most frequently heard concepts in the database world. Secrecy refers to safety of data against unauthorized disclosure, and integrity refers to accuracy of data. Preserving the accuracy of data is extremely important in any database. In the relational model, preserving accuracy of data is achieved by the use of integrity constraints. Entity integrity, referential integrity, foreign key integrity, and polyinstantiation integrity are four of the most important integrity constraints. They apply to all relations and should be enforced by the database management systems [Lunt, 90].

Entity integrity guarantees a unique representation of each entity in the database through specification of primary key attributes for each relation. Referential integrity assures that if there exist any reference between two or more entities, then the related

entities do exist in the database. Referential integrity is an inter-relation integrity constraints and is achieved with the use of foreign attributes. Foreign key integrity guarantees a uniform classification for the attributes that belong to the foreign keys. Polyinstantiation integrity specifies that there must never be two tuples with the same primary key unless they represent polyinstantiated tuples or elements and controls the effects of polyinstantantion [Sand, 92]. Before defining entity, referential, foreign key , and polyinstantiation integrity constraints, it is necessary to define some variables that will be used in this section. Furthermore, it is important to understand the concepts of candidate key, primary key, and foreign key.

Ai : denotes the ith attribute in relation R, t: a tuple $\in$ Rc

Ci : classification of attribute Ai, Cj: classification for the jth attribute.

ai : attribute i in instance Rc, ci classification of ith attribute in instance Rc

tc : tuple classification in Rc, Di : domain of ith element.

Rc : A relation instance, which is a set of distinct tuples of the form

(a1,c1,a2,c2,...,an,cn,tc) where each ai $\in$ Di or ai = null, c $\geq$ ci and

tc = lub { ci : i=1,n}

R_{1c}, R_{2c} : relation instances.

AKc : classification for primary key, FKc : classification for foreign key

FK : Foreign key, AK : Primary key, t1 : tuple $\in$ R_{1c}, t2 : tuple $\in$ R_{2c}

Candidate key: A candidate key of a relation R is a minimal set of attributes that serves as the unique identifier for each tuple in relation R [Date, 90].

Primary key: A primary key for a relation R is a candidate key of R. It is possible that a relation R has more than one candidate key, in which case exactly one candidate key must be chosen and designed as the primary key of R. The primary key of a relation serves the purpose of selecting a specific tuple from the relation as well as linking tuples from different relations [Date, 90].

Foreign Key: The definition of a foreign requires two relations, a referencing relation R1 and a referenced relation R2. Let PK denotes the primary key of R2, and let FK denotes one or more attributes of the relation R1. FK is said to be a foreign key of R1 if given any tuple t1 in R1, the following two requirements are met [Date, 90]:

1) t[FK] is either wholly null or wholly non-null

2) Whenever t1[FK] is non null, there is a tuple t2 in R2 such that t1[FK] = t2[PK].

The entity integrity constraints states that no primary key value can be null. This is because we use the primary key value to identify individual tuples in a relation; having null values for the primary key implies that we cannot identify some tuples. For example, if two or more tuples had null for their primary keys, we might not be able to distinguish them [Dosh, 92].

The Referential Integrity Constraint is a constraint that is specified between two relations and is used to keep the consistency among tuples of the two relations. the referential integrity constraints states that a tuple in one relation that refers to another relation must refer to an exiting tuple in the relation [Dosh, 92].

Entity Integrity constraints in Multilevel secure relations : A multilevel relation schema R is said to satisfy entity integrity if for all relation instances Rc of R and tuple t $\in$ Rc :

1) if Ai $\in$ PK then t[Ai] $\neq$ null, i.e., classification of primary key can not be null.

2) if Ai, Aj $\in$ PK , then t[Ci] = t[Cj] , i.e., PK is said to be uniformly classified, and

3) if Ai $\notin$ PK then t [Ci] $\geq$ t[C [PK]] where C[PK] is the classification of the apparent primary key PK [Lunt, 90].

Foreign Key Integrity Constraints: A multilevel secure relation schema R is said to satisfy foreign key integrity constraints if it satisfies the following conditions:

1) Either ($\forall$ Ai $\in$ FK) [t[Ai] = null] OR ($\forall$ Ai $\in$ FK) [t[Ai] $\neq$ null]

2) If Ai, Aj $\in$ FK then t[Ci] = t [Cj] [Lunt, 88].

The first condition states that all attributes belong to foreign key should be all null or non of them should be null, The second condition states that the classification for all attributes belong to the foreign key should be the same.

Referential integrity constraints in MLS relations : A multilevel relation schema R is said to satisfy Referential integrity [Dosh, 92] if for all relation instances R_{1c} of R1, R_{2c} of R2 and tuple t1 $\in$ R_{1c}, t2 $\in$ R_{2c} :

1) if t1 $\in$ R_{1c} then t1[FK] $\neq$ null, foreign key classification can not be null.

2) if Ai, Aj $\in$ FK , then t1[Ci] = t2[Cj], FK is said to be uniformly classified.

3) t1[FKc] $\geq$ t2[PKc] , i.e. the access class of the foreign key should always dominate the access class of the primary key of the referenced tuple,

4) t1[FKc] = t2 [AKc] , the access class of the foreign key must be same as the access class of the referenced primary key.

Polyinstantiation Integrity: Polyinstantiation is a natural consequence of a multilevel secure relation. SeaView introduced the concept of polyinstantiation, by which different versions of the same real-world entity (for example, a person) can be represented in the database, where the different versions represent what is known to users at different clearance levels. Polyinstantiating has two fundamentals forms: polyinstantiated tuples and polyinstantiated elements [Sand, 90] .

Polyinstantiated tuples occurs when a relation contains multiple tuples with the same primary key value, but having different access class values for the apparent primary key. Polyinstantiated elements occurs when a relation contains two or more tuples with identical primary key and the associated access class values, but having different values for one or more remaining attributes [Sand, 90].

Polyinstantiated integrity constraints: Let AK be the apparent primary key of R. R satisfies polyinstantion integrity if and only if for every Rc, we have for all Ai

AK, C_{AK}, Ci $\rightarrow$ Ai

This constraints stipulates that the user-specified apparent key AK, in conjunction with the classification attributes C_{AK} and Ci, determines the value of the Ai attribute [Sand, 90].

Chapter III

The New Decomposition Technique

During the past decade, there has been much interest in multilevel secure (MLS) database management systems, and in particular MLS relational database models. This has resulted in several MLS relational database models such as SeaView, LDV, and Pernul model. Multilevel secure DBMSs are subject to a number of security-related architectural and functional factors that affect performance and storage overhead. These factors include, among others, the distribution of data among security levels and how the database is physically partitioned into files.

The current models concentrate on update semantics, design, and integrity issues while ignoring performance and storage overhead which are the major parts of any security model. Several designers deliberately sacrifice performance, and storage requirements to achieve update semantics, design, and integrity issues.

After realizing the important of decomposition techniques and their role in performance and storage requirements, few papers were published that study decomposition techniques. SeaView model [Lunt, 90] decomposes a MLS relation into

single level base relations using horizontal and vertical fragmentation. The reconstruction of the MLS relation from its single level base relations requires repeated join operations, that is not only expensive but also results in an inefficient performance. Novel decomposes a MLS relation using horizontal fragmentation, the reconstruction of MLS relation requires union operations. The major problem with Novel technique is that the partition is large which requires large I/O that degrade the performance. Another disadvantage of Novel is that during update operations, Novel replicate the entire tuple when one attribute or more is updated which increases the storage overhead. MLR as well as LDV model have the performance penalty due to expensive join operations. The major problem of Pernul and Chinese models is the limitation of those models for specific organizations such as hospitals and consulting companies. Pernul and Chinese models do not support discretionary access control which is a fundamental categories of database security. As a result of the previous discussion I have designed a decomposition technique that minimize the number of join operations to one; Furthermore, the partition size is relatively small compare to partition size in Novel which minimize the storage requirement. On the other hand, the performance point of view, the new technique is better than the other in regard of updating operations. In this chapter the new decomposition and new recovery algorithm is discussed and an analytical comparison between the new technique and the existing ones is performed based on Time and Space complexity.

The New Decomposition Algorithm

The decomposition of a multilevel secure relation into single level base relations is as follows:

Notations

R1,c and R2,c : base relations with classification c, c $\in$ { U, C, S, TS }.

R1,c : Primary group base relations with classification c.

Ri,c : Attribute group base relations with classification c.

The other notations have been defined in page 11 in chapter 2.

Input : R (A1, C1, A2, C2,..., An, Cn, TC) .

Output : A collection of single-level secure base relations :

R1,c (A1, C1, A2, C2, A3, C3..., A ($\lceil n/2 \rceil$) , C ($\lceil n/2 \rceil$))

R2,c (A1, C1 , A ($\lceil n/2 \rceil + 1$) , C ($\lceil n/2 \rceil + 1$) , A ($\lceil n/2 \rceil + 2$) , C ($\lceil n/2 \rceil + 2$) ..., A(n), C(n))

$\forall$ c $\in$ { U, C, S, TS }

a) Create R1,c with = $\lceil n/2 \rceil$ attributes

$\forall$ TC = c

For i = 2 to = $\lceil n/2 \rceil$

Insert A (i) into R1.

End

b) Create R2,c with $\lceil n/2 \rceil$ attributes

$\forall$ TC = c

For i = $\lceil n/2 \rceil$ + 1 to n

Insert A (i) into R2.

End

End

The New Recovery Algorithm :

The recovery algorithm of a multilevel secure relation from its single-level base relations is as follows :

Notations:

R1,c : (A1, C1, A2, C2,..., A $_{(\lceil n/2 \rceil)}$, C$_{(\lceil n/2 \rceil)}$)

R2,c : (A1, C1, A $_{(\lceil n/2 \rceil + 1)}$, C$_{(\lceil n/2 \rceil + 1)}$, ..., An, Cn)

$\cup$: Union operation.

JOIN : Natural Join operation.

The other notations have been defined in page 22 in chapter 2.

Input : A collection of single level secure base relations.

Output: A multilevel secure relation R, R (A1, C1, A2, C2,..., An, Cn, TC)

$R1 = R_{1,U} \cup R_{1,C} \cup R_{1,S} \cup R_{1,TS}$

$R2 = R_{1,U} \cup R_{2,C} \cup R_{2,S} \cup R_{2,TS}$

R = (R1 JOIN R2)

The New Decomposition algorithm decomposes the MLS relation in table 1, page 13, into the following single level base relations

EMPLOYEE NO.	C1	NAME	C2	JOB	C3
333	S	OMER	S	JANITOR	S
666	S	SONIA	S	SECRETARY	S
444	S	ALI	S	SALESMAN	S

Table 4 R1, s

EMPLOYEE NO.	C1	BDATE	C4	SALARY	C5
333	S	12-99-55	S	$20,000	S
666	S	05-05-48	S	$28,000	S
444	S	02-19-65	S	$35,000	S

Table 5 R2, s

EMPLOYEE NO.	C1	NAME	C2	JOB	C3
555	S	DAVID	S	MANAGER	S
333	S	OMER	S	SPY	TS
666	TS	MIKE	TS	PRESIDENT	TS
444	S	ALI	S	SPY	TS

Table 6 $R1_{,TS}$

EMPLOYEE NO.	C1	BDATE	C4	SALARY	C5
555	S	02-10-67	S	$65,000	TS
333	S	12-19-55	S	$69,000	TS
666	TS	10-28-45	TS	$99,000	TS
444	S	02-19-65	TS	$75,000	TS

Table 7 $R2_{,TS}$

The New recovery algorithm constructs the multilevel secure relation R from its single level secure base relations by performing two union operation and only one join operation.

$R1 = R1_S \cup R1_{,TS}$

EMPLOYEE NUMBER	C1	NAME	C2	JOB	C3
555	S	DAVID	S	MANAGER	S
333	S	OMER	S	SPY	TS
333	S	OMER	S	JANITOR	S
666	TS	MIKE	TS	PRESIDENT	TS
666	S	SONIA	S	SECRETARY	S
444	S	ALI	S	SPY	TS
444	S	ALI	S	SALESMAN	S

R1

$R2 = R2_{,S} \cup R2_{,TS}$

EMPLOYEE NUMBER	C1	BDATE	C4	SALARY	C5
555	S	02-10-67	S	$65,000	TS
333	S	12-19-55	S	$69,000	TS
333	S	12-19-55	S	$20,000	S
666	TS	10-28-45	TS	$99,000	TS
666	S	05-05-48	S	$28,000	S
444	S	02-19-65	TS	$75,000	TS
444	S	01-20-60	S	$35,000	S

R = R1 JOIN R2

The result will be the original relation R in table 1.

Discussion

In Chapter 2, SeaView, and Novel algorithms decompose a MLS relation R into single level secure base relations, and reconstruct the MLS relation from its single level base relation. In the previous section, the New technique decomposes a MLS relation into single level base relations and reconstructs it from its single level base relations. The MLS relation R (table 1) has five attributes, seven tuples, and two security

classification (S, TS). SeaView decomposes R into 10 small single level base relations, Novel decomposes R into two large single level base relations, and the New technique decomposes R into 4 moderate size single level base relations. SeaView requires **five union** operations and **five join** operations to reconstruct MLS relation R, Novel requires two union operations. On the other hand, the New technique requires only **two union** operations and only **1 join** operation.

The number of join and union operations to reconstruct MLS in SeaView, and the New technique depends on several factors: number of attributes and number of security levels in the MLS relation. Assume R which is a MLS relation has **N** attributes and **X** security levels, then SeaView requires **N-1** number of join operations and **N* (X - 1)** union operation. On the other hand, the New technique requires **one** join operation and **(X - 1) * 2** union operations. That is the New technique requires less join and union operations which means that the performance of the new technique is better than SeaView model. That is New technique requires one join operation compares to N-1 join operation for SeaView, and (X-1) * 2 union operations for the New technique compares to N* (X-1) union operations for SeaView. Another example, let R has 7 attributes and 4 security level, then N = 7, X = 4. SeaView requires 7-1 = 6 join operations and the New technique requires one join operation. On the other hand, SeaView requires 7* (4-1) = 21 union operations while the New technique requires (4-1) * 2 = 6 union operations.

Multilevel Secure Operations

Having defined the decomposition and recovery algorithms for the New technique and the Novel technique shows how each technique works in decomposing MLS relation into single level base relations, we now focus our attention on the operations available for manipulating the information stored in the multilevel secure relation. The operations that will be discussed are INSERT, DELETE, and UPDATE.

Table 1, table 4 through table 7 will be used in this section :

The following operations are used in the **New decomposition technique**.

A Secret user is issuing this command :

INSERT

INTO Rs

VALUES ("555", "JOHN", " PROGRAMMER", "01-25-70", "$40,000")

First, the database director guarantees that the primary key classification does not contain any null value. Second, for all tuples belong to Rc , the new primary key does not exist in that relation.

The new tuple is added to table 4 and table 5 and the new base relation is

EMPLOYEE NO.	C1	NAME	C2	JOB	C3
333	S	OMER	S	JANITOR	S
666	S	SONIA	S	SECRETARY	S
555	S	JOHN	S	PROGRAMMER	S
444	S	ALI	S	SALESMAN	S

Table 8 R1, s

EMPLOYEE NO	C1	BDATE	C4	SALARY	C5
333	S	12-99-55	S	$20,000	S
666	S	05-05-48	S	$28,000	S
555	S	01-25-70	S	$40,000	S
444	S	02-19-65	S	$35,000	S

Table 9 R2, s

A Secret user is issuing this command :

UPDATE

SET SALARY = " $25,000"

WHERE EMPLOYEE NO. = 333.

Table 9 will be changed to the following base relation

EMPLOYEE NO.	C1	BDATE	C4	SALARY	C5
333	S	12-99-55	S	$25,000	S
666	S	05-05-48	S	$28,000	S
555	S	01-25-70	S	$40,000	S
444	S	02-19-65	S	$35,000	S

Table 10 R2, s

A Secret user is issuing the following command :

UPDATE Rs
SET JOB = " SALESMANAGER"
WHERE EMPLOYEE NO. = "444"

Table 8 will be changed

EMPLOYEE NO.	C1	NAME	C2	JOB	C3
333	S	OMER	S	JANITOR	S
666	S	SONIA	S	SECRETARY	S
555	S	JOHN	S	PROGRAMMER	S
444	S	ALI	S	SALESMANAGER	S

Table 11 R1, s

A Top Secret user is issuing this command

UPDATE Rrs
SET SALARY = " $85,000"
WHERE EMPLOYEE NO. = " 444"

Table 7 will be changed :

EMPLOYEE NO.	C1	BDATE	C4	SALARY	C5
555	S	02-10-67	S	$65,000	TS
333	S	12-19-55	S	$69,000	TS
666	TS	10-28-45	TS	$99,000	TS
444	S	02-19-65	TS	$85,000	TS

Table 12 R2,TS

A Top Secret user issues this command

UPDATE Rs
SET JOB = " SUPERVISOR"
WHERE EMPLOYEE NO. = "555"

Since a Top Secret is updating a secret base relation, a new tuple is created and added to Table 6 That is one of the major difference between the new technique and Novel, because Novel will create a new tuple with six attributes while the new technique create a tuple with three attributes only, several examples will discuss the update procedure in Novel later in this section.

The new base relation is

EMPLOYEE NO.	C1	NAME	C2	JOB	C3
555	S	DAVID	S	MANAGER	S
333	S	OMER	S	SPY	TS
666	TS	MIKE	TS	PRESIDENT	TS
555	S	JOHN	S	SUPERVISOR	TS
444	S	ALI	S	SPY	TS

Table 13 R1, TS

A Secret user issues this command:

DELETE

FROM Rs

WHERE EMPLOYEE = "444"

Table 10 and Table 11 will lose one tuple each, so the new relations are :

EMPLOYEE NO.	C1	NAME	C2	JOB	C3
333	S	OMER	S	JANITOR	S
666	S	SONIA	S	SECRETARY	S
555	S	JOHN	S	PROGRAMMER	S

Table 14 R1, s

EMPLOYEE NO.	C1	BDATE	C4	SALARY	C5
333	S	12-99-55	S	$25,000	S
666	S	05-05-48	S	$28,000	S
555	S	01-25-70	S	$40,000	S

Table 15 R2, s

The following operations are used in **Novel decomposition technique**:

Table 1 through Table 3 are used in this section:

A Secret user is issuing this command :

INSERT

INTO Rs

VALUES ("999", "ALEX", " TEACHER", "02-15-50", "$22,000")

Table 1 will be changed:

EMPLOYEE NUMBER	C1	NAME	C2	JOB	C3	BDATE	C4	SALARY	C5	TC
333	S	OMER	S	JANITOR	S	12-19-55	S	$20,000	S	S
666	S	SONIA	S	SECRETARY	S	05-05-48	S	$28,000	S	S
444	S	ALI	S	SALESMAN	S	02-19-65	S	$35,000	S	S
999	S	ALEX	S	TEACHER	S	02-15-50	S	$22,000	S	S

Table 16 R1, s

A Top Secret user issues this command

DELETE

FROM R_{TS}

WHERE EMPLOYEE = "666"

Table 3 will lose one tuple whose primary key is 666, the new relation is:

EMPLOYEE NUMBER C1		NAME C2		JOB C3		BDATE C4		SALARY C5		TC
555	S	DAVID	S	MANAGER	S	02-10-67	S	$65,000	TS	TS
333	S	OMER	S	SPY	TS	12-19-55	S	$69,000	TS	TS
444	S	ALI	S	SPY	TS	02-19-65	TS	$75,000	TS	TS

Table 17 $R2_{,TS}$

A Top Secret user issues this command

UPDATE Rs

SET JOB = " SUPERVISOR"

WHERE EMPLOYEE NO. = "999"

In Novel, when one attribute is updated by a higher level, a new tuple is created and inserted in the high level relation. Table 17 will be changed and the new relation is:

EMPLOYEE NUMBER C1		NAME C2		JOB C3		BDATE C4		SALARY C5		TC
555	S	DAVID	S	MANAGER	S	02-10-67	S	$65,000	TS	TS
333	S	OMER	S	SPY	TS	12-19-55	S	$69,000	TS	TS
444	S	ALI	S	SPY	TS	02-19-65	TS	$75,000	TS	TS
999	TS	ALEX	TS	SUPERVISOR	T S	02-15-50	TS	$22,000	TS	TS

Table 17 $R2_{,TS}$

Discussion

In the previous section several multilevel SQL operations have been used in both the New technique and the Novel technique. The examples shows that INSERT and DELETE operations have the same effect on the single level base relations. On the other hand, the major different is in the UPDATE operation. In Novel, updating one attribute or more in any tuple by a high level user requires the duplication of the entire tuple and inserting it in the high single level base relation. In the New technique, updating one or two attributes requires duplicating 50 percent of attributes and inserting them in the high single level base relation. Replicating the entire tuple in Novel increase the storage overhead and increases the partition size which degrade the performance.

Storage requirements (Space Complexity)

The storage requirement is the amount of storage required to store a relation. In this thesis is the amount of storage in bytes required to store the single level base relations. The storage requirements can be calculated by using equations to find how many bytes it requires to store the MLS relation after the decomposition procedure in both the New and the Novel technique. The reason for the calculation is to perform a comparison between the New decomposition technique and the Novel technique and finds out which one requires less storage.

The following notations will be used in this section:

Du, Dc, Ds, Dts : tuples created by unclassified, confidential, secret, and top secret users.
P : Probability of a tuple to be updated by a high level user.
α : The size of primary key divided by the size of non-key attribute.

S : Size of attribute A in bytes.

N : Number of attributes in relation R.

Stn: Storage requirements of relation R using Novel Technique.

Sts: Storage requirements of relation R using SeaView Technique.

Stnt1: Storage requirements of relation R using New Technique 1.

Stnt2: Storage requirements of relation R using New Technique 2.

The storage requirements can be measured by modifying equation 1 [Mukk, 94] to make it suitable for the new technique.

$$Stn = S *(N+2*\alpha) *[Du+Dc+Ds+Dts) + P *(3Du+2Dc+Ds)] . \quad (1)$$

In equation 1 : N, Du, Dc, Ds, Dts, α are constant and therefore, the equation is a function of P. In equation 1, Mr. Mukkamala [Mukk, 94] multiply α by two because his technique requires storage for two primary keys for each tuple, one for the primary key in the partitions and the other for the primary key stored in the index file.

Since the new technique requires storage for three primary keys for each tuple (one in the index file, and two in the partitions that store the tuple), we multiply α by three. We have two new equations that apply to the new technique based on number of attributes updated; when the number of attributes updated are less than 50% equation 2 will be used, when number of attributes updated are more than 50%, equation 3 will be used.

$$Stn1 = S *(N+3*\alpha) *[Du+Dc+Ds+Dts) + (P/2) *(3Du+2Dc+Ds)] . \quad (2)$$

$$Stn2 = S *(N+3*\alpha) *[Du+Dc+Ds+Dts) + P *(3Du+2Dc+Ds)] . \quad (3)$$

Example :

The following example is borrowed from [Mukk, 94] to show the storage requirements between the new technique and the Novel technique.

Let S = 20, N= 10, α = 0.5, Du=4, Dc=3, ds=2, Dts=1.

Stn= 20 *(10+2*0.5) * (4+3+2+1) + P (3*4+2*3+2)]

= 20 *(11) * [10+20* P]

= 2200 + 4400 * P (4)

Stnt1 = 20 * (10 + 3* 0.5) * [(4+3+2+1) + (P/0.5) * (3*4+2*3+2)]

= 20 * (11.5) * [10 + (P/0.5) * 20]

= 2300 + 2300* P (5)

Stnt2 = 20 * (10 + 3* 0.5) * [(4+3+2+1) + P * (3*4+2*3+2)]

= 20 * (11.5) * [10 + P * 20]

= 2300 + 4600 * P (6)

Probability	**Novel Technique**	**New Technique 1**
P	Stn = 2200 + 4400 * P	Stnt1 = 2300 + 2300 * P
P = 1	Stn = 2200 + 4400*1 = 6600	Stnt1 = 2300 + 2300 * 1 = 4600
P = .09	Stn = 2200 + 4400*0.9 = 6160	Stnt1 = 2300 + 2300 * 0.9 = 4370
P = .08	Stn = 2200 + 4400*0.8 = 5720	Stnt1 = 2300 + 2300 * 0.8 = 4140
P = .07	Stn = 2200 + 4400*0.7 = 5280	Stnt1 = 2300 + 2300 * 0.7 = 3900
P = .06	Stn = 2200 + 4400*0.6 = 4840	Stnt1 = 2300 + 2300 * 0.6 = 3680
P = .05	Stn = 2200 + 4400*0.5 = 4400	Stnt1 = 2300 + 2300 * 0.5 = 3450
P = .04	Stn = 2200 + 4400*0.4 = 3960	Stnt1 = 2300 + 2300 * 0.4 = 3220
P = .03	Stn = 2200 + 4400*0.3 = 3520	Stnt1 = 2300 + 2300 * 0.3 = 2990
P = .02	Stn = 2200 + 4400*0.2 = 3080	Stnt1 = 2300 + 2300 * 0.2 = 2760
P = .01	Stn = 2200 + 4400*0.1 = 2640	Stnt1 = 2300 + 2300 * 0.1 = 2530
P = 0.0	Stn = 2200 + 4400*0.0 = 2200	Stnt1 = 2300 + 2300 * 0.0 = 2300

The difference in storage requirements of Novel and the New technique is expressed as the percentage difference % Δ S, where %Δ S represent the percentage increase or decrease in storage requirements when one switch from Novel to New technique.

% Δ S = (Storage for Novel - Storage for New Technique_1) * 100 / Storage for Novel

% Δ S = (Stn - Stnt1) * 100 / Stn

P = 1, % Δ S = (6600 - 4600) * 100 / 6600 = 30.30%

P = 0.9, %Δ S = (6160 - 4370) * 100 / 6160 = 29.05%

P = 0.8, %Δ S = (5720 - 4140) * 100 / 5720 = 27.62%

P = 0.7, %Δ S = (5280 - 3900) * 100 / 5280 = 26.14%

P = 0.6, %Δ S = (4840 - 3680) * 100 / 4840 = 23.97%

P = 0.5, %Δ S = (4400 - 3450) * 100 / 4400 = 21.60%

P = 0.4, %Δ S = (3960 - 3220) * 100 / 3960 = 18.67%

P = 0.3, %Δ S = (3520 - 2990) * 100 / 3520 = 15.06%

P = 0.2, %Δ S = (3080 - 2760) * 100 / 3080 = 10.39%

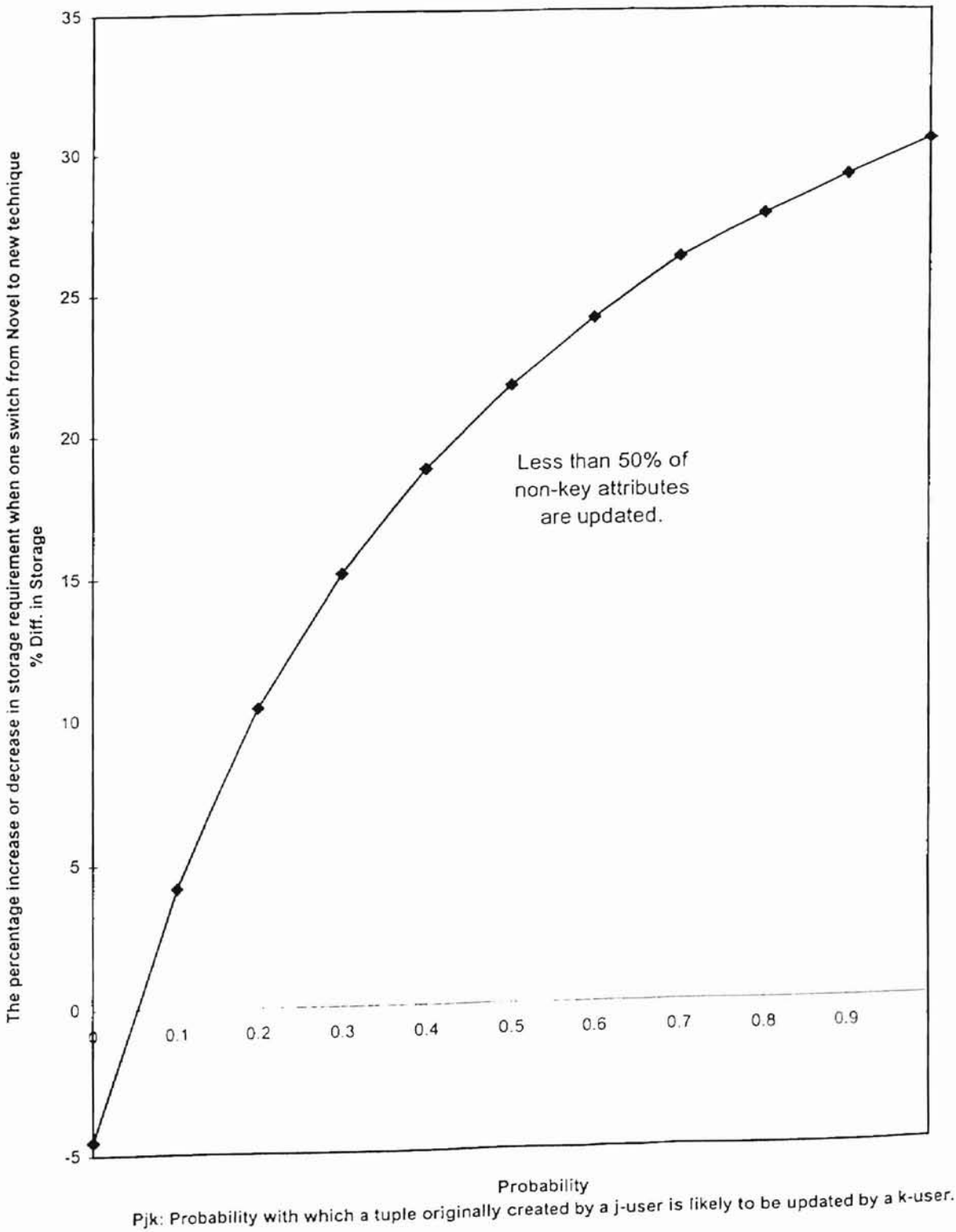
P = 0.1, %Δ S = (2640 - 2530) * 100 / 2640 = 4.17%

P = 0.0, %Δ S = (2200 - 2300) * 100 / 2200 = -4.55%

Figure 3 (page 51) shows a comparison between % Δ S and the probability of update P. When the probability of update is more than 0.1, Novel requires more storage

Figure 3

Comparison of Storage Requirements



Pjk: Probability with which a tuple originally created by a j-user is likely to be updated by a k-user.

than New technique; furthermore, the percentage difference increases by the increase of probability of update. That is the New technique requires less storage than Novel technique.

On the other hand, the worst case of the new technique is that when more than 50% of attributes are updated, then equation Stnt2 will be used.

Stnt2 = 2300 + 4600 * P

Probability	**Novel Technique**	**New Technique 2**
P	Stn = 2200 + 4400 * P	Stnt2 = 2300 + 4600 * P
P = 1	Stn = 2200 + 4400*1 = 6600	Stnt2 = 2300 + 4600 * 1 = 6900
P = .09	Stn = 2200 + 4400*0.9 = 6160	Stnt2 = 2300 + 4600 * 0.9 = 6440
P = .08	Stn = 2200 + 4400*0.8 = 5720	Stnt2 = 2300 + 4600 * 0.8 = 5980
P = .07	Stn = 2200 + 4400*0.7 = 5280	Stnt2 = 2300 + 4600 * 0.7 = 5520
P = .06	Stn = 2200 + 4400*0.6 = 4840	Stnt2 = 2300 + 4600 * 0.6 = 5060
P = .05	Stn = 2200 + 4400*0.5 = 4400	Stnt2 = 2300 + 4600 * 0.5 = 4600
P = .04	Stn = 2200 + 4400*0.4 = 3960	Stnt2 = 2300 + 4600 * 0.4 = 4140
P = .03	Stn = 2200 + 4400*0.3 = 3520	Stnt2 = 2300 + 4600 * 0.3 = 3680
P = .02	Stn = 2200 + 4400*0.2 = 3080	Stnt2 = 2300 + 4600 * 0.2 = 3220
P = .01	Stn = 2200 + 4400*0.1 = 2640	Stnt2 = 2300 + 4600 * 0.1 = 2760
P = 0.0	Stn = 2200 + 4400*0.0 = 2200	Stnt2 = 2300 + 4600 * 0.0 = 2300

As mentioned before %Δ S represent the percentage increase or decrease in storage requirements when one switch from Novel to New technique.

% Δ S = (Storage for Novel - Storage for New Technique 2) * 100 / Storage for Novel

% Δ S = (Stn - Stnt2) * 100 / Stn

P = 1, $\%\Delta S = (6600 - 6900) * 100 / 6600 = -4.545\%$

P = 0.9, $\%\Delta S = (6160 - 6440) * 100 / 6160 = -4.545\%$

P = 0.8, $\%\Delta S = (5720 - 5980) * 100 / 5720 = -4.545\%$

P = 0.7, $\%\Delta S = (5280 - 5520) * 100 / 5280 = -4.545\%$

P = 0.6, $\%\Delta S = (4840 - 5060) * 100 / 4840 = -4.545\%$

P = 0.5, $\%\Delta S = (4400 - 4600) * 100 / 4400 = -4.545\%$

P = 0.4, $\%\Delta S = (3960 - 4140) * 100 / 3960 = -4.545\%$

P = 0.3, $\%\Delta S = (3520 - 3680) * 100 / 3520 = -4.545\%$

P = 0.2, $\%\Delta S = (3080 - 3220) * 100 / 3080 = -4.545\%$

P = 0.1, $\%\Delta S = (2640 - 2760) * 100 / 2640 = -4.545\%$

P = 0.0, $\%\Delta S = (2200 - 2300) * 100 / 2200 = -4.545\%$

Figure 4 (page 55) shows a comparison between $\%\Delta S$ and the probability of update P, Figure 4 shows that the difference is constant regardless the value of probability of update, the percentage difference is -4.545%, that is Novel requires less storage than the New technique. This is the worst case in the New technique.

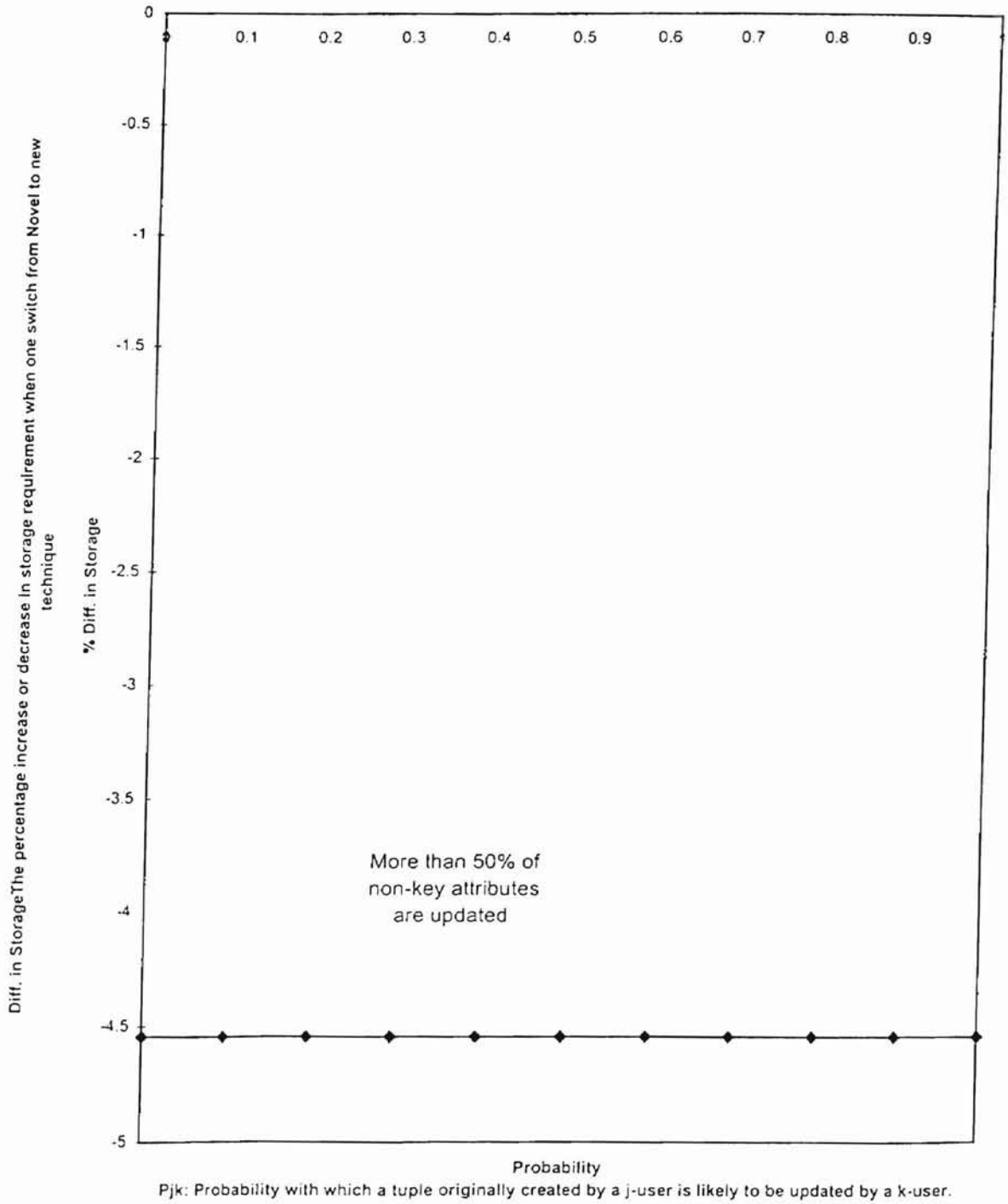
Cost of Reconstruction (Time Complexity)

The cost of reconstruction is measured in terms of the I/O cost of reading and writing the required partitions from the database. The I/O cost involved the sizes of the partitions and the size of the main memory. We assume that we have sufficient main memory to store the intermediate relations of the join operations. The I/O cost is also measured in terms of number of join and other operations involved in the decomposition and the creation of the partitions and the relations [Mukk, 94].

Since the cost of reconstruction depends on the cost of replication of the primary key and the cost of attribute key replication, it follows the same as the storage requirements, that is it depends on the probability of attributes updated by a higher level. Moreover, the

Figure 4

Comparison of Storage Requirements



Pjk: Probability with which a tuple originally created by a j-user is likely to be updated by a k-user.

cost of join operations have been discussed in detailed in the previous section. SeaView requires several join operations to reconstruct a multilevel secure relation from its single level base relations while the New technique requires only one join operation. For example, assume R has 6 attributes and 3 security level, then SeaView requires 5 join operations and 12 union operations. On the other hand, the New technique requires only one join operation and 4 union operations. That is the New technique requires less time than SeaView model in reconstructing a MLS relation from its single level base relation.

Chapter IV

Conclusion

A multilevel secure system is a system where each data item is assigned a security classification and every user a security clearance. The role of a multilevel secure database management system is to ensure that users manipulate only those data to which their clearance entitles them.. The multilevel secure relations are decomposed into a collection of single level base relations which are physically stored in the database. The primary advantage of the decomposition is to enforce mandatory controls with respect to single level base relations by Trusted Computing Base (TCB), a small part of the operating system that must always be invoked, and must be shown to perform only its intended functions. The TCB is responsible for enforcing mandatory controls with respect to single-level base relation . A new decomposition technique was developed to improve performance and storage requirements for MLS database systems.

The first part of these thesis focus on the design issues of several security models. This is followed by a detailed study of database security models by analyzing those

models and pointed out the advantages and disadvantages of each security models. The second part of this thesis analyze multilevel secure relational database security and all relevant subjects such as integrity constraints and polyinstantiation. The third part is developing a new decomposing technique that decompose a multilevel secure relation into single level base relations to be stored in the database. The fourth part is developing a recovery algorithm that recover the MLS relation from its single level base relations. And finally the fifth part is performing a comparison between the new technique and the exiting techniques based on space and time complexity.

Decomposition and Recovery techniques are fundamental issues of any security models, and these techniques are important in improving performance and storage requirement for any model. Therefore, I have studied the previous decomposition techniques and develop a new technique that improve both performance and storage requirements.

The new decomposition technique minimizes the time required to create a MLS secure relation from its single level base relations; therefore, it improves the performance of the system, performance issues has been scarified during designing of security models. The new technique has improved in general the storage requirement in comparison to other decomposition techniques because it doesn't duplicate the primary keys in each partition as in SeaView model which waste an enormous amount of storage.

The main focus of this research has been to develop a new decomposition and recovery algorithm for decomposing and recovering a MLS relation. A theoretical approach has been designed and an implementation of those techniques using a real

world relational database environment is a logical follow-up to this work. Although I have studied integrity in detail, security can be severely compromised by inference attacks. An inference attack occurs when a user can infer unauthorized information from a priori knowledge about the database and authorized query responses [Sand, 93]. It is therefore important to conduct a detailed investigation in inference-control.

References

[Bell,75] Bell, D.E. and LaPadula, L.J. " Secure Computer Systems: Mathematical foundation and models", Miter Corp. Report No. M74-244, 1975.

[Bert, 97] Bertino E., Samarati, P., Jajodia, S., " An Extended Authorization Model for Relational Databases", IEEE Transactions on Software Engineering, Vol. 9 No. 1, pages 85- 101, February 1997.

[Burns, 96] Burns, R. K. " A Comparison of Multilevel Structured Query Language Implementation" IEEE Symposium on Security and Privacy, Oakland, California, 1996, Pages 192-202.

[Caus, 90] Caustney, R. H " Factors affecting the availability of security measures in data processing system component" In Proc. 13 the National Computer Security Conference, Oct. 1990

[Brew, 89] Brewer, David, and Nash, Michael. " The Chinese Wall Security Policy" Proc. IEEE Symposium on Security and Privacy, Oakland, California, 1989, Pages 206-214.

[Chen, 95] Chen, F. , Sandhu, R. " The Semantic and Expressive Power of the MLR Data Model" IEEE Symposium on Security and Privacy, Oakland, California, 1995, Page 128-142.

[Date, 90] Date, C. J. " An Introduction to Database Systems" Addison Wesley, volume 1, fifth edition, 1990

[Denn,88] Denning, Dorothy, Lunt, T.F. " The SeaView Security Model" Proc. IEEE Symposium on Security and Privacy, Oakland, California, 1988, Pages 218-233.

[Dosh, 92] Doshi V. M. , Jajodia S. " Referential Integrity In Multilevel Secure Database Management Systems" Database Security, Gable, G. G. and Caelli, W.J.(Editors), Elsevier Science Publisher B.(North Hollond) 1992.

[Fole, 96] Foley, S. N., Gong, L. , Qian, X., " A security Model of Dynamic Labeling Providing a Tiered Approach to Verification" . IEEE Symposium on Security and Privacy, Oakland, California, 1996, Pages 142-153.

[Haig, 91] Haigh, J.T., O'Brien, R. C. and Thomsen, D. J. " The LDV Secure Relational DBMS Model. " Database Security IV : Status and Prospects, Jajodia and C. E. Landwehr (editors), North Holland, 1991, P 265-279.

[Hamm,93] Hammonds, G. L. " Confidentiality, Integrity, Assured Service: Trying Security All Together", ACM SIGSAC, 1993, Page 48-52.

[Hwan, 97] Hwang, M. S., Yang, W., " Multilevel secure database encryption with subkeys" Data & Knowledge Engineering, V22, 1997, Pages 117-131.

[Jajo, 91a] Jajodia, S. and Sandhu, R. S. " A Novel Decomposition of Multilevel Relations Into Single-level Relations." Proc. IEEE Symposium on Security and Privacy, Oakland, California, May 1991, Pages 300-313.

[Jajo, 91b] Jajodia, S. and Sandhu, R. S. " Toward a Multilevel secure Relational Data Model", ACM SIGMOD 1991 , V 20, Page 50-59

[Jajo, 90] Jajodia, S. and Sandhu, R. S. " Polyinstantiation Integrity in Multilevel Relations", Proc. IEEE Symposium on Security and Privacy, Oakland, California, 1990, Pages 104- 115.

[Kang, 95] Kang, I. E. " Concurrency Control for Federated Multilevel Secure Database Systems", IEEE Symposium on Security and Privacy, Oakland, California, 1995, Pages 118-135.

[Lin, 93] Lin, T. Y. " Bell and LaPadula Axioms: A New Paradigm for an Old Model" ACM SIG Security, Audit and control review. 1993, P 82-92.

[Liu, 87] Liu, L. C. "Elements of Discrete Mathematics" , McGraw- Hill International editions, second edition, 1987.

[Lunt, 93] Lunt, T. F. , Boucher, P. K. " The SeaView Prototype: Project Summary" 17th National Computer Security Conference, Oct. 11-14,1994. P 88-102.

[Lunt, 88] Lunt, T. F., " A Near-Term Design For The SeaView Multilevel database System" "Proc. IEEE Symposium on Security and Privacy, Oakland, California, May 1988, Pages 234-244.

[Lunt, 90] Lunt, T. F., Denning, D. E, Schell, R., Heeman, M. and Shockley, W. R. " The SeaView Security Model. " IEEE Transactions on Software Engineering, Vol. 16 No. 6, pages 593-607, June 1990.

[Mark, 96] Marks, D., G. " Inference in MLS Database Systems" IEEE Transaction On Knowledge & Data Engineering, V8, No. 1, February 1996.

[Mukk, 94] Mukkamala, R. and Jajodia S. " A Performance comparison of two decomposition techniques for multilevel secure database systems. "Database Security VII, Status and Prospects, T. F. Keefe and C. E. Landwehr (editors) North Holland, 1994, pages 199-215.

[Pern, 91] Pernul, G. and Luef, G. " A Multilevel Secure Relational Data Model Based on Views." Proc. IEEE Symposium on Security and Privacy, Oakland, California, May 1991, Pages 166-177.

[Pern, 91a] Pernul, G. " Security Constraint Processing During Multilevel Secure Database Design" IEEE Symposium on Security and Privacy, Oakland California, 1992, Pages 75- 84

[Pern, 94] Pernul, G., Quirchmayr, G., " Organizing MLS Databases from a Data Modeling Point of View" Proc. IEEE Symposium on Security and Privacy, Oakland, California, 1994, Pages 96- 105.

[Pess, 97] Pesati, V. R., Keefe, T. F., Pal, S. " The Design and Implementation of a Multilevel Secure Log Manager" Proc. IEEE Symposium on Security and Privacy, Oakland, California, 1997, Pages 55- 64.

[Qian, 97] Qian, X., Lunt, T. F. " A Semantic Framework of the Multilevel Secure Relational Model" IEEE Transactions On Knowledge & Data Engineering, V 9, No. 2, March 1997.

[Qian, 96] Qian, X., Lunt, T. F. " A MAC Policy Framework for Multilevel Relational Databases", IEEE Transactions on Knowledge and Data Engineering, V8, No. 1, 1996.

[Qian,96a] Qian, X., " View-Based Access Control with High Assurance", IEEE Symposium on Security and Privacy, Oakland, California, 1996, 85-93.

[Qian, 93] Qian, X., Lunt, T. F " Tuple-level vs. element-level classification" Database Security VI : Status and Prospects, B. M. Thuraisingham and C. E. Landwehr (editors), North Holland, 1993, Page 301-314.

[Sand, 93]Sandhu, Ravi, " Lattice-Based Access Control Models" Proc. IEEE Symposium on Security and Privacy, Oakland, California, 1993, P 9-18.

[Sand, 92]Sandhu, R. S. and Jajodia, S. " Polyinstantiation for cover stories" Proc. European Symposium on Research in Computer Security, Toulouse, France, November 1992, Pages 307-328.

[Sand, 90] Sandhu, Ravi, Jajodia, S. " A New Polyinstantiation Integrity constraint For Multilevel Relations" IEEE Symposium on Security and Privacy, Oakland, California,1990, Page 159-165.

[Silv, 95] Silvana C. , Mariagrazia F. , Giancarlo, M., Pierangela S. , Database Security , Addison-Wesley.

[Stac, 90] Stachour, Paul D. , Thuraisingham, B. " Design of LDV : A Multilevel Secure Relational Database Management System", IEEE Transactions on Knowledge and Data Engineering, V2, No. 2, 1990.

[Thur, 95]Thuraisingham, B. " Multilevel security for information retrieval systems II", Information & Management, V28, 1995. Pages 46-61.

VITA

Maher Abdel Fatteh Abdin

Candidate for the Degree of

Master of Science

Thesis: A NEW DECOMPOSITION TECHNIQUE FOR DECOMPOSING A MULTILEVEL SECURE RELATION INTO SINGLE-LEVEL RELATIONS.

Major Field : Computer Science

Biographical:

Personal Data: Born in Jerusalem on March 21, 1967.

Education: Graduated from Islamic Orphanage School, Jerusalem, in May 1985; received Bachelor of science degree in Computer Science from Oklahoma State University, Stillwater, Oklahoma in December 1991. Completed the requirements for the Master of Science degree with a major in Computer Science at Oklahoma State University in December 1998.