数据库安全模型的研究*)

张 剡 夏 辉 柏文阳

(软件新技术国家重点实验室 南京210093) (南京大学计算机科学与技术系 南京210093)

摘 要 在现有安全模型的基础上提出了NDMAC模型。该模型对主客体进行了明确的定义和分类,并采用最细客体粒度为属性和元组级的方案。该模型提出了隶属完整性规则、实体完整性规则和推理完整型规则,并给出了六种操作的处理策略。此外模型引入了同步约束、互斥约束和推理约束,并通过引入特权机制增强该模型的灵活性。

关键词 数据库安全,强制访问控制,多级安全模型

Research on Database Security Model

ZHANG Yan XIA Hui BAI Wen-Yang

(State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093)

(Department of Computer Science and Technology, Nanjing University, Nanjing 210093)

Abstract This paper presents a database security model named NDMAC, which is based on existing security models. After a brief introduction of current work, the database security model is presented and interpreted in detail. The main contributions of the model include: explicit definition and classification of subjects and objects; the finest granularity of objects is attribute-level and tuple-level; subjection integrity rule makes security levels of related objects satisfy definite relation; entity integrity rule ensures that the primary key of a visible tuple is entirely visible; inference integrity rule and inference restriction prevent information leakage caused by functional dependency; the disposal of six kinds of operation makes information flow from some subjects to others, which are associated with equal or higher security levels; the privilege mechanism makes the model more flexible.

Keywords Database security, Mandatory access control, Multilevel security model

1 引言

信息安全已经成为当前研究的热点课题,作为信息系统核心的数据库的安全尤其是网络环境下的数据库的安全则成为重中之重。目前,国内大部分企事业单位,包括国家的一些关键部门,大都使用国外进口的数据库产品,如ORACLE、DB2、SYBASE等。如何提高这类数据库产品的安全性则成为急需解决的问题。这种情况下,增强型安全数据库应运而生。NDMAC模型正是为了实现增强型安全数据库而设计的。

本文第2节简要介绍主要的安全模型并分析其优缺点,第3节详细介绍了NDMAC模型,第4节对NDMAC模型进行分析,最后进行小结和展望未来的工作。

2 主要安全模型分析

在数据库安全发展中,HRU存取矩阵模型、Bell-La Padula模型、Jajodia-Sandhu模型起着里程碑式的作用。由于NDMAC模型是用于实现增强型安全数据库,自主访问控制(DAC)将交由底层数据库管理系统实现,因此这里主要分析Bell-La Padula模型、Jajodia-Sandhu模型的强制访问控制(MAC)部分。

2.1 Bell-La Padula(BLP)模型

BLP模型是目前应用最广泛的MAC模型之一,当前很多MAC模型都是直接或者间接对BLP模型进行改进。BLP模型是一个状态机模型,它形式化地定义了系统状态及状态间的转换规则,引入安全级的概念,并制定了一组安全规则(或定理),进而构成对系统状态的限制和状态转换规则的约束。

在该模型中提出了主体和客体的由层次安全密级与非层次范畴组成的安全等级的概念,系统的元素分为主体与客体两部分,其中主体为系统中的主动元素,可以对客体作操作,客体为系统中被动元素,可以接受来自外界的操作。BLP模型对系统中的每个主体和客体分配一个安全级别,用偏序关系的格理论建立主体访问客体的数学模型。它的主要安全规则可以归纳为下读上写(No Read Up-No Write Down)规则,这样保证信息从低安全级主体向高安全级主体流动。

2.2 Jajodia-Sandhu(JS)模型

JS模型是以BLP模型为基础,它对标准的关系模型进行了扩充,借鉴了SeaView模型多级关系和多实例的概念。JS模型引入了基于多级关系和多实例的完整性规则,并定义了存取多级关系的处理策略。此外JS模型提供了由多级逻辑关系分解成单级物理关系的分解算法以及由单级物理关系重新生成多级逻辑关系的算法。

完整性规则包括实体完整性、空值完整性、实例间完整性和多实例完整性。存取多级关系策略用于主体对客体进行读、写(包括插入、修改、删除)操作,这里最关键的部分是对多实例的处理,其中插入、修改操作可能导致多实例的产生,读、删除操作也会牵涉到多实例。

2.3 BLP模型和JS模型分析

BLP模型引入了安全级的概念,并提出了简单安全规则、星规则和自主安全规则,模型通过分析证明一个初始状态

*)国家"八六三"高技术研究发展计划基金项目(项目编号2002AA141091)。张 剡 硕士,研究方向:数据库安全、数据挖掘。夏 辉 硕士,研究方向:数据库安全、移动数据库。柏文阳 副教授,研究方向:数据库安全、数据挖掘、数据仓库。

安全的系统只要在满足上述三个规则的条件下进行操作，则进行操作后的系统状态仍然是安全的。该模型中提出的安全级的概念成为所有强制访问控制模型的基石，而从简单安全规则和星规则总结出来的"不向上读，不向下写"规则也成为其它所有强制访问控制模型的指导思想，自主安全规则明确了强制访问控制同自主访问控制的关系。BLP 模型通过安全级和强制访问控制策略解决了"特洛伊木马"问题。但是 BLP 模型是针对操作系统安全提出的，它在很多方面无法满足数据库安全的需要。首先 BLP 模型对强制访问控制的客体粒度没有明确的定义；其次它没有考虑关联客体安全级之间的层次关系；BLP 模型仅考虑了读操作、写操作、调用操作和添加操作，因此 BLP 模型定义的操作无法满足数据库安全的需要；BLP 模型中未考虑隐蔽信道问题和推理控制问题，而达到较高安全级的安全数据库要求处理此类问题。JS 模型正是在这种情况下产生的。

JS 模型是以 BLP 模型为基础而提出的一种适用于关系数据库系统的强制访问控制模型，它提供了元素一级的安全保护。该模型借鉴了 BLP 模型中的"不向上读，不向下写"思想，保证了信息从低安全级主体向高安全级主体的流动。同时模型使用了多级关系、关系实例和多实例的概念，通过完整性规则和操作处理策略保证了数据操作的影响范围局限于安全级支配或者等于主体安全级的关系实例。JS 模型明确了客体粒度达到元素级，同时给出了读取操作、插入操作、修改操作、删除操作的处理策略，其中插入操作、修改操作、删除操作可能产生多实例。但是 JS 模型同样未考虑连接数据库操作和调用存储过程/存储函数操作的处理策略；同时多实例的引入固然可以解决部分隐蔽信道问题，但是多实例同数据库的基本准则相违背，因为多实例可能引起数据的不一致，这样会产生误导作用；JS 模型同样没有考虑关联客体安全级之间的层次关系；该模型同样没有考虑推理控制问题；此外元素级安全会极大地影响采用该模型的系统的性能。

由于我们的研究内容是增强型安全数据库，这决定了我们对安全模型有了特别的要求。现有安全模型无法满足我们的要求，因此我们提出了 NDMAC 模型。

3 NDMAC 模型

3.1 基本概念

(1)客体　模型中的客体是增强型安全数据库中数据的载体，客体是受保护的对象，本模型中考虑的客体包括以下几类：数据库、表、视图、存储过程/函数、属性、元组。模型中的客体分为三层，第一层为数据库，第二层为表、视图、存储过程/函数，第三层为属性和元组，这样客体集合形成了一个三层的隶属层次树，隶属关系我们记为$\lhd$。对于任意两个客体 O_1、O_2，$O_1 \lhd O_2$表示 O_2在隶属层次树中是 O_1的父亲节点，例如客体 O_2代表一个数据库，客体 O_1代表一张表，$O_1 \lhd O_2$表示表 O_1是数据库 O_2中的一张表。

(2)主体　模型中的主体是增强型安全数据库的用户，主要包括三大类：安全管理员、审计管理员和数据库用户。安全管理员主要负责管理强制访问控制部分和推理控制部分；审计管理员主要负责管理审计部分；数据库用户有普通数据库用户和数据库管理员两种角色，普通数据库用户角色使得用户可以进行数据访问操作，而数据库管理员角色使得用户可以进行自主访问控制相关管理工作，任一数据库用户一定包含普通数据库用户角色，而数据库管理员角色仅有特定的数据库用户方可拥有。这三类用户职责是无交集的，它们只能进行各自职权范围内的操作。这里我们使用了可信主体的概念，所谓可信主体就是可以依靠，不会损害系统安全性的主体。这里安全管理员和审计管理员为可信主体，数据库用户在进行自主访问控制相关的管理工作时是可信主体，而数据库用户在进行数据访问操作时为不可信主体，仅有不可信主体和安全级相关联。模型中的安全规则和操作处理策略针对的都是不可信主体，可信主体不进行相关的检查，但是要求所有可信主体进行过的操作都是可以追查的。

(3)安全级　每个不可信主体或者客体都和一个安全级关联起来，安全级反映了不可信主体的置信度，也反映了客体的重要性。安全级定义同 BLP 模型中的安全级定义是一致的，安全级由一个二元组 $L=(S, C)$表示。S 是密级，C 为范畴的集合。安全级别的集合形成一个满足偏序关系的格，此偏序关系称为支配($\geqslant$)关系。以下是它的形式化定义：

定义 $L_1=(S_1, C_1)$，$L_2=(S_2, C_2)$。

L_1支配 L_2成立，当且仅当：①$S_1 \geqslant S_2$，②$C_1 \supseteq C_2$，记为 $L_1 \geqslant L_2$。

$L_1=L_2$当且仅当：①$S_1=S_2$，②$C_1=C_2$

$L_1>L_2$，当且仅当：①$S_1>S_2$，②$C_1 \supseteq C_2$

$L_1<L_2$，当且仅当：①$S_1<S_2$，②$C_1 \subset C_2$

$L_1 \leqslant L_2$，当且仅当：①$S_1 \leqslant S_2$，②$C_1 \subseteq C_2$

如果 L_1、L_2既不满足 $L_1 \geqslant L_2$，也不满足 $L_1 \leqslant L_2$，则称 L_1、L_2不可比。

模型中密级是全序集合中的任意一个元素，模型默认的全序集合是{TS(绝密)，S(机密)，C(秘密)，U(公开)}，该全序集合中各元素顺序为 $TS>S>C>U$。模型提供安全命令支持用户自定义全序集合。

(4)多级关系和关系实例　NDMAC 模型中多级关系和关系实例的定义同 JS 模型不尽相同。在本模型中多级的关系的形式化定义如下：$R(A_1, c_1, \cdots, A_n, c_n, TC)(cr)$，这里 A_i 是定义在域 D_i 上的数据属性，c_i 是 A_i 的安全级且 c_i 是个常量，TC 是元组的安全级，常量 cr 是关系 R 的安全级。

安全级为 c 的关系实例是形如$(a_{x1}, c_{x1}, \cdots, a_{xn}, c_{xn}, tc)$的元组集合，其中 $a_{xn} \in D_{xn}$，$c_{xn} \leqslant c$，$tc \leqslant c$。我们可以按如下方法从多级关系 R 产生安全级为 c 的关系实例 $R_c(A_{x1}, c_{x1}, \cdots, A_{xn}, c_{xn}, TC')(c)$：

1)生成临时关系 R_{temp}，要求对于每个满足 $t \in R$ 并且 $t[TC] \leqslant c$ 的元组 t 有 $t \in R_{temp}$。

2)生成关系实例 R_c，要求当 $c_i \leqslant c$ 时：$A_i \in \{A_{x1}, \cdots, A_{xn}\}$ 并且 $t'[A_i]=t[A_i]$；否则 $A_i \notin \{A_{x1}, \cdots, A_{xn}\}$，其中 t' 是元组 t 在 R_c 中的对应元组。

第一步是对 R 根据安全级 c 进行水平过滤，剩下元组安全级支配于 c 的元组，第二步是对第一步过滤结果进行垂直过滤，使得剩下的属性列的安全级别支配于 c。以上两步也可以是先进行垂直过滤再进行水平过滤，其结果是一致的。

3.2 安全规则

(1)隶属完整性规则　安全数据库满足隶属完整性规则，当且仅当，对于任意两个客体 O_1、O_2，其对应的安全级 L_{O1}、L_{O2}满足：

$O_1 \lhd O_2 L_{O1} \geqslant L_{O2}$

该规则是对隶属层次树中客体安全级之间的支配关系的定义，该规则要求隶属层次树中的任一节点安全级支配其父节点的安全级。例如要求 $cd \leqslant cr$，其中 cr 是多级关系 R 的安

全级，cd 是关系 R 所在数据库的安全级；再如要求 $c_i \leqslant c_i$，$c_i \leqslant t[TC]$，其中 c_i 是 R 中第 i 个属性的安全级，$t[TC]$ 是 R 中元组 t 的安全级。

(2)实体完整性规则　设 AK 是定义在关系模式 R 上的主键，多级关系 R 满足实体完整性，当且仅当，对 R 的任一关系实例 R_c 中的每一个 $t \in R_c$ 的元组 t 有：

1) $A_i \in AK \Rightarrow t[A_i] \neq null$；

2) $A_i, A_j \in AK \Rightarrow c_i = c_j$，即组成 AK 的各属性的安全级相同；

3) $A_i \notin AK \Rightarrow c_i \geqslant c_{AK}$（$c_{AK}$ 为 AK 的安全级）；

4) $t[TC] \geqslant c_{AK}$（c_{AK} 为 AK 的安全级）。

这一完整性规则是对标准关系中的实体完整性的扩展，使其能处理安全级。第一个条件和标准关系模型中的实体完整性类似；第二个条件保证了所有组成主键的属性的安全级相同；第三个条件和第四个条件保证了主键的安全级支配于所有属性的安全级，并且元组的安全级支配主键的安全级，这样使得主键一定可见。该规则是 JS 模型中的实体完整性规则在 NDMAC 模型中的扩展。

(3)推理完整性规则　多级关系 R 满足推理完整性，当且仅当，对 R 中属性 $A_{i1}, \cdots, A_{in}, A_j \notin AK$ 有：

$$A_{i1}, \cdots, A_{in} \rightarrow A_j \Rightarrow c_{im} \geqslant c_j$$

其中 $c_{im}\{c_{i1}, \cdots, c_{in}\}$，这里 $A_{i1}, \cdots, A_{in}$ 是组成函数依赖的决定因素的属性，而属性 A_j 是函数依赖的被决定因素。该规则要求所有非主键参与的函数依赖集中，组成函数依赖的决定因素的属性中，至少有一个属性的安全级支配函数依赖的被决定因素的安全级。满足推理完整性规则的多级关系不会产生函数依赖带来的隐蔽信道。

3.3　操作处理策略

这里讨论的操作是指对安全数据库进行的数据访问操作。NDMAC 模型要求安全数据库必须满足安全规则中的隶属完整性规则，同时要求所有的多级关系满足实体完整性规则，此外要求安全数据库能够提示使得多级关系不满足推理完整性规则的函数依赖。这里的操作主要分为以下六种操作：连接数据库、查询、修改、删除、插入、调用存储过程/函数。下面我们将先分别讨论这六种操作在主体无特权情况下的处理策略，然后将讨论主体有特权情况下的操作处理策略。

(1)连接操作处理策略　连接操作的语句格式如下所示：

CONNECT sdatabasename

这里 sdatabasename 是数据库名，安全级为 c 的主体可以执行连接操作，当且仅当 $cd \leqslant c$，cd 是 sdatabasename 对应的数据库的安全级。

(2)查询操作处理策略　查询操作的语句格式如下所示：

SELECT A_{x1}, A_{x2}, …, A_{xn} FROM $R1, R2, \cdots, Rm$ [WHERE p]

这里 p 是谓词表达式，$R1, R2, \cdots, Rm$ 表示关系，A_{x1}, A_{x2}, …, A_{xn} 表示属性。

在介绍查询操作处理策略前我们先介绍三个约束：

1)同步约束：同步约束可以用一个三元组 $SYN=(O1, O2, SL)$ 表示，这里 $O1$、$O2$ 表示两个不同的客体，SL 代表一个安全级。同步约束要求客体 $O1$、$O2$ 在安全级支配于 SL 的主体执行的查询语句中，要么 $O1$、$O2$ 都出现，要么都不出现。

2)互斥约束：互斥约束可以用一个三元组 $MUT=(O1, O2, SL)$ 表示，这里 $O1$、$O2$ 表示两个不同的客体，SL 代表一个安全级。互斥约束要求客体 $O1$、$O2$ 在安全级支配于 SL 的主体执行的查询语句中，$O1$、$O2$ 不能同时出现。

3)推理约束：推理约束可以用一个二元组 $FD=(FFDSET, BFD)$ 表示，其中 $FFDSET$ 为属性的集合，它表示导致多级关系不满足推理完整性的函数依赖的决定因素，BFD 为一个属性，表示函数依赖的被决定因素。推理约束要求当使得多级关系不满足推理完整性的某个函数依赖的决定因素都出现在安全级为 c 的主体执行的查询语句中，而被决定因素未出现在该主体执行的查询语句中时，主体的安全级 c 应该支配组成决定因素的属性的安全级和被决定因素的安全级的最小上界。

假设安全级别为 c 的主体对多级关系 R 执行查询操作，A_{x1}, A_{x2}, …, A_{xn} 为查询操作涉及的属性，A_{c1}, A_{c2}, …, A_{cm} 为 R 的关系实例 Rc 的属性，t 为查询结果集合中的任意一条元组，当且仅当满足下列条件时查询操作被允许：

①对于每个 $A_{xn} \in \{A_{x1}, A_{x2}, \cdots, A_{xn}\}$ 有 $A_{xn} \in \{A_{c1}, A_{c2}, \cdots, A_{cm}\}$；

②$t[TC] \leqslant c$；

③查询操作满足同步约束、互斥约束和推理约束。

(3)插入操作处理策略　插入操作的语句格式如下所示：

INSERT INTO $R[A_{x1}[, A_{xn}]\cdots]$ VALUES $(a_{x1}[, a_{xn}]\cdots)$

设 t 是要插入的元组，AK 为关键字集合，c_{AK} 是关键字的安全级，c 为执行插入操作的主体的安全级，当且仅当满足下列条件时，插入操作被接受：

①对于每个 $A_{xn} \in AK$，$t[A_{xn}] \neq null$；

②对于关系中的现有任一元组 t'，$t'[AK] \neq t[AK]$；

③$t[TC] = lub(c_{AK}, c)$，这里 $lub(c_{AK}, c)$ 是 c_{AK} 和 c 最小上界。

这里我们可以发现插入操作要求新插入元组的关键字不含空值，并且关键字是唯一的，此外要求新插入元组安全级等于关键字安全级和主体安全级的最小上界。

(4)修改操作处理策略　修改操作的语句格式如下所示：

UPDATE R

SET $A_{x1}=s_{x1}[, A_{xn}=s_{xn}\cdots]$

[WHERE p]

这里 p 是谓词表达式，表示需要修改的元组的属性取值需要满足的条件。修改只能针对属性的数据值，修改后的元组的安全级等于主体的安全级。设 t 是满足修改条件的任一元组，t' 为 t 在修改后对应的元组，$\{A_{x1}, \cdots, A_{xn}\}$ 为修改操作涉及的属性，AK 为关键字集合，c_{AK} 是关键字的安全级，c 为执行修改操作的主体的安全级，当且仅当满足下列条件时，修改操作被接受：

①对于每个 $A_{xn} \in \{A_{x1}, \cdots, A_{xn}\}$，$c_{xn} \leqslant c$；

② $t[TC] \leqslant c$，$t'[TC] = c$。

我们可以发现修改操作针对安全级为 c 的关系实例，并且将被修改元组安全级提升为 c。

(5)删除操作处理策略　删除操作的语句格式如下所示：

DELETE FROM R [WHERE p]

这里 p 是谓词表达式，表示需要删除的元组的属性取值需要满足的条件。设 t 是满足删除条件的任一元组，$\{A_1, \cdots, A_n\}$ 为关系 R 的属性的集合，c 为执行修改操作的主体的安全级，当且仅当满足下列条件时，删除操作被接受：

①对于每个 $A_i \in \{A_1, \cdots, A_n\}$，$c_i \leqslant c$；

(下转第149页)

对于每组输入数据的算法平均运行时间。从图6中可以看出，随着最大深度的增加，平均运行时间逐渐降低。而且，输入的模式树的节点数越多，运行时间降低速度越大。

5.2 不同节点数时的优化算法的运行时间比较

我们令输入的模式树最大深度恒为15。那么，节点个数与叶节点数大体上成比例，节点个数也就与 $\max\{m,n\}$ 成比例。由于 k 是常数，优化算法的运行时间 $O(k\times\max\{m,n\}) = O(\max\{m,n\})$。在图6中，横轴代表输入模式树的节点个数，纵轴代表算法运行时间。随着节点数的增加，运行时间线性增长。图中的实验结果验证了我们的分析。

结束语 在基于查询的 XML 缓存替换策略的应用背景下，本文研究了 $XP^{\{[],//,*\}}$ 的包含问题。特别地，我们的研究对象是一类特殊的模式树。我们提出一个正确完整的 PTIME 算法，以判定这类模式树的包含问题。而且，从理论和实验的角度分析了算法的性能。实验结果表明，即使输入数据量很大时，算法也是有效的。目前，我们在研究当 XML 树有递归结构时的包含问题。这种情况下，path 中的节点可以重复，那么 paths 的包含算法运行时间达到 O(n ^ 2)。

参 考 文 献

1 W3C. XPath 1.0: XML Path Language. http://www.w3.org/TR/xpath, Nov. 1999
2 W3C. XLink 1.0: XML Linking Language. http://www.w3.org/TR/xlink/, June 2001
3 W3C. XPointer: XML Pointer Language. http://www.w3.org/TR/xptr/, Aug. 2002
4 W3C. XQuery 1.0: An XML Query Language. http://www.w3.org/TR/xquery/, Nov. 2003
5 Chen L, Rundensteiner E A. A Fine-Grained Replacement Strategy for XML Query Cache. In WIDW, MeLean, Virginia, 2002
6 Yannakakis M. Algorithm for acyclic database scheme. In: Morgan Kaufman pubs. (Los Altos CA), Zaniolo and Delovel eds. Proc. of the 7th Conf. on Very Large Databases, 1981
7 Amer-Yahia S, choo S, Lakshmanan L V S, Srivastava D. Minimization of tree pattern queries. SIGMOD, 2001
8 Milo T, Suciu D. Index structures for path expressions. In ICDT, 1999. 277～295
9 Miklau G, Suciu D. Containment and Equivalence for an XPath Fragment. In: Proc. of the 21st ACM SIGMOD-SIGART Symp. on Principles of Database Systems(PODS), Madison, Wisconsin, USA, June 2002
10 Cormen T, Leiserson C, Rivest R, Stein C. Introduction to Algorithm. Second Edition, 2001

(上接第103页)

② $t[TC]\leqslant c$。

这里我们可以发现删除操作仅针对那些主体可以看到所有属性的元组。

(6)调用操作处理策略 调用操作的语句格式如下所示：

EXECUTE funname([*para*1[, *para*2]…])

这里 funname 是存储过程/函数的名字，*para*1、*para*2为存储过程/函数的参数。设 cf 是存储过程/函数的安全级，c 为执行调用操作的主体的安全级，当且仅当 $cf=c$ 时，调用操作被接受。

(7)特权情况操作处理策略 上面讨论的六种处理策略均为无特权情况下的处理策略，我们这里还要讨论各种操作在有特权情况下的处理策略。特权可以用一个五元组表示 $P = (Subject, Object, OPType, BTime, Etime)$，其中 *Subject*、*Object*、*OPType*、*BTime*、*Etime* 分别表示主体、客体、操作类型、起始时间、结束时间，这里客体的粒度最细为表/视图或者存储函数/过程。特权情况处理策略可以理解为当主体有相应特权的情况下，无论主体对特权中指定的客体的操作是否满足对应操作处理策略，该操作都被允许。

4 NDMAC 模型分析

NDMAC 模型是在 BLP 模型和 JS 模型的基础上发展起来的，它是为增强型安全数据库而设计的。在增强方式下，自主访问控制由底层数据库负责，因此 NDMAC 模型中没有自主访问控制相关的描述。这样模型考虑了底层数据库的安全功能，使得 TCB 尽可能小。粒度选择也是 NDMAC 模型的一个重点。模型中明确了客体的最细粒度是属性/元组级。其原因是：我们研究目标是 B2安全级安全，这决定了客体粒度要至少达到属性/元组级；而在增强方式下，元素级控制粒度下的资源消耗和效率损失远远大于属性/元组级粒度下的资源消耗和效率损失，这样为了提高系统性能，控制粒度要尽可能粗。

很多多级关系安全模型通过引入多实例来处理隐蔽信道问题。多实例在带来更高的安全性的同时，也会引起数据的不一致，这样可能会对主体产生误导作用。多实例虽然是现实世界的某些现象的反映，但是总体而言它还是弊大于利，因此 NDMAC 模型未采用多实例。但是模型考虑到推理控制引起的隐蔽信道问题，因此模型中通过引入推理完整性规则来局部消除函数依赖导致的隐蔽信道。由于推理完整性并非强制实现的，模型在查询操作处理策略中通过自动生成的推理约束来进一步消除函数依赖导致的隐蔽信道。

NDMAC 模型扩充了现有安全模型的客体类，明确了该模型中的六种客体：数据库、表、视图、存储过程/函数、属性和元组。此外 NDMAC 模型通过引入隶属完整性规则，使得关联客体的安全级满足一定偏序关系。模型扩充了现有安全模型讨论的操作的范围，模型中讨论了查询、插入、修改、删除、连接、调用这六种操作的处理策略。模型借鉴 JS 模型中的实体完整性，定义了自己的实体完整性规则，使得元组可见时，关键字一定可见。此外模型引入了特权机制以处理特殊情况的需要，这样增加了模型的灵活性和实用性。

结束语 论文的研究成果被应用于国家863计划信息安全领域的研究课题中，实现了一个基于 ORACLE 数据库的安全增强器，其强制访问控制功能已基本达到了 B2级安全要求。如何达到更细的客体粒度而不严重影响系统效率、数据库系统的入侵检测和对抗将成为论文进一步的研究重点。

参 考 文 献

1 Bell D E, Lapadula L J. Secure Computer Systems: Unified Exposition and Multics Interpretation, MTR-2997, MITRE, March 1976
2 Denning. The SeaView Security Model. IEEE Symposium on Research in Security and Privacy, IEEE Computer Society Press, Los Alamitos, CA, 1988. 218～233
3 Jajodia S, Sandhu R. Toward a Multilevel Secure Relational Data Model, Information Security: An Integrated Collection of Essays: IEEE Computer Society Press, 1995. 461～491
4 Sandhu R, Chen F. The Multilevel Relational (MLR) Data Mode. ACM Transactions on Information and System Security, 1998, 1(1)