江苏大学

硕士学位论文

基于组合访问控制的安全数据库设计

姓名：邵学军

申请学位级别：硕士

专业：计算机应用

指导教师：鞠时光

20040901

摘　要

安全对象关系数据库是当前信息安全研究的一个重要分支，具有广泛的应用前景。该领域的研究具有强烈地域性和保密性，信息技术发达国家对我国一直施行尖端安全产品禁止输出策略，数据库安全产品亦在其列，因此，研究和开发自主的安全数据库产品是进行自主信息保护的一个重要手段。

本文就安全对象关系数据库进行了深入的研究，对安全对象关系数据库的安全策略、安全模型、安全设计和安全数据库的实现进行了深入的讨论，提供了一个较为完整的逻辑设计方案，并在此基础上实现安全对象关系数据库系统——VISTA。

本文针对传统的安全模型进行分析和改进，提出了一种新的安全模型 TDM，并设计了该模型相应的规则组。TDM 安全规则从安全定义、数据安全访问、数据完整性、冲突协调四个方面对 TDM 安全模型进行了严格的定义，为模型的实现提供了依据。同时，通过 TDM 模型与传统安全模型的兼容性论证，说明了 TDM 模型的可行性和合理性。

本文简要介绍了项目组设计开发的安全对象关系数据库系统——VISTA。从安全存储机制、安全数据模式、安全访问和审计设计四个方面，对 VISTA 的设计方案进行了阐述，首次提出了可组合安全访问控制方案，根据具体安全访问控制需求，对自主访问控制、强制访问控制和角色访问控制三种传统的访问控制方案进行合理的改进和设计，使之可以自由组合，以适合不同安全强度的实际应用的需要。

本工作在理论上具备以下创新点：

⑴提出了新的安全模型——TDM；

⑵第一次提出了可组合安全访问控制策略；

⑶对传统的自主安全访问控制和角色访问控制进行了改进，增强了其安全控制的约束，在自主安全访问控制中增加了有效时间域，将角色域分解为业务域和职责域，使得访问策略更贴近和适合实际需要。

关键词：安全模型，安全规则，数据库管理系统，访问控制，审计，安全存储

ABSTRACT

The security object-relational datebase is an important branch of the present study of information security. It has broad application foreground .The study in this region has strong character of district and secrecy. The countries with developed informative technology have been prohibiting from exporting the advanced security products to China, including the security datebase products. Thereby it is an important mean to study and develop the security datebase products for protecting our self-determined information.

In this thesis it carries through an embedded study to the datebase security products .It also has a thoroughgoing discussion about the security policy, security model, security design and security datebase .It provides a relatively complete logical designing blue print and realizes the security object-relational datebase system--VISTA .

This thesis aims at analyzing and improving the traditional security model and comes out a new security model (TDM). It designs the new model's regulate group .The TDM's security regulation defines it's meaning strictly from the four aspects as the security defines, the date security access, date integrality and the coordinating of confliction. It provides the basis for model realizing. At the same time, this article explains the feasibility and rationality through the discussing on the compatibility of TDM model.

It introduces the security object-relational datebase system VISTA , which is developed by the designing group. It expatiates the designing blue print of VISTA from the four aspects as below: the security storage mechanism, the security data model, security access and audit designing. For the first time it promotes the combinatorial security access control design. According to the need of security access control it goes along a rational amelioration and design to the three conventional security visiting control design, such as discretionary access control, mandatory access control and role-based access control, which can be combined freely in order to fit the practical applied need of different security intensity.

Theoretically it has some innovated features in this article :

Put forward a new security model

Firstly put forward combinatorial security access control design.

It improves the traditional discretionary access control and role-based access control and enforces its restriction of security control, which increases the effective domain in discretionary access control. It departs the role domain into operational domain and functional domain, which makes the accss policy more suitable to the practical need.

Key Wards: security model security regulation access control
The DataBase Management System audit security storage

第一章 绪论

对数据库安全的研究由来已久，尤其进入网络时代，数据密集管理的安全问题愈发突出，为保证国家安全、商业信息安全，各国及其大型 IT 行业纷纷投入大量经费从事数据库安全项目研究。

出于国家安全及其他缘由，一些信息技术大国对安全技术进行严格封锁，对这些技术的出口进行了严格的控制。由于技术的封锁，以及安全技术的特征，为保证信息的安全性，我们有必要开发自己的安全产品。

本课题主要就对象关系数据的安全构架进行了基础研究。

1.1 对象关系型数据库安全问题

自上世纪 80 年代起，基于对象的信息处理技术逐步为人们所接受，人们开始广泛研究如何以对象为基本信息单位，对信息进行存储、处理、加工和应用。在此基础上，对象关系数据库技术运用而生。

随着网络时代的到来，协同工作、多媒体技术、GIS 系统等越来越多采用对象关系数据库对数据进行存储和管理，大对象、复杂对象的安全性引起了这些领域的工作人员的关注。与此同时，数据库攻击者注意力也集中到对象关系数据库上，对象关系数据库的安全问题屡见报端。因此，对象关系数据库的安全防范问题成为数据库安全领域的一个备受关注的问题。

对象关系数据库由于其操作客体对象具有封闭性这一特征，人们往往主观认为其安全性高于关系型数据库。然而，事实并非如此，对象关系数据库可能存在的安全问题并不会少于关系型数据库，甚至，在特定环境中存在更多安全隐患。只是封闭性使得这些隐患更具有隐蔽性。

封闭性是在一个原子单位中将数据和与其相联系的操作连接起来，利用用户细节数据的隐藏和操作与应用分离，赋予数据关于其领域的完整性、有效性和一致性。然而，从数据库管理底层可能出现的对存储媒体直接攻击，到数据库管理的应用层，攻击者借助对象提供的服务的安全缺陷对对象属性值的攻击，这些安全问题封闭性是无法解决的。

同时，对象关系数据库中的数据的原子单位(或记录)的比特量一般远大于关系数据库中的数据原子单位。这样，攻击者对信息的获取、篡改、删除、统计分析的安全攻击的可能性就更大。

对象关系数据库安全框架可以一般包括以下几个安全部分(见图 1.1)：

(1)数据存储安全，指数据在存储介质上安全存储方式，包括数据的存储规则和数据加密。如数据库的加密技术，多实例问题等。

(2)数据后备安全，指对数据的备份，以及使用备份对数据的恢复。如数据库

镜像存储技术。

(3)数据访问安全，指用户访问数据库的安全规则。如自主访问控制、强制访问控制、基于角色的访问控制等。

(4)事务安全，指数据库管理系统根据在处理用户访问请求时，如何保证数据的完整性、有效性、一致性和信息的不泄漏。如事务回滚、事务锁等。

(5)对象解释安全。指用户在定义数据库模式时，如何建立主客体间的安全访问规则。如安全视图定义。在对象关系数据库中，该部分的安全规则主要解决如何与数据访问安全规则组合使用，防止攻击者借助对象提供的服务的安全缺陷对对象属性值的攻击。

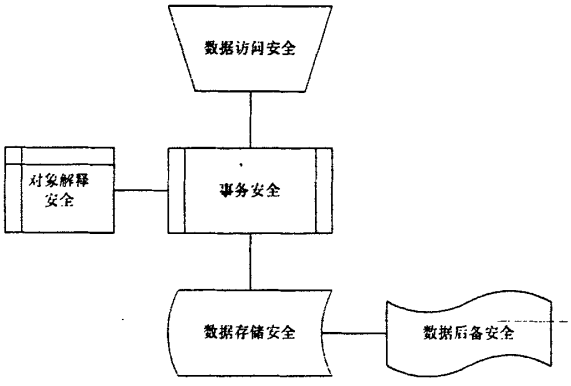

图 1.1 对象关系数据库安全框架的组成

上述五个部分中任何一个部分的安全缺陷，均会给对象关系数据库的应用带来不良后果。为保证对象关系数据库的安全使用，需要对上述方面进行深入研究。本文试图通过对我们开发的安全空间数据 VISTA 设计思想的介绍，探讨对象关系数据库的安全技术。

1.2 国内外研究的现状

数据库技术从 60 年代产生至今，已得到快速的发展和广泛的应用，由于数据库系统是保管信息系统核心内容的关键工具，大量数据集中存放，担负着日益艰巨的集中处理大量信息的任务，而且数据为许多用户直接共享，是宝贵的信息资源，从而使数据库系统的安全性问题日渐突出，其安全保密研究无疑应当具有极其重要的意义。

伴随着 TCSEC[5]的发布，国外的研究者们作了大量的研究，出现了一些高安全性的原型系统。基于这些研究，美国于 1991 年发布了 TCSEC 在数据库管理系统的解释 TDI[6]，为商用系统的开发和评估奠定了基础。

本节将回顾和总结国内外数据库安全发展情况，并展望了未来的发展趋势。

1.2.1 国外安全数据库研究状况

国际上对数据库安全研究起步较早，积累了丰富的理论经验。自 70 年代起，Bell，Biba，Lapadula 和 Denning 等人对信息安全进行了大量的基础研究，提出了多种系统安全模型，并在多个系统中得以实现。

Bell 等人的为实现数据库安全提供了基础。1978 年，Gudes 等人提出了数据库的多级安全模型，把计算机安全保密研究扩展到数据库领域。1986 年，Denning 等人提出了安全数据视图模型，该模型是针对关系数据库系统设计的，采用基于强制存取策略和自主存取策略控制对数据库进行访问控制。

80 年代，美国国防部基于军事计算机系统的保密需要，制订了"可信计算机系统安全评价准则"（TCSEC）[5]，形成了安全信息系统体系结构的最早原则。1994 年，美国国家计算机安全中心（NCSC）颁布了 TDI，即“可信计算机评估标准在数据库管理系统的解释[8]”，它将 TCSEC 扩展到数据库管理系统，并从安全策略、责任、保证和文档四个方面进一步描述了每级的安全标准。这表明，到 90 年代数据库安全已引起足够重视。

按照 TCSEC 标准，安全数据库研究原型一般是指安全级别在 B1 级以上的以科研为目的，尚未产品化的数据库管理系统原型。至今美国已研究出达到 TCSEC 要求安全系统（包括安全操作系统、安全数据库、安全网络部件）的产品多达 100 多种。目前在国外已有不少上市的数据库管理系统得到 B1 级的认证，如 Oracle 公司的 Trusted Oracle 7，Sybase 的 Secure SQL server version 11.0.6 等，而 B2 级及以上认证的安全数据库产品尚属少见。在美国的大型 DBMS 中，多数产品已经通过美国 NCSC 的安全认证，达到 B1 或相当于 B1 的级别，个别的系统已达到 B2 级。另外还有一些具有高安全级别的数据库原型，如：安全数据视图原型和 A1 安全 DBMS 等。然而，他们的高安全级别的产品对我国是封锁禁售的。

90 年代初，英、法、德、荷四国针对 TCSEC 准则只考虑保密性的局限，联合提出了包括保密性、完整性、可用性概念的"信息技术安全评价准则"（TISFC），但是该准则中并没有给出综合解决以上问题的理论模型和方案。

近年来六国七方（美国国家安全局和国家技术标准研究所、加、英、法、德、荷）共同提出了"信息技术安全评价通用准则"（CC for IT SEC）。CC 综合了国际上已有的评审准则和技术标准的精华，给出了框架和原则要求，是当前最新的信息安全标准，它是一系列开发准则的努力的结果。然而，将作为取代 TCSEC 用于系统安全的评测的国际标准，它仍然缺少综合解决信息的多种安全属性的理论模型依据。

迄今，国外在数据库安全模型上已做了很多工作，但许多难题尚未解决：安全体系结构方面的研究工作刚刚开始；安全机制上仍保持着传统的机制，未增加新的安全机制。90 年代以来，数据库安全的主要工作围绕着关系数据库系统的

访问控制模型展开。访问控制模型的研究工作主要分为三个方向：关系数据库管理系统中的自主访问控制模型（DAC）、强制访问模型（MAC）和基于角色的访问控制模型（RBAC）。对数据库安全的更深层次的研究尚未全面展开。同时，数据库安全研究的焦点集中在关系数据模型上，对非关系数据模型的数据库安全研究比较匮乏。

1.2.2 国内安全数据库研究状况

我国在网络信息安全方面的研究起步较晚，投入少，研究力量分散，与技术先进国家有差距，特别是在系统安全和安全协议方面的工作与国外差距更大。国外对于安全产品的出口又有严格的限制，就如 Oracle 的 Trusted Oracle,直到 1998 年才从国外引入。这些国外的高安全级产品即使能够出售给我们，其安全性也是令人担心的。对于信息安全产品，我们走自主研究和开发之路。

我国政府对计算机系统的安全性给予了高度的重视。1994 年 2 月国务院发布了“中华人民共和国计算机系统信息安全保护条例”，1999 年 2 月 9 日，正式成立了“中国国家信息安全测评认证中心（CNISTEC）”，同时，国家质量技术监督局成立了“国家信息安全测评认证管理委员会”，并批准《国家信息安全测评认证管理办法》、国家信息安全测评认证标志和《第一批实施测评认证的信息安全产品目录》。1999 年 10 月发布了“计算机信息系统安全保护等级划分标准[17]”，该准则为安全产品的研制提供了技术支持，也为安全系统的建设和管理提供了技术指导。

数据库系统方面，我国已经根据 TDI 和 TCSEC 编制了相应的适用于数据库管理系统安全标准的国标。目前国内的系统软件和应用软件的安全性级别基本在 C2 级，部分在 C1 级，而 B 级尚处于开始研究阶段。总体来讲，我国自己的安全产品目前还基本是空白。

国产数据库软件，具有一定影响力的除国信贝斯公司的 iBASE 数据库外，还有国家早先立项支持的由中国人民大学、中软总公司合作完成的 COBASE、东软公司的 OPENBASE、中国人民大学与知识工程研究所推出的 EasyBASE/PBASE 和华中理工大学的 DM 系统等。

这些产品大多提供了安全数据库应用环境。然而，它们的实现主要采用外挂式的技术方案，即内核采用国外研制的产品，安全控制自主研发，用户通过外挂的安全控制模块访问数据库内核，从而达到安全控制目的。数据库产品的自主安全内核依然受制于人，难以保证没有漏洞。另外，这种外挂式的设计方法，对于某些安全要求往往难以实现，比如 B1 级安全的“客体重用”要求就很难通过外挂的方式实现。

按照我国现有的水平，在相当的一段时期内，系统升级的典型体现应当是从 C2 级（或更低级别）更新至 B1 级。因此，从实用角度和自主性出发，我们自主开发一个可信的数据库管理系统。

本课题着重研究的对象是具有自主版权的可信数据库管理系统 SEC_VISTA，为实现多级对象关系型数据库的 B1 级安全。

1.3 研究内容

本课题的主要研究内容是参照已有的安全标准和控制模型，设计实现 B1 级多级对象关系型数据库 SEC_ VISTA。

目前，对于传统的关系型数据模式已经有了很多经典的访问控制方法。如 Sea View 安全视图模型，基于强制存取策略控制和自主存取策略控制等。这些模型很好地解决了对关系数据模型的访问控制问题。但如何管理多级安全环境下的复杂数据对象，如空间数据、多媒体数据等，目前还没有较好的访问控制模型。这需要对基于关系数据库的经典控制方法进行改造和扩充，实现基于对象数据的访问控制。

尽管目前普遍认为面向对象技术是解决对象关系型数据库的访问控制的途径，但是具体实现安全的面向对象数据库管理系统非常困难，更多的方法是直接改进关系数据库管理系统的安全数据模型，使之能够处理对象关系型数据。目前多级安全（Multilevel Secure，MLS）对象关系型数据库管理系统(Database Management System, DBMS)的研究还是主要基于关系数据库管理系统，这主要是由于关系数据库管理系统无论从理论上还是技术上都已经成熟，并且得到广泛应用。另外，由于数据库安全技术的研究往往滞后于数据库技术的研究，吸收基于关系型数据库安全模型的控制方法，扩充关系型 DBMS 的安全性是一条可行的途径。结合 SEC_VISTA 的研究，有必要研究这些系统在多级安全环境下的安全性。

我们以自主开发的空间数据库管理系统 SEC_VISTA 为内核，研究对象关系型数据库的安全模型，增加达到 B1 级主要特性的安全机制，从而提供一个安全的数据库应用开发环境 SEC_VISTA。在 SEC_VISTA 的开发过程中，重点解决了对象型数据的属性和方法的控制的处理，保证其安全性。

同时，传统的安全模型，如 Bell-LaPadula[1, 2]（以下简称 BLP）模型，不能满足对象数据的封装性要求，给访问对象带来了不便。在 SEC_VISTA 中，我们将对传统的安全模型进行适当的修改，使其更加适用于对象关系型数据的安全访问。

本课题主要解决以下几个问题：

(1)对象关系数据库安全框架的建立。

(2)对象关系型数据库访问控制策略的研究。

(3)改进传统安全模型，提出一种适宜对象操作的安全模型。

(4)SEC_VISTA 安全数据库的安全模型研究

(5)SEC_VISTA 安全数据库的安全策略的研究及其实现方法

1.4 内容安排

本文将从对象关系型数据库安全框架、SEC_VISTA 安全数据库安全模型、SEC_VISTA 的安全策略以及系统的实现等几个方面来介绍作者的工作：

⑴对象关系型数据库安全框架。第二章介绍了对象关系型数据库安全框架，讨论对象关系数据库安全体系、安全框架的各组成部分在安全数据库中的作用及其相互关系。在本章中着重研究对象关系数据库的安全访问策略及其基本技术。

⑵SEC_VISTA 安全数据库安全模型。第三章将介绍 SEC_VISTA 安全数据库的安全模型定义，讨论空间数据库 VISTA 的体系结构，根据其体系特征增加对象关系型数据库的安全需求，建立 SEC_VISTA 的安全模型。

⑶SEC_VISTA 的安全策略。第四章将详细介绍多级安全数据库 SEC_VISTA 安全策略的设计方案，包括安全访问策略和安全存储策略及其信息过滤机制等。

⑷SEC_VISTA 系统环境简介。第五章简单介绍了 SEC_VISTA 系统的安全功能，及其主要交互界面。

第二章 对象安全数据库安全概论

对象关系型数据库管理系统，是在关系数据库的基础之上增加了面向对象的特征，其数据模型比关系模型要丰富的多，数据间的关系也比较复杂。因此，传统的针对关系型数据库管理系统所采取的安全机制需要进行修改和补充，才能够适用于对象关系型数据库。

本章简要讨论对象关系型数据库安全访问策略，及其安全体系模型。

2.1 对象关系数据库安全定义

2.1.1 对象关系型数据库系统的安全定义

数据库系统的安全性是指数据库的数据及其组成部分不得受到侵害或篡改。其安全机制核心是：提供数据的安全存取服务和可信用户的安全访问服务，以保证所管理的数据具备可用性、完整性和一致性。需要满足下述基本的安全要求：

(1) 保密性，即数据不被直接或间接的泄露，防止非授权访问。

(2) 完整性，即数据的物理、逻辑乃至数据元素是完整的、正确的。即信息在存储或传输过程中不篡改、不缺损、不丢失。

(3) 可用性，即在任何时刻，可信用户可以通过合法途径访问授权数据。

由此可见，数据库系统的安全问题主要集中在存取和访问两个方面。

传统关系数据库系统的安全防范和攻击主要通过两个途径实现：(1)直接作用于数据；(2)借助数据库的数据操纵指令间接作用于数据。而对象关系型数据库系统则在保留这两种途径的基础上，由于对象的封闭型，还存在第三种途径，即通过对象提供的方法(服务)作用于对象的属性(数据)。

因此，我们可以给出对象关系数据库系统(ORDBS)安全的简要描述：ORDBS 的安全性是指数据库的数据及其组成部分不得受到侵害或篡改，并且数据对象提供的方法是可信的。其安全机制的核心是：提供数据的安全存取服务、提供数据对象方法的可信管理、提供可信用户的安全访问服务，以保证数据具备可用性、完整性和一致性。

2.1.2 对象关系型数据库的安全特点

数据库系统由于其自身的特征，在安全需求上有着特殊的要求，主要特点有：

(1)安全需求的复杂性。表现为：不同数据具有不同的安全等级；不同用户具有不同的访问域；不同的数据生存期具有不同的安全周期等。

(2)安全客体的复杂性。由于模式与视图间的多对多的映射关系，造成了安全

客体具有复杂的拓扑结构。每个视图的安全依赖于组成该视图拓扑结构的基本单位的安全因素的代数运算。

(3)安全操纵的复杂性。对数据库的操作是通过操纵指令的组合实现的，指令组合的多样性可能带来语义解释上的安全漏洞，造成信息泄漏。同时，数据库允许使用统计指令，可能产生出非敏感数据推理得出敏感数据的推理攻击。

(4)数据规范矛盾。数据库是建立在特定的数据模型上的数据密集型存储和管理机制，要求数据满足数据模型的规范，这可能导致安全隐患。如关系数据库中的一致性导致出的多实例问题。

(5)独立性矛盾。数据库要求数据对象独立于访问环境，这会导致对数据文件的直接安全攻击，导致安全问题。

在包含上述安全特征的基础上，由于对象关系型数据库还具备有别于其他数据库的特点，还有一些扩充的安全特点。ORDBS 除了具有原有关系数据库的各种特点以外，还具有以下特点：

(1)可扩充数据类型。允许用户在关系型数据库系统中扩充数据类型，即允许用户根据应用需求自己定义数据类型、函数和操作符。且一经定义，这些新的数据类型、函数和操作符将存放在数据库系统核心中，可供所有用户共享，如同基本数据类型一样。

(2)支持复杂对象。关系数据库以二维表作为数据模型，简单、清晰，但难以直观、全面地描述客观世界中的复杂事物。对象关系型数据库支持复杂对象。

(3)基类扩充。所谓基类扩充，是指用户在操作原语的基础上创建带有相应操作符和函数的新数据类型，以实现数据类型的扩充和复杂对象的创建。

(4)支持继承的概念。在事物分类中，继承描述了不同分类粒度下事物间特征的相关性。对象关系型数据库提供了继承机制，可以更好地模拟实际问题。

(5)通用规则系统。规则系统指 DDL、DDM 和 DDC 中原语的组合和解释规则的集合。由于对象关系数据对基类扩充和继承的支持，在此基础上产生的用户定义必须满足规则系统。

基于上述特征，ORDBS 还具有如下扩充的安全特点：

(1)安全定义的复杂性。传统数据库中，安全定义只需给出数据的安全等级和用户的访问域。在对象关系数据库中，还需要定义对象提供的服务(方法)的安全，包括服务与用户间的关系。

(2)安全继承问题。传统数据库中，数据间的关系比较简单，只是通过外码构成关联关系，而在对象关系数据库允许继承，使得数据类型间还存在包含关系。然而，数据对象的包含并不能简单地推广到数据对象安全的包含，这就引发了安全继承问题。

(3)安全实例问题。对象关系数据库的最终用户关心的是复杂对象类型的实例。如何给出实例的安全定义也是需要解决的问题，因为，实例安全的定义不是简单地重复类型的安全定义。通常实例的安全强度要高于类型的安全强度。

(4)规则可信问题。用户定义是否满足规则系统，或者规则系统是否能判断用户定义是可信的，这是对象关系数据库需要解决的问题。这里的可信判定包括用户定义的可用性和安全性两个方面的含义。

2.2 对象关系型数据库安全体系模型

任何信息管理系统可以通过三个互相支持的技术可以达到保护信息免遭破坏或篡改，对象关系型数据库也是如此。这三个技术为：鉴别，访问控制和审计。鉴别对用户身份合法性进行判别。访问控制定义和控制一个对象对另一个对象的访问权限。存取控制往往需要鉴别作为先决条件。审计过程收集系统中所有的数据操作，并且分析它们以发现系统的安全弱点。

图 2.1 是在安全对象关系数据库中这些安全服务和它们相互作用的逻辑结构图。图中给出了三种安全服务如何在对象关系数据库中作用的逻辑关系，以及对象关系数据库基本安全控制体系。

用户在请求访问时首先通过身份鉴别，在得到合法访问权后根据安全访问规则访问数据库，并由系统反馈相应信息。同时，在用户请求事件发生的同时，系统触发审计事件，对用户请求事件进行审计。

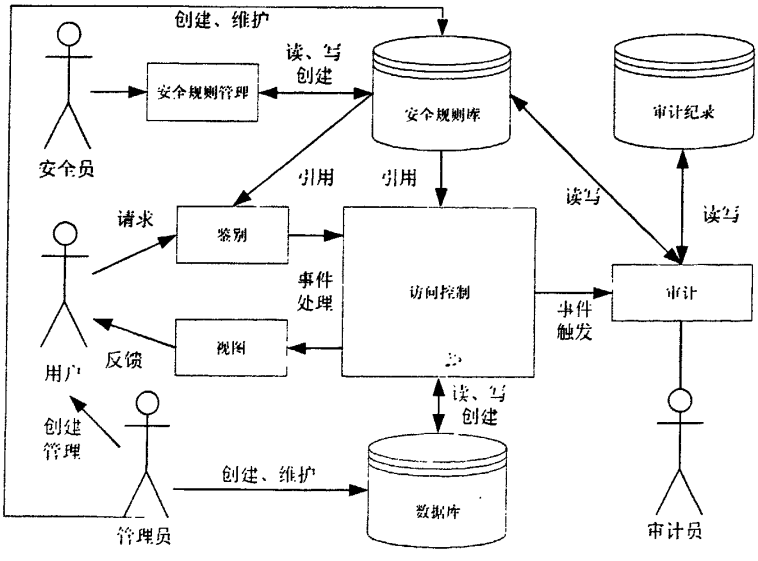

图 2.1 安全对象关系数据库的安全服务的逻辑结构图

系统中，系统安全员对系统中的客体的安全进行管理，规定用户对客体访问的规则。审计员对系统发生的历史事件进行审计，防范系统漏洞和对外部攻击机进行检测。管理员对系统主体进行管理，确定主体在系统中的活动域。通过安全员、审计员、管理员三者的合作，实现安全管理。

在安全体系中，三种安全服务是研究关注的焦点。其中：

鉴别是对系统中的主体进行验证的过程，根据用户的私有信息来确定用户的真实性，防止欺骗。主要采用三种方法验证主体身份：

(1) 只有该主体了解的秘密，如口令或加密密钥；

(2) 主体携带的物品，如智能卡；

(3) 只有该主体具有的独一无二的特征或能力，如指纹、声音、视网膜或签字等。

访问控制是一种加强授权的方法(所谓授权，是指资源的所有者或控制者准许其他人访问这种资源)。存取控制在身份识别的基础上，根据身份对提出的资源访问请求加以控制。在访问控制中，对其访问必须进行控制的资源称为客体；必须控制它对客体的访问的活动资源，称为主体，主体即访问的发起者，通常为用户、进程等。访问控制的目的是为了限制访问主体对访问客体的访问权限，从而使系统在合法范围内使用。访问控制需采取两种措施：一是识别与确证访问系统的用户；二是决定该用户对某一系统资源可进行何种类型的访问（读、写、删、改、运行等）。可具体描述如下：

(1) 规定需要保护的资源，又称客体

(2) 规定可以访问该资源的实体又称主体

(3) 规定可对该资源执行的动作（如读、写、执行或不许访问）

(4) 通过确定每个实体可对哪些资源执行哪些动作来确定该安全方案。

安全审计是对系统记录和过程的检查和审查，其目的是测试安全策略是否充足，证实安全策略的一致性，建议安全策略的改变，协助攻击的分析，收集证据以用于起诉攻击者。安全审计追踪是记录用于入侵检测和安全审计的相关事件的一个日志。它可以自动记录一些重要安全事件。记录此事件应包括试图联机的每个用户所在的主机和时间，同时对管理员的活动也要记录，以便于研究入侵事件。有些入侵的成功可能是由于管理员的错误所造成的，如管理员误将访问权给了某一个用户。审计追踪是检测入侵的一个基本工具。设计审计系统的关键是：

(1) 确定必须审计的事件，建立软件记录这些事件，并将其存储，防止随意访问；

(2) 审计机构监测系统的活动细节并以确定格式进行记录；

(3) 对试图联机，对敏感文件的读写，管理员对文件的删除、建立和访问权的授予等每一事件进行记录；

(4) 管理员在安装时对要记录的事件做出明确规定。

2.3 安全数据库访问策略

三种安全服务中，对于访问控制由于其涉及对象众多，对象间需要建立的安全指非常的复杂，备受关注。在基于安全研究过程中，经受了客体存取过程

为主要研究对象的访问控制策略：自主策略（DAC）、强制策略（MAC），和以主体作用域为主要研究对象的访问控制策略：基于角色的策略（RBAC）。本节我们将讨论这三种不同的策略。

2.3.1 自主策略（DAC）

自主型访问控制（Discretionary Access Control）是基于一种访问控制规则实现主体对客体的访问。这种控制规则是自主的，自主是指某一主体能直接或间接的将访问权或访问权的某些子集授予其他主体。用户对信息的控制是基于用户的鉴别和存取访问规则的确定。

DAC 基于用户的身份和访问控制规则。自主保护策略管理用户的存取，这些信息是以用户的身份和授权为基础的，它们详细说明了对于系统中的每一个用户（或用户组）和每一个客体，允许用户对客体的存取模式（例如读，写或执行）。根据指定的授权，用户存取客体的每一个要求都被检查。如果存在授权状态，则用户可以按指定的模式存取客体，存取被同意，否则被拒绝。DAC 之所以被称为自主的，是因为它允许用户将其访问权力赋予其它的用户。自主策略的灵活性使它们适合于多种系统和应用。

在自主存取控制中，用户对于不同的数据对象有不同的存取权限，不同的用户对同一对象也有不同的权限，而且用户还可将其拥有的存取权限转授给其他用户。因此自主存取控制非常灵活。自主存取控制能够通过授权机制有效地控制其他用户对敏感数据的存取。但是由于用户对数据的存取权限是“自主”的，用户可以自由地决定将数据的存取权限授予何人、决定是否也将“授权”的权限授予别人。在这种授权机制下，仍可能存在数据的“无意泄露”。

自主访问控制的实现主要有三种方式：1.访问控制表（ACL）；2. 访问能力表（Capability）3.授权关系表。

2.3.2 强制策略（MAC）

自主控制的最大问题是没有对信息的传播加以控制。为了避免这种情况的发生，引进了强制型安全控制。

强制型访问控制（Mandatory Access Control）通过无法回避的存取限制来防止各种直接的或间接的攻击。系统给主体分配了不同的安全属性，并通过主体和客体的安全属性的匹配比较决定是否允许访问继续进行。

强制访问控制施加给用户自己客体的严格的限制，也使用户受到自己的限制，它可以防止在用户无意或不负责任的操作时，泄露机密信息。一般强制访问控制采用以下几种方法：

(1) 限制访问控制。由于自主控制方式允许用户程序来修改他拥有文件的存取控制表，因而为非法者带来可乘之机，因而，系统可以不提供这一方便，在这

类系统中，用户要修改存取控制表的唯一途径是请求一个特权系统调用。该调用的功能是依据用户终端输入的信息，而不是靠另一个程序提供的信息来修改存取控制信息。

(2) 过程控制。在通常的计算机系统中，只要系统允许用户自己编程，就没办法杜绝特洛伊木马。但可以对其过程采取某些措施，这种方法称为过程控制。需要说明的一点是，这些限制取决于用户本身执行与否。因而，自愿的限制很容易变成实际上没有限制。

(3) 系统限制。显然，实施的限制最好是由系统自动完成。要对系统的功能实施一些限制。比如，限制共享文件，但共享文件是计算机系统的优点，所以是不可能加以完全限制的。再者，就是限制用户编程。事实上，有许多不需编程的系统都是这样做的。

在安全数据库系统中，有许多安全模型采用了强制存取控制策略，如贝尔-拉帕丢拉(Bell-La Padula)模型、第昂模型和施密斯-温斯莱特模型等。

2.3.3. 基于角色的策略（RBAC）

从保护能力来看，由于自主型控制太弱，强制型控制太强，而且二者的工作量都很大，不便于管理。于是提出了基于角色的策略[10]，这种策略在当代的商业环境中特别有意义。

角色控制与 DAC 和 MAC 相比，角色控制相对独立，根据配置可使某些角色接近 DAC，某些角色接近 MAC。基于角色的策略以系统中用户执行的行为为基础，管制用户对信息的存取。基于角色策略需要系统中角色的认证。一个角色可以作为一个行为和与特殊工作行为关联的责任集合而被定义。因而，代替指定每一个用户允许执行的所有存取，对客体的存取授权被指定给角色。

RBAC 提供了解决具有大量用户、数据客体和访问权限的系统中的授权管理问题。RBAC 涉及用户（user）、角色（role）、访问权限（permission）、会话（session）这几个主要概念。角色是访问权的集合。当用户被赋予一个角色时，用户具有这个角色所包含的所有访问权。用户和角色之间是多对多的关系，角色与访问权间也是多对多的关系。在 RBAC 模型系统中，每个用户进入系统时得到一个会话，一个用户会话可能激活的角色是该用户的全部角色的子集。对此用户而言，在一个会话内可获得全部被激活的角色所包含的访问权。角色和会话的设置带来的好处是容易实施最小特权原则。

由于对象关系数据库中数据结构的特殊性，我们可以将对象类提供的服务指定给角色，实现对象类的安全访问。

第三章 VISTA 数据库系统的安全模型

VISTA[24][26]数据库系统是一个对象关系型数据库管理系统，我们在该数据库的基础上设计了一整套安全策略，实现了一个安全对象关系数据库。不同于传统的建立在商业数据库系统上的安全数据库，VISTA 在数据库内核上直接开发安全功能，而不是采用外包完全模块的设计方案，因此，VISTA 更为安全。

本章简要介绍了 VISTA 的体系结构，并重点讨论 VISTA 数据库所遵循的安全模型。

3.1 VISTA 数据库系统简介

VISTA 系统是我们课题组自主开发的一个空间数据库管理系统，并于 92 年在该系统的基础上为墨西哥石油公司开发的一个用于石油勘探资源管理的 GIS。该系统由两部分组成：图符化数据库查询语言 CQL[27]和空间数据库管理系统。CQL 是用可视化的方法进行数据库查询操作，空间数据库管理系统部分主要实现对空间数据库的维护操作。

经过多年的技术积累和改造，VISTA 逐步形成了一个较为完善的对象关系型数据库。VISTA 系统结构如图 3.1 所示。

VISTA 包含了数据库管理系统的基本功能。主要分为四部分：对象关系数据的存储机制、DDL 解释机制、DML 解释机制和数据管理维护功能。

3.1.1 VISTA 的存储机制

数据库的存储结构是数据库的物理基础，对数据库系统的性能影响很大。存储结构中涉及记录的物理表示、物理块的磁盘分配、数据压缩技术以及文件形式等。在 VISTA 中将物理块的磁盘分配交给操作系统管理，仅考虑数据的逻辑存储机制。

VISTA 将数据存储定义为三段式[见图 3-2]。首先，进行数据模型处理；其次，进行文件形式处理，决定数据类型采用的文件形式，确定记录在文件中存放位置；最后，进行记录物理表示形式处理，决定记录的存储格式。再由 OS 将记录存储到磁盘。

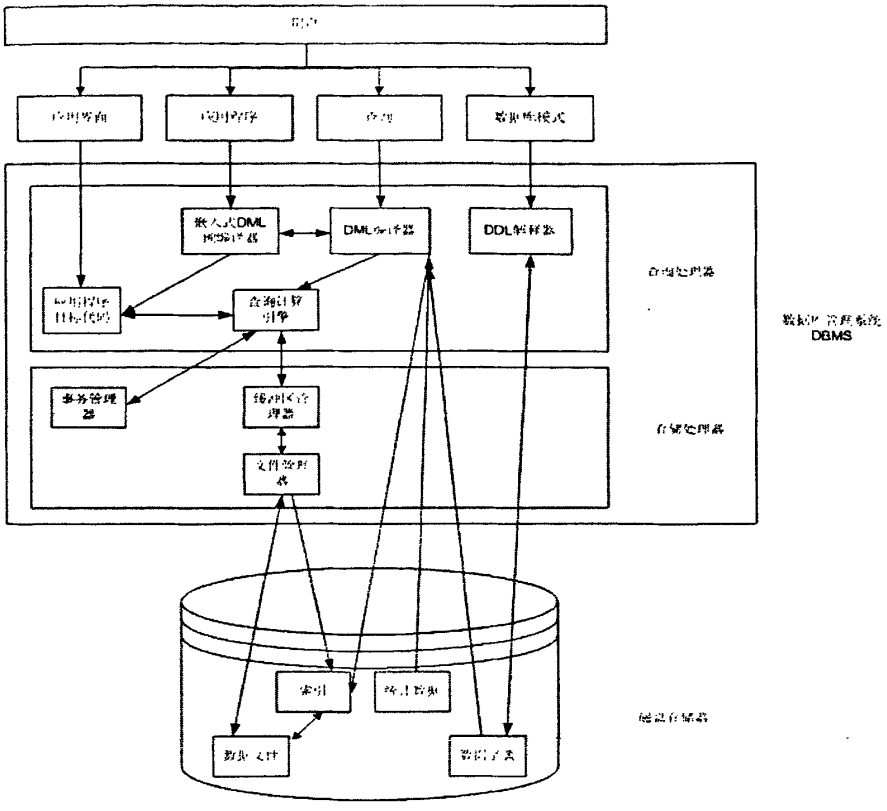

图 3.1：VISTA 系统结构图

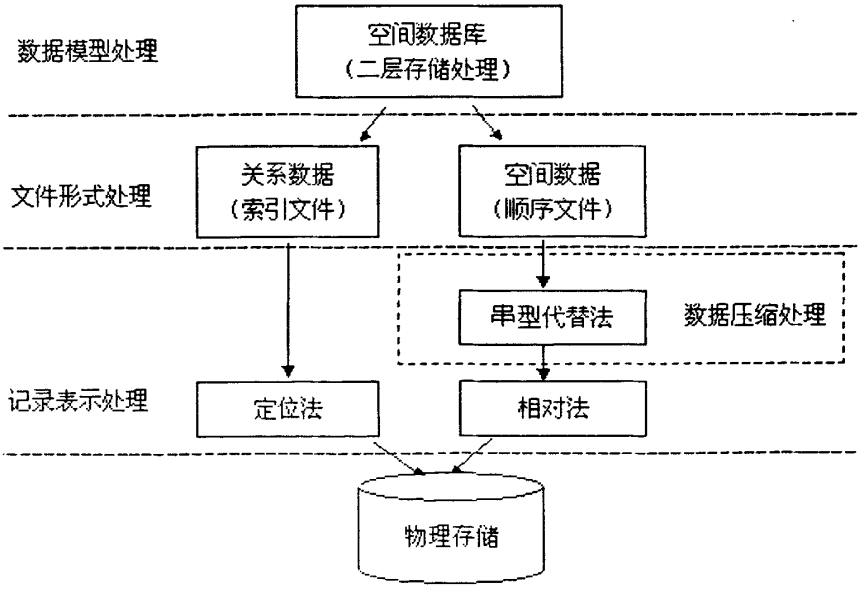

图 3.2 VISTA 数据存储流程

由于 VISTA 处理的数据为对象关系型数据，我们采用二层存储处理方案对数据进行存储。如图 3.3 所示，这种结构将传统的关系数据存取结构和空间数据存取结构相融合，能同时有效地管理和处理传统数据类型和新型空间数据类型。第一层是关系信息层，主要用来存取空间对象间的关系信息。这些信息均为正文信息，该层使用传统的 B 树来实现存取操作。第二层是对象数据层，该层用来存放各种异质数据，如地物坐标、图像、图形等。

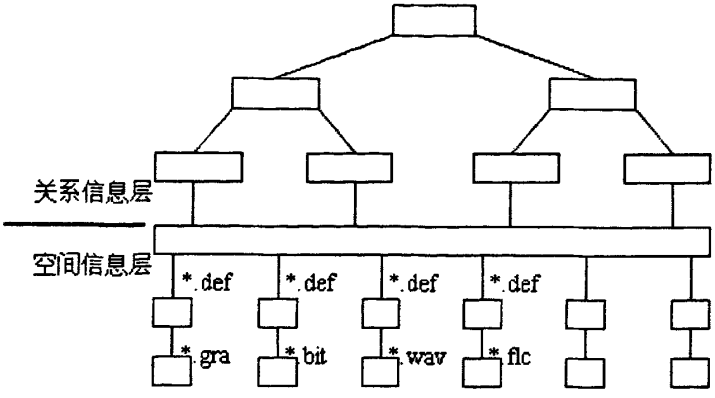

图 3.3 VISTA 二层存储处理方案示意

3.1.2 DDL 解释机制

数据定义语言(DDL)主要实现数据库的结构描述，包括外模式、模式、内模式的定义；数据库完整性定义；安全保密定义；存取路径定义。这些定义存储在数据字典中，是 DBMS 运行的依据。

DDL 的定义均与数据字典有着密切的关系，每个定义与数据字典间存在数据交换，数据字典的定义是否合理，直接影响到系统的正确性、通用性和重用。VISTA 将该部分的每个定义设计为：一个定义模块对应一个数据字典文件。定义模块通过数据字典管理模块访问对应的字典文件，数据字典管理模块通过数据字典表管理数据字典文件。图 3.4 给出了字典管理的结构示意。主要包括三个方面：

(1)外模块、模块、内模块、存取路径定义

VISTA 中外模式、模式、内模式定义是一个解释程序模块，即模式定义模块。该模块将相应的数据字典内容解释为数据库的数据结构，数据的存取过程由存储处理模块实现。同时，在内模式中定义了数据库的数据模型处理、文件形式处理和记录表示处理相关信息，其中文件形式处理包含了存取路径的计算方法。

(2)数据库完整性定义

数据库完整性是指数据的正确性和相容性，通过完整性约束条件的规定和检查来实现的。VISTA 的完整性定义包含值/结构约束、动态约束、执行约束等。VISTA 允许用户通过修改完整性数据字典，定义满足用户需要的数据库完整性约

束。

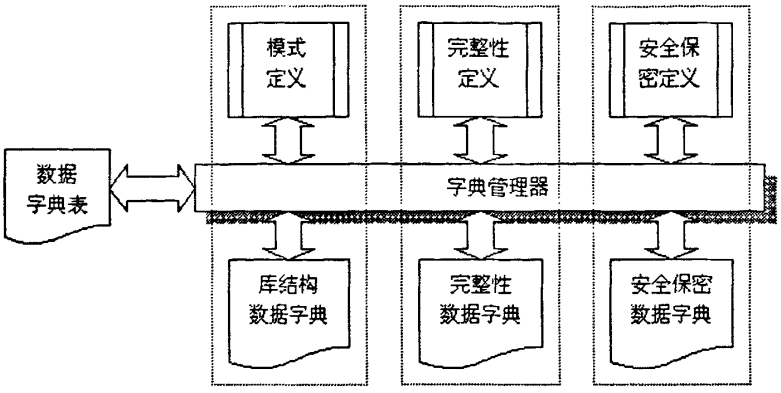

图 3.4 VISTA 中的 DDL 定义模块对数据字典的访问示意

(3)安全保密定义

保护数据库的安全，本质上就是保证合法访问，阻止非法访问。对数据库安全问题，可归结为对数据库系统中各种对象存取权的合法性和对数据库内容本身的安全（防泄露、篡改或破坏）两个方面。数据库的安全验证是建立在各种安全规则表上的，如用户安全、访问规则、审计日志、授权表等。VISTA 通过安全保密定义实现其安全。在此基础上建立的一系列的安全规则，就构成了 VISTA 的安全体系。

在后面的章节，我们将着重讨论这些规则的建立方法，以及 VISTA 如何通过对安全规则的解释来实现数据库的安全。

3.1.3 DML 解释机制

数据库操纵语言(DML)完成对数据库数据的检索、插入、修改、删除操作。DML 面向用户，将用户输入的请求解释为相应的 DML 指令，调用存储处理功能对数据库的数据进行操作。其工作过程如图 3.5。

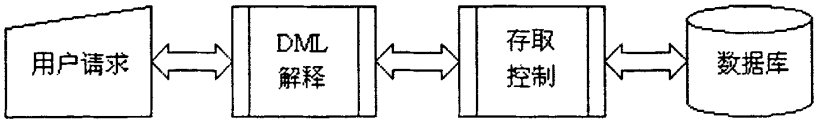

图 3.5 DML 工作过程

DML 功能的实现可以划分三个部分：用户接口、解释、存取调用。存取调用指 DML 根据用户请求的解释调用存取操作，向数据库写入或读取数据。解释部分将用户请求解释相应的存取指令，或将由数据库读取的数据解释为用户可识别形式。用户接口部分主要是获取用户请求和反馈数据信息。

从数据库安全角度看，可以在 DML 解释环节和存取控制环节加入安全约束，

VISTA 在 DML 解释部分实现安全约束，其策略将在第四章介绍。

3.1.4 数据管理维护功能

管理维护是 DBMS 设计中最为复杂、选择余地最大的一组功能模块，数据库管理系统的功能强弱主要由这部分体现。VISTA 系统的管理维护功能包括事务处理、安全控制、完整性控制、后备管理、数据恢复、错误处理和保护、系统信息维护和管理等。

(1)事务处理(Transaction Manager)模块可以保证数据库在并发操作时保持数据库的一致性。事务处理具备必须完成调度任务、同步控制、协调事务通讯、提交事务、放弃事务、事务约束检查、事务卷回(rollback)等功能。事务通过事务日志进行管理。

(2)安全控制(Security Manager)是为防止用户对数据库的非法访问和恶意攻击，采取的控制动作，包括用户识别、认证、授权、存取控制以及审计。

(3)完整性控制(Integrity Manager)。该功能是对完整性定义进行解释执行，实现数据的正确性和相容性。

(4)后备管理(Backup Manager)是数据库综合保护策略的一个重要组成部分，以便在文件损坏或丢失时重建该文件。数据库的备份手段多种多样，有完全数据库备份、表空间备份、逻辑数据库备份等，备份的手段不同对数据库的保护程度也不同，可以根据需要建立不同的备份构件。

(5)数据恢复(Data Recovery)主要包括两个方面：当事务卷回时将数据恢复到事务前的状态；当数据受损时出后备数据进行修复性恢复。前一种恢复机制在事务功能中实现，在此讨论后一种机制。

根据系统设计不同和数据受损情况的不同，后备恢复的处理方法不同。如完全恢复、不完全恢复，基于时间的恢复、基于改变的恢复等。VISTA 提供基于时间的数据恢复功能。

(6)错误处理和保护功能也是数据库保护措施之一，是对系统运行中的错误进行预警和保护性控制的一种管理机制。

(7)系统信息维护和管理是指对 DBMS 系统信息的操纵，包括系统信息的检索、插入、修改、删除、备份、恢复等操作。DBMS 的系统信息存放在数据字典中，该功能也就是对数据字典的操纵。

3.2 VISTA 数据库的安全模型

安全模型的目标就是精确地描述系统的安全策略。安全策略即是要对系统的安全需要，以及如何设计和实现安全控制有一个清晰的，全面的理解与表述。安全模型应具备如下特点：

(1) 是精确的，无歧义的；

(2) 是简单抽象的，容易理解的；

(3) 是一般的，只涉及到安全性质，不过度地抑制系统的功能或其实现。

安全模型是安全策略的明显表现。

数据库安全是指机密性、有效性、真实性、完整性和信息可用性的结合。而早期的数据库安全模型，主要针对数据库的保密性和数据的完整性两个角度进行研究，形成了多种数据库安全模型。如：贝尔-拉帕丢拉(Bell-La Padula)模型、第昂模型和施密斯-温斯莱特模型等以强制存取控制作为安全模型的出发点解决数据库的保密性，实现可信主体访问策略。而 Denning-Lunt 模型、Jajodia-Sandhu 模型、基于信赖的语义模型、基于数据语义的模型等以数据库中多值依赖作为研究的出发点，旨在解决数据库中数据完整性问题。这些安全模型的研究解决了数据库安全中某个方面的问题，但未就数据库整体安全给出模型。我们在吸取前人安全模型思想的基础上，对 VISTA 安全模型的设计旨在给出一个能体现数据库整体安全控制策略的安全模型。我们将该模型简称为 TDM（Total DB Security Model）。

TDM 模型将从三个角度考虑数据安全控制策略，即从(1)存取访问控制；(2)数据完整性；(3)用户安全视图等三个角度考虑如何解决数据库的机密性、有效性、真实性、完整性和可用性问题。

3.2.1 TDM 模型的定义

TDM 是基于操作原语的安全模型。其基本思想是将主体的操作请求分解为基本的操作原语，根据安全规则约束，决定操作原语是否执行。

TDM 的操作原语集为{创建、写、读、锁}，鉴于目前我们开发的 VISTA 中尚未实现并发操作，故本文只讨论创建、写、读三个原语，对锁操作原语暂不讨论。其他系统中的基本操作，在 TDM 中可分解基本原语，如删除操作分解为写 NULL 操作；对于对象实体的方法执行操作分解为对对象实体属性的读写操作。

在 TDM 模型下，对数据库的安全访问过程可以描述如下：

访问请求→操作原语→安全进程控制→访问数据库→消息重组→安全进程控制→消息反馈

即将“访问请求”分解为若干条“操作原语”；“安全进程控制”模块对“操作原语”中的主客体的安全约束进行甄别，决定是否执行原语；可执行原语访问数据库；对获取的客体信息进行重新组合，经过“安全进程控制”模块信息处理后，反馈信息给用户。

安全进程控制是一些控制器，其功能是根据安全规则对操作进程进行安全甄别和处理，决定操作进程是否进行或如何以安全的方式与用户进行交互。TDM 模型中定义了四个安全进程控制，分别为：约束检测、引用检测、视图监视和消息填充，其定义如下：

(1) 约束检测

定义 3.1： 约束检测，指在 TDM 模型中执行创建客体或向客体写信息操作，得到的客体是否满足安全规则规定的约束条件。如主客体安全等级匹配约束、数据完整性约束、数据一致性约束等。

该安全进程主要控制主体对客体是否具有创建权或写操作权限，以保障数据的有效性、真实性和完整性。

(2) 引用检测

定义 3.2： 引用检测，指在 TDM 模型中在执行写操作和读操作时，主体是否对客体具有访问权的约束。主要检测主体是否对客体具有特权，即主体是否是客体的宿主，或获得访问授权。

该安全进程主要控制主体对客体是否具有读、写操作权限，以保障数据的机密性和可用性。

(3) 视图监视

定义 3.3： 视图监视，指在 TDM 模型中执行读操作时，控制被访问客体在出现多值映射运算时应满足的安全访问规则。如多实例访问控制、主客体安全等级匹配约束、关系型数据元组的代数运算结果是否满足安全访问约束、对象型数据类的继承、聚合是否满足安全访问规则等。

该安全进程主要控制客体对主体的消息反馈，在保障数据可用性的同时，防止信息泄漏。

(4) 消息填充

定义 3.4： 消息填充，指在 TDM 模型中，系统向主体反馈信息时，对主体限制访问信息填充安全约定值，使得主体获得信息满足虚拟的数据完整性和一致性。即用虚拟信息替代限制访问信息，使得主体对限制访问没有觉察。

该安全进程主要解决数据库中常见的统计泄密的安全问题。

消息填充通常不是基于安全规则的，而是基于知识或约定的。TDM 模型中可以根据需要定义消息填充机制，如空值(NULL)填充、盲填充或基于知识库的填充等，前两种在解决统计泄密方面并不理想，后一种则需要借助人工智能来实现。在 VISTA 中。我们目前使用的是空值填充机制，即对受限访问信息均以 NULL 值代替。

图 3.6 给出了 TDM 模型的结构示意图。图中包括四个安全进程控制，分别为：约束检测、引用检测、视图监视和消息填充，前三个为安全约束控制，最后一个为安全防范控制。

由图可见 TDM 模型的安全约束是通过安全进程来实现的，而这些进程除“消息填充”进程以外，均与安全规则库有着密切的联系，换言之，是对安全规则库中的相应规则进行检验，因此，我们需要对安全规则库进行阐述。

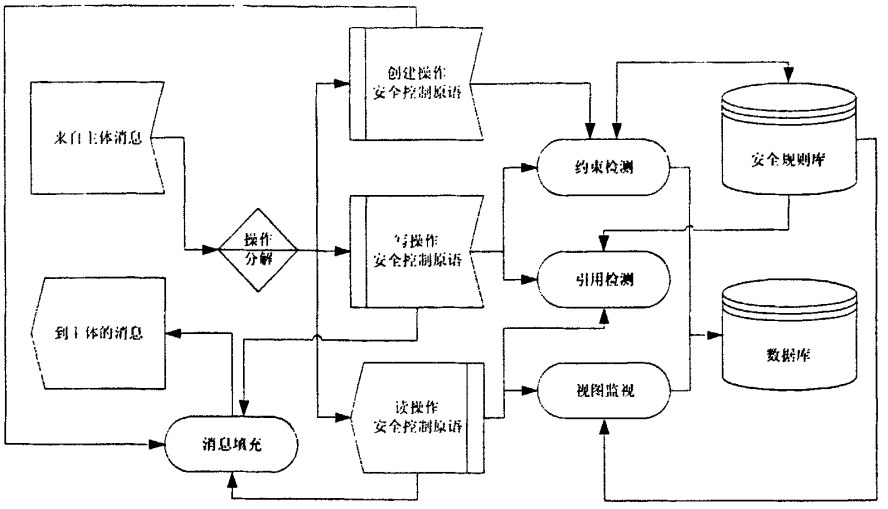

图 3.6 TDM 模型结构图

3.2.2 TDM 模型的规则定义

3.2.2.1 符号描述和基本定义

为便于描述，我们首先约定一组符号描述和有关的概念定义。

约定 3.1(符号描述)：

(1) 用如下符号描述安全数据库中的名词：M 表示“主体”，O 表示“客体”，R 表示“角色”，S 表示“密级”，P 表示“权力”。

(2) 用 RD 表示数据库中满足关系型数据结构的数据对象，OC 表示对象类，OD 表示满足对象型数据结构的数据对象。Table(a)表示数据表 a，Record(a)表示记录 a 或元组 a，Field(a)表示字段，Attribute(a)表示对象 a 的属性，Operate(a)表示对象 a 的服务。View 表示视图。

(3) 用 OP(name, M, O)表示操作原语，其中：name 为原语名，name∈{write, read, create}；M 为操作发起的主体，O 为操作对象，即操作施加的客体。

(4) 用 dom(a)表示对象的定义域，其中 a 为对象。如 dom (P_R)表示定义给角色 R 的特权域。dom(O)表示系统辖域中的所有客体，而 dom(O_S) 则表示密级为 S 的对象构成的集合。

定义 3.5(组合操作)：

组合操作，为多个操作原语之间的运算关系，可划分为三种运算：串行、并行、往复。定义如下：

(1) 操作 OP_1 和 OP_2 串行指：OP_1,OP_2 的操作是有序的，即 OP_1 必须在 OP_2 之前发生。记作<OP_1 , OP_2 >

(2) 操作 OP_1 和 OP_2 并行指：OP_1,OP_2 的操作是无序的，即 OP_1 与 OP_2 有

执行时间无先后约束。记作[OP_1 , OP_2]。

(3) 操作 OP 往复指：连续多次执行操作 OP，即<OP,<OP,<.....>>>。有两种记号方法：①记为：OP*，表示 OP 是一个往复操作，我们只关心操作的组合方式，不关心组合次数；②记为：OP^n，表示 OP 是一个 n 次往复操作，我们关心操作的组合方式次数。特别：当 n=1 时 OP^n 是一个原子操作，即 OP。

定义 3.6(空操作)：

为解决安全识别问题，我们引入空操作概念。所谓空操作：指系统不对任何客体进行操作，只是从一个状态转换到另一个状态。

定义空操作的三种记号方式如下：

① OP_b^a 表示一个由状态 a 转移到状态 b 的空操作。如 $OP_{\overline{HALT}}^{\overline{WAIT}}$ 表示系统从等待状态转移到停机状态。

注意：在空操作中，我们同时引入扩展操作名，这些操作不再是上文提到的操作原语，而是指系统内部封装的操作。这些操作对主体不可见，从安全角度看，我们认为这些操作是可信的。为区别起见，这些操作名为带上划线的大写字母表示，如 $\overline{HALT}$ 。

② $OP(\overline{NAME},\varepsilon,\varepsilon)$ 表示一个空操作，我们关心的只是该操作转换结果。

③ OP^0 表示一个空操作，我们只关心操作形式，而不关心操作方式。通常该表示用于抽象操作。也可以记为ε_{OP}。

约定 3.2(规则的符号描述)：

(1) $f :\subseteq a_1 \times a_2 \times ... \times a_n \rightarrow a_0$ 表示名为 f 的规则，a_i (i=1,2,...,n)为规则的约束条件，a_0 为规则的作用结果。在不产生歧义的情况下用(f)表示规则名。

(2) 用 $f_{[a_1,a_2,...,a_n]}^{[A_1,A_2,...,A_n]}$ 表示规则的实例，其中 A_i 是约束条件 a_i 的值。在不发生歧义的情况下，简写为 $f[A_1,A_2,...,A_n]$。

(3) 用 $f(A)\vdash B$ 表示对 A 施加规则 f 得到结果 B，A,B 为客体。可以简写为 A→B (f)。

(4) . 表示规则运算。用 f.g 表示规则 f 和规则 g 的复合运算，即 f.g(a)=f(g(a))，表示对 a 施加规则 g 后，再施加规则 f。

定义 3.7 (安全测试函数 TEST)：

该函数测试操作是否满足安全规则，如果结果为真，则转到下一个操作；如果为假，则终止操作。即：

$$TEST(OP, Rule) = \begin{cases} \overline{HALT} & if \quad false \\ MOVE & if \quad true \end{cases}$$

在不发生歧义的情况下，省略规则 Rule，简写为 TEST(OP)。

定义 3.8(虚码)：

我们引入虚码的概念。所谓虚码指对数据库中的码，尤其是关系元组中的码进行处理，得到对元组或对象进行唯一标识的码值。虚码对应于 VISTA 中的内码、外码和对象标志码。虚码由码和其密级构成，虚码的码值等于：VISTA 码值+密级。

TDM 规定记录必须定义虚码。如果元组或对象未定义码，则 VISTA 指定默认虚码，以顺序码作为虚码。

约定 3.3(码的记号描述)：

用 KEY(a)标志客体 a 的码，用 iKEY(a)表示元组 a 的内码，用 oKEY(a)表示元组 a 的外码，用 fKEY(a)表示对象 a 的标识码，用 vKEY(a)表示客体 a 的虚码。

3.2.2.2 TDM 安全规则

TDM 安全模型定义了 10 组安全规则，从安全定义、数据安全访问、数据完整性、冲突协调四个方面定义了 TDM 中安全操作需要遵循的规则。安全进程控制根据这 10 组规则对数据库操作进行控制，以实现数据库安全。

安全定义主要给出如何定义数据库中各要素的安全属性，包含角色定义规则(Role Definition Rule)、密级定义规则(Security Level Definition Rule)、授权规则(Authorization Rule)等三个规则组。这三组规则是其他规则的基础，定义了数据库主体和客体的安全属性，以便其他规则可以进行安全甄别。

数据安全访问主要给出主体对客体的访问限制规则，包含写规则(Write Rule)、读规则(Read Rule)、建规则(Create Rule)、视图重组规则(View Rule)、操作迁移规则(Operate Migration Rule)等五组规则。这五组规则是约束检测、引用检测、视图监视三个安全进程进行安全控制的依据。

数据完整性是指数据的正确性和相容性，通过完整性约束条件的规定和检查来实现的。数据完整性定义规则(Data Integrity Definition Rule)组给出了约束条件定义的规则。

冲突协调指当安全甄别引用多条安全规则时，各规则间出现安全约束不协调的情况，如何选取适合的安全规则进行安全甄别。我们通过冲突协调约定(Clash Coordination Treaty)组给出。严格意义上，这不是安全规则，只是约定，但为保证安全规则的完备性，我们将其放在规则定义中。

3.2.2.2.1 角色定义规则(R)

角色分组规则(R1)：允许对角色定义分组，即：$R1 \subseteq R \rightarrow dom(R)$。如果角色 R_1 和 R_2 在同一角色组，则 $dom(R_1)= dom(R_2)$

特权角色规则(R2)：将[illegible]

$$R2 :\subseteq \underline{R_1} \times \underline{R_2} \rightarrow dom(P_{\underline{R_1}}) \cap dom(P_{\underline{R_2}}) = \phi$$

权力分配规则(R3)：包含两条规则：①初始化角色时，角色继承所在角色组的权限；②只有特权角色可以访问或操纵角色的权限。即：

(R3-1) $R3-1 :\subseteq \overline{INITIALIZE}(R) \rightarrow P_R := P_{dom(R)}$

(R3-2) $R3-2 :\subseteq \underline{R_1} \times R_2 \times P_{R_2} \rightarrow P'_{R_2}$

角色会话规则(R4)：包含两条规则：①具有相同角色组的主体间允许会话；②普通角色主体对特权角色主体进行盲会话，即特权角色主体不可见。规则描述如下：

(R4-1) $R4-1 :\subseteq M_1 \times M_2 \times (dom(R_{M_1}) = dom(R_{M_2})) \rightarrow \overline{TALK}(M_1, M_2)$

(R4-2) $R4-2 :\subseteq M_1 \times M_2 \times \underline{R(M_1)} \rightarrow \overline{TALK}(M_2, \overline{HIDE}(M_1))$

注意：此处的“会话”与传统的角色会话含义不同，传统“会话”的含义是根据用户的要求负责将它所代表的用户映射到多个角色去；而 VISTA 中的“会话”是指：主体[用户]间允许进行直接信息交流的通道，主体间允许通过“会话”进行留言或发送请求。

3.2.2.2.2 密级定义规则(S)

主体密级规则(S1)：只有特权角色主体可以定义或操纵主体的密级。即

$$S1 :\subseteq M_1 \times \underline{R_1(M_1)} \times M_2 \rightarrow S(M_2)$$

客体初始化密级规则(S2)：包含四条规则：①客体继承宿主主体的密级；②子类客体继承所有父类客体密级的最低密级；③对象实例客体继承对象类的密级；④密级协调规则，当由规则(S2-1)、(S2-2)或(S2-3)组合定义客体密级，所得密级不一致时，以最低密级作为客体密级。规则描述如下：

(S2-1) $S2-1 :\subseteq \overline{INITIALIZE}(O) \times \underline{M_o} \rightarrow S(O) = S(M_o)$

(S2-2) $S2-2 :\subseteq \overline{INITIALIZE}(OC_0) \times \bigcup_i OC_{i\ father(OC_0)} \rightarrow S(OC_0) = \min_i (S(OC_{i\ father(OC_0)}))$

(S2-3) $S2-3 :\subseteq \overline{INITIALIZE}(OD) \times OC_{OD} \rightarrow S(OD) = S(OC_{OD})$

(S2-4) $S2-4 :\subseteq \overset{3}{\underset{i=1}{\times}} S(OD)_{(S2-i)} \rightarrow S(OD) = \overset{3}{\underset{i=1}{\min}} S(OD)_{(S2-i)}$

客体密级修改规则(S3)：包含两条规则：①客体密级可以由宿主主体修改，且新密级高于原密级及其主体密级；②客体密级可以由特权角色主体修改。规则描述如下：

(S3-1) $S3-1:\subseteq S(O)\times M_o \rightarrow S'(O)\cap(S'(O)\geq \max(S(O),S(M_o)))$

(S3-2) $S3-2:\subseteq M_1\times \underline{R_1(M_1)}\times S(O)\rightarrow S'(O)$

面特性规则(S4)：组合客体的密级不高于其组成客体的密级。即：

$$S4:\subseteq O\times(O=\bigcup_i O_{i_o})\rightarrow S(O)\leq \min_i(S(O_{i_o}))$$

规则(S4)是数据库密级定义的一个重要属性，如：记录的密级不高于构成该记录的所有字段值的密级；对象实例的密级不高于其属性和方法的密级。该规则也可称为密级敏感性规则，即集体的成员至少和集体一样敏感。

3.2.2.2.3 授权规则(G)

角色授权规则(G1)：具有相同角色组的主体间允许授权，受权方只有使用权，而无拥有权。即：

$$G1:\subseteq M_1\times M_2\times(dom(R(M_1))=dom(R(M_2)))\times P_{M_2}\rightarrow \overline{GRANT}(M_1,P_{M_2})$$

特权授权规则(G2)：特权角色主体可以对普通角色主体授权，授权成功后，该权利为普通角色的所有。即：

$$G2:\subseteq M_1\times M_2\times \underline{R(M_1)}\times P_{M_1}\rightarrow \overline{GRANT}(M_2,P_{M_1})\cap P_{M_2}$$

单级授权规则(G3)：TDM 只支持单级授权，即只有权力所有者可以授权，权力接受者不得转授。即

$$G3:\subseteq M_1\times M_2\times(dom(R(M_1))=dom(R(M_2)))\times P_{M_3}\rightarrow \overline{GRANT}.False.$$

权力回收规则(G4)：包含两条规则：①权力宿主对权力回收；②特权角色主体对权力回收。规则描述如下：

(G4-1) $G4-1:\subseteq M_1\times M_2\times P_{M_2}\rightarrow \overline{REVOKE}(M_1,P_{M_2})$

(G4-2) $G4-2:\subseteq M_1\times M_2\times \underline{R(M_1)}\times P\rightarrow \overline{REVOKE}(M_2,P)$

3.2.2.2.4 写规则(Wr)

主体 M 对客体 O 进行写操作，必须满足两个条件：①M 对 O 有写访问权；②M 的密级不高于 O 的密级，或称 M 的密级被 O 的密级所支配。该规则可以描述为：

$$Wr:\subseteq M\times O\times P_{WRITE(M,O)}\times(S_M\leq S_O)\rightarrow OP(write,M,O)$$

3.2.2.2.5 读规则(Re)

主体 M 对客体 O 进行读操作，必须满足两个条件：①M 对 O 有读访问权；②M 的密级不低于 O 的密级，或称 M 的密级支配 O 的密级。该规则可以描述为：

$$\mathrm{Re} :\subseteq M \times O \times P_{\overline{READ(M,O)}} \times (S_M \geq S_O) \rightarrow OP(read, M, O)$$

3.2.2.2.6 建规则(Cr)

主体 M 对客体 O 进行建操作，只需满足主体 M 具有创建客体的权力。该规则可以描述为：

$$Cr :\subseteq M \times P_{\overline{CREATE}} \rightarrow OP(create, M, O) \times P_{\overline{WRITE(M,O)}} \times P_{\overline{READ(M,O)}}$$

注意：主体 M 是客体 O 的宿主，M 对 O 拥有读和写的权力，并且是这两个权力的宿主。同时，根据规则(S2-1)，此时被创建客体的密级等于主体的密级，即 *S(O)=S(M)*。

3.2.2.2.7 视图重组规则(V)

视图元组密级规则(V1)：构成视图的元组密级最大上界不超过访问视图的主体密级。即：

$$V1 :\subseteq View(a) \times M \rightarrow \max_i(S(\mathrm{Re}cord(b_i)_{View(a)})) \leq S(M)$$

注意：根据(S4)这些元组中可能存在高于主体密级的字段值。

最大满足性规则(V2)：在(V1)的前提下，满足主体访问请求的所有元组应当在视图中全部出现，即使存在多实例的情况。规则描述为：

$$V2 :\subseteq View(a) \times M \rightarrow \forall \mathrm{Re}cord(b)_{View(a)}$$

部分隐蔽规则(V3)：视图中出现高于主体密级的客体信息是隐蔽的，即视图的原子组成的密级被主体密级支配。即：

$$V3 :\subseteq View(a) \times M \times O_{View(a)} \times (S(O) \succ S(M)) \rightarrow \overline{HIDE}(O)$$

全部隐蔽规则(V4)：视图中的一个元组的所有字段值的最小密级高于主体密级，则该元组对主体不可见，即视图中不出现“NULL”元组。即：

$$V4 :\subseteq View(a) \times M \times \mathrm{Re}cord(b)_a \times Field(c_i)_b \times (\min_i S(c_i) \succ S(M)) \rightarrow \overline{HIDE}(b)$$

3.2.2.2.8 操作迁移规则(M)

操作迁移规则主要定义，操作序列在安全检测控制下的运作进程是否中断，该组规则对操作组合的安全检测方式给出定义。

组合操作测试规则(M1)：其规则如下：

(M1-1) $M1-1 :\subseteq TEST(< OP_1, OP_2 >) \rightarrow TEST(OP_1) \times TEST(OP_2)$

(M1-2) $M1-2 :\subseteq TEST([OP_1, OP_2]) \rightarrow TEST(OP_1 \times OP_2)$

(M1-3) $M1-2 :\subseteq TEST(OP^*) \rightarrow (TEST(OP))^*$

注意：(M1-2)的含义是：并行操作，对并行操作的结果作检测。

空操作测试规则(M2)：其规则如下：

(M2-1) $M2-1:\subseteq TEST(\{OP,OP^0\})\rightarrow TEST(OP)$

(M2-2) $M2-2:\subseteq TEST(\{OP^0,OP\})\rightarrow TEST(OP)$

其中：{}为抽象的操作组合。

3.2.2.2.9 数据完整性规则(I)

多级实体完整性规则(I1)：该规则定义在多级安全中元组或对象的标识值的密级定义规则。组成规则有三：

①虚码值不为空；

②虚码密级为最低级，从而，不同虚码具有相同密级；

③客体密级与虚码的密级无关，但遵循规则(S4)，即虚码不参加密级运算，但码参加密级运算。

规则描述如下：

(I1-1) $I1-1:\subseteq vKEY(a)\rightarrow VALUE(a)\neq NULL$

(I1-2) $I1-2:\subseteq vKEY(a)\rightarrow S(vKEY(a))=\min(dom(S))$

$corollary \quad I1-2':\subseteq vKEY(a)\times vKEY(b)\rightarrow (S(vKEY(a))\equiv S(vKEY(b)))$

(I1-3) $I1-3:\subseteq S(O)\times vKEY(a_O)\rightarrow S(O)$

参照完整性规则(I2)：如果一个外码在给定密级是可见的，则包含所参照主码的客体在此密级也必须是可见的，并且外码元素的密级必须支配所参照主码诸元素的密级，即访问必须是按照密级向下访问。描述如下：

$$I2:\subseteq S(oKEY(a))\rightarrow \bigcup(O_{iKEY(oKEY(a))}\times(S(iKEY(oKEY(a)))\leq S(oKEY(a))))$$

多实例规则(I3)：

系统中多实例现象可划分为如下三种：

①多实例关系(类)：指具有相同关系(类)名但其模式(定义)具有不同密级的多个关系(类)；

②多实例元组(对象)：也称实体多实例，指具有相同主码，但其主码密级不同的多个元组(对象)；

③多实例元素：也称属性多实例，指一个属性中具有不同密级但与相同主码和码密级相联系的多个元素。

VISTA 中允许第二中多实例的情况出现，该情况下多实例规则(I3)有二：①多实例由虚码唯一标识；②主体密级支配多实例对象密级，即按密级向下访问。即：

(I3-1) $I3-1:\subseteq(\overline{VALUE}(KEY(a))=\overline{VALUE}(KEY(b)))\times(S(KEY(a))\neq S(KEY(b)))\rightarrow(vKEY(a)\neq vKEY(b))$

(I3-2) $I3-2:\subseteq S(M)\times KEY(a)\rightarrow\bigcup(O_{KEY(a)}\times(S(O_{KEY(a)})\leq S(M)))$

3.2.2.2.10 冲突协调约定(C)

冲突协调约定主要是在系统可以使用多个安全规则对操作安全性进行检测时，选取安全规则的约定。这些约定只在系统无法选取安全规则时采用。约定如下：

①强规则约定(C1)，当侯选规则的约束条件中存在主体，以主体密级最高的规则为检测规则；

②弱规则约定(C2)，当(C1)无法判别，以约束条件中客体集中含元素最多的规则为检测规则；

③零规则约定(C3)，当(C2)无法判别，系统对规则进行随机选择。

3.2.3 TDM 模型性质的评价

对安全模型的评价通常采用两种方法：攻击检测和性能评价。攻击检测通过审计试图登录的失败记录和试图连接特定资源的失败记录来发现外部和内部攻击，通过主体间的授权、会话、对客体的访问审计来甄别权利滥用者。攻击检测的有效性在很大程度上建立在对已知的攻击信息的知识的数据库的丰富上，同时，还依赖检测结果提供时机的实时性上。鉴于 VISTA 系统尚处于开发测试阶段，攻击检测法难以全面有效地反映 TDM 模型的可行性和完备性。而使用 A 级所要求的形式上生成或形式证明 VISTA 的安全程度，目前也是难以实现的。但我们可以通过 TDM 安全模型与传统安全模型的关系，从性能上对其进行评价。

如果通过对 TDM 安全规则的组合使用，可以实现传统模型 A 的安全控制，则可以认为 TDM 模型对模型 A 是兼容的，TDM 安全模型至少可以达到模型 A 的安全要求。我们通过对 TDM 模型与部分传统模型的兼容关系的陈述，简要说明 TDM 模型的可行性。

[1]对存取矩阵模型的兼容性

根据 HRU 存取矩阵模型，其核心是授权状态三元组{主体，客体，存取矩阵}。TDM 模型中定义的主体和客体与 HRU 中的主客体定义一致。同时，根据规则组(S)、(Wr)、(Re)、(Cr)生成存取矩阵，(S)定义了主客体的密级，(Wr)、(Re)、(Cr)生成主体对客体的操作授权(在 TDM 中解释为主体是否可以对客体执行操作原语)。

由此，可以推导出 TDM 模型对存取矩阵模型兼容。

[2]对贝尔-拉帕杜拉模型的兼容性

贝尔-拉帕杜拉模型的核心由三条基本安全规则组成，可简述如下：

①No read-up secrecy，即一个主体仅能读取其安全级别受此主体安全级别支配的客体信息。在 TDM 模型中，(Re)规则与之对应。

②No write-down secrecy，即一个主体仅能向其安全级别支配此主体安全级别的客体写信息。在 TDM 模型中，(Wr)规则与之对应。

③Tranquility Principle，即没有主体可以修改激活客体的密级。该规则在以后的模型版本中被去掉。但控制密级改变的规则在安全模型中是有用的，虽然它可能会依赖于特定的应用。在 TDM 模型中，(S3)定义了密级修改规则。

④ Discretionary Security Property，即一个主体只能在获取了所需的授权后才能执行相应的存取。该规则称为自主安全规则，与 HRU 存取矩阵模型的授权状态相兼容，根据上面的讨论，TDM 模型可以通过规则组生成存取矩阵与该规则保持一致。

在此基础上，可以推导出 TDM 模型对贝尔-拉帕丢拉模型模型兼容。

[3]对安全数据视图模型的兼容性

安全数据视图模型分为两层，下层为强制存取控制模型(MAC)，相当于实施贝尔-拉帕丢拉模型的强制安全策略的访问控制器；上层是可信计算基(TCB)，定义了多级关系的概念，支持对多级关系和视图的自主存取控制。

TDM 模型通过三个安全约束控制进程，即约束检测、引用检测、视图监视实现 MAC 的基本功能。TDM 中的五组数据安全访问规则与安全数据视图模型的强制存取控制策略对应。安全数据视图模型中的 TCB 核心部分可以由 TDM 模型中的(V)和(I)两个规则组和(S4)规则来体现。

这样，安全数据视图模型的两层结构在 TDM 模型中均可以得以实现，故 TDM 模型对安全数据视图模型也存在兼容性。

通过对上述三个常见安全模型的兼容性通论，可以得出一个结论，TDM 安全模型是可行的。

同时，我们知道一个数据库系统当满足下面四个条件时，被认为是安全的：

①不存在一个主体能不通过授权而获得信息；

②不存在一个主体能不通过授权而修改信息；

③不存在一个机制，在该机制中一个授权主体可以与未被授权的主体进行信息交流；

④不存在一个主体能不通过授权而激活一个方法。

根据 TDM 安全规则定义，可知：规则(R3)、(G)、(Re)、(Wr)、(Cr)约束了条件①②④；规则(R4)、(Re)、(Wr)约束了条件③。因此，支持 TDM 模型的数据库系统是安全的。

第四章 VISTA 数据库安全设计

在第三章我们定义了 VISTA 所遵循的安全模型，借助 TDM 模型可以设计一个安全对象关系数据库。本章主要讨论在 TDM 模型的基础上，安全 VISTA 的设计方案。

本章从数据库的安全存储、安全模式、访问控制和审计设计四个方面全面介绍 VISTA 安全设计方案。

4.1 安全存储设计

数据存储是数据库安全的一个重要环节，由于数据库的独立性，数据与应用环境、访问语言是相分离的。独立性的特征使得数据具有更为灵便的使用价值，同时，也带来了一定的安全隐患。如传统的小型关系数据库 dBASE、FoxBase 等，由于其数据库在存储机制上未进行安全控制，用户可以借助程序设语言不需要通过 DBMS 直接访问数据库中的数据，这样常常带来信息泄漏。在 VISTA 中，我们首先解决数据库独立性可能带来的安全隐患。

解决这一隐患的方法通常有两个策略：[1]对数据进行加密处理；[2]对数据存储的空间关系进行运算，而非采用顺序存储。第一个策略使用密码学原理使得非合法用户无法正确对数据进行解释，从而实现数据安全；第二个策略对数据的空间关系进行重组，在不知道数据空间分布算法的情况下，非法用户无法恢复原有的数据的空间关系，也就无法获得全部数据信息，因此无法对数据进行正确的解释。

这两个策略的安全保障在于，涉及的算法是与 DBMS 相关的，即算法存在于 DBMS 中，借助独立性的访问是不能够越过算法直接给出数据的正确解释的，也就是说，数据的解释必须由 DBMS 来实现。

在 VISTA 中，我们使用了这两个策略对数据库的数据存储安全进行了控制。两种策略实现所用的算法由 DBMS 系统给出，与数据文件分离，VISTA 安全数据存储由三部分文件组成：数据目录、数据文件、索引文件。

数据目录是数据文件的清单，其数据结构为五元组{ID, FILE_NAME, FILE_TYPE, FILE_NUMBER, FILE_MASTER}其中：ID 为顺序号；FILE_NAME 为文件名；FILE_TYPE 为数据文件类型，如库文件、自由表等；FILE_NUMBER 构成该数据文件的子文件数，即一个数据文件可能由若干个子文件组成；FILE_MASTER 为文件的宿主用户名。用户可以通过对数据目录的检索访问到具体的数据文件。

数据文件是对象的存储，由三部分组成：文件头、数据区、[illegible]

示意图见图 4.1。

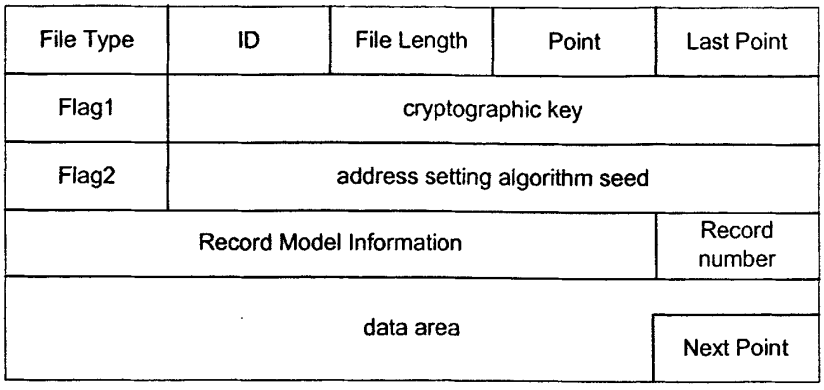

图 4.1 VISTA 数据文件结构示意图

文件头由 11 部分构成：File Type 为文件类型标志；ID 为子文件的顺序号，即该子文件是数据文件的第几个子文件；File Length 为该数据子文件的长度；Point 为数据区的尾指针；Last Point 为上一个子文件地址/名，当其值为"NULL"时，表示该文件为第一个子文件，即该数据文件的起始文件；Flag1 标志，是否使用数据加密控制；cryptographic key 加密算法的密钥；Flag2 标志，是否使用空间组合变换控制；address setting algorithm seed 空间组合变换算法的种子数；Record Model Information 为记录的模式信息；Record number 为数据文件包含的记录数。

数据区：存储经过数据安全存储控制处理的数据值。

尾指针 Next Point 指向下一个子文件，当其值为"NULL"时，表示该文件为最后一个子文件。

索引文件是对数据进行查询的一个重要工具，索引算法是数据库计算的一个重要研究内容，我们仅研究索引文件的数据结构，相关索引算法可参看数据库的相关资料。VISTA 的记录索引可以用四元组表示：{ID, key, re_id, address}。其中 ID 为索引顺序号；key 为索引字；re_id 为记录号；address 为记录存储的入口地址，用于对记录数据的访问。

通过数据目录、数据文件、索引文件三种文件可以实现 VISTA 数据的安全存储。图 4.2 给出了在这三种文件基础上的三个操作原语的执行流程。

图[a]是建操作原语的操作流程。该流程由 DBMS 生成密钥和空间组合变换种子，生成子文件数和子文件的长度，并在这些初始数据的基础上建立子文件，改写数据清单，建立索引文件；

图[b]为写操作原语的操作流程。该流程将记录进行加密；并将加密的记录数据进行空间组合变换产生的存储信息，将记录分解为若干数据块，分散存储到不同存储区域(存储区域可以在同一个子文件中，也可以在若干个子文件中)。同时，该流程还根据记录信息和记录的存储空间地址建立索引信息。在该流程中，还有

一个操作细节，当数据存储空间不足时，根据一定的算法，VISTA 系统可以在初始的子文件的基础上，追加子文件，称为扩展子文件。

由于在 VISTA 系统中未给出“删除”操作原语，删除操作由写操作完成，当写操作向文件写入"NULL"时，等价于传统的删除操作。此时，写操作原语流程还需要甄别扩展子文件是否为空，若为空则使用操作系统的删除指令删除子文件，并改写文件清单的相关值。

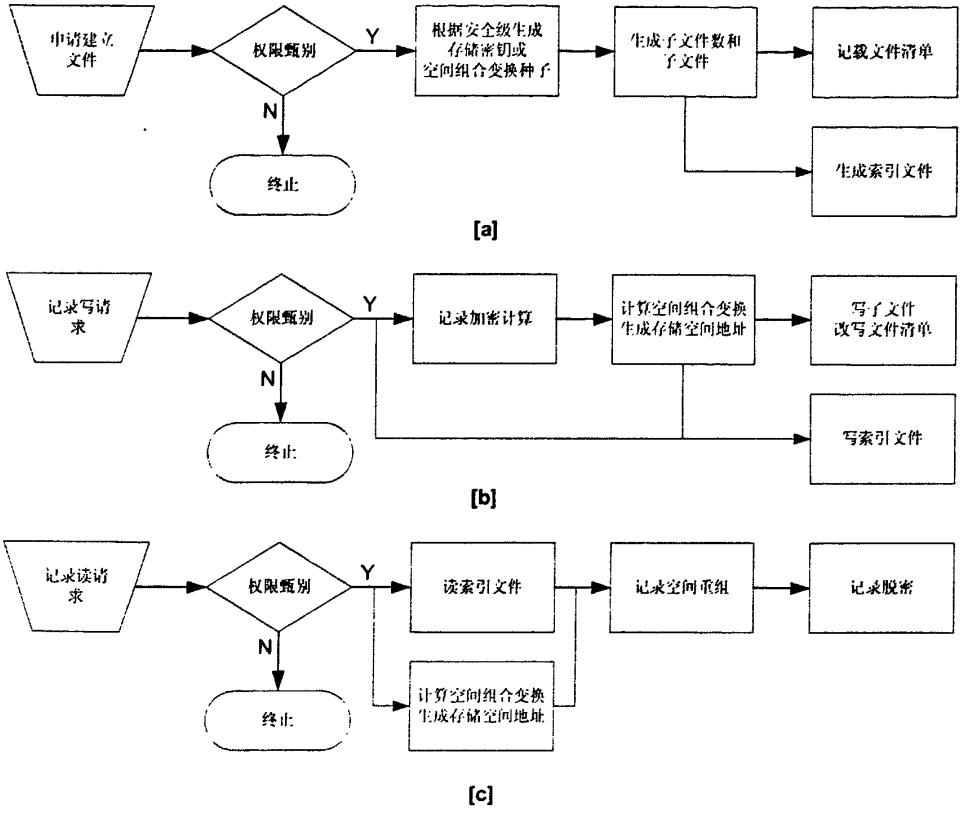

图 4.2 VISTA 数据安全存储的操作原语的操作流程

图[c]是读操作原语的操作流程。该操作流程根据索引提供的记录的入口地址信息和空间组合变换算法得到的存储信息，将分散的记录数据块进行重新拼装，即记录重组，然后，再将重组后的记录进行脱密，供合法用户使用。

VISTA 系统的安全存储设计主要解决数据库内模式(物理模式)的安全问题。根据 VISTA 的安全存储设计方案，可以克服由数据库独立性带来的信息泄漏的隐患。

4.2 数据库模式中安全属性设计

数据库模式描述了数据库的数据结构，给出了数据表的元组结构。模式通常在数据字典中定义，通过对元数据及其关系的定义，确定数据表的模式，决定数据库的逻辑关系。在安全数据库中，可以在模式中定义数据库数据的基本安全属性。

当我们对数据库进行安全控制时，安全粒度是一个重要的指标，粒度大小反映了安全控制可以达到的最小数据范围。数据库安全粒度大小通常有三种：文件(数据表)、记录(元组/对象)和记录属性(字段/对象属性及方法)。VISTA 中安全粒度定位为：关系型数据以记录属性作为粒度，对象型数据以对象的属性及方法作为粒度。

在传统的数据库模式定义基础上，我们扩展了安全属性的定义。

定义 4.1： VISTA 模式定义如下：

①{记录号，字段列表，密级}

②{字段名，字段类型，字段长度，码标识，默认值，密级}

③{对象号，对象名，对象属性列表，对象方法列表，密级}

④{对象属性号，数据类型，数据长度，默认值，密级}

⑤{对象方法号，方法参数列表，密级}

定义中①给出了记录的定义，②给出了记录属性的定义，③给出了对象的定义，④给出了对象属性的定义，⑤给出了对象方法的定义。各定义中的“密级”给出了各数据对象的安全属性，VISTA 根据这些安全属性值给出的安全密级进行安全控制。

由于，组成模式的各数据对象存在包含关系，根据安全规则中的面特性规则(S4)可知：

①记录密级 <= 字段密级

②记录密级 <= 对象密级

③对象密级 <= 对象属性密级

④对象密级 <= 对象方法密级

需要注意的是，VISTA 是对象关系型数据库，其记录的字段的数据类型除了基本数据类型外，还可以是对象型的；一个对象属性也可以是另一个对象。针对这两种情况，我们规定情况中出现具有对应关系的数据对象的密级是相同的。

在该数据库模式定义下的数据存储的逻辑结构如图 4.3 所示。

由图可以数据对象及其密级同时保存。字段 2 为对象型数据，其值为指向相关对象的指针，字段 2 的密级与对象密级相同。"#"表示对象属性域的终止，"##"表示给对象域的终止。

定义 4.1 给出了 VISTA 数据字典中与数据库模式定义相关的元数据的定义方法，图 4.3 给出了数据表结构

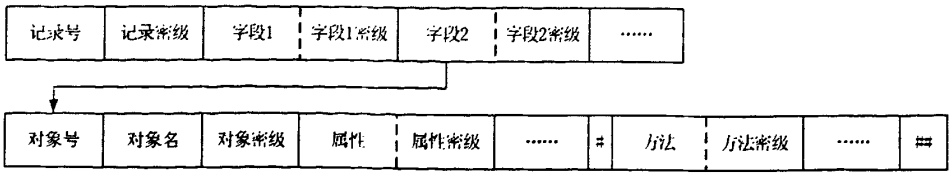

图 4.3 VISTA 数据逻辑存储结构

4.3 安全访问设计

我们上节讨论了 VISTA 系统的内模式、模式的安全设计问题，对于子模式即视图的安全设计，可以根据视图重组安全规则组(V)进行设计，限于篇幅，本文不再叙述。

在数据库模式安全设计的基础上，本节主要讨论对数据的安全访问。VISTA 安全访问由自主安全访问、强制安全访问和角色控制三种机制综合而成。我们将分别进行阐述。

4.3.1 自主安全控制设计

自主安全控制其核心是：用户对资源对象“拥有”权限，系统通过检查用户对资源的特权，决定用户是否可以访问资源。自主安全控制三个部分构成：用户权限集、授权机制、访问甄别。用户权限集定义了用户对资源的“拥有”关系，通常通过用户权限矩阵实现，该部分是自主安全控制进行安全甄别的依据；授权机制指用户可以自主地将他说拥有的权限传授给其他用户；访问甄别指自主安全控制进行安全甄别的策略。

4.3.1.1 用户权限矩阵设计

典型自主安全控制使用存取矩阵[11]进行安全访问控制。存取矩阵模型是一个状态机模型，将系统的安全状态描述为一个大型的矩形阵列，在此矩阵中，行表示系统的主体，列表示系统的主体和客体。阵列的每个单元填入的数值，表示主体对客体或其他主体的存取方式。

在 VISTA 中，借鉴存取矩阵模型设计用户权限矩阵，该矩阵可由三元组表示：Q=(M,O,P)，其中 M 为用户、O 为资源、P 为权限。用户权限矩阵的行表示 DBMS 中的用户，列表示数据对象，阵列的每个单元的值为用户对数据对象的权限。图 4.4 给出了用户存取矩阵的逻辑存储结构。

{M}为用户集，由特权用户和普通用户组成。在 VISTA 中，特权用户包括系统管理员、系统安全员和系统审计员三种，这三种用户映射为三种特权角色，根据特权角色规则(R2)，这三种用户对数据资源的权限是互斥的。特权用户可以访问系统中的所有数据资源，只有特权用户可以访问的数据资源称为系统表，如用户权限表、审计日志等。特权用户以外的用户为普通用户，根据用户的职责可

映射为各种角色，普通用户只能访问用户定义的数据表，不能访问系统表。

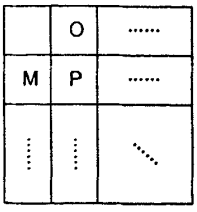

图 4.4 VISTA 用户权限矩阵的逻辑存储结构

O 为资源集，其元素根据其存在方式可分为两类：静态数据对象和动态数据对象。静态数据对象指系统初始化时产生的数据集，如数据字典、系统表等；动态数据是指系统在使用过程中，用户定义产生的各种数据对象，如数据表、对象集等。在系统使用过程中，动态数据是可以增减的，而静态数据不可以增减。

P 为权限集，定义了用户可以施加于资源的动作的集合，如读(read)、写(write)、建(establish)、修改(modify)、删除(clean)、销毁(destroy)、禁用(forbid)等。在 VISTAL 中，P={{A},T)，A 为动作列表，T 为动作的有效时间。值得说明的是，动作对于不同的对象其实际含义存在差异，如“建”操作对于数据表，则为根据数据表建立视图；而对于对象，则为建立新的动作。动作含义的解释是由系统的规则库给出的。

图 4.5 给出了一个用户权限矩阵的例子。该例子中定义了两个用户，user_1 是审计员，为特殊用户，user_2 是普通用户。定义了 4 个资源：DD 是库结构数据字典(见§3.1.2 中图 3.4)，User_P 是用户权限表，List_1 为一个自由表，Object_1 为一个对象。其中 DD、User_P 为静态数据对象；List_1、Object_1 为动态数据对象。表中符号“-”表示不赋予权限，即不可访问；符号“a”表示所有，当 A="a"表示赋予所有权限，当 T="a"表示在任意时间均可访问。

	DD	User_P	.....	List_1	Object_1	.....
User_1	- -	M a	.....	R a	R a	.....
User_2	W a	- -	.....	REC 030101- 040101	a a	.....
.....	.....	.....	.....	.....	.....	.....

图 4.5 用户权限矩阵的例子

根据图 4.5，我们可知 User_1 对 User_P 在任意时刻拥有修改权；User_2 对 List_1 在 2003 年 1 月 1 日到 2004 年 1 月 1 日期间拥有读、建和删除的权限。User_2 对 DD 在任意时刻拥有写权限，即 User_2 可以在 DD 中写入数据，意味着 User_2 可以创建数据库、自由表等。

用户权限矩阵的建立和修改遵循安全规则(G2)，即用户的权限由特权用户赋

予和管理，矩阵中用户拥有的权限是拥有权。为了简化矩阵的赋值，约定宿主，即数据对象的创建者，对该对象的初始权限为“读”和“写”，其他权限(除 forbid 外)由系统安全员赋给。

“禁用”权限由系统审计员对用户进行实施。

4.3.1.2 权限的授予和回收设计

授权管理，是自主访问控制的一个重要组成部分，使得主体可以将自己拥有的权限进行扩散，满足更为广泛的数据共享需求，同时，不合理的授权行为也会造成信息的泄漏。VISTA 在安全规则中定义了严格的授权规则(见 § 3.2.2.2.3)，在这组规则的限定下，我们对自主访问控制的授权管理进行设计。

一、授权定义

授权可以用三元组定义：{M1,M2,{G}}。M1 是授权者，M2 是受权者，{G}权力列表，即 M1 将权限集合{G}授予 M2。当然，该授权应遵循安全规则(G1)。

权力 G 可以描述为 G={O,{P}}，O 为对象，{P}为权限列表。同上，P 描述为{A,T}。

根据安全规则(G3)可知 P 对于 M2 只是使用权，而非拥有权，即 M2 不能将权限 P 再转授给其他用户。

二、授权表的逻辑存储设计

授权表记载了用户间的授权情况，由两个表构成，如图 4.6 所示。

(a)表为索引表，由三部分组成：{M,P_1,P_2}。M 为用户号，P_1 为授权记录指针，指向由 M 提供的第一个授权记录，当指针值为“NULL”表示 M 未向任何用户提供授权。P_2 为受权记录指针，指向由 M 接受的第一个受权记录，当指针值为“NULL”表示没有任何用户向 M 授权。

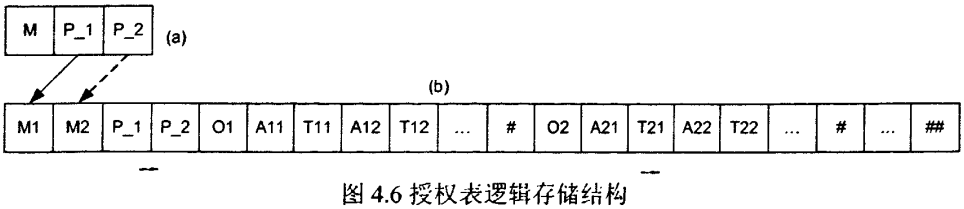

图 4.6 授权表逻辑存储结构

(b)表为授权记录表，由五部分组成：{M1,M2,P_1,P_2,{G}}，其中 M1,M2,{G}的定义同上，P_1 为授权记录指针，指向由 M1 提供的下一个授权记录，当指针值为“NULL”表示该记录为 M1 提供最后一个授权。P_2 为受权记录指针，指向由 M2 接受的下一个受权记录，当指针值为“NULL”表示该记录为 M2 接受的最后一个授权。表中“##”为一条授权记录的终止符，“#”为一个权力 G 的终止符。

根据设计，一个用户数偶的授权可以由一条记录表示。

二、授权回收设计

授权的回收是一个比较复杂的问题，在权限回收中可能会出现各种问题。为提高效率和安全性，VISTA 设计了两种回收方法：[1]自动回收，[2]强行回收。

自动回收，是根据动作 A 的有效时间 T，由系统自动回收。系统根据计时器，自动扫描授权记录表，对 T 失效的授权进行回收。

强行回收，是授权用户或特权用户根据安全规则(G4)对授权进行回收。这种回收通常出现在系统出现安全隐患和安全攻击时。由于，这种回收是人为进行的，我们将其称为强行回收。

在系统运作中，可能会出现一些与授权回收相冲突的问题，我们在此讨论最常见的 3 种情况。

[1]销毁、删除客体时授权回收管理

当客体被删除或销毁时，用户动作施与的客体不存在，用户的权限将被悬空，这时需要对权限进行回收。回收算法见图 4.7(a)，首先在用户权限矩阵中，搜索对删除客体"O" 拥有权限的所有主体集{M}；然后，根据{M}搜索授权表，将授权表中授权者为{M}的元素，客体是"O"的所有授权；最后，删除用户权限矩阵中与删除客体"O"相关的列。

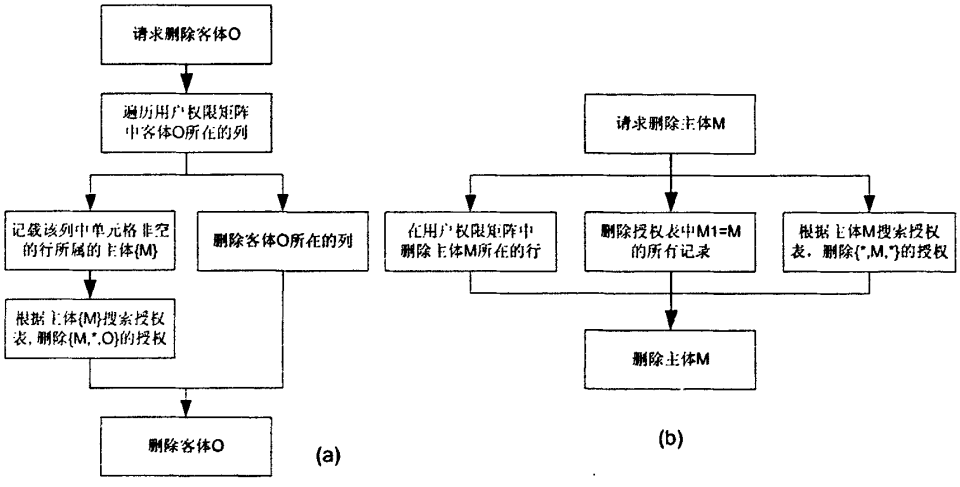

图 4.7 主客体删除时的授权回收算法

[2]删除主体时授权回收管理

当主体被删除或销毁时，部分客体将无动作的发起着，客体被悬空，这时需要对权限进行回收。回收算法见图 4.7(b)，首先删除用户权限矩阵中主体为删除主体"M"所在的行；然后，在授权表中，删除授权者 M1="M"的所有记录；最后，搜索授权表，将授权表中受权者 M2="M"的所有授权删除。

[3]事务进行中的授权回收

在事务进行中，可能会发生与事务相关的授权回收。由于此时进行授权回收主要是进行安全控制，为此，在 VISTA 中，回收算法设计为：先终止事务并将其回滚，然后进行授权回收，算法如图 4.8 所示

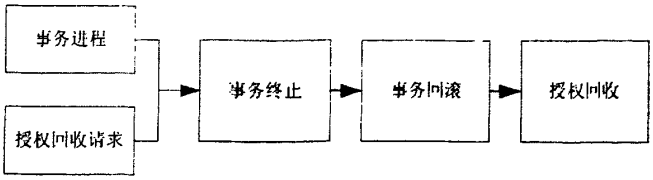

图 4.8 事务进行中的授权回收算法

4.3.1.3 自主访问甄别设计

在用户权限矩阵和授权表的控制下，对用户请求进行甄别，判别用户请求是否合法，如果合法则进入事务进程，否则终止用户请求。甄别算法如图 4.9 所示。

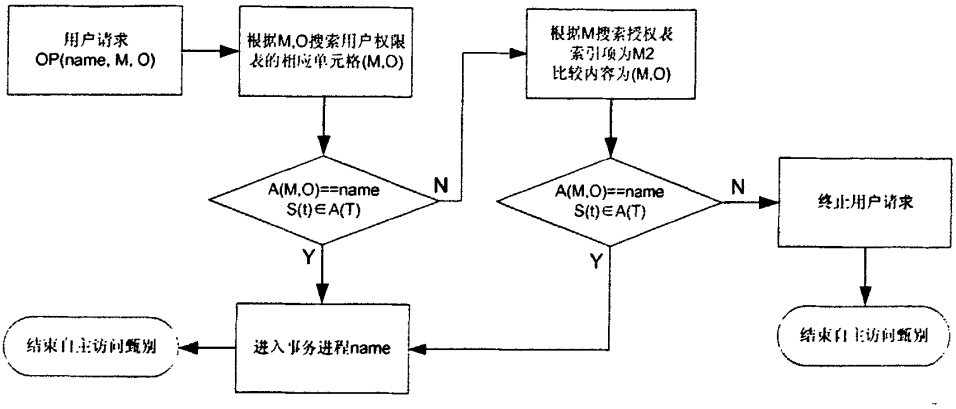

图 4.9 自主访问甄别算法

当用户对系统进行操作，发出用户请求 OP(name,M,O)，其中 name 为操作名；M 为操作主体，即用户；O 为操作对象，即客体(见§3.2.2.1)。自主访问甄别首先搜索用户权限表，搜索 M 所在的行和 O 所在的列，读取行列交叉的单元格(M,O)中的值(A,T)，当 A=name 且系统当前时间在动作有效时间 T 的区间内，用户请求是合法请求，则系统执行用户请求，进入事务进程。

当根据用户权限矩阵无法甄别用户请求是否合法时，则搜索授权表，判断用户请求是否合法。甄别算法根据主体 M，通过授权表中的 P_2 指针遍历授权表，比较各记录客体 O 是否为操作请求的对象，若是则比较(A,T)是否合法，若合法则系统执行用户请求，进入事务进程。否则，用户请求不合法，则终止用户请求。

4.3.2 强制安全控制设计

当系统的安全策略命令如下时，出现了对强制存取控制机制（MAC）的需要：[1]保护决策不是由客体的属主决定；[2] 系统必须加强保护决策。

强制安全存取控制通过无法回避的存取限制来防止各种直接和间接的攻击。在强制安全存取控制之下，系统给主体和客体分配不同的安全属性。这些属性在

单位安全策略没有改变之前是不可能被轻易改变的。系统通过对主体和客体的安全属性的匹配比较决定是否允许访问继续进行。

4.3.2.1 系统安全级的定义

强制安全访问策略是基于系统元素密级(Classification)的。VISTA 模型的密级是如下四元素集合中的任一元素：{绝密(Top Secret)，机密(Secret)，秘密(Confidential)，公开(Unclassification)}，此集合是全序的，即：

绝密 > 机密 > 秘密 > 公开

一、主体密级的定义、修改和存储

策略对系统中的每个用户分配一个安全级别，称为允许安全级(Clearance)，分配给用户的允许安全级反映了对用户不将敏感信息泄漏给不持有相应允许安全级用户的置信度，用户能以受允许安全级支配的任意安全级别向系统注册，用户激活的进程将被授予此注册安全级别。

VISTA 模型中，主体密级定义遵循安全规则(S1)(见§3.2.2.2.2)，根据规则主体密级只能由特权角色定义，其定义、修改过程如图 4.10 所示。

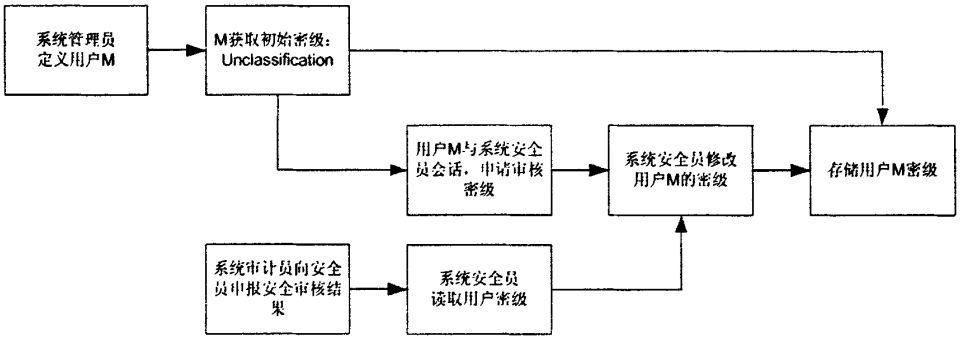

图 4.10 主体密级定义算法

当系统管理员定义一个新主体(用户)M 时，该用户获得初始化密集“公开”。主体 M 可以通过角色会话(规则 R4-2)向系统安全员申请修改密级，此外，系统审计员根据安全审计结果，要求系统安全员对主体 M 的密级进行操作。只有系统安全员才能对主体密级进行修改。

VISTA 的主体密级存储在用户权限矩阵中，我们对图 4.4 进行改造，增加主体密级一列，用于存储主体的密级，得到图 4.11。

	S	O	.....
M	S(M)	P	.....
⋮	⋮	⋮	⋱

图 4.11 带主体密级的用户权限矩阵的逻辑存储结构

二、客体密级的定义、修改和存储

策略对系统中每个客户也分配一个安全级别，客体的安全级别反映了存储在客体内信息的敏感度，也反映了未经授权向不允许存取该信息的用户泄漏这些信息造成的潜在损坏度。

VISTA 模型中，客体密级定义遵循安全规则(S2、S3 和 S4)(见§3.2.2.2.2)，其定义、修改过程如图 4.12 所示。

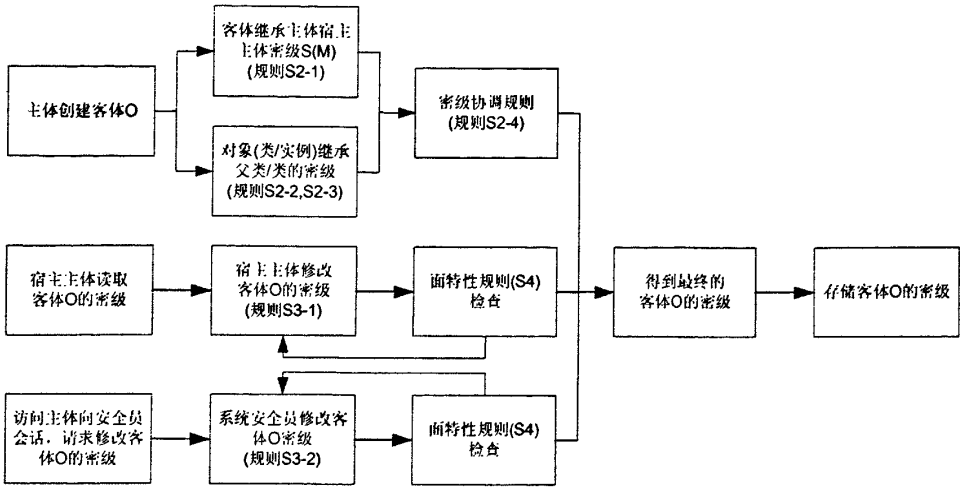

图 4.12 客体密级定义算法

根据图 4.12 可见，客体的定义和修改分三种情况，[1]创建主体，此时客体主要通过继承获得密级；[2]宿主主体修改所属客体的密级，此时，遵循密级向上修改规则(规则 S3-1，见§3.2.2.2.2)；[3]特权修改，这种情况下只有系统安全员可以修改客体密级。

在 VISTA 系统中，用户定义的客体密级存储是我们关心的问题之一，用户定义的客体可以分为：库、表、索引、记录、对象类、对象实例、字段值、对象属性值等。在§4.2 中，我们讨论了数据库安全模式存储问题，图 4.3 给出了记录、对象实例、字段值、对象属性值和对象实例的方法等客体的密级存储结构。数据库、表、对象类的密级存储定义见图 4.13。

图[a]给出了库存储结构，DB_S 定义了库的密级，TA_ID_i 给出了隶属于该数据库的表。("#"为终止符，下同)

图[b]给出了表(含自由表)存储结构，TA_S 定义了表的密级，TA_Str_P 指向构成表的第一个字段存储结构的地址，IND_ID_i 定义该表的索引序号。

图[c]给出了字段(对象属性)存储结构，字段为顺序存储，Security 定义了字段的密级，End_FLAG 定义了该字段(属性)是否为宿主表(宿主类)的最后一个字段(属性)。

图[d]给出了对象类的存储结构，CL_S 定义了类的密级，AT_Str_P 指向构成类的第一个[illegible]存储结构的地址，SE_Str_P 指向构成类的第一个[illegible]存储结

构的地址，CL_Fa_List 为该类的父类列表，其值为"NULL"表示无父类，CL_So_List 为该类的子类列表，其值为"NULL"表示无子类，IND_ID_i 定义该类表的索引序号。

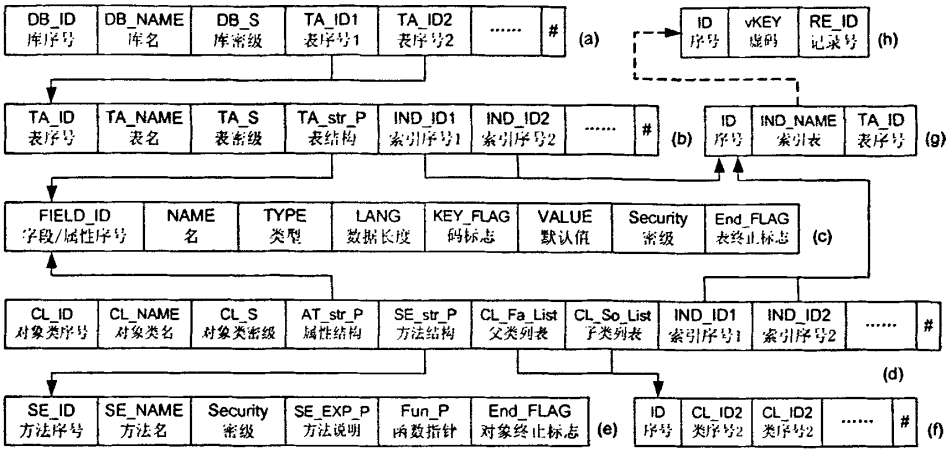

图 4.13 部分客体的逻辑存储结构

图[e]给出了类的方法的存储结构，方法为顺序存储，Security 定义了字段的密级，End_FLAG 定义了该方法是否为宿主类的最后一个字段。

图[f]给出了类列表的存储结构，CL_ID_i 为对象类序号。

图[g]给出了索引文件列表的存储结构。

图[h]给出了索引文件的逻辑结构。

图 4.3 和图 4.13 给出了 VISTA 模型中用户定义各种对象的存储结构，标志了强制访问控制所需的客体密级。

4.3.2.2 强制访问控制设计

强制访问控制通过判断主体密级与客体密级关系是否与操作规则匹配来控制操作是否合法。VISTA 中，规定了与强制访问控制相关的安全规则有写规则(Wr)、读规则(Re)和视图重组规则(V)，前两者激活客体，后者反馈访问结果。强制访问控制算法如图 4.14 所示。

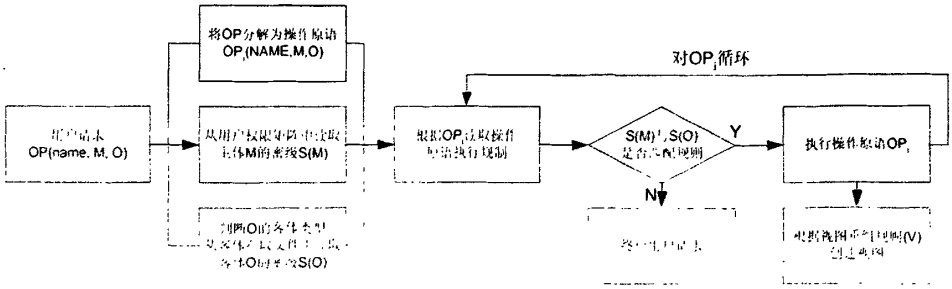

图 4.14 强制访问控制算法

强制访问控制首先将用户请求分解为操作原语请求序列，并读取相关客体和主体的密级；然后，依据操作原语请求序列逐个对操作原语请求的合法性进行甄别，若某个操作原语请求不合法，则终止用户请求；最后，若序列中所有原语请求均通过合法甄别，根据视图重组规则将请求结果反馈给用户。

4.3.3 角色安全控制

所谓角色，用一般业务系统中的术语来说，实际上就是业务系统中的岗位、职务或者分工。

用户组和角色的最主要的区别在于，用户组是作为用户的一个集合来对待的，并不涉及它的授权许可；而角色则既是一个用户的集合，又是一个授权许可的集合。

4.3.3.1 角色安全定义

角色安全控制通过“范围(Categories)”集合描述。范围集合是系统中非分层元素集合，该集合的元素依赖于所考虑的环境和应用领域。

角色安全描述可以通过二元组描述：Ca={Bu,Re}，称为角色域。Bu 为角色业务(Business)，反映用户工作领域；Re 为角色的职责(Responsibility)，反映了用户的工作岗位或职务。例如：{用户 1,{计算机学院，教师}}，{用户 2,{外国语学院，学生管理}}分别定义了两个用户的角色，用户 1 和用户 2 的业务范围分别为计算机学院和外国语学院，其职责范围分别为教师和学生管理工作者。

一、角色定义

VISTA 系统定义了两类角色：特权角色和普通角色。

特权角色是对系统进行安全管理的用户，共有四种角色：系统管理员、系统安全员、系统设计员和可信角色。管理员负责系统维护与管理；安全员负责访问控制；审计员负责安全审计；可信角色也称隐蔽角色，对应用户称为虚拟用户，即非实际用户，该用户由系统自动生成和销毁，负责当普通用户销毁时，接受隶属于被销毁用户的客体，以防这些客体被挂起，造成“死数据”。

普通角色对应系统中的实际使用者，是进行基于角色访问控制的访问主体，也是安全控制主要研究的对象之一。普通角色的角色域由角色业务域 Bu 和角色职责域 Re 中的元素唯一确定。即

$$Ca(R_1) \neq Ca(R_2) \Leftrightarrow Bu(R_1) \neq Bu(R_2) and \operatorname{Re}(R_1) \neq \operatorname{Re}(R_2)$$

因此，角色定义首先由系统管理定义角色业务域 Bu 和角色职责域 Re，如图 4.15 中的[a][b]。Bu 只需定义元素名和业务说明。Re 不仅需要定义元素名，还需定义该元素可以激活的客体的属性(如表的字段、对象类的属性)，即 Re 限定了角色可以访问的客体属性子集。

在角色业务域 Bu 和角色职责域 Re 定义的基础上定义角色表，表结构见表

4.15[c]。表中定义了两个特殊角色“Null”和“All”，“Null”这个角色的角色域为空，即该角色不能访问任何客体，当一个新用户被定义时，该用户的角色为“NULL”；“All” 这个角色的角色域为全部，即该角色可以访问任何客体。函数 Bu×Re→R 不是满射，既存在数偶(Bu,Re)与 R 不存在映射关系，在角色表中用“-”表示。

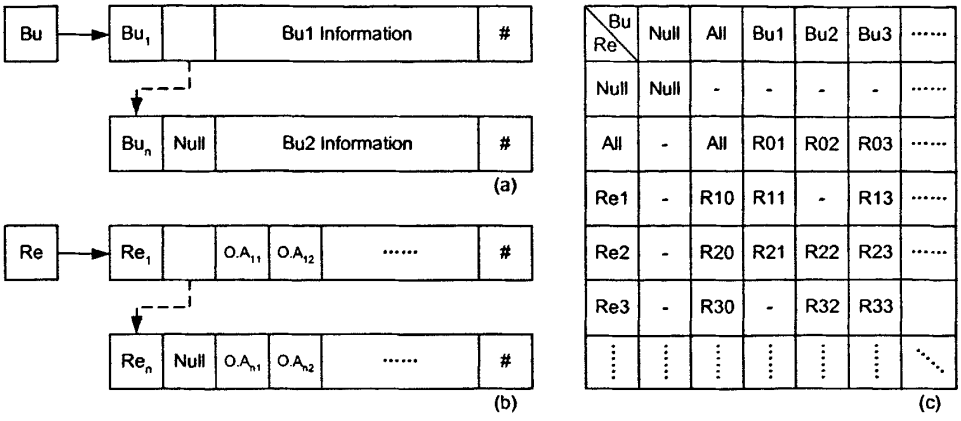

图 4.15 角色表示意图

在角色表的控制下，可以实施角色访问控制。

二、角色的偏序关系

在实际生活中，角色间存在一定的关系。我们将定义角色间的关系，这种关系被定义为一种偏序关系，记为{Ca,≤}。

[一]、Bu 和 Re 集合上的关系定义

已知角色域 Ca={Bu,Re}，我们首先定义角色业务域 Bu、角色职责域 Re 的关系。Bu 和 Re 两个集合分别存在偏序关系，分别记为{Bu,≤},{Re,≤}。关系“≤”定义为业务包含，即 a≤b 表示为 b 包含 a，记为<a,b>，对于 Bu 则 a 的业务范围均包含在 b 的业务范围内，对于 Re 则 a 可以激活的客体的属性均可以被 b 激活。关系“≤”可以用图 4.16 中的虚框内的关系图表示。

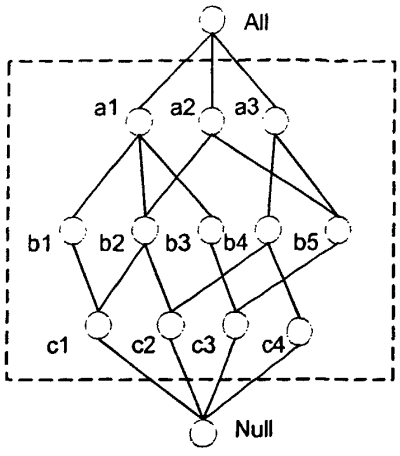

图 4.15 角色的 Bu,Re 域偏序关系示意图

从图中可知 b2≤a1,c3≤b5,又根据偏序的传递性可知 c3≤a2；但(c3,b2),(c4,a1)两个元素偶均不存在“≤”关系。

在 VISTA 中，{Bu,≤}由系统管理员定义，{Re,≤}通过对客体属性集合的包含关系进行判定。

同时我们在 Bu,Re 中分别引入特殊元素“Null”和“All”，“Null”为所有元素的下界，表示不具体任何角色域；“All”为所有元素的上界，表示拥有任何角色域(见图 4.16)。在这种定义下，{Bu,≤}和{Re,≤}构成格，可以用格的运算规则对 Bu 和 Re 两个集合中的元素进行运算。

[二]、Ca 集上的关系定义

设角色 Ca1=(Bu1,Re1)，Ca2=(Bu2,Re2)，称 Ca1 支配 Ca2 成立，当且仅当：Bu2≤Bu1 且 Re2≤Re1，记为 Ca1≤Ca2。

若给定角色 Ca1,Ca2，若 Ca1≤Ca2 和 Ca2≤Ca1 均不成立，则称 Ca1 与 Ca2 是不可比的(incomparable)。

可以证明在 Ca 集上定义满足“≤”的偏序关系。

三、用户角色分配

实施角色访问控制，首先是对用户分配角色，该工作由系统安全员执行，角色分配过程见图 4.17[a]。系统管理员定义一个新的用户 M，用户 M 得到初始角色“NULL”，M 使用自会话向系统特权用户申请角色，系统安全员根据 M 的情况分配角色 R(M)。用户角色信息存储在用户权限表中，将图 4.11 进行修改得图 4.17[b]。当用户角色定义后，系统安全员可以根据用户申请或系统审计申请修改用户角色 R(M)

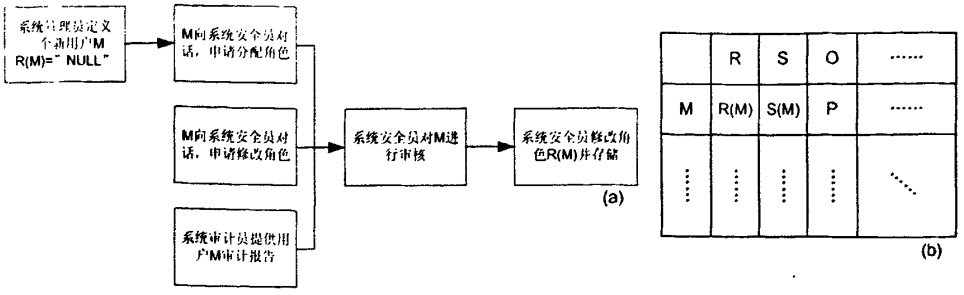

图 4.17 用户角色分配及存储

四、客体 Bu 域的逻辑存储

进行角色访问控制时，主要比较用户访问的客体是否属于角色域的管辖。根据角色定义，可知角色职责域 Re 定义了明确的管辖域，可以通过比较激活的客体属性集合是否与 Re 的定义域一致来判断访问的合法性。但角色业务域 Bu 没有明确定义其管辖域，该管辖域是由客体的 Bu 值给出的。

当一个主体定义一个客体时，客体继承主体的 Bu 值。由于客体的多样性，在安全控制时不需要对每个客体定义 Bu 值，根据具体需求约定需要给出 Bu 值的客体。VISTA 系统约定，只有记录定义 Bu 值，其他客体不定义，也就是说，角色访问是 Bu 甄别只对记录实施。

我们对图 4.3 中的记录逻辑存储结构进行修改，用来存储客体 Bu 值，其存储结构如图 4.18 所示，其中“记录 Bu 域”存储客体的 Bu 值。

记录号	记录Bu域	记录密级	字段1	字段1密级	字段2	字段2密级	……

图 4.18 客体角色 Bu 域的逻辑存储结构

五、角色分组定义

根据安全规则，我们可知主体间进行授权的前提是授权者和受权者必须在同一个角色组，因此需要对角色分组进行定义。

在 VISTA 中，约定以 Ca 的 Bu 域为分组依据，记角色 R 的 Bu 值为 dom(R)。根据安全规则(R1)，VISTA 系统的角色分组定义如下：

定义 4.2：两个角色 R1,R2 属于同一角色组，当且仅当 dom(R1)=dom(R2)。

从角色表可见，当角色 R1,R2 属于同一角色组，当且仅当 R1 和 R2 位于表中的同一列。

推论：两个主体 M1,M2 间可以进行授权操作，当且仅当 dom(R(M1))= dom(R(M2))，即 M1 和 M2 的业务域相同。

4.3.3.2 角色访问设计

角色访问控制通过角色的业务域和职责域读取客体信息。角色访问控制算法如图 4.19 所示。

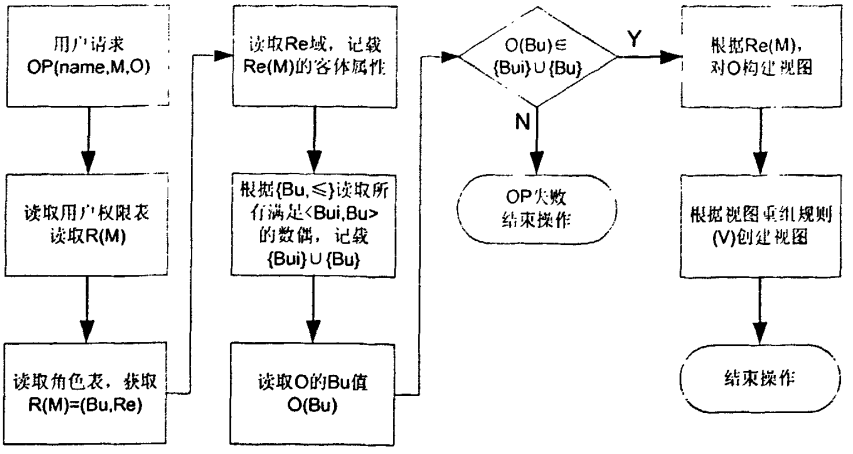

图 4.19 角色访问控制算法

角色访问控制首先读取用户的角色信息 R(M)=(Bu,Re)。然后，计算 Bu 的偏序关系，记载所有满足<Bu_i,Bu>的关系，形成新的业务集{Bu_i,Bu}，即主体 M 可以访问的业务域为{Bu_i,Bu}。读取客体 O，判断客体 O 的 Bu 值是否属于 M 的业务域，若不属于，则用户操作请求不合法，操作失败；若属于，则操作合法。若操作合法，则根据 M 的 Re 域的定义，在 O 中读取可以访问的属性，根据视图重组规则(V)创建视图，并将请求结果反馈给用户。

4.3.4 VISTA 访问控制设计

我们分别讨论了三种访问控制：自主访问控制、强制访问控制和角色访问控制。这三种访问控制分别从不同角度对系统访问安全进行了管理：自主访问控制从信息操作角度进行控制，强制访问控制从信息敏感角度进行控制，角色访问控制从信息内容分类进行控制。每种控制可以解决一些安全问题，但均存在不足，为此，我们将三种控制有机结合在一起，以便能更为全面地对数据进行安全控制。

图 4.20 给出了三种控制访问的关系，我们期望能得到 D 域的访问控制。

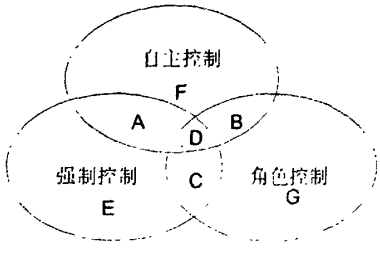

图 4.20 三种控制访问的关系

4.3.4.1 VISTA 整体访问控制设计

图 4.21 给出了 VISTA 整体访问控制的流程

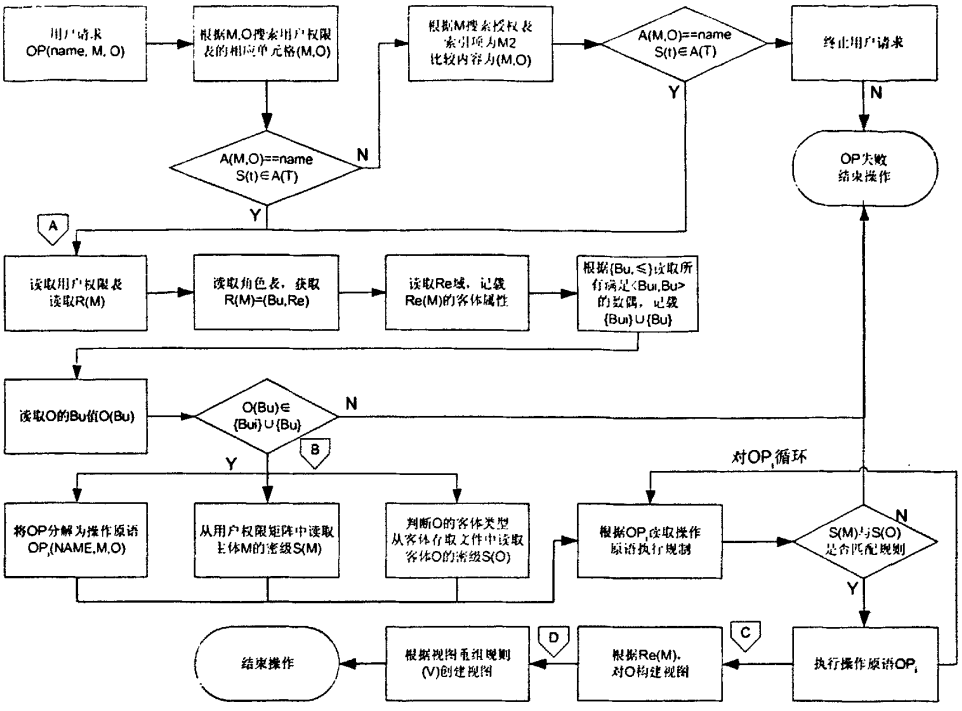

图 4.21 VISTA 整体访问控制流程

首先执行自主访问控制，甄别用户是否可以执行用户请求；然后执行角色访问控制，搜索用户的业务域，缩小数据的访问范围，在图中为从标志“A”起的流程；接着执行强制访问控制，判断用户是否可以访问敏感数据，在图中为从标志“B”起的流程；在此基础上，再执行角色访问控制，甄别用户的职责域，在图中为从标志“C”起的流程；最后，执行视图重组规则，向用户反馈信息，在图中为从标志“D”起的流程。

在访问控制中，需要注意的是，可能存在外码不属于用户的职责域，从而导致一些数据无法通过外码关联被访问到，造成合法访问不完整。因此，需要先进行强制访问，然后再进行角色控制的 Re 甄别。

4.3.4.2 VISTA 整体访问控制方案的修正

VISTA 访问控制综合了三种访问控制策略，安全控制的强度较高，但可能在实际使用中，不需要如此高强度的安全控制，我们需要进行方案的修正，以适合各种实际应用。

通过对访问控制参数的设定，可以实现各种访问控制的组合策略。控制访问参数的设置如下：

[1]自主访问控制通过对参数 P 进行安全甄别，在§4.3.1 中，我们定义了 P={{A},T)，可以进行如下定义：A="a"和 T="a"，表示用户在任意时间均可访问客体并拥有所有访问权限，如图 4.5 中的(User_2,Object_1,{a.a})。当所有用户对所有客体的权限 P 均设置为{a,a}，则可以证明自主访问控制无效

[2]强制访问控制通过对主体和客体的密级 S 的关系进行安全甄别，从安全规则可以知当主客体密级相等时，主客体的敏感度一样，这样不存在密级甄别问题，从而可知强制访问控制无效。由此，我们将系统中所有的主体和客体的密级均定义为“公开”，就可以使强制访问控制失效。

[3]角色访问控制通过对角色域 Ca 的定义进行安全甄别，由于 Ca 是一个格，格中的关系具有传递性，只需将角色域定义为格中所有元素的上界，就可以访问所有的数据域。在§4.3.3 中，定义了特殊的域元素“All”，表示拥有任何角色域，可以证明当所有用户的角色均设置为“All”时，即 Ca={All,All}，可以证明角色访问控制失效。

当对三种访问控制的参数进行设置，使得控制失效和有效，就构成访问控制的不同组合。特别当三种控制均失效时，系统不进行访问控制。通过修正，VISTA 访问控制可以适合多种实际应用。

4.4 审计设计

B1 级的安全性设计的另一个特点是对用户的操作行为进行审计。

VISTA 系统中，审计子系统根据所设置的审计开关和审计阈值，对系统中的事件（即用户的操作）进行收集和审计。审计开关用来确定事件收集的范围，或确定审计事件的类型，而审计阈值则用来进一步确定需审计的事件。审计事件是与系统安全性有关的事件，审计子系统从审计事件中提取信息，将这些信息记录到审计日志中，供系统审计员查询和获得审计报告。

4.4.1 审计表及其操作设计

一、审计表存储设计

审计记录存放在一个名为审计表(审计日志)的系统表中，VISTA 审计表的逻辑结构如图 4.22 所示。

T	M	O	A	DAC_F	MAC_F	REAC_F	S_Edit	State	Msg
记录时间	操作主体	操作客体	主体动作	自主控制标志	强制控制标志	角色控制标志	安全参数修改	状态信息	错误信息

图 4.22 审计表逻辑存储结构

审计表记载了事件的发起时间(T)、发起主体(M)、操作客体(O)和主体动作(A)等基本信息，记载三种访问控制甄别是否合法的标志(*_F)，同时，记载安全参数的变更情况(S_Edit)，该信息包括授权变更情况，以及系统状态信息(State)和事件出错信息(Msg)。

二、审计表的操作

系统支持系统审计员和用户对审计日志的查询，并按要求提供审计报告。

审计子系统根据事件主体和客体的安全级对审计日志的内容进行分级管理。

系统审计员可以查询所有的审计内容，其他用户对审计内容的查询必须事先经过多级安全和角色安全检查。任一审计事件 A 的审计信息的安全级 S(A)定义为 $S(M_A)$（事件 A 中主体的安全级）和 $S(M_O)$（客体的安全级）的最小上界，即 $S(A)= lub(S(M_A), S(M_O))$。仅当用户 M 的安全级 $S(M)\geq S(A)$且 $R(M)\geq R(M_A)$ 时，用户 M 才能查询有关事件 A 的审计信息。

对审计日志设置以下方式的查询：

(1) 关系查询：对指定的关系，查询其上的操作引起的事件的审计内容。

(2) 用户查询：对指定的用户，查询它所引起的事件的审计内容。

(3) 安全级查询：查询具有指定安全级的用户或具有指定安全级的关系所涉及的事件的审计内容。

(4) 时间查询：对指定的时间区间，查询期间所发生的事件的审计内容。

系统审计员根据需要，可删除档案日志中的部分或全部信息，其他用户无权对审计日志中的信息进行删除。

4.4.2 审计参数的设置

系统审计员对审计参数进行设置，控制审计进程。审计参数包括两个部分：审计粒度阈值和审计控制阈值。

一、审计粒度阈值设置

审计粒度的设置决定需要进行审计的事件，VISTA 根据三种访问控制的安全参数进行审计粒度设计。粒度阈值表如图 4.23 所示：

R 角色	S_M 主体密级	S_O 客体密级	T 时间	A 动作
[R]	[S_M]	[S_O]	{T}	{A}

图 4.23 审计粒度阈值设置表

若存在用户请求 OP(name,M,O)，则如下情况之一对用户请求 OP 进行审计：

(1) $[R] \leqslant R(M)$

(2) $[S\_M] \leqslant S(M)$

(3) $[S\_O] \leqslant S(O)$

(4) $T(OP) \in \{T\}$

(5) $name \in \{A\}$

二、审计控制阈值设置

审计控制阈值决定审计预警触发机制，以便对危及系统安全的事件进行报警，该阈值分报警阈值和惩罚阈值。报警阈值是一个主体在一定时间内出现的系统可容忍的报警事件的最大次数，惩罚阈值是一个主体在一定时间内出现的系统

可容忍的对其报警的最大次数。

控制阈值表如图 4.24 所示，其中 Lij=(a,b)，j>=2，a 为报警阈值，b 为惩罚阈值，a 和 b 均为整数值；Li1 为时间长度，如 3 天：

	T 时间阈	DAC_L 自主控制 非法阈	MAC_L 强制控制 非法阈	REAC_L 角色控制 非法阈	S_Edit_L 安全参数 修改阈	Msg_L 错误阈
M 操作主体	L11	L12	L13	L14	L15	L16
O 操作客体	L21	L22	L23	L24	L25	L26
A 主体动作	L31	L32	L33	L34	L35	L36

图 4.24 审计控制阈值设置表

审计控制算法如下：

(1)对审计表进行汇总，计算主体、客体和动作在相应的时间阈 Li1 中放生的控制甄别值为非法的次数、安全参数修改的次数和事件出错的次数，记为 Nij

(2)如果 Nij >= Lij.a，则触发审计报警机制

(3)如果 Nij >= Lij.b，则触发审计惩罚机制

4.4.3 基于审计的安全管理方案设计

我们讨论了审计信息的存储和审计参数设置，讨论了审计控制的算法，本节将讨论如何根据审计进行系统安全管理。

基于审计的安全管理，是根据审计的警告，通过系统安全员、审计员和管理员一起实现安全管理。其管理方案如图 4.25 所示。

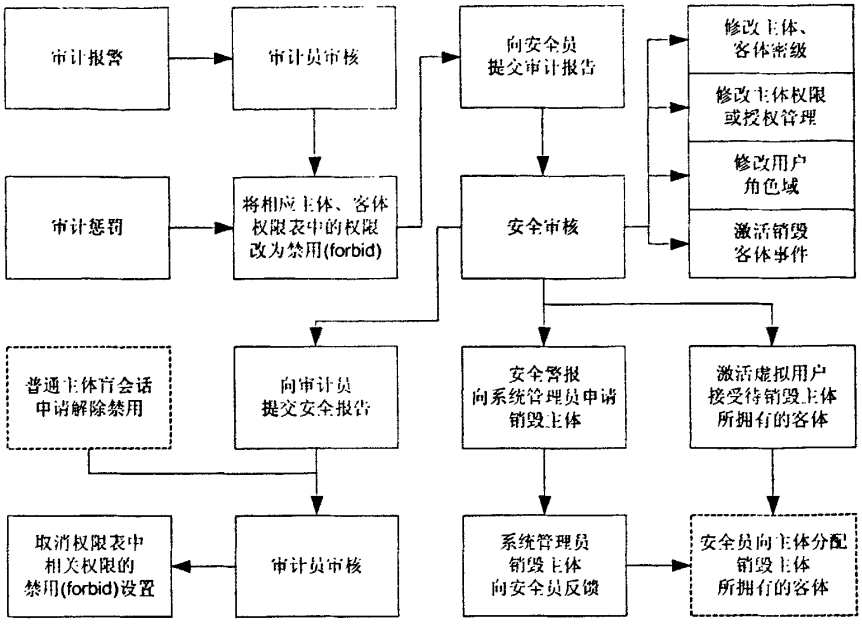

图 4.25 基于审计的安全管理方案

从图中可以看出该安全管理存在以下几个主要的管理要求：

[1]审计报警和审计惩罚的差异在于审计惩罚直接将主体对客体的权限改为“禁用”，而审计报警需要审计员对事件安全异常进行审核后再修改权限表。

[2]审计员不能对安全参数进行管理，只有向安全员提交审计报告，由安全员对安全参数进行管理。

[3]安全管理员对审计报告进行安全审核后，对安全事件向审计员提交安全报告，要求审计员取消“禁用”权限的设置。安全员对非安全事件的处理分两类：可以直接修改安全参数或销毁客体，或申请销毁主体。

[4]安全员不能直接销毁主体，只有向系统管理员申请销毁主体。安全在申请销毁主体前，需要激活虚拟用户接受待销毁主体所拥有的客体，以防这些客体被挂起，成为“死数据”。

[5]虚拟用户不能对其拥有的数据进行管理，这就要求系统安全员工根据实际情况将虚拟用户拥有的数据分配给普通用户，使得数据存在宿主，可以接受管理。

通过对安全存储、安全模式管理、安全访问控制和安全审计管理基本可以实现安全数据库的基本安全需求。

第五章 VISTA 的安全体系的实现

前两章讨论了 VISTA 的安全规则和安全设计，陈述了如何管理和存储安全信息、如何进行安全控制。我们在安全对象关系数据库安全规则和模型设计的基础上，开发了一个安全数据库的试验系统 VISTA，本章将讨论 VISTA 的安全体系的实现。限于篇幅，我们仅讨论 VISTA 的安全功能体系，安全操作方式和部分功能实现的方法。本文主要介绍功能内容和用户界面，对实现细节和代码不加以阐述。

5.1 系统功能设计

VISTA 系统由若干功能模块构成，这些模块主要包括数据库常规的数据管理模块和数据库安全管理模块。图 5.1 给出了主要的安全管理模块的组织结构，安全访问控制模块是后台控制模块，在数据管理和数据查询时，由相关功能模块调用，图中未标出。

安全管理模块主要由五个部分组成：

[1]用户登录。根据用户名和用户密码甄别用户是否合法，是否允许用户使用 VISTA 系统。

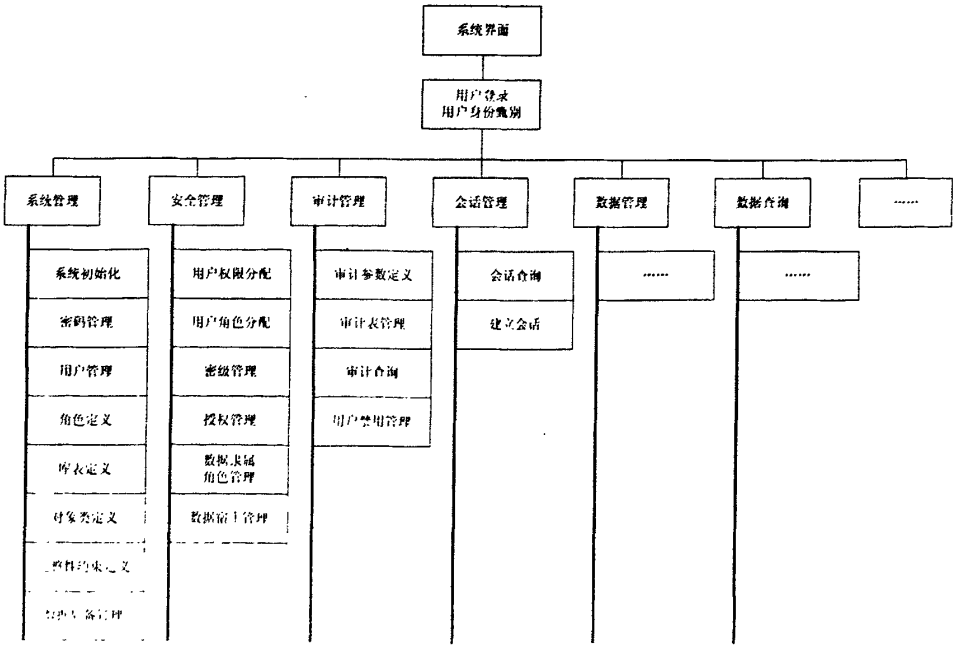

图 5.1 VISTA 系统的功能结构示意图

[2]系统管理。该功能主要由系统管理员执行，部分可由普通用户使用，主要是对系统一些主要参数、约定和数据字典进行定义。包括系统初始化、密码管理、用户管理、角色定义，库表定义、对象类定义、完整性约束定义和数据后备管理等，其功能概述如下：

系统初始化，由系统管理员执行，将系统参数恢复到初始状态，删除所有用户数据、清空各种字典、删除所有用户，只保留初始三个特殊用户：系统管理员、系统安全员和系统审计员。该操作不能轻易操作，操作之前，系统管理员需要对所有数据执行后备操作。该操作当系统发生严重故障时，进行恢复性操作。

密码管理，由所有用户执行，主要是对当前用户的密码进行修改。

用户管理，由系统管理员执行，对用户进行管理，主要有如下功能：定义、删除、修改、浏览用户信息等。一般在以下情况使用：①定义一个新用户；②删除一个非法用户；③修改用户密码等。

角色定义，由系统管理员执行，定义角色业务域 Bu 和角色职责域 Re、角色表以及 Bu 域和 Re 域上的偏序关系。

库表定义、对象类定义和完整性约束定义，由管理员和普通用户执行，主要对数据库模式、数据库的关联、索引、完整性约束等进行定义。该操作由两个后续操作：①安全员对模式中客体密级进行调整；②管理员对模式中的客体进行角色职责域 Re 定义。

数据后备管理，由系统管理员执行，主要用户数据的备份和恢复，防止因故障造成数据流失。

[3]安全管理，该功能主要由系统安全员执行，部分可由普通用户使用，主要是对系统的一些安全参数和安全事项进行管理，包括用户权限分配、角色分配、主客体密级管理、授权管理、客体隶属角色管理以及数据宿主管理等，其功能概述如下：

用户权限分配，由系统安全员执行，主要对每个主体权限进行管理。

用户角色分配，由系统安全员执行，主要对每个主体进行角色管理。

密级管理，由系统安全员执行，主要对主客体的密级进行管理。

授权管理，由系统安全员和普通用户执行，主要对主体进行授权管理，包括授权操作和授权回收操作。

数据隶属角色管理，由系统安全员执行，主要对客体的隶属角色进行管理，即修改数据域记载的 Bu 和 Re 值。

数据宿主管理，由系统安全员执行，当发生主体销毁时，进行该操作，主要有两个操作：①将销毁主体所拥有的客体交付虚拟用户管理；②将虚拟用户拥有的数据分配给普通用户。

[4]审计管理，该功能主要由系统审计员执行，部分可由普通用户使用，主要是对系统相关的审计操作进行管理，包括审计参数定义、审计表管理、审计查询和用户禁用管理等，其功能概述如下：

审计参数定义，由系统审计员执行，主要对审计粒度阈值和审计控制阈值进行定义和修改。

审计表管理，由审计员执行，主要对审计表进行管理，包括记录的备份、删除操作。

审计查询，由系统安全员和普通用户执行，主要是对审计表进行浏览和数据分析，供系统安全员和普通用户了解用户对数据访问的安全状况。

用户禁用管理，由审计员执行，主要是对用户“禁用”权限的设置，包括设置和取消操作。

[5]会话管理，该功能由所有用户使用，主要是进行用户间的会话管理，包括会话查询和建立会话等，其功能概述如下：

会话查询，供用户阅读和浏览会话，允许会话接受者删除相应的会话。

建立会话，供用户建立会话，在建立会话时，系统对会话合法性进行甄别。

在系统执行过程中，一些功能不是孤立的，根据操作的执行顺序可以主动激活相关功能。如，库表定义操作，则根据操作的发起者不同，激活不同的事件，如果是系统管理员执行库表操作，操作完毕则激活角色定义操作，并生成会话，要求安全员进行安全修改；如果是普通用户执行库表操作，则生成会话，要求系统用户和系统安全员进行相应的管理操作。

5.2 部分功能简介

在前面各节我们介绍了 VISTA 系统的安全设计和主要的安全功能，给出了系统中主要安全操作的控制流程，本节就一些安全管理功能的实现进行简要介绍。

5.2.1 用户管理

用户管理主要是由系统管理员对系统用户进行管理，包括用户的建立、修改、浏览和删除操作，用户界面如图 5.2 所示。

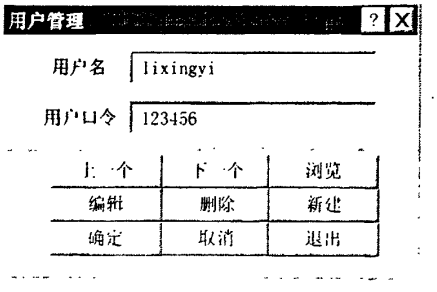

图 5.2 VISTA 用户管理功能交互界面

在该对话框中，“浏览”功能以列表方式显示所有用户；“编辑”功能用来修

改用户的名和密码；“删除”功能用以删除当前显示的用户，该功能具有破坏性，系统需要进行甄别是否将用户名列入虚拟用户的表，若列入则虚拟用户接受了待销毁主体所拥有的客体，可以删除该用户，并向安全员报告用户删除成功，否则，向安全员会话，请求进行数据宿主管理，并退出删除操作，暂缓执行。“新建”功能用以建立一个新的用户。

5.2.2 角色定义

角色定义由系统管理员执行，定义角色业务域 Bu 和角色职责域 Re、角色表以及 Bu 域和 Re 域上的偏序关系。VISTA 采用向导的方式进行引导用户进行角色定义，用户界面如图 5.3(a)所示。

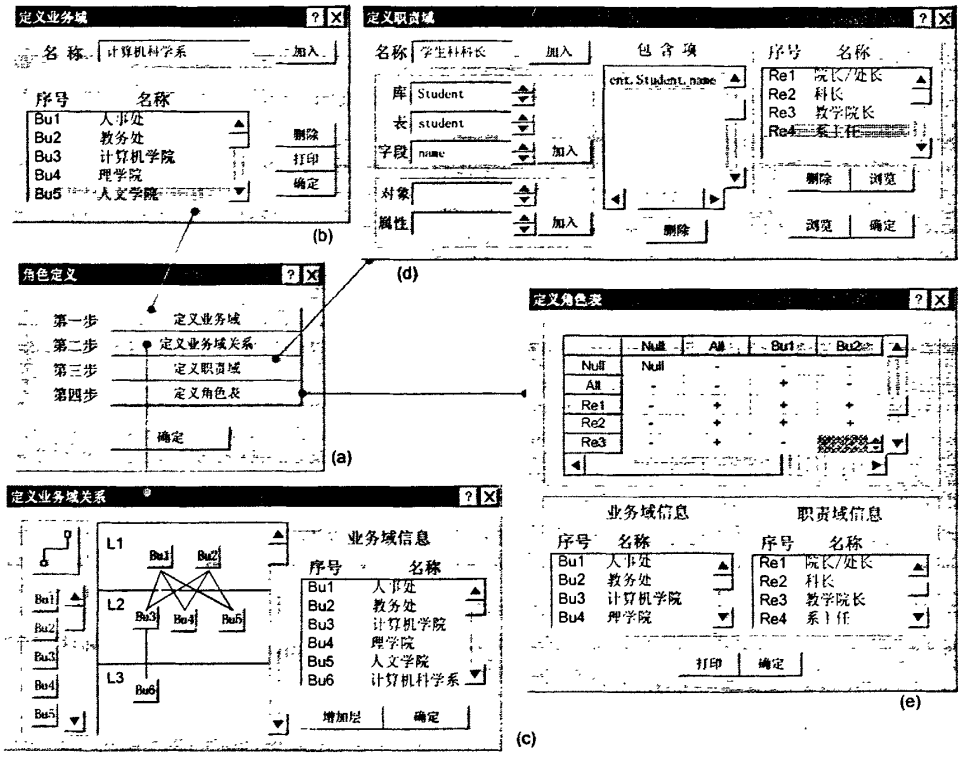

图 5.3 VISTA 角色定义的交互界面

图(b,c,d,e)分别为业务域 Bu 定义、Bu 偏序关系定义、角色职责域 Re 定义和角色表定义。

在图(b)只需添加新的名称就可以定义新的业务域，业务域代码由系统主动给出。图(c)通过交互的方式定义业务的偏序关系，首先将不同的业务代码拖到各自所在的层 L_i，然后用连线工具在业务代码上进行连线，就定义了角色的偏序集，当用户按下“确定”键后系统保存该偏序关系。

图(d)通过名称的“加入”按钮建立新的职责域，通过其他两个“加入”按钮将对象加入到该职责域；在包含项区域用来查询该职责域包含的项，可以通过

“删除”按钮删除当前选中的项；在最右边的区域是对所有职责域进行管理，“删除”按钮用以删除当前选中的职责域，“浏览”按钮用来察看当前选中的职责域的定义情况；右下角的“浏览”按钮用来察看系统中所有职责域的定义情况。

图(e)定义角色表，表的下部用来浏览各域的代码和名称的对应关系，上部用来定义角色，“-”表示不定义角色，“+”表示定义了角色。在安全员对用户分配角色时，方法是分别选择 Bu 和 Re，系统甄别该(Bu,Re)偶对是否被定义了角色，若定义则角色分配成功，否则不成功。

通过图 5.3(a)的向导，系统管理员可以完成角色的定义。

5.2.3 用户权限管理

用户权限管理由系统安全员执行，是将客体的各种操作权限分配给用户，所得结果是进行自主访问控制的依据，其操作界面如图 5.4 所示。

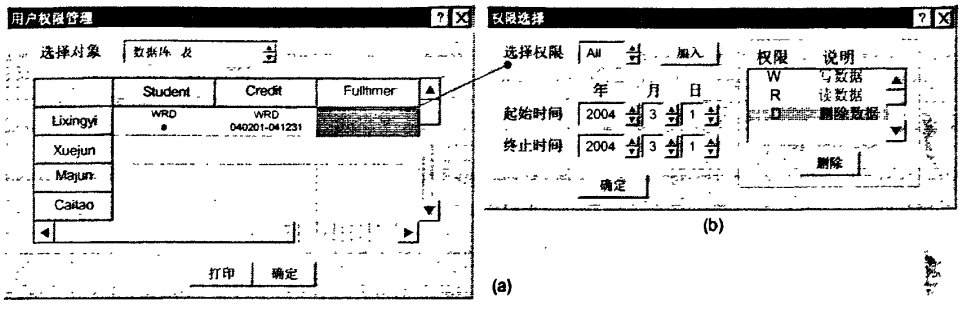

图 5.4 VISTA 用户权限管理的交互界面

在图(a)中，首先选择客体类，VISTA 定义了三种选项：系统菜单，用以控制用户可以使用的系统操作；数据库/表，用以控制用户访问的数据范围；对象类，用以控制用户对对象的访问。

然后在权限矩阵中选取相应的单元格，双击进行相应主体对客体的权限定义，此时进入图(b)，进行权限定义。

图(b)通过选择权限然后加入的方法来定义权限，在权限选择中有两个特殊的权限“All”和“Null”，“All”表示可以使用任何权限，在权限矩阵中用“a”表示；“Null”表示不可以使用任何权限，在权限矩阵中用“-”表示。

图(b)通过选取的方法定义用户权限的有效时间，若起始时间和终止时间中的年度选择“0”，则表示无时间限制，在权限矩阵中用“a”表示。

对于其他安全参数的管理，如角色分配管理、密级定义管理等，方法与之相似，这里不再叙述。

5.2.4 审计查询

审计是对安全状况进行分析的一个重要手段之一，在§4.4 中，我们详细讨论了审计的设计问题。在 VISTA 的实现中，我们给出了专门的审计管理功能，

图 5.5 和图 5.6 给出了“审计查询”功能的用户界面。

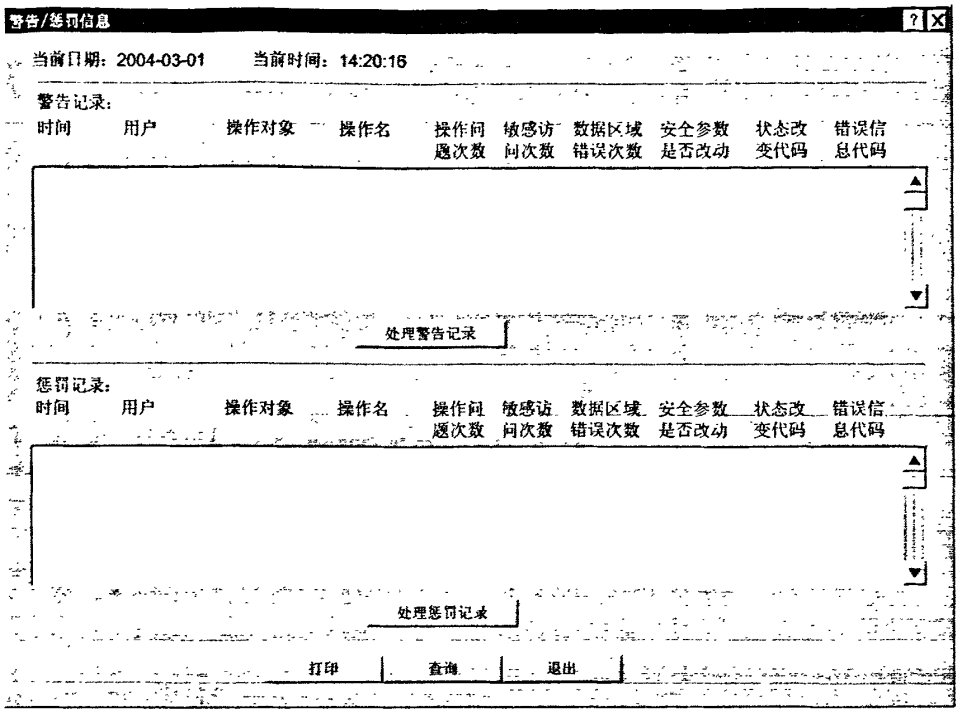

图 5.5 VISTA 用户权限管理的交互界面

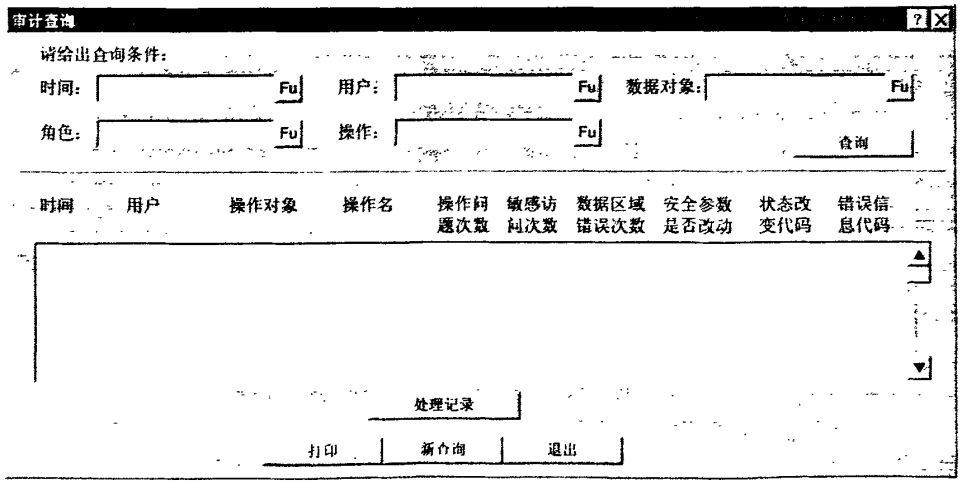

图 5.6 VISTA 用户权限管理的交互界面

当用户调用“审计查询功能”时，系统根据用户的类型决定操作方式，当用户是系统审计员时，系统首先进入“警告/惩罚信息”对话框，再进入“审计查询”对话框；而用户是普通用户时，系统直接进入“审计查询”对话框，同时对话框中的“处理记录”按钮变为“会话”按钮。即只有系统审计员才有权处理审计记录。

审计查询的条件由用户在相应的文本框中输入常量，也可以通过“Fu”按

钮输入表达式，进行更为复杂的查询。查询的五个条件项(时间、角色、用户、操作、数据对象)间的关系是“与”的关系，即查询是以：
select ..... from ..... where f(时间)∩f(角色)∩f(用户)∩f(操作)∩f(数据对象)
为查询指令的。当文本框为空时，表示相应项允许所有值。

5.2.5 建立会话

会话是 VISTA 系统的一个特色，它允许用户间进行信息交换，是系统具有更大的灵活性。也为以后 VISTA 系统向分布式发展进行了基础准备。

图 5.7 给出了“建立会话”功能的用户交互界面。一个会话由三个部分组成：会话者、会话内容和请求时间域，前两者不言而喻，时间域主要用来告知对方，需要在什么时间域中对会话中的请求作出答复。

当用户发送会话时，系统会自动甄别会话的合法性，若合法，系统自动加上会话标识信息：会话发起者、会话发起时间，发送会话。会话标识信息用来让用户对会话进行查询。

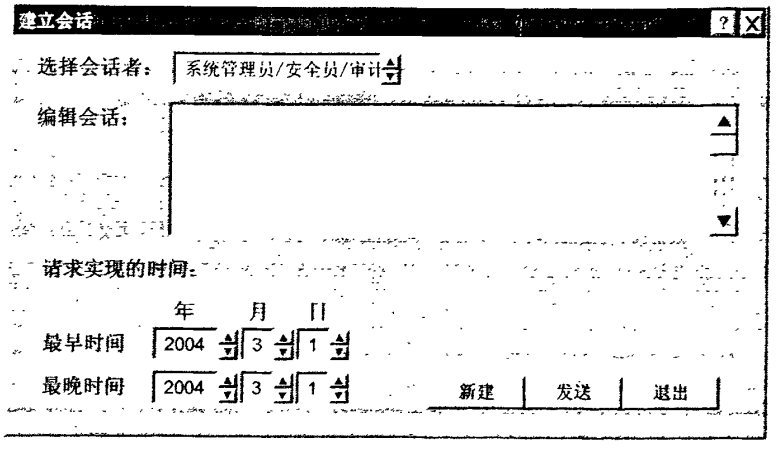

图 5.7 VISTA 建立会话功能的交互界面

5.3 测试和结论

VISTA 的实现环境如下：

- 硬件环境：Alpha 工作站
- 软件环境：Unix，X-window，Motif
- 编程语言：C 语言

为证明 VISTA 的可行性，我们对 VISTA 进行了测试，本节简要介绍测试情况。

、测试数据背景

[1]数据设计。我们设计了两套测试数据，一套是教学管理数据，一套是地理

数据。前者是纯关系型数据，后者是对象关系型数据。

[2]用户设计。定义了三个系统用户：管理员、安全员、审计员；一个特殊用户：虚拟用户；五个普通用户

[3]角色域的定义。对于两套数据分别定义了 5 个 Re 域和 7 个 Bu 域。

[4]安全访问的数据粒度定义。自主访问的安全粒度为表和对象；强制访问的安全粒度为表的字段域和对象的属性域；角色访问的安全粒度为：Re 域达到表的字段域和对象的属性域，Bu 域达到表的记录。

二、测试方式设计

测试方案设计如下：

[1]访问控制组合测试。四组：①三种控制综合；②没有设置访问控制；③基于自主和强制的访问控制；④基于自主和角色的访问控制。目标：测试访问控制组合是否可行和正确。

[2]攻击测试。三组：①存储攻击，不通过 VISTA 直接访问数据文件；②操作攻击，使用非合法的访问权限、角色身份访问数据和敏感数据；③挂起攻击，在系统运行过程中，借助联机方式，对内存和数据文件进行访问，获取数据。

[3]审计测试。给出一组非法操作，检验在阈值范围内，审计是否全部检测到。

[4]主客体销毁测试，销毁一些主客体，检验在该情况下，系统数据是否安全，是否存在挂起和信息泄漏的情况。

[5]会话测试。是否存在非法会话。

三、结果及其结论

以上诸方案除挂起攻击外，均满足系统设计要求，基本办证数据的一致性和安全性。

挂起攻击出现的问题如下：①内存攻击造成一些敏感信息的泄漏，其原因是数据驻留内存时间过长，此外，一些事件完成后，未对内存中的数据进行清"0"处理；②文件攻击，造成系统挂起，其原因是数据动态后备管理设计不当，在文件被破坏性攻击后，不能自动恢复。

从整体测试分析，VISTA 系统可以满足对象关系型数据库的基本安全需要，可以实现基于自主、强制和角色三种访问控制方式的组合下安全访问控制。说明，该设计是合理的、可行的和安全的。

第六章 总结和展望

安全对象关系数据库是当前信息安全研究的一个重要分支，具有广泛的应用前景。该领域的研究具有强烈地域性和保密性，信息技术发达国家对我国一直施行尖端安全产品禁止输出策略，数据库安全产品亦在其列，因此，研究和开发自主的安全数据库产品是进行自主信息保护的一个重要手段。

本文就安全对象关系数据库进行了深入的研究，对安全对象关系数据库的安全策略、安全模型、安全设计和安全数据库的实现进行了深入的讨论，提供了一个较为完整的逻辑设计方案，并在此基础上实现安全对象关系数据库系统——VISTA。

6.1 创新和特色

本文针对传统的安全模型进行分析和改进，提出了一种新的安全模型 TDM，并设计了该模型相应的规则组。TDM 安全规则从安全定义、数据安全访问、数据完整性、冲突协调四个方面对 TDM 安全模型进行了严格的定义，为模型的实现提供了依据。同时，通过 TDM 模型与传统安全模型的兼容性论证，说明了 TDM 模型的可行性和合理性。

项目组较为全面地设计了一个安全对象关系数据库系统 VISTA。从安全存储机制、安全数据模式、安全访问和审计设计四个方面，对 VISTA 的设计方案进行了阐述，首次提出了可组合安全访问控制方案，根据具体安全访问控制需求，对自主访问控制、强制访问控制和角色访问控制三种传统的访问控制方案进行合理的改进和设计，使之可以自由组合，以适合不同安全强度的实际应用的需要。

引入用户会话机制，使得用户可以通过安全的会话通道进行需求请求，解决了不同用户间的合法通讯问题。同时，为进一步设计分布式安全数据库作准备。

在研究的基础上，开发了 VISTA 系统，并对其进行了较为系统的安全测试，验证了 TDM 安全模型的正确性和实用性。

因此，本工作在理论上具备以下创新点：

(1)提出了新的安全模型——TDM；

(2)第一次提出了可组合安全访问控制策略；

(3)对传统的自主安全访问控制和角色访问控制进行了改进，增强了其安全控制的约束，在自主安全访问控制中增加了有效时间域，将角色域分解为业务域和职责域，使得访问策略更贴近和适合实际需要。

6.2 展望

TDM 提供了一个安全对象关系数据开发的安全模型，借助对 TDM 安全规则的实现和应用可以开发一个满足 B1 安全标准的安全数据库产品。

VISTA 是在 TDM 基础上开发的一个安全对象关系数据库试验产品，对指导开发安全数据库具有指导意义。VISTA 系统涵盖了安全数据库开发中涉及的用户权限分配、特权用户定义、访问控制、安全存储控制、审计、安全通讯、数据完整性问题和数据后备处理等一系列安全设计问题，对 VISTA 的研究、设计和开发为进一步开发自主产权的商用数据库系统提供了理论基础和实践基础。

通过 VISTA 系统的研发和测试可以证明，TDM 模型可以用于支撑多种类型安全数据库管理系统的开发，具有广阔的使用前景和理论价值。

目前，我们只是在理论上探讨了安全数据库模型和安全数据库实现方法，并进行了一些实验性的设计和开发，在此基础上，要实现一个完整的安全数据库管理系统尚有许多工作要做：

⑴安全事务控制的研究；

⑵并发机制下的安全控制问题；

⑶分布数据库的安全访问问题；

⑷查询优化与安全控制组合及其控制顺序间的关系，等。

基于上述理论问题的提出，实现安全数据库需要还开展大量的工作。

总之，TDM 模型在安全数据库领域具有广泛的应用前景。安全数据库的开发是一个相当大的软件项目，有许多工作要做。VISTA 系统目前只是一个试验性的产品，要实现实际意义上的安全数据库管理系统还需要进一步研究和开发。

参考文献

[1] Bell, LaPadula L., "Secure Computer Systems: Mathematical Foundation and Model", M74-244, Bedfrod, Mass: The MITRE Corp., Oct.1974.

[2] Bell, LaPadula L., "Secure Computer Systems: Unified Exposition and Multics Interpretation", Project NO.522B, Bedford, Mass: The MITRE Corp., Contract NO.F19628-76-C-0001, MARCH.1976.

[3] Denning D E., "A lattice model of secure information flows", Communications of ACM, 1976,19(5): 236-243.

[4] Biba K J., "Integrity consideration for secure computer system", Mitre Corp, Tech Rep:TR-3153,Bedford,Mass,1977.

[5] Department of Defense, "Trusted Computer System Evaluation Criteria", Dec.1985.

[6] LaPadula L J., "Formal modeling in a generalized framework for access control", In: Proc of the IEEE Computer Security Foundations Workshop III. Los Alamitos, CA, 1990. p100-109.

[7] Department of Defense, "Trusted Database Management System of TCSEC", 1991.

[8] National Computer Security Center, "Trusted Database Management System Interpretation of TCSEC", Apr.1991.

[9] John McLean, "Security Models", Encyclopedia of Software Engineering, Wiley Press, 1994.

[10] R.S.Sandhu, E.J.Conye, H.L.Feinstein and C.E.Youman, "Role-Based Access Control Models", IEEE Computer, Vol.29, No.2, p38-47, Feb.1996.

[11] Ravi Sandhu, Pierangela Samarati, "Authentication, Access Control, and Audit", ACM Computing Surveys, Vol.28, No.1, March 1996.

[12] Donald G.Marks, "Inference in MLS Database System", IEEE Transactions on Knowledge and Data Engineering, Vol.8, No.1, Feb 1996.

[13] Ravi Sandhu, Qarnar Munawer, "How to do Discretionary Access Control Using Roles", Proceedings of 3rd ACM Workshop Role-Based Access Control, Fairfax, Virginia, October 22-23,1998.

[14] Ravi Sandhu and Fang Chen, " The Multilevel Relational(MLR) Data Model", ACM Transaction on Information and System Security, Vol.1, No.1, pp: 93-132,November 1998.

[15] David F.Ferraiolo, Ravi Sandhu, Serban Gavrila, D.Richard Kuhn and Ramaswamy Chandramouli, "A Proposed NIST Standard for Role-Based Access Control", ACM Transactions on Information and System Security, Vol.4, No.3, August 2001.

[16] Ravi Sandhu, "Future Directions in Role-Based Access Control Models",2001.

[17] 计算机信息系统安全保护等级划分准则.GB17859-1999，北京：国家质量技术监督局，1999-09-01.

[18] 刘启原，刘怡编，数据库与信息系统的安全，北京：科学出版社，2000.

[19] 陆庆元，吉增瑞，信息存储系统中的信息安全技术，计算机研究与发展，Vol.37，2000.10.

[20] 冯登国，国内外信息安全研究现状及其发展趋势，网络安全技术与应用，2001，No.1,p8-13.

[21] 冯登国，计算机通信网络安全，北京：清华大学出版社，2001

[22] 宋志敏，南相浩，唐礼勇，余嘉宁，数据库安全的研究与进展，计算机工程与应用，pp:85-87,2001.1.

[23] 鞠时光，Sergio Chapa, The card-box Metaphor for Huaman Spatial Database communication. The proc. of 1997 IEEE PACRIM'97, Aug, 1997,Victoria, Canada, 1997, Vol.2, pp669-677.

[24] 鞠时光，Sergio Chapa, Visualization of a spatial database, The proc of visual computation'97, Mar. 1997, Mexico City, pp86~98.

[25] 鞠时光,Sergio Chapa and W. Chen, "Extensible motor of a object-relational DBMS: design and implementation", The proc. of technology of object-oriented languages and systems, IEEE computer society, 1999, pp372-379.

[26] 鞠时光，可视化空间数据库查询语言 CQL，计算机学报，1999，Vol.22, No.2, P205-211.

[27] 鞠时光， Define of CQL, a Visual Query Language, Computer and Systems, 1999, Vol.3, No.2, pp.77-87.

[28] 鞠时光，对象关系型数据库管理系统开发技术，北京：科学出版社，2001.5

[29] 马建平，余祥宣，多级安全关系数据库系统的分析和设计，计算机工程和科学，Vol,19. No.3, Aug, 1997.

[30] 洪帆，蔡蔚，多级安全关系数据库系统审计功能的设计，小型微型计算机系统，Vol.17,No.2,Feb.,1996.

致　谢

本文是在鞠时光教授的悉心指导下完成的。导师广博的学识，严谨的治学态度，踏实的科研精神、高尚的人格都使我受益匪浅，让我铭记一生。在课题的选题及研究过程中，都倾注了导师的心血。研究工作和论文得以顺利完成，离不开导师学业上的倾心相授和悉心教诲。值此论文完成之际，谨向辛勤培养我的导师表示崇高的敬意和衷心的感谢。

感谢计算机与通信工程学院的老师对我的教育与帮助。李星毅老师在课题研究过程中给我提出了宝贵的建议，提供了许多相关资料，工作还得到施化吉老师的帮助，在此一并感谢。

必须感谢在我的学习生活中给予我关心帮助的同学和朋友，尤其是马俊、岳小平、朱金伟、翁正岭、陈伟鹤等，他们在我的课题实现和专题研讨的过程中给了我很多的有益的启发。

最后，我衷心感谢所有在学业上给予过我关心和帮助的人，谨向他们表示深深的谢意！

基于组合访问控制的安全数据库设计

作者：邵学军
学位授予单位：江苏大学