多级安全关系数据库系统客体标记粒度问题的研究^{*}

The Analysis of the Granularity Problem of Labels Associated with the Storage Object in MLS Relational Database

程万军 郑军 彭成宝 张霞

(东北大学软件中心 沈阳110004)

Abstract The granularity problem of labels associated with storage object in multilevel relational database which is a bone of contention of the researchers has the direct relationship with the system architecture and work module, and it directly affects the system integrity, evaluate about security, etc. This paper compares and analyzes in detail several applied granularity schema and indicates their characteristic, applicability and involved problems. The researchers with different interesting and intents can adopt different granularity schema to protect the objects in the system for the different application environment.

Keywords MLS DBMS, Security label, Label granularity, MAC, Polyinstantiation

1. 引言

针对信息系统安全性需求的增长,国内外都制订了相应的安全评估和分级标准。这些标准都规定了对系统安全规格、功能等的可靠性和可行性要求,而对涉及具体实现的细节最多只给出指导性的建议。多级(安全)关系数据库系统的客体标记粒度问题就是其中之一。而这个问题的重要性是不言而喻的,对它的透彻研究无论对安全模型和策略的研究者,还是对数据库系统的研究者和开发者以及应用者都具有非常重要的指导和实践价值。

应用于多级数据库系统的标记(Labels)与应用于操作系统的标记的基本不同点在于它标示在存储客体类型上,而不是标记在处理问题上^[1,2]。文[1]中对多级关系数据库系统客体保护(即客体标记)粒度问题做了如下的说明:它没有规定或者约定客体的标记粒度,而是将这个问题交由具体数据库系统的设计开发者来决定。

一般而言,在数据库系统中,存储数据库基本数据的客体(例如:文件、档案、关系、元组等)以及存储元数据(metadata)的客体(例如:目录、索引、模式、数据字典、DAC 授权表)必须被标记。而那些用户不可见的内部资源(如:资源分配表等)则不需要被标记。多级关系数据库客体标记粒度问题直接关系到系统的安全性和完整性问题,并对系统的安全性评估以及系统的隐蔽通道分析等方面产生影响。

2. 多级关系数据库可供采用的客体粒度模式

不同数据库系统的研究开发者所面对的是不同体系结构、不同接口类型和不同可修改程度(自主知识产权)的数据库系统,而且他们所开发的系统目的和目标用户不同,因此,使得他们在进行多级关系数据库系统开发时采用了不同的客体标记粒度。

在多级操作系统中,被保护客体的粒度是文件。而在多级关系数据库研究领域,客体保护粒度可概括为如下四种可能的模式:1)整个关系(relations);2)关系的各单独元组(行, tuple);3)关系的各属性(列, column);4)关系的各单独元素(element)。

在多级关系数据库系统中,由于各种客体的内在层次性,

使得 TCB 的设计和评估者对客体的显式标记和隐含标记进行了区分。这种差别仅仅在 TCB 的边界范围内是有意义的;而从 TCB 之外的来看,所有的存储客体都具有显式标记^[1]。

目前,元组级标记被所有经过评估的主流数据库系统(Sybase、Oracle、Informix)的可信或安全版本所采用。

不同系统采用不同客体粒度的依据除了上述的系统体系结构、可移植度、接口类型、目标用户等因素外,还有一个重要的因素,就是系统所采用的多级关系模型对真实世界(Real World)的描述、表达能力,以及这种能力的完备性和完整性。换个角度来说,就是数据库系统的管理者和应用开发者可以使用多级数据库所提供的接口功能来完整、全面、方便地表达应用的逻辑要求,而无需利用其它处理和转换手段来间接地表达应用所需的逻辑。

多级安全数据库系统的两种主要的体系结构是:“内核化”体系结构和“复制”体系结构,限于篇幅,本文将不对该问题进行过多讨论。

下面的内容将对各种模式的特点和相关的问题进行分析和比较,其中的例子仍沿用在多级关系安全数据库研究中被广泛使用的“飞船宇航”关系。

3. 客体标记粒度模式的比较

3.1 关系和属性级标记

关系级粒度是指整个关系只有唯一定义的分级级别。换句话说,关系中的所有数据元素被分为相同安全级别。关系级标记粒度的一个例子:

表1 关系级客体标记模式

SECURE 用户看到的的关系 SOD:

Starship	Objective	Destination
Enterprise	Exploration	Talos
Voyager	Spying	Mars

UNCLASSIFIED 用户看到的的关系 SOD:

Starship	Objective	Destination
Enterprise	Exploration	Talos
Voyager	Exploration	Talos

属性级粒度意味着关系的每个属性(列)分配一个单一的

^{*} 国家863计划项目(863-301-6-5)资金资助。

安全级别。整个关系中某属性的所有值都被分为系统安全级别。属性级标记的例子：

表2 属性级客体标记模式

SECURE 用户看到的一个关系 SOD:

Starship	U	Objective	C	Destination	S
Enterprise		Exploration		Talos	
Voyager		Spying		Mars	

由上可知,关系级标记和属性级标记粒度都是基于“复制结构”的实现,即:不同敏感度(安全级别)的数据进行物理或半物理的分割,形成“多库”结构,然后再由某个多级数据库管理系统(MLDBMS)对整个数据信息逻辑进行协调整合。

关系级客体标记粒度从直觉上来看更容易实现,可以利用通常的商用数据库产品来实现。但其缺点就是牵强、不灵活。它适合于安全级别较少且相对固定的应用情况,不能灵活变动以适应应用和管理的要求。而元素级标记则引发和加重了一系列相关问题(如:复杂多实例等)^[2,3]。

对属性级标记,由于其显见的表达缺陷(如:它只能适应非常少的安全级别;如果固定关系数量,则可能无法对大型系统进行建模),使它的应用受到极大的限制,或者说,它只适合对具体的应用需求进行设计和开发,无法对各种需求不断变化的应用提供灵活的支持。

正是由于上述关系级标记和属性级标记的缺点,使得它们对数据库厂商和数据库内核的研究者失去了吸引力。对它们进行研究和实现的,主要集中于通用安全领域和较上层软件的研究和开发者,他们利用通用的商业级数据库系统,通过数据库系统的多重再配置和二次开发,实现上述模型的表达能力,为较具体的应用提供安全数据管理支持。而各数据库厂商和数据库内核技术的研究者则把注意力集中到了元组级和元素级标记粒度上。

3.2 元组及元素级标记

元组级标记是目前国内外各主流数据库厂商多级安全系统所共同采用的模式。在这种标记粒度下,关系中的每个元组(记录)被标记以一个安全标记,元组中的各元素(字段)被隐含定义了与元组标记相同级别的标记。下面是该粒度标记的一个例子:

表3 元组级客体标记模式

SECURE 用户看到的关系 SOD:

Starship	Objective	Destination	TC
Enterprise	Exploration	Talos	U
Voyager	Spying	Mars	S

UNCLASSIFIED 用户看到的关系 SOD:

Starship	Objective	Destination	TC
Enterprise	Exploration	Talos	U

元素级标记则进一步将标记引入到关系中元组的每个字段中,下面是该标记粒度的一个简单例子。

表4 (单纯)元素级客体标记模式

SECURE 用户看到的关系:

Starship	Objective	Destination			
Enterprise	U Exploration	U Talos			U
Voyager	U Spying	S Mars			S

UNCLASSIFIED 用户看到的的关系:

Starship	Objective	Destination			
Enterprise	U Exploration	U Talos			U
Voyager	U NULL	U NULL			U

在实际设计和开发中,由于多级关系模型描述真实世界以及系统完整性的需要,在元组一级通常增加元组标记,因为在关系数据库中,数据的访问和调度是以记录和记录集为单位进行的,所以在访问控制机制中就要求在元组级提供标记,以作为元组所有和引用关系的控制依据。这种元素级标记粒度中提供元组标记的例子也在下面示出。

表5 提供元组标记的元素级客体标记模式

Starship	Objective	Destination	TC
Enterprise	U Mining	M ₁ Talos	U M ₁
Enterprise	U Spying	M ₂ Talos	U M ₂
Enterprise	U Exploration	U Talos	U U

由上面的例子可以看出,元素级标记粒度为多级安全关系数据库提供了最大的灵活性和充分的逻辑表达能力。但当前的研究表明它还缺乏完备性,即它的灵活表达能力还缺少有力和完整的理论和语义支持。较为直接的一个问题就是在这种粒度情况下的双重多实例问题——元组级多实例和属性级多实例^[4,5]。

关于客体标记粒度问题的研究和争论一直是多级关系数据库领域的中心内容。相对于早已成熟的关系代数和关系数据库理论,元素级标记的多级关系理论还缺乏完整性、完备性和可靠性。另外的一个问题就是这种标记粒度下的多级关系对真实世界的表达方式的不可理解性。关系数据库的发展和流行,其主要原因就是关系理论和语义的成熟、完整,以及它的良好理解性和直观性。

关于上述问题的讨论和研究一直是多级安全关系数据库系统研究和争论的热点,并针对该问题已经提出了多种模型和处理方法。但是这些模型的一个问题就是:它们都抛开了数据库基本操作和基本语义的理论模型,距离模型的实际应用还有许多问题需要解决。

主流数据库的多级安全版本都是以元组级标记粒度为出发点和立脚点,其重要原因就是这些系统以通用性、可用性和系统性能保证为应用目标,并以模型的成熟性、灵活性、可用性作为系统功能设计的基础。

4. 元组级和元素级标记的详细分析

前面已经说明,元组级和元素级标记粒度模式所实现的是 DBMS MAC 控制机制,即数据库内核的安全控制,它们为数据库服务器厂商和研究者所研究和运用。下面将结合我们在安全数据库管理系统 OpenBASE Secure 上的经验,对内核化结构多级关系数据库系统中的元组级和元素级标记模型进行详尽的分析。

要全面权衡元组级标记和元素级标记,首先要对多级关系数据库中的两类多实例进行揭示和分析。在多级关系数据库系统中有两类多实例:实体多实例(entity polyinstantiation)和元素多实例(element polyinstantiation)。多实例是多级数据库系统防止推论(存储隐蔽通道)的基本手段,同时也是数据管理的一个有力手段。实体多实例在关系中会发生多个主键(AK)相同而主键的安全标记不同;而元素多

实例是指关系中有多个主键和主键标记相同、非主键属性值不同的元组存在情况^[6,7,8]。

众所周知,单级关系中的主键是元组的唯一性标识,即主键起到标识真实世界唯一性事物的作用。而在多级关系中,实体多实例的实际意义就在于:对真实世界中的唯一事物,出于安全考虑,它有不同的侧面;或者可以解释为:同一名称与多个敏感程度和保密性不同的实际事物相联系。为了在关系中唯一确定实体标识,实际系统中常用的方案就是将元组的外在主键和与主键相联系的标记联合作为“复合实体标识”。

元素多实例情况的产生实际上涉及到高级别主体对低级别主体的信任和数据借用及引用的问题。这在实际中是很容易理解并经常发生的。比如情报的收集人会从不同级别的提供者那里提取情报数据,他们之间就会有信任度以及数据借用和引用情况的存在。同样,为解决关系中元组的唯一标识问题,可能的方案就是将用户定义主键与元组标记结合作为“复合主键”,起到唯一确定元组的作用。下面是一个例子:

表6 多实例情况的例子

Starship	Objective		Destination		TC	
Enterprise	S	Spying	S	Rigel	U	S
Enterprise	U	Exploration	U	Null	U	TS
Enterprise	U	Exploration	U	Talos	U	U

分析了上述两种多实例,可以看出,从直觉上,元素级标记可以非常好地同时支持两种多实例的并存。而元组级标记则只能实现一种多实例情况:或者实体多实例或者元素多实例。多级关系理论的证明表明,元素级和元组级访问控制(提供相应级别的标记实现)之间通过关系分解存在着确定性的转换关系,元组级标记可以分别实现两类多实例中的任何一种,其原因就在于此^[5]。

从多级安全的目标和实际系统的功能性方面来说,元素级标记粒度模式是多级关系数据库系统的最理想情况。在这种标记粒度模式下,关系数据库系统的最小数据单位与安全标记发生联系,可以为系统管理者和应用开发者提高最灵活的控制和数据表达能力。但应用该粒度模型的首要问题就是解决该粒度条件下多级关系形式化模型的完整性和语义的完备性,也就是模型的可用性问题。

完整性的保证是多级关系数据库系统的最大挑战。标记粒度的细化带来了完整性控制的复杂化。元素级标记在提供最大的灵活性和表达能力时,不可避免地对系统数据的完整性控制施加了各种限制,它们包括:多级实体完整性、标记完整性、主键完整性、多级引用完整性等等,而且这些完整性机制的性能代价也是不可低估的。而在完整性控制中所采用的任何折中方案都是以牺牲系统的安全性、增加系统的隐蔽通道(信号通道)为代价的。

实现元素级标记的挑战不仅在数据库设计开发者需要理清安全策略的完整语义,而且数据库系统的管理和应用者还要清楚地了解所运行系统的安全特性,能够准确把握和运用这些特性来描述真实世界的数据库,为应用逻辑提供一致的数据管理服务^[4]。正是由于元素级标记使得系统管理和应用设计工作的复杂化以及系统性能的不可预见,因此商用大型多级关系数据库系统均以元组级的标记作为系统特性目标。

从上面的元素级和元组级标记的说明和例子中可以看出,元组级标记是元素级标记的基础,基于元素级标记的访问控制和完整性控制都是以元组级标记的访问控制和完整性控

制机制为基础展开的。因此从这一点而言,元组级标记及以它为基础的访问控制和完整性控制机制的完整、可靠实现可以为元素级标记的实现打下坚实的理论和实际工程基础。

总结 在前面,对多级关系数据库系统可能的4种客体标记粒度进行了整体情况的比较和分析。粗略来说,关系级标记主要被信息安全研究和工程人员所采用,他们以现有商用一般安全性的数据库系统为特定的应用场合实现多级安全的“多”数据库系统。这样的多级数据库系统所提供的灵活性是有限的,而且由于各单级数据库系统本身最多提供了审计保护级的功能,因此它们需要底层高安全级别操作系统环境的功能支持和保护。

而属性级标记,由于其显见的的能力限制,它只是作为客体标记粒度的一种可能情况而被提出,作为与其它标记模式的对比。对该粒度模式的研究很少,而且主要在早期研究者的文献中出现。这种标记模式与关系标记粒度同属于复制体系结构,以它们为基础所实现的系统强制访问控制策略被称为 OS MAC。

元组级和元素级标记粒度是数据库内核研究者和数据库厂商所提出的标记模式,而且它们对底层操作系统的依赖较上述两种客体标记粒度要弱很多,相对于 OS MAC,它们所实现的是 DBMS MAC,即数据库系统内核的强制访问控制机制。

元组级标记和元素级标记之间存在着逻辑的承袭性以及分解和转换关系。围绕元组级标记所实现的强制访问控制和完整性控制机制是元素级标记的基础。对标记粒度的过分追求而忽视了与之相对应的完整性控制和隐蔽通道控制,或者在完整性上的折中和妥协,都是以牺牲系统的安全性为代价的。针对不同目的的应用,研究组织会按不同的需要作出不同的选择。

参 考 文 献

- 1 Trusted Database Management System Interpretation of the Trusted Computer System Evaluation Criteria, NCSC-TG-021. National Computer Security Center, April 1991
- 2 Jajodia S, Sandhu R S, Blaustein B T. Solutions to the Polyinstantiation Problem. In Information Security: An Integrated Collection of Essays. M. Abrams, S. Jajodia, H. Podell, eds., IEEE Computer Society, 1995
- 3 Jajodia S, Sandhu R. Polyinstantiation Integrity in Multilevel Relations. In: Proc. of the 1990 IEEE Symposium on Research in Security and Privacy, 1990. 104~115
- 4 Qian Xiaolei, Lunt T F. Tuple-level vs. element-level classification. Database Security VI: Status and Prospects, 301~315
- 5 Jajodia S, Sandhu R. Toward a Multilevel Secure Relational Data Model. In: Proc. of the ACM SIGMOD Intl. Conf. on Management of Data, 1991. 50~59
- 6 Lunt T F. The True Meaning of Polyinstantiation: Proposal for an Operational Semantics for a Multilevel Relational Database System. In: Proc. of the Third RADC Database Security Workshop, 1990. 26~36
- 7 Lunt T F. Polyinstantiation: An Inevitable Part of a Multilevel World. In: Proc. of the Fourth IEEE Workshop on Computer Security Foundations, 1991. 236~238
- 8 Sandhu R, Jajodia S, Lunt T F. A New Polyinstantiation Integrity Constraint for Multilevel Relations. In: Proc. of the IEEE Workshop on Computer Security Foundations, 1990. 159~165