多级安全 OS 与 DBMS 模型的信息流及其一致性分析

李 斓[1),2)] 冯登国[3)] 徐 震[3)]

[1)](上海交通大学信息安全工程学院 上海 200030)

[2)](上海市信息安全综合管理技术研究重点实验室 上海 200030)

[3)](中国科学院软件研究所信息安全国家重点实验室 北京 100080)

摘 要 数据库安全与操作系统安全密不可分,如果多级安全 DBMS 的安全策略不违反 OS 的安全策略,那么可以使用多级安全 OS 的安全机制来实现 DBMS 的部分安全功能,如强制访问控制.信息流分析使我们能更好地理解安全策略的意义和内容.该文给出了多级安全 OS 模型和以该模型为基础的多级安全 DBMS 模型,首次详细分析了它们在强制访问控制策略下的信息流集合.经过主客体的映射后,证明了数据库与操作系统的信息流集合是一致的,这个结论保证了利用 OS 的机制来实现 DBMS 的强制访问控制的合理性.

关键词 信息流;多级安全数据库;多级安全操作系统;一致性;多级关系模型

中图法分类号 TP309

Information Flow Analysis and Consistency of Multilevel OS and DBMS Model

LI Lan[1),2)] FENG Deng-Guo[3)] XU Zhen[3)]

[1)](*School of Information Security Engineering, Shanghai Jiaotong University, Shanghai* 200030)

[2)](*Key Laboratory of Integrate Administration Technologies for Information Security, Shanghai* 200030)

[3)](*State Key Laboratory of Information Security, Institute of Software, Chinese Academy of Sciences, Beijing* 100080)

Abstract The security of database system (DBMS) is closely related to security of operation system (OS). Multilevel security DBMS can implement its security functions, such as mandatory access control (MAC), using the mechanisms of multilevel security OS only when the security policy of DBMS not violate the security policy of OS. Information flow analysis is beneficial to understanding the meaning and content of the security policies. The theory of information flow analysis is introduced at first. Before discussing the information flows, the multilevel security OS model and multilevel security DBMS model are presented. Then the information flows allowed in multilevel security OS and multilevel security DBMS are analyzed for the first time. The methods of analysis in OS and DBMS are similar: After concluding the objects that contain information, all the information flows among those objects are discussed according to the operations in the OS and DBMS. Because any object of DBMS can be mapped to one or a group of objects in OS, the category set of DBMS is a subset of the category set of OS. Finally the set of DBMS's information flows is proved consistent with the one of OS's information flows based on the definition of the consistence between two information flow sets. The conclusion guarantees the soundness of implementing mandatory access control of multilevel security DBMS using the mechanisms of multilevel security OS.

Keywords information flow; multilevel security DBMS; multilevel security OS; consistency; multilevel relation model

---

收稿日期:2003-06-26;**修改稿收到日期**:2005-01-27.**本课题得到国家"九七三"重点基础研究发展规划项目基金**(G1999035802)、**国家自然科学基金**(60025205、60273027)、**国家"八六三"高技术研究发展计划项目基金**(2002AA141080)**资助**.**李 斓**,**男**,1977 **年生**,**博士**,**讲师**,**主要研究方向为系统与网络安全**.**冯登国**,**男**,1966 **年生**,**博士**,**研究员**,**博士生导师**,**主要研究方向为信息安全和计算机网络安全**.**徐 震**,**男**,1976 **年生**,**博士**,**主要研究方向为系统与网络安全**.

1 背 景

数据库管理系统(DBMS)离不开操作系统(OS)的支持,要保证数据库有较高的安全特性,操作系统同样也要有较高的安全保证.如果操作系统能对数据库的文件做很好的保护,那么数据库可以专注于其特有的安全机制.如果将安全数据库建立在安全操作系统的基础之上,我们能利用操作系统已有的安全机制来实现数据库的安全功能.

多级安全操作系统的强制访问控制(MAC)与多级安全数据库的强制访问控制有着非常多的共性,只是针对的主客体范围不同.但是数据库管理系统是操作系统上的应用程序,它的主客体最终会归结为操作系统上的主客体,数据库的访问操作也最终要映射为操作系统的访问操作.既然操作系统已经定义好了自己的强制访问控制规则,我们就可以利用它的功能来实现数据库的强制访问控制.但是,如果数据库中的某些操作导致的信息流向无法用操作系统的操作来表达,那么我们就不能保证这种方法的合理性.因此,我们需要找到一种手段分析信息在系统中的传递和变迁,并能证明操作系统的各种操作足以实现数据库中所有的信息流动.

信息流分析的方法使我们能够清楚地理解系统内部信息的流动方向.每个安全系统都有它的安全策略,这些安全策略定义了系统信息流动的规则,通过信息流分析,我们可以更好地理解安全策略的内容与意义.在安全系统的信息流分析方面已经有许多的研究工作,例如信息流的层次模型[1]、面向对象系统的信息流分析[2,3]、RBAC 的信息流分析[4]以及基于角色的安全系统的信息流分析[5]等等.多级系统的安全策略主要是 BLP 模型[6],既适用于操作系统也适用于数据库.本文首次通过对多级安全操作系统模型与其上的多级安全数据库模型的信息流进行分析,描述强制访问控制策略下信息的流动方向,并证明数据库的信息流与操作系统的信息流是一致的.

本文第 2 节介绍信息流分析的基础;第 3 节描述安全操作系统模型及其信息流分析;第 4 节描述安全数据库管理系统模型及其信息流分析;第 5 节证明数据库管理系统模型的信息流与操作系统的信息流是一致的;最后一节是结论.

2 信息流分析基础[1,5]

信息流的分析包括三个元素:信息的发送者、信息的接收者以及促使信息流动的操作,三者缺一不可.只有在对象之上的操作才能引起信息的流动,因此必须有一个程序或者操作命令作为信息流动的手段.

定义 1. 信息的流动:如果有一个包含了对象 A 和 B 的程序执行完毕之后,对象 B 中包含了原来存在于对象 A 中的信息,那么我们就说信息从对象 A 流动到了对象 B.记为 $A \xrightarrow{I} B$.

系统中的信息流有两种:合法信息流与非法信息流.合法信息流是符合系统安全策略的所有信息流,而不合法的信息流都属于非法信息流.通过对信息流进行分析,我们可以判断某种信息的流动是否合法,或者说是否符合系统的安全策略.在安全策略中,总是认为一些对象之间的信息流肯定是合法的,这样的对象就可以形成一个个集合,我们定义为范畴.

定义 2. 范畴(category)是一些对象的集合,在系统的安全策略下信息可以在这些对象之间自由流动.

如果把一个安全策略记为 P,那么系统合法的信息流集合可以记为 $F(P)$,那么存在一个函数 f 可以列举出这个信息流集合:

$$f: P \times 2^{CO} \times F \times 2^{OP} \longrightarrow F(P),$$

其中 P 是安全策略,CO 是系统中的范畴,F 是所有可能的信息流,OP 是系统中的操作.但是安全策略允许的信息流集合与实现安全策略的系统的信息流集合并不一定相同.假设函数 Φ 用来列举实现的信息流,则有

$$\Phi: P \times 2^{CO} \times I \times 2^{OP} \longrightarrow F(I).$$

我们可以如下定义两个信息流集合之间的相等关系和一致关系.

定义 3. 信息流集合相等:两个信息流集合 $F(P)$ 和 $F(I)$ 相等,记为 $F(P) \equiv F(I)$,当且仅当(1)它们处理同样的范畴集合,即 $CO_P \equiv CO_I$;(2)$F(P)$ 中的每一条信息流都在 $F(I)$ 中有一条信息流与之对应,反之亦然.相等关系满足交换率.

定义 4. 信息流集合一致:信息流集合 $F(I)$ 与另一个信息流集合 $F(P)$ 是一致的,当且仅当(1)$F(I)$ 处理的范畴集合包含于 $F(P)$ 处理的范畴集合,即 $CO_I \subseteq CO_P$;(2)$F(I)$ 中的每一条信息流在 $F(P)$ 中都有一条信息流与之对应,即 $F(I) \subseteq F(P)$.一致关系不满足交换率.

3 安全操作系统模型及其信息流集合

多级安全操作系统模型采用 BLP 模型作为其安全策略,它定义了操作系统的强制访问控制策略.

3.1 Bell-La Padula 模型[6]

Bell-La Padula 模型(BLP 模型)是 1976 年由 Bell 和 La Padula 提出的强制访问控制模型.我们

把系统中的对象称为客体(object)，用户称为主体(subject)，BLP 模型中的所有主体和客体都有一个标签，用 $\lambda(o)$，$\lambda(s)$ 分别表示客体和主体的标签，标签包含两个元素：密级 C(class)和范围 G(category)，因此我们也可以用一个二元组 (C,G) 来表示，标签之间的关系有两种：支配关系与不可比关系. 一个级别 $\lambda_1(C_1,G_1)$ 支配另一个级别 $\lambda_2(C_2,G_2)$ 当且仅当 $C_1\geq C_2$ 并且 $G_1\supseteq G_2$，我们记为 $\lambda_1\geq\lambda_2$. 如果 $\lambda_1\neq\lambda_2$，那么标签 λ_1 支配标签 λ_2 也可以记为 $\lambda_1>\lambda_2$. 如果两个级别不存在支配关系，则它们不可比. BLP 模型关于读写控制的规则如下：

(1)(简单安全特性)主体 s 能够读客体 o 必须满足 $\lambda(s)\geq\lambda(o)$；

(2)(星特性)主体 s 能够写客体 o 必须满足 $\lambda(o)\geq\lambda(s)$.

简单安全特性保证系统中的主体对客体“不能上读”，而星特性保证主体对客体“不可下写”. 这两个特性控制信息只能在同级之间或从低级向高级流动.

3.2　多级安全操作系统模型

不失一般性，我们认为操作系统的一切客体都可以用文件来表示，而访问文件的主体都为操作系统上的进程. 进程与文件都有自己的标签，记为 $\lambda(p)$ 与 $\lambda(f)$. 操作系统中存在最小标签 $\lambda_{\min}$ 和最大标签 $\lambda_{\max}$：

$$\exists\lambda_{\min}\lambda_{\max}\ \forall p\in os, f\in os, \lambda(p)\geq\lambda_{\min}, \lambda(f)\geq\lambda_{\min}$$

且 $\lambda_{\max}\geq\lambda(p)$，$\lambda_{\max}\geq\lambda(f)$.

进程与文件之间的操作有 3 种：读、写和添加. 根据强制访问控制策略，这 3 个操作分别应该满足下面的 3 条规则.

规则 1.　根据 BLP 模型的简单安全规则，进程 p 能读文件 f 的前提是 $\lambda(p)\geq\lambda(f)$.

规则 2.　对文件内容进行添加不需要获取文件的原有内容，根据 BLP 模型的星特性，进程 p 能添加文件 f 的前提是 $\lambda(f)\geq\lambda(p)$.

规则 3.　写文件的前提是读取文件的内容，根据 BLP 模型的简单安全规则与星特性，进程 p 能写文件 f 的前提是 $\lambda(p)=\lambda(f)$.

3.3　多级安全操作系统中强制访问控制允许的信息流

在分析信息流之前，我们需要归纳出操作系统中拥有信息的对象. 毫无疑问，文件 f 是信息的容器，它储存的内容就是信息. 此外，操作系统的主体也是与信息密切相关的. 进程通过输入集 I 与输出集 O 来与系统进行信息交换. 因此操作系统中拥有信息的对象包括文件 f、输出集 O 与输入集 I. 在 BLP 模型的安全策略下，所有的对象都应该有标签，但是多级操作系统只给文件定义了标签，输入集与输出集的标签必须间接地获取. 作为系统与外界交换信息的中介，进程把从外部引入的信息作为它的输入，把系统呈现给它的信息作为输出，因此我们认为输入集与输出级的标签与进程的标签是相同的，即

进程 p 的输入集用 I^p 表示，且 $\lambda(I^p)=\lambda(p)$；

进程 p 的输出集用 O^p 表示，且 $\lambda(O^p)=\lambda(p)$.

范畴是对象的集合，因此在操作系统中，文件、输入集与输出集都是范畴的组成部分. 根据强制访问控制策略，系统允许的信息流有两种：同级流动，或者从低级流向高级. 一个范畴内的对象之间信息流动是自由的，我们可以把所有具有相同标签的对象放在同一个范畴中. 因此多级安全操作系统的范畴 CO 满足下列条件：

(1)系统中的每一个标签都有一个范畴与之对应，这个标签就是范畴的标签.

$$\forall\lambda_i \text{ in } os, \exists CO_i \text{ in } os: \lambda(CO_i)=\lambda_i.$$

(2)系统中的任何对象都属于某个范畴，对象的标签与所属范畴的标签相同.

$$\forall f \text{ in } os, \exists CO_i \text{ in } os: f\in CO_i\wedge\lambda(f)=\lambda(CO_i);$$

$$\forall I^p \text{ in } os, \exists CO_i \text{ in } os: I^p\in CO_i\wedge\lambda(I^p)=\lambda(CO_i);$$

$$\forall O^p \text{ in } os, \exists CO_i \text{ in } os: O^p\in CO_i\wedge\lambda(O^p)=\lambda(CO_i).$$

(3)系统中没有两个范畴的标签是一样的.

$$\forall CO_i,\ CO_j \text{ in } os, \text{且 } i\neq j: \lambda(CO_i)\neq\lambda(CO_j).$$

既然只有操作能引起信息流动，那么我们就以操作系统中三种操作来分析它的信息流：

(1)进程 p 读取文件 f 的时候，信息从文件 f 流向进程 p 的输出集：$f\xrightarrow{I}O^p$. 根据系统的规则要求，要么 $\lambda(f)=\lambda(p)$，要么 $\lambda(p)>\lambda(f)$：

如果 $\lambda(f)=\lambda(p)$，那么 $\exists CO_i$ in os，使得 $f\in CO_i\wedge O^p\in CO_i$，信息流为 $CO_i\xrightarrow{I}CO_i$.

如果 $\lambda(p)>\lambda(f)$，那么 $\exists CO_i$，CO_j in os，使得 $f\in CO_i\wedge O^p\in CO_j\wedge\lambda(CO_j)>\lambda(CO_i)$，信息流为 $CO_i\xrightarrow{I}CO_j$.

(2)进程添加文件 f 的内容无需获取文件的其他信息，只是把进程自己的信息传递给文件，因此信息从进程的输入集流向文件 f：$I^p\xrightarrow{I}f$. 系统的规则要求或者 $\lambda(f)=\lambda(p)$，或者 $\lambda(f)>\lambda(p)$：

如果 $\lambda(f)=\lambda(p)$，那么 $\exists CO_i$ in os，使得 $f\in CO_i\wedge I^p\in CO_i$，信息流为 $CO_i\xrightarrow{I}CO_i$.

如果 $\lambda(f)>\lambda(p)$，那么 $\exists CO_i, CO_j$ in os，使得 $f\in CO_i\wedge I^p\in CO_j\wedge\lambda(CO_i)>\lambda(CO_j)$，信息流为 $CO_j\xrightarrow{I}CO_i$.

(3)进程 p 写文件 f 的操作包含着读取文件的内

容,因此信息流包括两个方面:$f \xrightarrow{I} O^p$ 和 $I^p \xrightarrow{I} f$. 根据规则的要求,必须满足 $\lambda(f)=\lambda(p)$:

$\exists CO_i \text{ in } os$,使得 $f \in CO_i \wedge I^p \in CO_i \wedge O^p \in CO_i$,信息流为 $CO_i \xrightarrow{I} CO_i$.

信息直接流动只能出现在不同类的对象之间,也就是或者从进程流向文件,或者从文件流向进程. 文件之间如果不通过进程的操作是不可能进行数据交换的,同样进程之间如果需要数据交换必须借助操作系统的资源进行,这种资源在操作系统中也是当成文件来看待,因此,多级安全数据库的直接信息流只有上述三种情况.

综上所述,多级安全操作系统的信息流有两种:范畴内的信息流动与范畴间的信息流动,根据范畴的定义,范畴内的信息可以任意流动,但是范畴间的信息流动受到强制访问控制策略的限制,在我们的操作系统中如果允许一个范畴的信息流动到另一范畴,那么后者的标签必须支配前者的标签,因此我们可以归纳出多级安全操作系统的直接信息流集合:

$$F(I)_{os}\{CO_i \xrightarrow{I} CO_j \mid \lambda(CO_j) \geq \lambda(CO_i) \wedge (\exists p, f:(\lambda(p)=\lambda(CO_i) \wedge \lambda(f)=\lambda(CO_j)) \vee (\lambda(p)=\lambda(CO_j) \wedge \lambda(f)=\lambda(CO_i)))\}.$$

4 多级安全数据库模型

为了保证数据库系统的数据完整性,我们采用严格的 BLP 模型作为多级安全数据库模型的安全策略.

4.1 严格的 BLP 模型

严格的 BLP 模型是原有模型的改进,它采用了严格的星特性:

(1)(简单安全特性)主体 s 能够读客体 o 必须满足 $\lambda(s) \geq \lambda(o)$;

(2)(严格星特性)主体 s 能够写客体 o 必须满足 $\lambda(o)=\lambda(s)$.

严格的 BLP 模型同样不允许上读,但是也不允许上写,主体写客体的条件是他们拥有相同的标签.

4.2 多级安全数据库模型[7,8]

关系数据库中的客体有很多种,但是真正存储数据的客体只有关系中的记录. 因此我们分析一种简单的数据库模型:假如数据库系统中的数据库 D 是一个所有人都可以访问的容器,数据库中包含多个关系 T,这些关系在创建的时候都有自己的标签 $\lambda(T)$,关系中存储的是数据单元——记录. 系统中具有标签的客体的最小粒度是记录 E,一个关系 T 可以包含不同级别的记录,但必须满足

$$\forall E \in T, \quad \lambda(E) \geq \lambda(T).$$

4.3 多级安全数据库模型的信息流分析

多级安全数据库的体系结构有两种:可信主体与 TCB 子集[9]. 如果采用可信主体的体系结构,安全数据库完全依赖自身来完成强制访问控制,所以我们分析可信主体的安全数据库与安全操作系统的信息流集合之间的一致性是没有必要的. 因此,我们只考虑采用 TCB 子集这种体系结构的安全数据库. 在这种多级安全数据库中,关系是一个跨级别的对象,它包含的记录的标签不一定相同. 针对这样的客体进行强制访问控制是不方便的,因此有必要将关系分解成多个小的客体,这些客体中信息的标签是一致的. 于是我们提出了下面的多级关系模型.

4.3.1 多级关系模型

Sandhu 等提出了标签粒度在字段级别上的多级数据模型[8],而我们的数据库模型中标签级别只达到了记录级,因此可以采用更简单的多级关系模型. 因为数据库的关系 T 可以拥有不同级别的记录,所以它是一个多级关系 MR. 为了方便强制访问控制,我们将多级关系分解成更小的客体,称为单级关系 SR.

定义 5. 多级关系 MR:包含有不同标签的记录的关系. 多级关系包含的记录的标签都要支配该关系的标签.

定义 6. 单级关系 SR:某个 MR 的成员,它拥有自己的标签,而且包含了这个 MR 中所有该标签上的记录.

一个多级关系 MR 由多个单级关系 SR 组成,每个 SR 包含的是与自己的标签相同的记录,我们可以通过下面的规则定义多级关系模型:

(1)每个多级关系分成多个单级关系,单级关系的标签都支配多级关系的标签. 多级关系的表示如下:

$$MR(N,(A_1,A_2,\cdots,A_n),K,\lambda_{MR},SRL(SR_1,SR_2,\cdots,SR_n)),$$

其中 N 是关系的名称;$(A_1,A_2,\cdots,A_n)$ 是关系的模式,即属性列表;K 是关系的主键,可以为空;λ_{MR} 是关系的标签;$SRL(SR_1,SR_2,\cdots,SR_n)$ 是多级关系的单级关系列表,满足

$$\forall SR_i \in SRL, \quad \lambda(SR_i) \geq \lambda_{MR}.$$

(2)多级关系的每个 SR 标签都不相同,关系中的记录分别存放在与之标签相同的 SR 中:

$$\forall SR_i, SR_j \in SRL: \lambda(SR_i) \neq \lambda(SR_j);$$

$$\forall E \in MR, \exists SR_i \in SRL: \lambda(E)=\lambda(SR_i) \wedge E \in SR_i.$$

多级关系包含了不同级别的信息,如果数据库与操作系统采用相同的标签系统,我们不能将一个多级关系映射为单一的操作系统文件,但是单级关系却可以和文件很好地对应起来. 因此根据这个多级关系模型,我们可以用多个操作系统文件表示一个多

级关系.同样,数据库的主体与操作系统的进程对应,每个数据库用户都有一个或多个活跃的进程来表示,而且它们的标签也是相同的.经过这些映射后,我们就可以利用操作系统的强制访问控制来实现数据库的强制访问控制.

4.3.2　多级安全数据库中强制访问控制允许的信息流

与操作系统类似,我们首先分析数据库中包含信息的对象.因为多级关系的记录标签不同,因此我们只把单级关系看成对象,同样数据库的用户也有输入集 I^U(用户对系统的输入)和输出集 O^U(系统对用户的输出),而且它们的标签与用户相同.因此数据库范畴的成员是单级关系、输入集和输出集.与操作系统一样,我们也把相同标签的对象归到同一个范畴当中,这些范畴满足操作系统的范畴应该满足的三个条件.

同样,只有客体上的操作才能导致多级安全数据库的信息流动.而在数据库模型中,操作都是针对关系这一级目标的,不会在某一条记录上定义一种操作.这些操作包括查询关系(select table)、在关系中插入记录(insert table)、删除关系中的记录(delete table)和更新关系中的记录(update table).用户对某个关系实施一个操作的先决条件是能够看到关系的存在,也就是说必须满足 $\lambda(U)\geq\lambda(T)$[11].下面通过这些操作来分析多级安全数据库中的信息流:

(1)查询关系是用户获取关系中记录信息的过程,根据规则要求,用户查询到的多级关系的记录必须满足 $\lambda(U)\geq\lambda(E)$,因此信息是从多个单级关系流向用户的输出集,但是这些单级关系的标签都被用户的标签所支配:

$$U \text{ select table } MR:\{SR_i\xrightarrow{I}O^U \mid SR_i\in SRL_{MR}\wedge \lambda(U)\geq\lambda(SR_i)\}.$$

(2)在关系中插入记录相当于用户利用自己的信息生成一条新的记录,这条记录的标签与用户标签相同,需要插入到同级别的单级关系中:

$$U \text{ insert table } MR:\{I^U\xrightarrow{I}SR_i \mid SR_i\in SRL_{MR}\wedge \lambda(U)=\lambda(SR_i)\}.$$

(3)从关系中删除记录分为三个步骤:读取关系中的记录,选择满足条件的记录,然后删除这些记录.根据规则,用户只能删除与其有相同标签的记录,于是信息流分为三个部分:

$$U \text{ delete table } MR:\{SR_i\xrightarrow{I}O^U \mid SR_i\in SRL_{MR}\wedge \lambda(U)=\lambda(SR_i)\}\cup\{O^U\xrightarrow{I}I^U\}\cup \{I^U\xrightarrow{I}SR_i \mid SR_i\in SRL_{MR}\wedge\lambda(U)=\lambda(SR_i)\}.$$

(4)更新关系的记录可以分为三个步骤:读取关系中的记录,修改满足条件的记录的某些内容,然后将新的记录写入关系中.根据安全策略的要求,用户可以更新所有能访问的记录,但是生成的新记录的标签与用户相同.因此信息流分为三部分:

$$U \text{ update table } MR:\{SR_i\xrightarrow{I}O^U \mid SR_i\in SRL_{MR}\wedge \lambda(U)\geq\lambda(SR_i)\}\cup\{O^U\xrightarrow{I}I^U\}\cup \{I^U\xrightarrow{I}SR_i \mid SR_i\in SRL_{MR}\wedge\lambda(U)=\lambda(SR_i)\}.$$

5　多级安全数据库与多级安全操作系统信息流的一致性

数据库作为操作系统上的应用程序,所有的主客体都将对应到操作系统上的主客体,所有的访问操作也都最终归结为操作系统上的访问关系.因此,数据库中的信息流映射到操作系统上之后,不能违反操作系统的信息流规则.上述的多级安全数据库模型的信息流应该和多级安全操作系统模型的信息流保持一致.证明这个结论之前我们先证明一个引理.

引理 1.　多级安全数据库的范畴集包含于多级安全操作系统的范畴集,即 $CO_{\text{db}}\subseteq CO_{\text{os}}$.

证明.　多级安全数据库中的主体是用户,客体是单级关系.通过两个映射关系可以把数据库的主客体对应为操作系统的主客体.用户登录数据库系统后,将会打开一个会话,所有的操作都将在这个会话中进行.而每个会话都是一个数据库的进程,最终也表现为操作系统的某个进程 p.一个用户可以打开多个会话,因此数据库用户与操作系统的进程之间是一对多的关系,我们用 φ 来表示这个映射:

$$\varphi: U\longrightarrow 2^p,$$

于是用户的输入集和输出集也映射为多个进程的输入集和输出集,分别记为 φ_I,φ_O:

$$\varphi_I: I^U\longrightarrow 2^{I^p};\ \varphi_O: O^U\longrightarrow 2^{O^p}.$$

数据库的每个单级关系映射为操作系统上的某个文件,用 ψ 来表示:

$$\psi: SR\longrightarrow f.$$

因为数据库采用与操作系统完全相同的标签系统,所以经过映射,我们可以保持对象标签的一致性:

$$\forall p\in\varphi(U):\lambda(p)=\lambda(U);$$
$$\forall f=\psi(SR):\lambda(f)=\lambda(SR).$$

根据输入集与输出集标签的性质,我们有

$$\forall I^p\in\varphi_I(I^U):\lambda(I^p)=\lambda(I^U);$$
$$\forall O^p\in\varphi_O(O^U):\lambda(O^p)=\lambda(O^U).$$

数据库系统的范畴内容包括单级关系、用户输入集和输出集,而操作系统的范畴包括文件、进程的输入集和进程的输出集.

$\forall CO_i^{db} \in CO^{db}$,假设它的标签为 λ_i.

$\forall I^U \in CO_i^{db}: \lambda(I^U)=\lambda_i$,所以 $\forall I^p \in \varphi_I(I^U): \lambda(I^p)=\lambda_i$;

$\forall O^U \in CO_i^{db}: \lambda(O^U)=\lambda_i$,所以 $\forall O^p \in \varphi_O(O^U): \lambda(O^p)=\lambda_i$;

$\forall SR \in CO_i^{db}: \lambda(SR)=\lambda_i$,所以对于 $f=\psi(SR)$: $\lambda(f)=\lambda_i$.

因此,CO_i^{db}中的所有对象最终映射为操作系统中的一个对象集,这个集合的所有成员的标签都为 λ_i,根据操作系统范畴的特性,一定 $\exists CO_j^{os}: \lambda(CO_j^{os})=\lambda_i$,而且上述集合中的所有成员都属于这个范畴,那么 CO_i^{db}可以映射到 CO_j^{os}. 既然所有的数据库范畴都能映射为操作系统的范畴,我们就可以定义范畴之间的一对一的映射关系 $\theta: CO^{db} \longrightarrow CO^{os}$.

因为映射关系 θ 的存在,我们可以得出数据库系统的范畴集合包含于操作系统的范畴集合,即 $CO_{db} \subseteq CO_{os}$. 证毕.

定理 1. 多级安全数据库模型的强制访问控制允许的合法信息流集合 $F(I_{db})$ 与多级安全操作系统模型的强制访问控制允许的合法信息流集合 $F(I_{os})$ 是一致的.

证明. 多级安全数据库模型遵循的强制访问控制策略是严格的 BLP 模型,而多级安全操作系统模型的强制访问控制策略是 BLP 模型,根据定义,严格的 BLP 模型是 BLP 模型的子集,凡是严格的 BLP 模型允许的信息流 BLP 模型也都允许,因此有 $P_{db} \subseteq P_{os}$.

根据引理 1,数据库的范畴集包含于操作系统的范畴集:$CO_{db} \subseteq CO_{os}$.

数据库模型中的合法信息流都是上述四种操作引起的,下面我们分别来说明这些信息流与操作系统信息流的关系.

(1)对于查询关系造成的信息流:$\{SR_i \xrightarrow{I} O^U \mid SR_i \in SRL_{MR} \wedge \lambda(U) \geq \lambda(SR_i)\}$,假定 $SR_i \xrightarrow{I} O^U$ 是其中的任意一条信息流,根据映射关系,令 $f_i=\psi(SR_i)$,$O^p \in \varphi_U(O^U)$,因为 $\lambda(U) \geq \lambda(SR_i)$,所以这条信息流最终可以映射为

$$\{f_i \xrightarrow{I} O^p \wedge \lambda(p) \geq \lambda(f_i)\}.$$

这等同于代表用户的进程 p 读取文件 f_i 的信息流集合,这样的进程可能有多个,根据 $F(I_{os})$ 的内容,我们可以确定 $\{f_i \xrightarrow{I} O^p \wedge \lambda(p) \geq \lambda(f_i)\} \subseteq F(I_{os})$.

因为任何一条查询关系的信息流都可以像上面描述的情况一样进行映射,所以查询关系造成的信息流集合包含于 $F(I_{os})$.

(2)插入数据的信息流为 $\{I^U \xrightarrow{I} SR_i \mid SR_i \in SRL_{MR} \wedge \lambda(U)=\lambda(SR_i)\}$,假定 $I^U \xrightarrow{I} SR_i$ 是其中的任意一条信息流,根据映射关系,令 $f_i=\psi(SR_i)$,$I^p \in \varphi_I(I^U)$.

因为 $\lambda(U)=\lambda(SR_i)$,所以信息流最终可以映射为

$$\{I^p \xrightarrow{I} f_i \wedge \lambda(p)=\lambda(f_i)\}.$$

因为 I^p 与 f_i 的标签相等,那么它们属于同一个范畴,于是这个信息流集合属于范畴内流动,因此插入数据的信息流对应的操作系统信息流是范畴内部流动,根据 $F(I_{os})$ 的内容,我们知道 $\{I^p \xrightarrow{I} f_i \wedge \lambda(p)=\lambda(f_i)\} \subseteq F(I_{os})$.

因为任何一条插入数据的信息流都可以同上面描述的情况一样进行映射,所以插入数据造成的信息流集合包含于 $F(I_{os})$.

(3)删除关系中数据的信息流包括三部分,其中第三部分相当于插入数据的信息流,而第一部分是 $\{SR_i \xrightarrow{I} O^U \mid SR_i \in SRL_{MR} \wedge \lambda(U)=\lambda(SR_i)\}$,假定 $SR_i \xrightarrow{I} O^U$ 是其中的一条信息流,根据映射关系,令 $f_i=\psi(SR_i)$,$O^p \in \varphi_U(O^U)$,因为 $\lambda(U)=\lambda(SR_i)$,所以这条信息流最终可以映射为

$$\{f_i \xrightarrow{I} O^p \wedge \lambda(p)=\lambda(f_i)\}.$$

因为 O^p 与 f_i 的标签相等,属于同一个范畴,于是这部分信息流属于范畴内流动,根据 $F(I_{os})$ 的内容,我们可以确定 $\{f_i \xrightarrow{I} O^p \wedge \lambda(p)=\lambda(f_i)\} \subseteq F(I_{os})$.

第二部分是 $\{O^U \xrightarrow{I} I^U\}$,因为删除操作总是在一个会话中进行,所以输出集到输入集之间的信息流动总是局限于同一个会话中,也就是说局限于同一个进程中,因此这部分信息流映射为

$$\{O^p \xrightarrow{I} I^p \mid p \in \varphi(U)\}.$$

而操作系统中,信息在同一个进程中的流动并不是真正的信息流动. 因此删除数据的信息流集合包含于 $F(I_{os})$.

(4)更新关系中数据的信息流包括三部分,其中第一部分相当于查询关系的信息流集合,第二部分和第三部分相当于删除关系数据的第二、三部分的信息流集合,因此更新关系中数据造成的信息流集合也是包含于 $F(I_{os})$ 的.

数据库模型的所有合法信息流都分别属于上述四种信息流集合,于是 $F(I_{db})$ 中任何一条信息流都在 $F(I_{os})$ 中有一条与之对应.

综上所述,多级安全数据库模型的强制访问控制允许的合法信息流集合 $F(I_{db})$ 与多级安全操作系统

模型的强制访问控制允许的合法信息流集合 $F(I_{os})$ 是一致的. 证毕.

6 结　论

数据库的安全以操作系统的安全为基础，使用多级安全操作系统的强制访问控制可以更方便地实现多级安全数据库的强制访问控制，但是必须保证数据库的信息流都可以用操作系统的信息流来表达. 信息流分析的方法可以帮助我们更好地理解系统的安全策略，本文首次根据主体对客体的操作分析了多级安全 OS 与多级安全 DBMS 模型在强制访问控制策略允许下的信息流集合，因为数据库的操作最终归结为操作系统上的操作，同时根据我们给出的多级关系模型以及数据库用户表示方法，关系映射为操作系统上不同标签的多个文件，数据库用户也映射为一组进程，于是数据库的信息流最终会转化为操作系统的信息流，然后我们证明了数据库的合法信息流集合与操作系统的合法信息流集合是一致的. 它们的一致性保证了用多级安全 OS 的强制访问控制机制来实现多级安全 DBMS 的强制访问控制的合理性.

参　考　文　献

1 Denning D. E.. A lattice model of secure information flow. Communications of the ACM, 1976, 19(5): 236～243

2 Bertino E., de Capitani Di Vimercati S., Ferrari E., Samarati P.. Exception-based information flow control in object-oriented systems. ACM Transactions on Information and System Security, 1998, 1(1): 26～65

3 Samarati P., Bertino E., Ciampichetti A., Jajodia S.. Information flow control in object-oriented systems. IEEE Transactions on Knowledge and Data Engineering, 1997, 9(4): 524～538

4 Osborn S. L.. Information flow analysis of an RBAC system. In: Proceedings of the 7th ACM Symposium on Access Control Models and Technologies, Monterey, California, 2002, 163～168

5 Nyanchama M., Osborn S. L.. Information flow analysis in role-based security systems. In: Proceedings of the 6th International Conference on Computing and Information, Peterborough, Ontario, Canada, 1994

6 Elliott Bell D., LaPadula L. J.. Bell-LaPadula model for secure computer systems. The MITRE Corporation, Bedford, MA: Technical Report ESD-TR-75-306, 1976

7 Sandhu R.. Design and implementation of multilevel databases. In: Proceedings of the 6th RADC Workshop on Multilevel Database Security, Southwest Harbor, Maine, 1994, 1～5

8 Sandhu R., Chen F.. The multilevel relational (MLR) data model. ACM Transactions on Information and System Security, 1998, 1(1): 93～132

9 Notargiacomo LouAnna. Architectures for MLS database management systems. In: Information Security: An Integrated Collection of Essays, Essay 19. Los Alamitos: IEEE Computer Society Press, 1995, 439～459

10 Xu Zhen, Feng Deng-Guo. Architecture of SKLOIS multilevel secure DBMS. In: Proceedings of the CCICS'2003, Wuhan, 2003, 334～328(in Chinese)
(徐　震，冯登国. SKLOIS 多级安全数据库管理系统的体系结构. 见：CCICS'2003 论文集，武汉，2003，334～328)

11 Li Lan, Feng Deng-Guo. RBAC in multilevel relation databases. In: Proceedings of the CCICS'2003, Wuhan, 2003, 329～333(in Chinese)
(李　斓，冯登国. 多级关系数据库中的 RBAC. 见：CCICS'2003 论文集，武汉，2003，329～333)

LI Lan, born in 1977, Ph. D., lecturer. His current research interests include system and network security, database, XML and XML security.

FENG Deng-Guo, born in 1965, professor, Ph. D. supervisor. His current research interests include information security and computer network security.

XU Zhen, born in 1976, Ph. D.. His current research interests include system and network security, database.

Background

This work is mainly supported by the National High Technology Research and Development Program (863 Program) under grant No. 2002AA141080. The title of the project is "Research on Key Security Technologies of OS and DBMS Platform", which aims to develop and improve the key security technologies on OS and DBMS.

When developing security DBMS platform, we can use the security mechanisms of security OS to implement the security functions of DBMS, such as mandatory access control (MAC). But there is not any theory to prove the soundness of that. Information flow analysis is beneficial for us to understanding the meaning and contents of security policy. After providing the security OS and DBMS model, authors analyze the information flows of security OS and DBMS model allowed by MAC policy and prove their consistency. The result guarantees the soundness of implementing MAC function of DBMS using security mechanisms of OS.