ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.17, No.4, April 2006, pp.750−758
DOI: 10.1360/jos170750

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

安全数据库的推理控制*

严和平+, 汪 卫, 施伯乐

(复旦大学 计算机与信息技术系,上海 200433)

Inference Control in Secure Database

YAN He-Ping+, WANG Wei, SHI Bai-Le

(Department of Computer and Information Technology, Fudan University, Shanghai 200433, China)

+ Corresponding author: Phn: +86-21-55076190, Fax +86-21-65642219, E-mail: 031021055@fudan.edu.cn, http://www.fudan.edu.cn

Yan HP, Wang W, Shi BL. Inference control in secure database. *Journal of Software*, 2006,17(4):750−758.
http://www.jos.org.cn/1000-9825/17/750.htm

Abstract: This paper investigates the inference problems due to functional dependencies (FD) and multi-valued dependencies (MVD) in a multilevel relational database with element classification schemes. The algorithms presented can greatly improve the data availability. To further deal with the indirect privacy violation, the view-based inference control method is proposed which can eliminate the secure problem brought by multi-view collusions. The theories of splitting views into the view dependency basis are the foundation of future work on the view-based inference control.

Key words: secure database; inference control; view; view collusion

摘 要: 首先对按元素划分安全级的多级数据库上由函数依赖(FD)和多值函数依赖(MVD)引起的推理问题进行了研究,所提出的推理控制算法在很大程度上提高了数据的可用性.为进一步有效防范推理所导致的敏感信息泄露,给出了基于视图的推理控制方法.该方法能够处理多视图合谋带来的安全问题.最后给出了视图依赖基划分原理,它是以后有关视图推理控制的基础.

关键词: 安全数据库;推理控制;视图;视图合谋
中图法分类号: TP311 文献标识码: A

1 引 言

信息技术的发展,使访问数据库的用户越来越多、访问控制粒度越来越细.更细粒度访问控制的实现[1],对敏感信息的安全提出了更新要求.安全数据库策略旨在保护敏感信息和数据完整性,在多级关系数据库(MRD)中,是通过BLP模型[2]的强制访问控制(MAC)机制实现的.MAC能够控制对敏感信息的直接访问,但非法用户却能以间接方式访问敏感信息,这涉及到推理控制.推理控制能够防止蓄意破坏的用户利用历史访问或相互交换查询信息,借助推理通道实现对敏感信息的间接访问.因此,安全数据库应使用推理控制来检测和清除推理

* Supported by the National Natural Science Foundation of China under Grant Nos.60303008, 69933010 (国家自然科学基金); the National High-Tech Research and Development Plan of China under Grant No.2002AA4Z3430 (国家高技术研究发展计划(863))
Received 2004-10-25; Accepted 2005-08-24

通道.

1.1 相关工作及其局限性

早在 1980 年,科研工作者们就已开始对安全数据库的推理控制问题进行研究,其工作主要集中在问题的定义及框架的建立上.Jajodia[3]证实了数据库自身约束是造成多级关系数据库中大多数推理通道的原因.目前,有关推理通道的问题不仅仅局限于多级数据库,而是已拓展到其他有关统计数据库、一般数据库、数据挖掘以及基于 Web 的推理问题等领域.

一般来讲,检测并消除推理通道技术可以分为两类:第一类是在设计阶段发现推理通道,通过修改设计提高某些敏感数据的安全级来解决[4−8],但可能导致给属性或原始数据分配过高的安全级,削弱数据的可用性;第二类是在查询期间消除推理通道[9],如果检测到潜在的推理通道就拒绝查询或修改查询,以保证敏感信息安全.这种基于查询的推理控制问题代价高,且仍会导致敏感信息的泄露(从拒绝的查询来推理敏感信息[10]).Hinke[6]提出了基于语义图的检测工具来表示语义关系和检测推理通道的方法.Brodsky[11]结合数据和数据库约束,基于用户查询历史对推理控制进行了研究;Tzong[4]针对特定的约束:函数依赖(FD)与多值函数依赖(MVD)进行了推理控制研究.但是,他们的推理控制研究工作在以下几个方面还有待深入和完善:(1) 他们的推理控制算法分别针对两种不同的函数依赖——基于 FD 的算法解决对属性进行安全分级的推理通道问题,而基于 MVD 的算法解决对元组进行安全分级的推理通道问题.在同一数据库既按属性划分安全级,又按照元组划分安全级时,应考虑按元素来划分安全级的推理通道问题;(2) Tzong[4]提出的 FD 算法面向数据库设计阶段,与数据库实例无关,不利于数据的可用性;(3) 算法一开始就调整安全级,没有消除主键依赖,易出现给主键分配最高的安全级,还可能导致给其他属性分配过高的安全级;(4) 提出的最小信息损失评价方法是失准的,算法中调整安全级的信息损失与数据库实例无关.Rizvi[1]能够实现更细粒度的访问控制,利用参数化视图,粒度不仅可以是元组、属性,而且还可以是某些特定元素,并给出了有效查询的推导规则,但没有考虑推理控制问题.特别是在单个查询有效的情况下,利用历史查询信息或者多个用户有效查询的相互交换,通过推理通道导致敏感信息的泄露,危及数据库的安全.本文针对当前这种细粒度访问控制趋势,针对常规的推理通道,从按元素划分安全级、视图的角度对推理控制进行了研究.

1.2 本文贡献

本文针对以上推理控制工作出现的问题,尤其是 Tzong[4]推理控制所出现的问题以及 Rizvi[1]实现更细粒度访问控制后出现的推理控制问题,以最细粒度——元素级的访问控制为例,对最具代表性的函数依赖(FD)和多值依赖(MVD)所产生的推理通道进行了研究,贡献如下:(1) 给出了基于元素安全级调整的推理控制算法,算法解决了 FD 与 MVD 所引起的推理控制问题.因算法针对关系实例,不会出现在数据库设计阶段进行推理控制导致属性安全级分配过高的问题,且数据的可用性大大提高.算法一开始就对 FD 集规范化,效率得到很大提高.算法在对元素安全级进行调整时,由于信息损失的计算基于数据库实例,因而更加精确且符合实际;(2) 提出了安全和不安全视图依赖基的概念,这是定义有效查询规则的基础;(3) 给出了安全视图依赖基的划分算法,算法的时间复杂度只与关系模式 R 中的 FD 数和属性数有关.建立在物化后安全视图基上的查询,能够有效防范多个非法用户的视图合谋.以上推理控制方法稍加修改就可用于其他各种带安全参数数据(给数据分配相应的 label)的推理控制中,文中的安全、不安全视图依赖基可作为 Rizvi[1]中有条件、无条件查询规则的基础.因而,本文的推理控制是 Rizvi[1]在推理控制研究方面的深入.

本文第 2 节对文中所用术语进行定义并对相关理论进行阐述.第 3 节研究基于 FD 和 MVD 的推理控制算法及其正确性和复杂性.第 4 节描述基于视图的推理控制方法,给出视图依赖基的划分原理,解决多视图合谋的推理控制问题.第 5 节是本文的结论及今后研究方向.

2 术语的定义及相关理论

用户对信息的访问控制是通过赋予用户和被访问信息一定权限来实现的,用户只能访问其访问权限内的

各种信息,把分配给用户和信息的各种权限称为安全级.

定义 **2.1**. 安全级.设 SL 是一组安全级集合,如未分级(U)、已分级(C)、密级(S)、最高密级(TS)等.安全级为三元组$\langle O,S,\lambda\rangle$形式:$O$(*object*)是指被访问客体集合$\{O_1,O_2,\ldots,O_n\}$;$S$(*subject*)是指访问客体的主体集合$\{S_1,S_2,\ldots,S_n\}$;$\lambda$:$O\cup S\to SL$ 表示安全级映射.

为保护敏感数据,MAC 必须保证主体的安全级大于或等于客体的安全级,即λ(*subjects*)$\geq\lambda$(*object*).这里的客体可以是表、元组、属性、元素或 XML 数据[12]等,主体可以是访问客体的人或程序.

定义 **2.2**. 多级关系模式.多级关系模式用 $R(A_1,C_1,\ldots,A_n,C_n,TC)$表示.其中:$A_i$ 表示实体属性;C_i 是与属性 A_i 相关联的安全级;C_i 定义在$[L_i,H_i]$上,$H_i\geq L_i$;TC 为 $\cup_{i=1}^{n}(L_i,H)$,表示元组的安全级;H 为最高安全级;A_1 表示主键,这里的主键与关系数据库中的主键不同,允许键值重复存储,并假设已解决多实例问题.

定义 **2.3**. 多级关系实例.多级关系实例用 $r(A_1,C_1,\ldots,A_n,C_n,TC)$表示,是不同元组$(a_1,c_1,\ldots,a_n,c_n,tc)$的集合.其中:$a_i\in D_i$($D_i$ 为属性域),$c_i\in[L_i,H_i]$;或 $a_i=null,c_i\in[L_i,H_i]\cup null,tc\geq lub\{c_i|c_i\neq null:i=1\ldots n\}$(*lub* 表示最小上界).

函数依赖从语义上确切地表示了关系模式属性之间的对应关系,是泄露敏感信息的重要推理通道.但是,用户仅仅知道函数依赖 FD:$X\to Y$ 而不知道属性 X 和属性 Y 之间值的映射关系是不足以造成敏感信息泄露的,这里假设用户知道 FD:$X\to Y$ 的同时,也知道其属性值之间的相关映射.

数据库按元素划分安全级后,原来数据库模式中的约束发生了一些变化,为保持一致性[7,13],须对数据库的完整性定义作如下修改.

定义 **2.4**. 键完整性约束:关系模式 $R[X,K]$上的关系实例 r,X 是 R 上属性集合,对属性 $Y\subseteq X$,$t[Y]$表示元组 t 在属性 Y 上的取值,关系实例 r 满足键完整性约束,如果:1) 对每个元组 $t\in r,A\in K,t[A]\neq null;A_K,TC\to C_K;A_K,C_K,C_i\to A_i$;2) $A_i,A_j\in K\Rightarrow C_i=C_j$;3) $A_i\in K\Rightarrow C_i\geq C_j,A_j\subseteq X-K$.

定义 **2.5**. 外键完整性约束:数据库上的(R,C),$R=\{R_i[X_i,K_i]\}$,$1\leq i\leq n$,是一组关系模式集合,$\{r_i\}1\leq i\leq n$ 为对应的关系实例,即 r_i 是 $R_i[X_i,K_i]$上的关系实例,对外键依赖 $R_i[Y]\to R_j,Y\subseteq X$,满足外键完整性约束,如果:1) $t[Y]=null$ 或 $t[Y]\neq null$;2) 如果 $t[Y]\neq null$,则存在元组 $t'\in r_j$,且 $t[Y]=t'[K_j]$;3) $\lambda(Y)\geq\lambda(K_j)$.

造成敏感信息泄露的推理通道可形式化定义如下:

定义 **2.6**. 推理通道(inference channel): $InfCh(H)=\{X\in U_\alpha|\lambda(X)\langle\lambda(H)\wedge INFER(X\to H)\rangle\varepsilon\vee\exists Z\in U_\alpha[\lambda(Z)\langle\lambda(H)\wedge INFER((Z\cup\{X\})\to H)\rangle(INFER(Z\to H)+\varepsilon)]\}$.其中:$U_\alpha$表示数据集合;$INFER(X\to H)>\varepsilon$为推理函数.推理通道包含两部分内容:“或”的前半部表示由低安全级的 X 可以推出较高安全级的 H;“或”的后半部表示低安全级的 Z 自身并不能推出 H,但与 X 结合后,能推出较高安全级的 H.

并不是所有的函数依赖都会导致敏感信息泄露,只有那些不安全的函数依赖才是我们需要控制的对象.

定义 **2.7**. 安全的函数依赖推理:对于函数依赖 $X\to Y$,如果属性 X 的安全级不低于 Y 的安全级,即$\lambda(X)\geq\lambda(Y)$,能访问属性 X 的主体一定能访问属性 Y,称 $X\to Y$ 是安全的函数依赖推理.对于左部与右部有多个属性的函数依赖 $B_1,B_2,\ldots,B_n\in X,A_1,A_2,\ldots,A_m\in Y$,则要求 $lub(\lambda(B_1),\lambda(B_2),\ldots,\lambda(B_n))\geq lub(\lambda(A_1),\lambda(A_2),\ldots,\lambda(A_m))$.

对 FD,有一个正确和完备的推理规则集,于 1974 年由 W.W.Armstrong 提出.主要包括自反性、增广性、传递性.一般地,由函数依赖集 F 经推理规则推得的全部 FD(F^+)是原 FD 数的指数倍.如关系模式 $R(ABC)$,$F=\{A\to B,B\to C\}$,根据 FD 推理规则可以得到 43 个 FD.

原来安全的函数依赖经过推理后是否会产生新的不安全的函数依赖呢?

定理 **2.1**. 对函数依赖集 F 中的任意函数依赖及其相关联的映射,如果每个 FD:$X\to Y$ 是安全的,即$lub(\lambda(X_1),\lambda(X_2),\ldots,\lambda(X_i))\geq lub(\lambda(Y_1),\lambda(Y_2),\ldots,\lambda(Y_j))$,$X_i\in X,Y_j\in Y$,那么 F^+集中的每个 FD 是安全的.

证明:由于 F^+中所有的 FD 都是由 F 中的 FD 通过自反性、增广性、传递性推理得来,如果这些推理规则安全,那么 F^+中所有 FD 也安全.为了表示方便,对复合属性 $X:X_1X_2\ldots X_i$,用 $lub(\lambda(X)$来表示 $lub(\lambda(X_1),\lambda(X_2),\ldots,\lambda(X_i))$.

1) 对 $\dfrac{}{X\to Y}Y\subseteq X\subseteq U$, 因为 Y 是 X 的子集,所以有 $lub(\lambda(X))\geq lub(\lambda(Y))$,即自反性规则是安全的;

2) 对 $\frac{X \to Y, Y \to Z}{X \to Z}$,由已知:$X \to Y$ 和 $Y \to Z$ 是安全的,故 $lub(\lambda(X)) \geq lub(\lambda(Y))$,$lub(\lambda(Y)) \geq lub(\lambda(Z))$,由此可得 $lub(\lambda(X)) \geq lub(\lambda(Z))$,即传递性规则是安全的;

3) 对 $\frac{X \to Y}{XZ \to YZ} Z \subseteq U$,由已知:$X \to Y$ 是安全的,故 $lub(\lambda(X)) \geq lub(\lambda(Y))$,而 $lub(\lambda(Z)) \geq lub(\lambda(Z))$,所以有 $lub(\lambda(XZ)) \geq lub(\lambda(YZ))$,即增广性规则是安全的.

因此,安全的函数依赖集 F 推理产生的 F^+集是安全的. □

我们不必考虑 F^+集所有的函数依赖,因为其中有些是平凡的,有些由其他函数依赖推理得来,我们最终只需考虑规范的函数依赖集合,去掉无关属性后的规范集 F_C 和 F^+等价.Ullman JD 给出了求视图上函数依赖集 F 的规范覆盖 F_C 算法.

3 推理控制算法

3.1 FD推理控制

Tzong[4]基于 FD 的属性安全级调整算法,与关系实例无关,很大程度上限制了数据的可用性.本文的推理控制算法只对元素的安全级进行调整,粒度更细,面向数据,信息损失少,极大地提高了数据的可用性.

3.1.1 算法说明

为减少信息损失,算法在调整过程中,事先按照各元素值的安全级赋予各元素一定的权重.Tzong[4]已经证明,防止信息泄露,同时使得信息损失最小,实际上是一个 NP-complete 问题.所以,在元素安全级的调整过程中是局部最优的.实际应用中,FD 集规范化后,函数依赖左部所含的属性并不多,且元素的安全级按一定规律分配,如:通常给某些属性上的元素分配较高的安全级.

针对 FD 导致敏感信息泄露的情形,本文的元素安全级调整算法和 Tzong[4]提出的属性安全级调整算法有以下几方面的不同:1) 基于不同的访问粒度,一个按属性划分安全级,一个则按元素划分安全级;2) 属性安全级调整算法是在设计阶段进行的,与关系实例无关,而元素安全级调整算法与关系实例密切相关;3) 属性安全级调整算法中的权重是人为赋值给每个属性的,因而不能很好地反映调整过程中的信息损失,而元素安全级调整算法中的权重由需要调整的元素安全级决定,正确地反映了调整过程中的信息损失.

3.1.2 FD 推理控制算法

算法 **3.1**. FD-ADJUST.

Input: FD set; relational instances classified over element.

Output: The relation instance in which there is no FD-Compromise and the adjustment minimizes the information loss.

1) Normalize the FD set into F_C;
2) Right Decomposition: For each FD in F_C with a composite right-hand side, i.e., the right-hand side is not a single attribute, decompose it as follows: Let $X \to A_1A_2 \ldots A_n \in F_C$, then decompose it into $\{X \to A_1, X \to A_2, \ldots, X \to A_n\}$; Let the result of the right decomposition of all FD's be F_r;
3) for $i=L_l$ to L_h / L_l and L_h represent the lowest and highest clearance of user.
4) for each FD: $B_{i_1}B_{i_2}\ldots B_{i_m} \to A$ in F_r
5) for each set :$T_u = \pi_{B_{i_1},\ldots,B_{i_m},A_i}(\sigma_{count(*)>1, B_{i_1}=b_{i_1},\ldots,B_{i_m}=b_{i_m}, A_i=a_i}(r))$

if $\exists t \in T_u$, $lub(t(L_{B_{i_1}=b_{i_1}}),\ldots,t(L_{B_{i_m}=b_{i_m}}),t(L_{A_i=a_i})) \leq i$

AND $\exists t' \in T_u$, $lub(t'(L_{B_{i_1}=b_{i_1}}),\ldots,t'(L_{B_{i_m}=b_{i_m}})) \leq i$, $L_{A_i=a_i} > i$

then Begin

for each $t' \in T_u$

$W_{\max} = W_{B_{i_j}=b_{i_j}} = \max\{t'(L_{B_{i_j}=b_{i_1}}),\ldots,t'(L_{B_{i_m}=b_{i_m}})\}$;

/(j=1,…,m), find the element which has the max weight.
$t'(L_{B_{i_j}=b_{i_j}})$ =i+1; /adjust the element's classification level.

End

6) Terminate.

算法分析:1) 正确性.算法的第3行表示调整过程是根据用户的访问级别由低到高,将元素调整为具有更高的安全级,不会出现为了防止高安全级的用户推理访问敏感信息时,因元素安全级的调整而导致新的敏感信息泄露给低安全级的用户的现象;2) 复杂度.算法的复杂度主要体现在第5行的for循环,最坏情况下,循环次数是$n^{|UIN|}$,$|UIN|$表示关系实例中所含属性数.假定实际应用中函数依赖不是很多,故总时间复杂度为 $O(|L||F_r|n^{|UIN|})$,$|L|$表示分配给元素的安全级数,$|F_r|$为F_C右分解后所包含的函数依赖数.当关系实例中的元组数较多时,算法是不切实际的,可假定算法是在元组数不太多的情况下进行.以后向关系实例中添加元组时,由触发器触发FD-ADJUST算法,此时由于算法中内循环次数大为减少,因而总的时间复杂度降为$O(|L||F_r|n)$,算法现实可行.

3.2 MVD推理控制

3.2.1 算法说明

与基于FD的推理控制算法相同,在按元素划分安全级后,不需要像Tzong[4]的MVD-ADJUST算法那样将一条记录中所有元素的安全级都进行调整,只需调整其中部分元素的安全级.针对由于MVD导致的敏感信息泄露,我们提出的元素安全级调整算法和Tzong[4]的元组安全级调整算法有以下几方面的不同:1) 基于不同的访问粒度,一个是按元组划分安全级,一个是按元素划分安全级——一个是调整元组的安全级,一个是调整元素的安全级;2) 调整算法中的权重计算方法有明显区别,一个是按元组数来计算权重,一个则是按有相同元素值的元素计算权重.我们的算法调整粒度更细,从更大程度上保证了信息的可用性.

3.2.2 MVD推理控制算法

算法 **3.2**. MVD-ADJUST.

Input: The attributes in relation scheme: $UIN=\{A_1,A_2,\ldots,A_m\}$;
The MVD set in relation scheme;
The relation instance r in which the classification level is assigned over element.

Output: The relation instance r in which there is no MVD-compromise and the adjustment minimizes the information loss.

1) for $A_i=A_1$ to A_m
2) for k=1 to n
3) for $l=L_1$+1 to L_h
$w[A_i][a_k][l]=w[A_i][a_k][l]+l-L(A_i=a_k)$; /initialize the weight
4) find the MVD set's join dependency:$\bowtie(S_1,S_2,\ldots,S_k)$;
5) for $l=L_1$ to L_h
$R^{\vdash}=\{t|t\in r,L(A_i=a_k)\le l\}$; /$R^{\vdash}$ represents the tuples users can access
$R^{\dashv}=\bowtie(\pi_{s_1}(R^{\vdash}),\pi_{s_2}(R^{\vdash}),\ldots,\pi_{s_k}(R^{\vdash}))-R^{\vdash}$; /the tuples users can infer
6) repeat
$w[A_i][a_k][l+1]=\min_{L(A_{i'}=a_{k'})\le l;a_{k'}\in t'[A]}\left\{\sum_{l'\le l}l+1-w[A_{i'}][a_{k'}][l']\right\},t'\in R^{\dashv}$;
adjust all $(a_k,*)$ to $(a_k,l+1)$; /"*" represents the level lower than l+1
$R^{\dashv}=R^{\dashv}-\sigma_{A_i=a_k}(R^{\dashv})$;
until $R^{\dashv}=\varnothing$
7) Terminate.

算法分析:1) 正确性.算法将能推理得到的具有较高安全级元组中的元素安全级进行调整,使该元素具有更高的安全级,算法按照用户的访问权限由低到高进行调整,不会导致由于后面的调整而出现新的敏感信息泄

露,因而调整后所得到的关系实例是安全的,证明算法是正确的;2) 复杂度.算法的复杂度主要体现在第 5 行开始的两个嵌套循环上:第 1 个嵌套循环与关系模式中的属性数$|UIN|$、关系实例中的元组数 n 以及分配给用户的访问权限$|L|$有关,所以该嵌套循环的复杂度为 $O(|UIN||L|n)$;第 2 个嵌套循环与访问权限数$|L|$以及用户所能推理的元组集合中不同元素值相关,在最坏情况下,不同元素值的个数为整个关系实例的元素数,这样,该嵌套循环的复杂度为 $O(|UIN||L|n)$.综合以上嵌套循环的复杂度,最终整个算法的时间复杂度为 $O(|UIN||L|n)$.

4 基于视图的推理控制

FD 和 MVD 推理控制算法是在整个数据库实例上操作的,当 r 中元组数 n 非常大时,效率不高,而且多个用户可能相互交换查询信息、利用历史查询信息或用户的先验知识来推理敏感信息,为提高查询效率,解决多用户合谋引起的敏感信息泄露,在此引入视图,必要时可以将视图物化来提高查询效率.

4.1 相关定义

定义 **4.1**. 先验安全知识:设 P 是元组的概率分布,称在先验知识情形下是安全的,如果查询 S 和视图 V': $P[S(I)=S|K(I)]=P[S(I)=S|V'(I)=V'\wedge K(I)]$,这里,$K:S|_PV'$表示先验安全知识.

定义 **4.2**. 多视图合谋:将 n 个不同的视图 $V_1,V_2,\ldots,V_n$ 发布给 n 个不同的主体后,这其中的多个主体相互交换视图信息,从而达到访问敏感信息的目的.

定义 **4.3**. 安全的多视图合谋:对数据库模式 R 的任意实例 r,给定视图集合 $V'=V_1,V_2,\ldots,V_n$ 和另一视图 V'',在交换视图 $V_1,V_2,\ldots,V_n$ 信息后,得到 $V''=f(V')$.这里,f 为视图信息的计算函数,主要是指相互交换查询信息.若 V'' 仍为安全的视图,则称 $V_1,V_2,\ldots,V_n$ 为安全的多视图合谋;反之,则称为不安全的多视图合谋.

视图 V'' 的信息都可以从 V'得到,V'' 的信息内容不会超过 V',如果敏感信息对 V'安全,则对 V'' 也是安全的.

4.2 多视图合谋的划分

引理 **4.1**. 假设条件 1:对形如 V_{KXY} 的视图(其中 K 为键属性,且在视图中存在 FD:$X\rightarrow Y$),若视图中每个元组 t 在属性 X 上的取值唯一.结论:主体在此视图上所定义的多个子视图 $V_1,V_2,\ldots,V_n$ 为安全的多视图合谋.

证明:主体要想利用 FD:$X\rightarrow Y$ 作为推理通道来推得敏感信息,必须是拥有视图的多个主体在交换视图信息后,得到形如$(a_1\ u,b_1\ u)$,$(a_1\ u,x\ s)$的两个元组,这两个元组在属性 X 上的取值都为 a_1.由 FD:$X\rightarrow Y$ 可推得,处在高于主体安全级 u 的元素 x,在安全级为 s 上元素的值一定为 b_1,从而让低于安全级 s 的主体推得了对他而言为敏感信息的 x.但这不可能,因为即使在多个主体交换各子视图信息之后,所得视图中的元组也不会超过原视图中元组的信息.即通过计算后,新视图中的元组在属性 X 上的取值也唯一.因而,就不存在形如$(a_1\ u,b_1\ u)$,$(a_1\ u,x\ s)$的两个元组,所以杜绝了这样的推理通道.定义在假设条件 1 下的子视图 $V_1,V_2,\ldots,V_n$ 为安全的多视图合谋. □

引理 **4.2**. 假设条件 2:对形如 V_{KXY} 的视图(其中 K 为键属性,且在视图中存在 FD:$X\rightarrow Y$),若视图中元组 t 在属性 X 上的取值不唯一,但所有这些在属性 X 上取值相同的元组在与之相对应的属性 Y 上,元素的安全级相同.结论:主体在此视图上所定义的子视图 $V_1,V_2,\ldots,V_n$ 为安全的多视图合谋.

证明:主体要想利用 FD:$X\rightarrow Y$ 作为推理通道来推得敏感信息,必须是拥有视图的多个主体在交换视图信息后,得到形如$(a_1\ u,b_1\ u)$,$(a_1\ u,x\ s)$的两个元组.但这不可能.由引理的假设,在与之相对应的属性 Y 上,元素的安全级相同.那么,在定义各子视图的主体在交换视图信息后,只能得到形如$(a_1\ u,b_1\ u)$,$(a_1\ u,x\ u)$或者形如$(a_1\ u,b_1\ s)$,$(a_1\ u,x\ s)$的两个元组,而不存在形如$(a_1\ u,b_1\ u)$,$(a_1\ u,x\ s)$的两个元组.而这样的两个元组对某一安全级的主体来说,在属性 Y 上元素的取值要么全部可见,要么全部不可见.这样,杜绝了因 FD:$X\rightarrow Y$ 产生的推理通道.所以,子视图 $V_1,V_2,\ldots,V_n$ 为安全的多视图合谋. □

引理 **4.3**. 假设条件 3:对形如 V_{KXY} 的视图(其中 K 为键属性,且在视图中存在 FD:$X\rightarrow Y$),若视图中元组 t 在属性 X 上的取值不唯一,但所有这些在属性 X 上取值相同的元组在 X 和与之相对应的属性 Y 上,元素的安全级是相同的.结论:主体在此视图上所定义的子视图 $V_1,V_2,\ldots,V_n$ 为安全的多视图合谋.

证明:根据假设条件 3,视图 V_{KXY} 中的元组在属性 X 和属性 Y 上的表现形式为$(a_1\ c,b_1\ c)$,这里的 c 为安全级,那么对任意安全级的主体,在属性 X 和 Y 上的取值要么全部可见,要么全部不可见,这样就失去了推理通道存在

的条件,因而在此基础上定义的多个子视图 $V_1,V_2,\ldots,V_n$ 为安全的多视图合谋. □

引理 **4.4**. 主体定义在假设条件 1~3 视图之上的多个子视图 $V_1,V_2,\ldots,V_n$ 是安全多视图合谋.

证明:因为组成子视图的 $V_1,V_2,\ldots,V_n$ 既有来自假设条件 1 下主体所定义的子视图,也有来自假设条件 2、假设条件 3 下主体所定义的子视图,引理 4.1~引理 4.3 已经证明,那些完全在假设条件 1、假设条件 2 或假设条件 3 上主体所定义的子视图是安全的多视图合谋.那么,主体是不能通过仅交换假设条件 1 或仅交换假设条件 2 或仅交换假设条件 3 下所定义的子视图信息后,由产生的新视图来推得敏感信息的;只能是在交换不同假设条件下所定义的多个子视图后得到新视图,利用新视图来推得敏感信息.即利用推理通道 FD:$X\rightarrow Y,Y\rightarrow Z\vdash X\rightarrow Z$,设 $X\rightarrow Y$ 是假设条件 1 上的 FD,$Y\rightarrow Z$ 是假设条件 2 上的 FD.我们先看交换假设条件 1 和假设条件 2 下定义的视图之后的情形,用元组来表示:先有元组$(x_1\ u,y_1\ u)$(来自假设条件 1 的视图),然后有元组$(y_1\ u,z_1\ u)$,$(y_1\ u,z_2\ s)$(来自假设条件 2 的视图),计算后新视图中有元组$(x_1\ u,z_1\ u)$,$(x_1\ u,z_2\ s)$.以此推得处于安全级为 S 上的敏感信息 z_2.但这显然不可能.因为根本就不可能存在$(y_1\ u,z_1\ u)$,$(y_1\ u,z_2\ s)$的元组.采用类似的方法可以证明,其他两种情况下视图的合谋也是安全的.所以,主体在此之上定义的多个子视图 $V_1,V_2,\ldots,V_n$ 是安全的多视图合谋. □

定义 **4.4**. 安全的视图依赖基:把形如在假设条件 1、假设条件 2 和假设条件 3 下分别定义的视图称为安全的视图依赖基;否则,称为不安全的视图依赖基.

定理 **4.1**. 完全在安全视图依赖基上定义的多个子视图 $V_1,V_2,\ldots,V_n$ 为安全的多视图合谋;而在安全视图依赖基和不安全视图依赖基上定义的多个子视图 $V_1,V_2,\ldots,V_n$ 为不安全的多视图合谋.

4.3 视图依赖基等价类划分算法

由此,我们可以对视图进行等价类划分:安全的视图依赖基和不安全的视图依赖基.以后处理查询时,在安全的视图依赖基之上的子视图或查询,不必考虑推理控制的问题,而只考虑不安全视图依赖基之上的子视图或查询的推理控制问题.从而,使原先在处理历史查询与合谋所遇到的问题得到了很好的解决,查询日志记录可以大幅减少,有效地提高了查询速度,且可使信息得到最大限度的利用,能阻止任意访问控制粒度的推理通道.这里的关键就是对任意关系模式 R 的关系实例 r,找到与之相应的安全视图依赖基和不安全视图依赖基.

算法 4.1 针对 FD 集中的每个函数依赖将关系实例分解,然后对分解后的关系实例划分,得到安全的视图依赖和不安全的视图依赖.最后,对划分后的视图物化,再合并,得到安全的视图依赖基和不安全的视图依赖基.

算法 **4.1**. 划分视图依赖基等价类.

Input: Relational scheme R, relational instances and FD set in r.

Output: Materialized view dependency basis that are equivalent.

Begin

Security-basis:={};

Insecurity-basis:={};

Normalize the FD set into F_C;

Delete primary dependency;

For each FD:$X\rightarrow Y$ in F_C

For each Y_i in Y

Create view V_{KXY_iC1} as /define the secure dependency basis according to the first condition

Select K, X, Y_i From R Where the value of $t[X]$ is unique;

Create view V_{KXY_iC2} as /define the secure dependency basis according to the second condition

Select K, X, Y_i From R Where $t_1[X]=t_2[X]$ and $\lambda(t_1[Y])=\lambda(t_2[Y])$;

Create view V_{KXY_iC3} as

Select K, X, Y_i From R Where $t_1[X]=t_2[X]$ and $\lambda(t_1[X])=\lambda(t_1[Y])$;

Create view V_{KXY_iUC} as /define the unsecure dependency basis

Select K, X, Y_i From R Where tuples are not satisfy the preceding three conditions;

Materialize $V_{KXY_iC1},V_{KXY_iC2},V_{KXY_iC3},V_{KXY_iUC}$;

Security-basis:=*security-basis*∪{ V_{KXY_iC1}, V_{KXY_iC2}, V_{KXY_iC3} };

Insecurity-basis:=*security-basis*∪{ V_{KXY_iUC} };

Return *Security-basis*, *Insecurity-basis*;

End.

4.4 算法分析

算法由两个循环组成:对于第 1 个循环,循环次数为 FD 集中的 FD 数.在对 FD 集规范化后,FD 集中的 FD 数大为减少.在实际数据库中,由为数不多的几个函数依赖组成(这里假设 F_C 集中函数依赖数为$|F_C|$);对于第 2 个循环,循环次数为组合属性中属性的个数.假设总的属性数为$|U|$,那么视图依赖基中总的视图数为 $4|F_C||U|$.其中,安全的视图依赖基数为 $3|F_C||U|$;而不安全的视图依赖基数为$|F_C||U|$.这与访问控制中为每个访问用户建立一个视图相比,数量大幅减少.在找到这样的安全视图依赖基之后,就可以利用文献[1]中的查询推理规则,有效地实现最细粒度信息的访问,防止因函数依赖所引起的推理问题.对于查询,只在涉及不安全视图时才做必要的修改查询或拒绝查询,这使得防范多用户合谋和单用户利用历史查询访问敏感信息所带来的安全问题易于控制.

4.5 举 例

例 4.1:设下面的关系模式 $T(A,B,C,D)$中有函数依赖($A\rightarrow B$,$C\rightarrow D$),属性 A 为主键,现在我们来看 T 的关系实例 r(与属性值相对应的 U 和 S 表示该值是非敏感信息和敏感信息)(如图 1 所示).

Attribute / Record	A		B		C		D	
1	a_1	u	b_1	u	c_1	u	d_1	s
2	a_2	u	b_2	u	c_1	u	d_1	s
3	a_3	u	b_3	c	c_2	u	d_2	u
4	a_4	u	b_2	c	c_2	s	d_2	s
5	a_5	u	b_1	s	c_3	u	d_3	u
6	a_6	u	b_3	s	c_3	u	d_3	s

Fig.1 Relational scheme classified by elements

图 1 按元素划分安全级的关系模式

从图中可以看出,虽然在关系模式中有 $A\rightarrow B$,但关系实例 r 在属性 A 上的取值唯一,符合假设条件 1,属于安全的多视图合谋.因为由引理 1 的结论,蓄意破坏的用户无法利用 $A\rightarrow B$ 进行推理得到 B 属性上为敏感信息的元素值;第 2 种情形,对 $C\rightarrow D$,关系实例 r 中在属性 C 上取值为 c_1 的只有记录 1 和记录 2,而在相对应的属性 D 上,元素的安全级均为 S,符合假设条件 2,因此是安全的多视图合谋;第 3 种情形,关系实例 r 中在属性 C 上取值为 c_2 的有记录 3 和记录 4,从中可以看出它符合假设条件 3,对蓄意破坏的用户而言,记录 3 和记录 4 在属性 C 和 D 上的取值要么全部可见,要么记录 4 上与之相对应的值全部不可见,因而不能利用 $C\rightarrow D$ 来推得敏感信息,属于安全的多视图合谋;最后一种情形,关系实例 r 在属性 C 上取值为 c_3 的元组只有记录 5 和记录 6,它不符合 3 个假设条件之中的任意一个,因而是不安全的多视图合谋.在查询时,应对记录 6 中属性 C 上的取值进行调整或者对记录 5 中属性 D 上的取值进行调整,可以按照信息受损最小的原则任选其一.

5 结束语

本文所提出的安全推理控制算法针对按元素划分安全级的数据库,与 Tzong[4]中分别建立在按属性或元组划分安全级的算法不同,这些算法更有效地保证了数据的可用性,可防范由于 FD 和 MVD 所引起的推理控制问题,扩展性好,修改后仍可用于其他各种粒度数据(如按 label 进行划分安全级)的推理控制,弥补了 Rizvi[1]在推理控制方面的不足,同时克服了各种安全级调整算法的弊端.为进一步防范由于交换查询信息或利用历史信息所引起的敏感信息泄露,引入了基于视图的推理控制方法,给出了安全视图依赖基的概念,建立在安全视图依赖基上的查询能够有效防止多视图合谋和利用历史查询信息所带来的安全问题.本文的研究工作是基于两种最典

型的推理通道——函数依赖和多值函数依赖进行的,在实际应用中,可能还存在多种其他推理通道.今后我们将从审计的角度来考虑推理控制的问题.

References:

[1] Rizvi S, Mendelzon A, Sudarshan S, Roy P. Extending query rewriting techniques for fine-grained access control. In: ACM SIGMOD Conf. Paris, 2004. 551−562.

[2] Bell DE, LaPadula LJ. Secure computer systems: Unified exposition and multics interpretation. Technical Report, ESD-TR-75-306, Bedford: MITRE Corp., 1976.

[3] Jajodia S, Meadows C. Inference problems in multilevel secure database management systems. In: Abrams M, Jajodia S, Podell H, eds. Information Security: An Integrated Collection of Essays. Los Alamitos: IEEE Computer Society Press, 1995. 570−584.

[4] Tzong, Ozsoyoglu G. Controlling FD and MVD inferences in multilevel relational database systems. IEEE Trans. on Knowledge and Data Engineering, 1991,3(4):474−485.

[5] Stickel M. Elimination of inference channels by optimal upgrading. In: Proc. of the 1994 IEEE Symp. on Research in Security and Privacy. Oakland, 1994. 168−174.

[6] Hinke T. Inference aggregation detection in database management systems. In: Proc. of the IEEE Symp. on Security and Privacy. 1998. 96−106.

[7] Lunt TF, Denning DE, Schell RR, Heckman M, Shockley WR. The seaview security model. IEEE Trans. on Software Engineering, 1990,16(6): 593−607.

[8] Morgenstern M. Controlling logical inference in multilevel database systems. In: Proc. of the IEEE Symp. on Security and Privacy. 1998. 245−255.

[9] Stachour P, Thuraisingham B. Design of LDV: A multilevel secure relational database management system. IEEE Trans. on Knowledge and Data Engineering, 1990,2(2):190−209.

[10] Kenthapadi K, Mishra N, Nissim K. Simulatable auditing. In: ACM Symp. on Principles of Database Systems. 2005. 118−127.

[11] Brodsky A, Farkas C, Jajodia S, Member S. Secure databases: Constraints, inference channels, and monitoring disclosures. IEEE Trans. on Knowledge and Data Engineering, 2000,12(6):900−919.

[12] Bertino E, Castano S, Ferrari E, Mesiti M. Protection and administration of XML data sources. Data & Engineering, 2002,43(3): 237−260.

[13] Chen F, Sandhu RS. The multilevel relational (MLR) data model. In: Proc. of the IEEE Symp. on Research in Security and Privacy. 1995. 128−142.

严和平(1970－),男,湖南南县人,博士生,工程师,主要研究领域为数据库技术,XML,数据挖掘.

施伯乐(1936－),男,教授,博士生导师,CCF 高级会员,主要研究领域为数据库与知识库,数据挖掘,数字图书馆.

汪卫(1970－),男,博士,教授,博士生导师,主要研究领域为数据库技术,XML,数据挖掘.