基于数字水印的数据库角色访问控制模型

郑吉平,秦小麟,崔新春
(南京航空航天大学信息科学与技术学院,江苏南京 210016)

摘 要: 角色授权是角色访问控制模型中一个重要部分,已有形式化方法大都集中在显式的描述上. 本文在研究已有的 RBAC 模型基础上,结合关系数据库水印技术信息隐藏的特点,提出一种基于数字水印的数据库角色访问控制模型及其实现框架,重点研究了与角色授权相关的水印算法,并给出模型在入侵容忍安全数据库模型中的实现.

关键词: 角色访问控制; 数字水印; 安全数据库模型; 入侵容忍
中图分类号: TP392 **文献标识码**: A **文章编号**: 0372-2112 (2006) 10-1906-05

Digital Watermark Based Database Model Using RBAC

ZHENG Ji-ping, QIN Xiao-lin, CUI Xin-chun
(*School of Information Science & Technology, Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu* 210016, *China*)

Abstract: Role authorization is very important in RBAC models. Existing formalized methods focus on explicit description. Based on current RBAC models and associated with information hiding technique of watermarking relational databases, a digital watermark based database model using RBAC (WRBAC) and its framework are provided. Then, digital watermark algorithms about role authorization are emphasized. Finally, implementation of WRBAC model in our intrusion tolerance database environment is provided.

Key words: Role-based Access Control (RBAC); digital watermark; security database model; intrusion tolerance

1 引言

访问控制模型是信息安全的一个重要方面. RBAC (Role-based Access Control) 模型[1]的提出可以减少授权管理的复杂性,符合更加复杂的安全策略的应用环境. 数字水印技术[2]是上世纪 90 年代提出,主要解决确定数字产品的所有权和检验数字内容的原始性等安全性问题. 已有的访问控制模型[1,3~6]显式表达系统授权和系统动态配置策略,往往成为非法用户主动攻击的目标. 利用数字水印信息隐藏的特点,将其应用到访问控制中,有效增加了系统的安全性,并且可以防止多个用户同谋获取更高权限的现象.

本文首先研究了 RBAC96 模型,在此基础上,结合数字水印技术,将数据库中的对象操作权限授权给相应角色,并提出基于数字水印的数据库角色授权访问控制模型. 重点讨论了模型中关系数据库数字水印算法和模型如何解决用户同谋现象,并给出模型在课题组入侵容忍数据库安全模型上的实现.

2 RBAC96 访问控制模型

定义 1 RBAC96 模型[1]的组成包括下列几个部分:

U, R, P 以及 S (用户,角色,授权和会话);

$PA \subseteq P \times R$, PA 是授权到角色的多对多关系;

$UA \subseteq U \times R$, UA 是用户到角色的多对多关系;

$roles(S_i) \subseteq \{ r \mid (user(S_i), r) \in UA \}$;

其中:

$User: S \rightarrow U$, 将各个会话映射到一个用户的函数 $user(S_i)$;

$roles: S \rightarrow 2^R$, 将各个会话 S_i 与一个角色集合联接起来的映射,随时间变化而变化,且会话 S_i 的授权为 $U_{r \in roles(S_i)} \{ p \mid (p, r) \in PA \}$.

在 RBAC96 模型中,角色和授权是多对多的关系,即对每个角色设置了多个授权关系,同时一个授权也可以赋予多个角色. RBAC96 规定:每个角色至少具备一个授权,而每个用户至少扮演一个角色.

定义 2 RBAC96 模型中授权的描述:

OP, 一组操作集合;

OBJ, 一组对象集合;

$P = OP \times OBJ$, 为一组权限集合;

S, 一组会话集合;

$operations: R \times OBJ \rightarrow 2^{op}$, 一个角色映射到客体对象的多个操作集合;

$object: P \rightarrow 2^{OBJ}$, 一个权限映射到一组对象集合;

$PA \subseteq P \times R = OP \times OBJ \times R$, 表示权限和角色之间的多

收稿日期:2005-11-11;修回日期:2006-02-22
基金项目:国家自然科学基金(No. 60673127);江苏省高技术研究计划(No. BG2004005);航空科学基金(No. 02F52033)

对多指定关系;

其中:

$roles(p_i)=\{r\in R \mid (p_i,r)\in PA\}$,任意一个权限 p_i 对应一个角色集合;

$permissions(r_i)=\{p\in P \mid (p,r_i)\in PA\}$,一个角色可以赋予多个权限;

$operations(r_i,obj_i)=\{op\in OP \mid op,obj_i,r_i)\in PA\}$,一个角色可以对一个对象进行一组操作.

授权机制通过特定的操作,如读、写、更新和执行等,将角色和用户联结起来.通过授权管理机制,可以给予一个角色多个授权,而一个授权也可以赋予多个角色.相对于将用户和授权之间直接关联的方法,RBAC 授权机制用更加简单的方法向最终用户提供语义更加丰富和完整控制的存取功能.

3　基于数字水印的数据库角色授权访问控制模型

RBAC96 模型在角色和权限管理上采用了角色、对象和操作的授权机制,授权和系统配置策略以显式的方式表达[6].非法用户往往将这些策略作为攻击目标,采用各种各样的非法手段获取不正当的权限或者进行恶意篡改,系统安全性受到严重威胁.采用数字水印技术[2]可以将授权和配置策略隐式和动态地表达出来[11],因而提出基于数字水印的数据库角色授权访问控制模型(WRBAC,Watermark Based Authorization Database Model Using RBAC)[1,9~11].

3.1　模型描述

定义 3　授权角色信息即角色版权(Role Copyright,简称 RC)为一个 6 元组 $RC=(roleID,DBOID,r,w,u,e)$,其中 $roleID$ 为角色标识,$DBOID$ 为数据库对象标识,r,w,u,e 分别表示角色对数据库关系的读、写、更新和执行操作,取值为 0(不具有此权限)和 1(具备此权限).

对于 WRBAC 中的每一个角色信息,包括角色标识、数据库对象标识以及角色对数据库对象的访问权限,定义为角色的版权信息,通过数字水印生成算法将 RC 嵌入到数据库对象中,角色版权信息隐式地嵌在数据库对象中.

WRBAC 中数据库对象与角色之间通过权限管理器(PM,Permission Manager)相互联系,权限管理器负责保存角色的版权信息,当拥有角色的用户需要访问数据库对象时,首先访问 PM,PM 对相应的角色进行检查,检查完毕之后确定角色(集)的数据库访问对象和权限.

定义 4　权限管理器(PM)在 t 时刻拥有状态 s,记为 $pm(s,t)$,且当 $t_1\neq t_2$ 时,$pm(s_1,t_1)$ 和 $pm(s_2,t_2)$ 无任何启发和关联信息.

根据定义 4,PM 在不同的时刻 t_1,t_2 具有不同的状态,即 t_1 时刻具有相同的角色(集)的用户与在 t_2 时刻拥有对数据库对象的访问权限之间没有关联.

3.2　模型框架工作原理

图 1 给出了基于数字水印的数据库角色授权访问控制模型框架,虚线框部分是 WRBAC 模型区别于 RBAC96 模型的核心部分.

定义 5　WRBAC 模型核心组成部分:

USERS,ROLES,SESSIONS,DBOIDS,OPS,WDAS,WGAS(用户,角色,会话,数据库对象,操作,水印检测算法,水印生成算法);

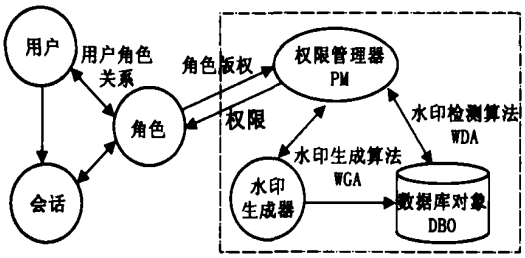


图 1　基于数字水印的数据库角色授权访问控制模型框架

$UA\subseteq USERS\times ROLES$,是用户到角色的多对多关系;

$assigned\_users(r)=\{u\in USERS \mid (u,r)\in UA\}$,角色 r 映射到一组用户;

$PRMS=2^{DBOIDS\times OPS}$;

$PA\subseteq PRMS\times ROLES$,是权限到角色的多对多关系,其中:

$request\_prms(role:ROLES,dboid:DBOIDS,wda:WDAS)\rightarrow 2^{PRMS}$;

$notify\_prms(dboid:DBOIDS,role:ROLES)\rightarrow 2^{PRMS}$;

$update\_prms(role:ROLES,dboid:DBOIDS,wga:WGAS)\rightarrow 2^{PRMS}$.

在 WRBAC 框架中,角色(集)将需要访问的数据库对象递交给 PM,PM 通过水印检测算法(WDA,Watermark Detecting Algorithm)将角色(集)提交的数据库对象的操作权限返回给角色(集).在任何时刻,PM 都可以通过水印生成器(WG,Watermark Generator)对每个数据库对象生成新的水印,在不同的时刻相同角色(集)所获取的权限不存在关联.

定义 5′　带角色层次结构的 WRBAC 模型[3,12]:

在定义 5 的基础上,增加角色层次(RH,Role Hierarchy)定义:

$RH\subseteq ROLES\times ROLES$;

$\succ$ 为 RH 的偏序关系,当 $r_1\succ r_2$ 时:

$(auth\_prms(r_2)\subset auth\_prms(r_1))\wedge(auth\_users(r_1)\subset auth\_users(r_2))$;

$auth\_prms(r)=\{p\in PRMS \mid r'\succ r,(p,r')\in PA\}$.

加入角色层次的 WRBAC 模型是偏序的,这样可以保证具有不同权限的角色之间线性关联,实现了权限层次之间的继承,方便管理.此外,WRBAC 模型可以大大减少用户版权嵌入数据库对象的数量,只须对低层次的角色版权进行嵌入即可完成相关水印操作.

3.2.1　水印生成算法(WGA)

水印生成算法[7,8]就是将角色版权信息 RC 嵌入到相应的数据库对象 DBOID 中.在关系数据库中,数据库对象包括数据库、关系、视图、存储过程等,RC 可以在不同粒度对象上进行嵌入.以数据库中关系为例,设待嵌入水印的数据库对象用 DBOID$(P,A_1,\cdots,A_v,\cdots)$ 表示,DBOID 由 n 个元组 $dboid_1\cdots dboid_n$ 组成,每个元组 $dboid$ 都由 1 个主码 $dboid.p$、v 个数值型属性 $dboid.A_1,\cdots,dboid.A_v$ 和其他属性组成.这样,改变少

量元组中的这些数值型属性中的 ξ 个位来嵌入水印,通过采用关于密钥 K 的 hash 函数来决定那些元组、属性以及属性位真正被用来嵌入水印.

算法中所采用的 hash 函数可以将任意长度的输入信息转化为固定长度的 hash 值,并且 hash 函数是单向的,无冲突的.为了保证数据库对象的使用不被破坏,引入参数 v 表示元组中可用于嵌入水印的属性个数;参数 ξ 表示一个属性中可以被用于水印嵌入的最少位数;1/ γ 表示元组中嵌入水印的比例.则嵌入水印算法 $wga(K, DBOID, \gamma, \upsilon, \xi, L, RC)$ 如下:

(1) 计算待嵌入水印 $\Gamma = H_0(K, RC)$,长度为 L;

(2) foreach $dboid \in DBOID$ do

(3) if $(H_2(K, dboid.P) \bmod \gamma = 0)$ then

(4) 属性索引 $i = H_i(K, dboid.P) \bmod \upsilon$

(5) 属性中位索引 $j = H_2(K, dboid.P) \bmod \xi$

(6) if $H_1(K, dbo.P)$ 为偶数时,嵌入水印位 $mask\_bit = 0$, else $mask\_bit = 1$

(7) 计算要嵌入水印 Γ 中的相应索引 $l = H_1(K, dboid.P) \bmod L$

(8) 获取水印中相应嵌入位的值 $f = f_1$

(9) 将 $dboid.A_i$ 的第 j 位的值设置为 m, $m = x \oplus f_l$,其中 $\oplus$ 为标准的 XOR 操作

(10) 返回数据库对象 $DBOID$.

水印嵌入算法 $wga(K, DBOID, \gamma, \upsilon, \xi, L, RC)$ 中,对象中属性位的定位以及标记位数值的确定是通过采用不同密钥的 hash 函数确定,因此算法具有很好的隐藏效果:a、位置隐藏,在没有多个复本和不知道密钥 K 的情况下,无法得出水印嵌入的具体位置;b、嵌入值隐藏,密钥隐蔽性决定用户无法猜测出水印标记位的数值;c、对象隐藏,在不知道密钥的情况下,水印嵌入到数据库中的具体对象无从得知;d、水印隐藏,秘密 hash 函数保证了根本无法知道计算后的水印.

3.2.2 水印检测算法(WDA)

根据角色(集)提交的数据库对象 DBOID,通过水印检测算法[7,8]得出相应的角色版权 RC.同样以数据库中的关系为例,在水印嵌入算法 $wga(K, DBOID, \gamma, \upsilon, \xi, L, RC)$ 中,嵌入的水印 $\Gamma = H_0(K, RC)$,因此水印检测算法的目标就是从数据库对象 $DBOID$ 中提取出 Γ 的值.设 $\Gamma = (f_0, \cdots, f_{L-1})$ (其中,f_i 取值 0 或 1),对于数据库对象中的每一个元组,依次计算其中相应的水印信息.引入阈值 τ,为检测出数据库对象中的水印所需的最少元组比例.通过投票的方式决定 $\Gamma = (f_0, \cdots, f_{L-1})$ 中每一个 f_i 的值.水印检测算法 $wda(K, DBOID, \gamma, \upsilon, \xi, \tau, L, RC)$ 具体如下:

(1) 要检测出水印表示为 $\Gamma = (f_0, \cdots, f_{L-1}) = (?, \cdots, ?)$, ? 为 0 或 1

(2) foreach $i = 0$ to $L-1$ do

(3) 初始化 f_i 投票值 $count[i][0] = count[i][1] = 0$

(4) foreach $dboid \in DBOID$ do

(5) if $(H_2(K, dboid.P) \bmod \gamma = 0)$ then

(6) 属性索引 $i = H_2(K, dboid.P) \bmod \upsilon$

(7) 属性中位索引 $j = H_2(K, dboid.P) \bmod \xi$

(8) if $dboid.A_i$ 第 j 位不存在,则忽略计算下一个元组 else 设定 $m = dboid.A_i.j$

(9) if $H_1(K, dbo.P)$ 为偶数时,嵌入水印位 $mask\_bit = 0$, else $mask\_bit = 1$

(10) 嵌入水印 Γ 的相应索引 $i = H_1(K, dboid.P) \bmod L$,其值为 $f = m \oplus x$

(11) $count[i][f] = count[i][f] + 1$

(12) foreach $i = 0$ to $L-1$ do

(13) $f_i = 0$ if $count[i][0]/(count[i][0] + count[i][1]) > \tau$;

(14) $f_i = 1$ if $count[i][1]/(count[i][0] + count[i][1]) > \tau$;

(15) 返回 $\Gamma = (f_0, \cdots, f_{L-1})$.

水印检测算法 $wda(K, DBOID, \gamma, \upsilon, \xi, \tau, L, RC)$ 中,$\tau \in (0.5, 1]$.根据投票,如果 f_i 既不等于 0 也不等于 1,则发现无相关的水印.从安全角度,水印可能被篡改.

定义 6 设 f 为启发函数,x_1, x_2 是组件状态,当 $f(x_1, x_2) = 0$ 时,x_1 和 x_2 无启发和关联信息.

定理 1 WRBAC 模型可以解决用户同谋现象.

证明 已知

$user_1$ 和 $user_2$ 是两个同谋用户;

$rc_{mn} = (roleID_{m1}, DBOID_{m1}, r, w, u, e) \cup (roleID_{m2}, DBOID_{m2}, r, w, u, e) \cup \cdots \cup (roleID_{mn}, DBOID_{mn}, r, w, u, e)$;

$rc_1 = rc_{11} \cup uc_{12} \cup \cdots \cup rc_{1m}$;

$rc_2 = rc_{21} \cup rc_{22} \cup \cdots \cup rc_{2n}$;

由 $UA \subseteq USERS \times ROLES$ 得:

$user_1 \rightarrow roles_1$, $user_2 \rightarrow roles_2$;

由定义 4,在 t_1 时刻:

$f(pm(rc_1, t_1), pm(rc_2, t_1)) \neq 0$

在 t_2 时刻:

$f(pm(rc_1, t_1), pm(rc, t_2)) = 0$

$f(pm(rc_2, t_1), pm(rc, t_2)) = 0$

根据定义 6,合谋用户并不能得到相关权限信息,所以 WRBAC 模型可以解决用户同谋现象.

4 模型应用

图 2 表示了本课题组研发的一个入侵容忍数据库原型系统 NHSDB[13].该系统基于多级关系数据库模型 MLR 和以基于水印的角色存取机制作为权限管理核心,采用可选的自主和强制存取控制的多策略安全模型.在 NHSDB 中,管理角色体系负责用户和角色的管理.权限管理器管理 WRBAC 中的权限:将版权 RC 提交给水印生成器 WG,WG 通过相应的水印生成算法将 RC 嵌入到数据库对象中;PM 通过相应水印检测算法检测数据库对象中相应的版权返回给 RBAC 角色体系.主安全管理员(CSSO,Central System Security Officer)负责生成数据库对象水印以及对管理角色体系的二级管理.

NHSDB 数据库安全模型核心函数[5,12,14](格式:函数名(谓词 1 ∧谓词 2 ∧…∧谓词 n)),其中"|→"表示用户、角色和

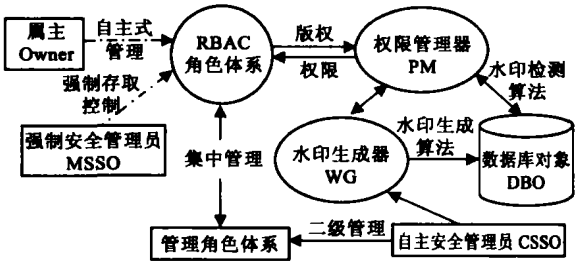


图 2　基于水印的 NHSDB 访问控制模型

权限的一对多关系.

GrantPermission 函数,对角色进行授权. 采用水印生成算法将数据库对象的操作权限授予给相应的角色,其中函数 *(op, dboid, wga)* 表示权限,同时更新 *assigned_perms* 函数.

GrantPermission((dboid, op, role, wga: NAME)

∧((op, dboid, wga) ∈PERMS; roles ∈ROLES)

∧(PA′= PA + ((op, dboid, wga)|→role))

∧(assigned_perms′= assigned_perms - (role|→assigned_perms(roles)) ∪(role|→(assigned_perms(role))) ∪(op, dboid, wga)))

RevokePermission 函数,进行角色权限回收. 采用水印检测算法确定角色权限,并更新角色权限列表:更新 *assign_perms* 函数,其中函数 *(op, dboid, wda)* 表示权限.

RevokePermission((op, dboid, role, wda: NAME)

∧((op, dboid, wda) ∈PERMS; role ∈ROLES; ((op, dboid, wda)|→role) ∈PA)

∧(PA′= PA - ((op, dboid, wda)|→role))

∧(assigned_perms′= assigned_perms - (role|→assigned_perms(role)) ∪(role|→(assigned_perms(role) - (op, dboid, wda))))

CheckAccess 函数,检查当前会话对数据库对象的操作权限.

CheckAccess ((session, op, dboid, wda: NAME; result: BOOLEAN)

∧(session ∈SESSIONS; op ∈OPS; dboid ∈DBOIDS, wda ∈WDAS)

∧(result = (∃r ∈ROLES & r ∈session_roles(session) & (((op, dboid, wda)|→r) ∈PA)))

RolePermissions 函数,返回一个角色被授予的权限集合.

RolePermissions((role: NAME; result: 2^{PERMS})

∧(role ∈ROLES)

∧(result = (op ∈OPS; dboid ∈DBOIDS; wda ∈WDAS | ((op, dboid, wda)|→role) ∈PA)))

RoleOperationsOnDbo 函数,返回给定角色对数据库对象的操作权限集合.

RoleOperationsOnDbo ((role: NAME; dboid: NAME; result: 2^{OPS})

∧(role ∈ROLES; dboid ∈DBOIDS; wda ∈WDAS)

∧(result = (op ∈OPS | (op, dboid, wda)|→role ∈PA)))

UserOperationsOnDbo 函数,返回给定用户对数据库对象的操作权限集合.

UserOperationOnDbo ((user: NAME; dboid: NAME; result: 2^{OPS})

∧(user ∈USERS; dboid ∈DBOIDS; wda ∈WDAS)

∧(result = ((r ∈ROLES; op ∈OPS | (user|→r) ∈UA)

∪(op ∈OPS | (op, dboid, wda)|→r ∈PA))))

5　总结

本文在 RBAC96 模型基础上结合数字水印技术提出一种新的访问控制模型-基于数字水印的数据库角色访问控制模型 WRBAC. WRBAC 模型在对角色进行授权以及实现数据库系统安全策略的过程中,利用数字水印信息隐藏的特点,实现了对用户透明的访问控制. 和已有访问控制模比较,WRBAC 模型具有更高的安全性,并且能够解决多个用户同谋的现象. WRBAC 模型已在课题组 NHSDB 安全数据库模型上实现,文中给出相关实现的核心函数. 目前,关系数据库水印技术实现比较复杂,寻找高效和切实可行的关系数据水印算法是将来研究的方向.

参考文献:

[1] R Sandhu, E coyne, H Feinstein, et al. Role-based access control models[J]. IEEE Computer, 1996, 29(2): 38 - 47.

[2] Fabien A P Petitcolas, Ross J Anderson, Markus G kuhn. Information hiding-a survey[J]. Proceedings of the IEEE, 1999, 87(7): 1062 - 1078.

[3] R Sandhu, David Ferraiolo, Richard Kuhn. The NIST Model for Role-based Access Control: Towards an Unified Standard [J/OL], ACM, 2000, 47 - 63.

[4] GJ Ahn, R Sandhu. Role-based authorization constraints specification[J]. ACM Transactions on Information and System Security, 2000, 3(4): 207 - 226.

[5] D Ferraiolo, R Sandhu, S Gavrila, et al. Proposed NIST standard for role-based access control[J]. ACM Transactions on Information and System Security, 2001, 4(3): 224 - 274.

[6] R Sandhu, P Samarati. Access control: principles and practice [J]. IEEE Communications, 1994, 32(9): 40 - 48.

[7] Rakesh Agrawal, Jerry Kiernan. Watermarking relational databases[A]. Proceeding of the 28th VLDB Conference[C]. Hongkong, 2002.

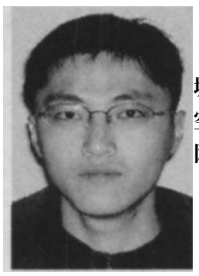
[8] Radu Sion, Mikhail Atallah, Sunil prabhakar. Rights protection for relational data[A]. Proceedings of the ACM SIGMOD International Conference on Management of Data San Diego[C]. San Diego, 2003. 98 - 109.

[9] Elisa Bertino, Sushil Jajodia, Picrangcla Samarati. A flexible authorization mechanism for relational data management systems [J]. ACM Transactions on Information Systems, 1999, 17(2): 101 - 140.

[10] Radu Sion, Mikhail Atallah, Sunil Prabhakar. On watermarking numeric sets[A]. Proceedings of the Workshop on Digital Wa-

termarking[C]. USA ,2002.
[11] Stefan Katzenbeisse ,Fabien A P Petitcolas. Information Hiding Techniques for Steganography and Digital Watermarking[M]. Boston ,London :Artech House ,1999.
[12] Wilfred Ng ,Ho-Lam Lau. Effective approaches for watermarking XML data[A]. DASFAA 2005[C]. LNCS 3453 ,2005.
[13] Sandhu R ,Munawer Q. The ARBAC99 model for administration of roles[A]. The Annual Computer Security Applications Conference[C]. Monterey ,California ,USA ,ACM Press ,1999. 229 - 238.
[14] Fredj Dridi ,Bjorn Muschall , Gunther Pernul. Administration of an RBAC system[A]. Proceedings of the 37th Annual Hawaii International Conference on System Sciences[C]. 2004.

作者简介:



郑吉平 男,1979 年 10 月出生于安徽省宣城市,现居住地为江苏省南京市,目前是南京航空航天大学博士,主要研究方向:数据库安全、网络安全. E-mail :zhengjiping @nuaa. edu. cn

秦小麟 男,1953 年 6 月出生于江苏省南京市,现为南京航空航天大学信息科学与技术学院教授,博士生导师,主要研究方向安全数据库、空间数据库、时空数据库、GIS 等.
E-mail :qinxcs @nuaa. edu. cn