基于概念格的 RBAC 模型中角色最小化问题的理论与算法

张 磊[1,2],张宏莉[1],韩道军[2],沈夏炯[2]
(1.哈尔滨工业大学计算机科学与技术学院,黑龙江哈尔滨 150001; 2.河南大学数据与知识工程研究所,河南开封 475004)

摘 要: 基于概念格的 RBAC 模型是角色挖掘中的一个重要方向,在概念格上找出满足最小权限原则的最小角色集合有助于降低安全管理的复杂性.本文研究了在概念格的 RBAC 模型上的角色最小化问题及其算法.首先将角色最小化问题引入概念格模型,并给出了概念格模型上最小角色集、角色替代和角色约简的定义,和相关定理的证明.在此基础上建立了一个基于角色替代的角色最小化问题求解模型,并设计了一个贪婪算法.该算法以对象概念集为初始集,逐个将集合中的概念用它的父概念来替代和约简,自底向上地迭代求解最小角色集.实验与分析表明了本文相关理论和算法的有效性.

关键词: 形式概念分析; 概念格; 基于角色的访问控制; 最小角色集
中图分类号: TP18 **文献标识码:** A **文章编号:** 0372-2112 (2014)12-2371-08
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2014.12.006

Theory and Algorithm for Roles Minimization Problem in RBAC Based on Concept Lattice

ZHANG Lei[1,2], ZHANG Hong-li[1], HAN Dao-jun[2], SHEN Xia-jiong[2]
(1. *School of Computer Science and Technology, Harbin Institute of Technology, Harbin, Heilongjiang* 150001, *China*;
2. *Institute of Data and Knowledge Engineering, Henan University, Kaifeng, Henan* 475004, *China*)

Abstract: Roles minimization problem and its algorithm based on RBAC model are studied in this paper. Roles minimization problem is introduced into concept lattice model. The minimal set of roles, roles replacement and roles reduction are defined, and the corresponding theorems are proved. Based on this, the model of solving roles minimization problem based on roles replacement is created and a greedy algorithm is proposed. In this algorithm, the object concepts set is regarded as initial set, concetps in roles set are replaced and reducted by their parents one by one, and the minimal set of roles is solved by iteration in bottom-up way. Experiments show that the theory and the proposed algorithm are effective.

Key words: formal concept analysis (FCA); concept lattice; RBAC; minimal set of roles

1 引言

基于角色的访问控制(Role Based Access Control, RBAC)[1]是应用最广泛的访问控制模型之一.RBAC 模型,将权限与角色关联,通过分配和取消角色来完成用户权限的授予和撤销,极大地简化了对系统访问控制进行管理的复杂度.由于角色的设定对用户的权限分配具有关键性的影响,因此寻找适合系统需要的角色及其对应的权限,使其能够精确地反映系统的功能和安全需求,是 RBAC 系统设计中最重要的一个环节.

随着新技术的不断发展和应用范围的不断扩大,信息系统变得越来越复杂,用户和待管理的信息资源也急剧膨胀,类型也更加多样化.同时,系统中的访问控制的主体和客体也往往会随着外部需求的变化而不断发生变化.这导致权限管理和角色的获取变得非常复杂.目前有两大类构建所需角色的方法:一类为自顶向下的角色工程方法,它通过分析系统的功能需求来创建角色并分配相应的权限,能较好地反映系统的业务逻辑和安全需求.另一类为自底向上的角色工程方法,它通过分析系统中已经存在的用户和权限之间的分配关系,利用

收稿日期:2013-06-07;修回日期:2014-02-23;责任编辑:梅志强
基金项目:国家 973 重点基础研究发展计划(No.2011CB302605);国家 863 高技术研究发展计划(No.2010AA012504,No.2011AA010705);国家自然科学基金(No.61272545,No.U1204605,No.61402149)

数据挖掘方法来得到能反映这种分配关系的角色,也被称为角色挖掘(Role Mining).后一类方法能够自动化和半自动化地发现角色,为寻找合适的角色提供辅助,因此成为目前在复杂的信息系统中获取角色的一个重要研究方向.

角色最小化问题是伴随角色挖掘研究而出现的一个热点问题,其目标是找出满足访问控制矩阵中用户-权限分配关系的最小角色集合,因为如果挖掘出过量的角色反而会增加系统管理的复杂性.目前已有不少算法和研究成果出现.Vaidya 等人[2]对基本的角色挖掘问题(RMP 问题)进行了系统的阐述和定义,分析了问题的理论边界,给出了 δ 近似和最小噪声两种方法,并指出该问题是 NP 难问题.在此基础上,Vaidya 等人[3]又提出了 edge-RMP 问题及其求解算法,进一步减轻管理员在角色分配和权限定义方面的负担.Lu 等人[4]提出了一个角色数目最小化问题的统一建模框架.Ene 等人[5]则提出了把最小角色数的求解转化为著名的最小 biclique 覆盖问题的方法,从而利用该问题的成熟方法来寻找最小角色集合.Zhang 等人[6]使用分解访问控制矩阵的方法来获取角色层次图,再利用图优化技术来寻找最合适的角色.

作为形式概念分析[7]的核心数据结构,概念格[8]模型和角色的层次模型具有天然的对应关系:用户对应于形式背景的对象,权限和资源对应于属性,形式概念的内涵对应于角色,Hasse 图对应于角色间的层次关系.这种对应关系使得利用概念格来进行角色获取的研究具有极大的便利性.文献[9]提出了一个利用概念格从访问控制矩阵中发现角色的方法,该方法还能判定系统的安全度并发现越权的恶意登陆.文献[10]利用概念格创建角色树进行角色挖掘,并在此基础上进行移动服务的推荐.文献[11]提出了一个利用形式概念分析来支持访问控制可视化的方法,该方法能够抽取角色-权限关系、发现潜在角色并绘制角色层次视图.文献[12]中,提出了一种利用概念格将角色的权限和属性进行关联分析的模型,根据用户属性将满足需求的角色自动分配给新用户.文献[13]通过各种实验证明利用概念格进行角色挖掘具有非常显著的性能.

由于概念格的完备性,在上述基于概念格的 RBAC 模型中,使得该模型将所有可能的角色都被挖掘出来,并以 Hasse 图的形式给用户提供一个完整的角色层次结构视图,从而为角色的设定提供辅助性作用.但是在实际应用中,过量的角色会使系统的管理更加复杂.因此在挖掘出所有可能角色及其层次关系的基础上,进一步推荐一个最小的角色集是一个很有意义的工作.

文献[14]给出了一个基于概念格进行角色发现的经典方法,该方法给出了挖掘角色层次的权重结构复杂度(Weighted Structural Complexity)作为最小代价函数,以此评判挖掘出的角色是否符合预定义的要求,并给出了一个贪婪算法挖掘有意义的角色信息.通过设定参数,该方法能够找出一个近似最小角色集,但是由于算法严格按照代价函数对概念角色进行剪枝,计算的中间过程完全忽略,使得管理员在进行角色筛选时,无法对角色进行调整.本文的方法则是在不改变原有概念格的基础上,致力于研究如何在基于概念格的 RBAC 模型所发现的角色集合中,找出满足最小权限原则的最小角色集合.由于最小角色集问题的求解是一个 NP 难问题[1],本文尝试在角色替代的最小角色集求解模型的基础上,设计一种贪婪算法来尽可能的降低时间复杂度.

2 基于概念格的 RBAC 模型

本节主要对概念格和基于概念格的 RBAC 模型的基本定义进行简单介绍.为了简化讨论,本文只考虑 NIST 标准中的 RBAC1 模型,并对一些基本概念稍作修改,去掉模型中的会话功能,相关定义参考自文献[7,8,14].

在形式概念分析中,形式背景是一个三元组 $K=(G,M,I)$.其中 G 是对象集合,M 为属性集合,$I\subseteq G\times M$. xIm 表示对象 x 具有属性 m.

设 $A\subseteq G$ 和 $B\subseteq M$,定义如下两个映射:

$f(A)=\{m\in M\mid \forall\, x\in A, xIm\}$;

$g(B)=\{x\in G\mid \forall\, m\in B, xIm\}$.

称二元组 (A,B) 为形式概念(简称概念),若满足 $A=g(B)$, $B=f(A)$.其中 A 称为外延,B 为内涵.对于概念 (A_1,B_1) 和 (A_2,B_2),若满足 $A_1\subseteq A_2$(等价于 $B_1\supseteq B_2$),记为 $(A_1,B_1)\leqslant(A_2,B_2)$.在形式背景 K 上,存在唯一的一个由 $\leqslant$ 诱导的格结构,称为概念格,记为 $L(K)$.在 $L(K)$ 中,若不存在 (A_3,B_3),使得 $(A_1,B_1)\leqslant(A_3,B_3)\leqslant(A_2,B_2)$,则称 (A_1,B_1) 为子概念,(A_2,B_2) 为父概念,并记为 $(A_1,B_1)\prec(A_2,B_2)$.将父概念和子概念用线段连接起来就构成了概念格的 Hasse 图.

称概念 (A,B) 为 m 属性概念,若满足 $A=g(\{m\})$.称概念 (A,B) 为 g 对象概念,若满足 $B=f(\{g\})$.

形式背景上的概念具有如下性质($\forall\, A,A_1,A_2\subseteq G,\forall\, B,B_1,B_2\subseteq M$):

性质 1 $A_1\subseteq A_2\Rightarrow f(A_1)\supseteq f(A_2)$, $B_1\subseteq B_2\Rightarrow g(B_1)\supseteq g(B_2)$.

性质 2 $A\subseteq g(f(A))$, $B\subseteq f(g(B))$.

性质 3 $f(g(f(A)))=f(A)$, $g(f(g(B)))=g(B)$.

性质 4 $(g(f(A)),f(A))$ 和 $(g(B),f(g(B)))$ 均为概念.

访问控制的三个基本要素为主体(用户)、客体(资源)和操作.为了简化讨论,这里直接使用权限表示操

作和客体的二元组,也即用户在对象(客体)上施加的操作.基于角色的访问控制的相关定义如下:

U、R、P 分别表示用户集合、角色集合、权限集合;

$PA \subseteq R \times P$,表示多对多的权限和角色对应关系;

$UA \subseteq U \times R$,表示多对多的用户和角色对应关系.

$\text{pers}(r) = \{p \in P \mid (r, P) \in PA)\}$表示角色 r 所具有的权限集.

$\text{PERS}(R) = \{p \in P \mid r \in R, (r, P) \in PA)\}$表示角色集 R 所具有的权限集.

对于一个访问控制矩阵可以用一个形式背景 $K(U, P, IA)$来表示,其中 $IA \subseteq U \times P$.对于 $u \in U, p \in P, (u, p) \in IA$ 表示用户 u 具有权限 p.这样的形式背景称为安全背景.其对应的概念格 $L(K)$称为安全背景 K 上的安全格.安全形式背景及安全格如表 1 与图 1 所示.

表 1　安全形式背景示例

用户	权限				
	a	b	c	d	e
1	1	1	1	0	1
2	1	1	0	0	0
3	0	0	1	1	1
4	1	0	1	1	0
5	1	0	1	0	1

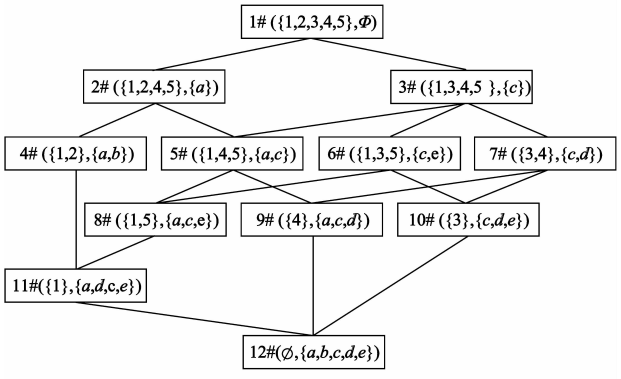


图1　表1所示安全形式背景的安全格

$L(K)$上的概念 $C = (A, B)$在本文中称为角色概念.其内涵 B 是权限的集合,对应于访问控制矩阵上的一个角色,而外延 A 则为具有该角色的用户.因此,角色与角色概念是一一对应的关系.角色概念 C 所对应的角色,在本文中称为概念诱导的角色,记为 $\text{int}(C)$.概念角色的集合 CS 所对应的角色集合,记为 $\text{INT}(\text{CS})$.文献[9]分析并证明了安全格 $L(K)$上的所有角色概念的集合,构成了整个访问控制矩阵中的所有可能的角色及其层次结构.

3　最小角色概念集及其求解

文献[2]中对 RBAC 模型中的角色挖掘问题作出了如下定义.

定义 1　角色挖掘问题　对于 $m \times n$ 的访问控制矩阵 A,将其分解为大小分别为 $m \times k$ 和 $k \times n$ 的两个矩阵 UA 和 PA,使得 k 在所有可能的矩阵分解中最小.

定义 1 将访问控制矩阵中用户与权限的关系表示为 UA 矩阵用户与角色的关系,而角色与权限的关系由 PA 矩阵表示.角色挖掘问题就是寻找满足用户和权限分配关系的最小的角色集合.在安全格上,由于概念与角色间存在一一对应关系,最小角色集的求解可以转化为在安全格中寻找最小角色概念集的问题.

定义 2　最小角色概念集　对于安全背景 K,如果其安全格的一个概念集合 S_m,满足以下两个条件,则称 S_m为安全背景 K 上的最小角色概念集:

条件 1　对于 K 中的每个用户所具有的权限,都可以由 S_m中的若干个概念的内涵的并集来表示.

条件 2　S_m中的概念个数是最少的.

由于安全格包含了整个访问控制矩阵中的所有可能的角色.因此,满足定义 2 的最小角色概念集即是定义 1 的角色挖掘问题的解.

定义 3　角色替代　设角色 $r \in R$,角色集 $\text{RS} \subseteq R$,若 $\text{pers}(r) = \text{PERS}(\text{RS})$,称 RS 是 r 的一个替代.

由定义 3 可知,一个角色可以由其他角色的组合来表示.在安全格上就表现为,一个概念的内涵是其他几个概念的内涵的并集.如图 1 中,5 # 概念的内涵就可以由 2 # 和 3 # 概念内涵的并集来表示.

定义 4　角色集替代　设角色集 $\text{RS}_1, \text{RS}_2 \subseteq R$,若 $\forall r \in \text{RS}_1$,都能在 RS_2中找到一个替代,且 $\text{PERS}(\text{RS}_1) = \text{PERS}(\text{RS}_2)$,称角色集 RS_2是 RS_1的一个替代.

定义 4 将定义 3 的角色替代推广到角色集与角色集之间.

定义 5　角色约简　设角色 $r \in R$,角色集 $\text{RS} \subseteq R$,若 $\text{pers}(r) = \text{PERS}(\text{RS})$,称 $R - \{r\}$是 R 的一个约简.

定义 5 实质上是定义 3 的一个子集替代.

在上述定义的基础上,我们给出如下的相关定理.

定理 1　安全格上所有对象概念的集合必然满足定义 2 条件 1.

证明　对于安全背景 $K(U, P, IA)$上的任意用户 $u \in U$,其在安全背景 K 上的权限集为$\{p \in P \mid (u, p) \in IA\} = f(\{u\})$,由对象概念的定义知,任意用户的权限集恰为该用户的对象概念的内涵,定理得证.

推论 1　安全格上所有对象概念所诱导的角色满足最小权限原则.

证明　由定理 1 可知,每个用户的权限恰为该用户的对象概念的内涵.因此满足最小权限原则.

定理 2　设角色集 $\text{RS}_1, \text{RS}_2, \text{RS}_3 \subseteq R$,若 RS_2 是 RS_1

的一个替代,RS_3 是 RS_2 的一个替代,则 RS_3 也是 RS_1 的一个替代.

证明 由于 RS_2 是 RS_1 的一个替代,RS_3 是 RS_2 的一个替代,由定义 4 得,$PERS(RS_1) = PERS(RS_2) = PERS(RS_3)$.下面证 $\forall r \in RS_1$,都能在 RS_3 中找到一个替代.由于 RS_2 是 RS_1 的一个替代,由定义 3 和 4 知若 $\forall r \in RS_1$,都能在 RS_2 找到一个对应的替代,记为 RSr,且有 $pers(r) = PERS(RSr)$.设 RSr 中有 n 个元素,由于 RS_3 是 RS_2 的一个替代,则对于 $\forall r_i \in RSr$,都能在 RS_3 找到一个对应的替代,记为 RSr_i,有 $pers(r_i) = PERS(RSr_i)$.所以有 $\forall r \in RS_1$,在 RS_3 存在 $\bigcup_{i=1\cdots n} RSr_i$ 满足 $pers(r) = PERS(RSr) = PERS(\bigcup_{i=1\cdots n}\{r_i\}) = \bigcup_{i=1\cdots n} PERS(\{r_i\}) = \bigcup_{i=1\cdots n} pers(r_i) = \bigcup_{i=1\cdots n} PERS(RSr_i) = PERS(\bigcup_{i=1\cdots n} RSr_i)$.因此,$\forall r \in RS_1$,都能在 RS_3 中找到一个替代.所以,RS_3 也是 RS_1 的一个替代.

推论 2 若一个角色概念集诱导的角色集是所有对象概念诱导的角色集一个替代,则角色概念集必然满足定义 2 条件 1.

证明 由定理 1 和定理 2 直接推论可证.

定理 2 实际上是角色替代的传递性证明.推论 2 则说明可以从对象概念的集合开始,不断寻找更小的角色替代集合来逐步缩小求解最小角色概念集.

定理 3 设 $CS \subseteq L(K)$,$C \in L(K)$,CS 是 C 的所有父概念构成的集合,若 C 不是属性概念,则 INT(CS)是 $int(C)$的一个替代.

证明 若要证明 INT(CS)是 $int(C)$的一个替代,只需证明 $PERS(INT(CS)) = pers(int(C))$.

设 $C = (A, B)$,CS 中有 n 个概念.由于 CS 是 C 的父概念集合,所以对 $\forall C_i = (A_i, B_i) \in CS$,有 $B_i \subseteq B$,所以有 $PERS(INT(CS)) = \bigcup_{i=1\cdots n} B_i \subseteq pers(int(C)) = B$.

下面用反正法证明.假定 $\bigcup_{i=1\cdots n} B_i \neq B$,设 $B' = B - \bigcup_{i=1\cdots n} B_i \subseteq B$,则有 B'不为空集,且 $B' \cap \bigcup_{i=1\cdots n} B_i = \varnothing$.对于 $\forall m \in B'$,由性质 4 可知,$(g(\{m\}), f(g(\{m\})))$为一个安全格上的概念.根据性质 1,$\{m\} \subseteq B \Rightarrow g(\{m\}) \supseteq g(B) \Rightarrow f(g(\{m\})) \subseteq f(g(B)) = B$,由概念格的偏序定义可得,$C \leqslant (g(\{m\}), f(g(\{m\})))$.由于 C 的所有父概念集合在 CS 中,$(g(\{m\}), f(g(\{m\})))$不是 C 的父概念,所以只有两种情况使得 $C \leqslant (g(\{m\}), f(g(\{m\})))$成立:$C = (g(\{m\}), f(g(\{m\})))$或存在 C_i 满足 $C \leqslant C_i \leqslant (g(\{m\}), f(g(\{m\})))$.由于 C 不是属性概念,有 $C \neq (g(\{m\}), f(g(\{m\})))$.根据概念格的偏序定义和性质 2,有 $C \leqslant C_i \leqslant (g(\{m\}), f(g(\{m\}))) \Rightarrow \{m\} \subseteq f(g(\{m\})) \subseteq B_i \subseteq B$,因此有 $\{m\} \subseteq B_i$,这与 $B' \cap \bigcup_{i=1\cdots n} B_i = \varnothing$ 矛盾.定理得证.

推论 3 设 $CS, CSP \subseteq L(K)$,$C \in CS$,CSP 是 C 的所有父概念的集合,若 C 不是属性概念,则 $INT(CS \cup CSP - \{C\})$是 INT(CS)的一个替代.

证明 由定理 3 可知 INT(CSP)是 $int(C)$的一个替代,再根据定义 4 可直接推论得出.

推论 4 设 $CS, CSP \subseteq L(K)$,$C \in L(K)$,CSP 是 C 的所有父概念的集合,C 不是属性概念,若 $CSP \subseteq CS$,则 $INT(CS - \{C\})$诱导的角色集是 INT(CS)的一个约简.

证明 由定理 3 和定义 5 可直接推论得出.

由于角色和概念的一一对应关系.角色的替代和约简其实就是角色概念的替代和约简.定理 3 和推论 3 实质上是给出了一个寻找角色替代的方法,即可以将角色概念集合中的每个概念用它的父概念来替代的方式寻找角色集合的替代.推论 4 则是给出了一个约简角色集合的方法.

定理 4 属性概念诱导的角色集不存在其他替代.

证明 设 C 为 m 属性概念,由属性概念的定义和性质 4 可知,$C = (A, B) = (g(\{m\}), f(g(\{m\})))$.下面用反证法证明不存在概念 $C' = (A', B')$ 满足 $pers(int(C')) = B' \subseteq pers(int C)) = B$,$m \in B'$ 且 $C' \neq C$.设存在概念 C',根据性质 1 和 3 有,由于 $B' \subseteq B = f(g(\{m\})) \Rightarrow g(B') = A' \supseteq g(f(g(\{m\}))) = g(\{m\}) = A$,所以 $A' \supseteq A$.再由 $m \in B'$,根据性质 1 有,$\{m\} \subseteq B' \Rightarrow g(\{m\}) = A \supseteq g(B') = A'$,所以有 $A \supseteq A'$.因此有 $A = A'$,与 $C' \neq C$ 矛盾.因此不存在角色概念,满足 $pers(int(C')) \subseteq pers(int C))$.定理得证.

定理 4 给出了利用定理 3 自下而上逐个用父概念的替代来进行角色替代的终止条件.

定理 5 同时为属性概念和对象概念的概念必然包含在最小角色概念集中.

证明 设 C 为 m 属性概念和 g 对象概念,$C = (A, B)$.由定理 1 的证明过程可知,用户的对象概念的内涵与该用户的权限集相等.由定理 4 可知,属性概念诱导的角色不存在其他替代.因此同时为属性和对象概念的概念所诱导的角色是唯一满足用户 g 的角色.所以必然包含在最小角色概念集中.定理得证.

定理 5 给出了必然包含在最小角色概念集中的角色概念,降低了角色替代的搜索范围.

4 最小角色概念集查找算法

本节主要描述在概念格上求解最小角色集的方法.需要说明的是,可以先用任意一种概念格的构造算法从访问控制背景中构造出概念格.

4.1 算法描述

本节在上述相关定义及定理的基础上,设计了一种贪婪算法来求解最小角色概念集.算法主要思想为:

从对象概念的集合开始,自下而上将集合中的角色概念逐个用父概念集合替代,直到遇到属性概念为止.在遍历过程中找到的元素个数最小的集合即是最小角色概念集.在角色概念替代的过程中,利用推论 4 缩减角色概念的数目.下面所述的算法 2 是整个算法的主算法,算法 1 是用于概念角色的约简.

算法 1　ReduceRoles 的算法描述

```
Function ReduceRoles(L(K),CandidateRoles)
输入:概念格 L(K);候选角色概念集 CandidateRoles
输出:约简后的候选角色概念集 ReduceRoleSet
BEGIN
1. ReduceRoleSet: = ∅;
2. DO
3.   FOR each C∈CandidateRoles
4.     IF(C 为属性概念) THEN
5.       记 C 为可选角色概念;将 C 从 CandidateRoles 移至 ReduceRoleSet;
6.     ELSE IF(C 的父概念均为可选或必选角色概念) THEN
7.       将 C 从 CandidateRoles 删除;
8.       ELSE IF(C 有且仅有一个非可选或必选角色概念的父概念) THEN
9.         C_P: = C;
10.      END IF;
11.  END FOR;
12. UNTIL CandidateRoles 中概念不发生变化;
13. 将 CandidateRoles 中的所有概念移至 ReduceRoleSet;
14. Return ReduceRoleSet;
END
```

在算法 1 中,CandidateRoles 保存原始的角色概念集合,ReduceRoleSet 保存约简后的角色概念集合.可选角色概念是指该概念是 ReduceRoleSet 集中的角色概念.必选角色概念是指该概念为最小角色概念集中的概念.

算法 1 第 3 ~ 11 行遍历 CandidateRoles 中的每个概念 C.若 C 为属性概念(第 4,5 行),则由定理 4 知 int(C)不存在替代,故 C 必然在 ReduceRoleSet 中,将其从 CandidateRoles 删除并加入到 ReduceRoleSet.若 C 的父概念均为可选角色或必选角色概念(第 6,7 行),则根据定理 3 和推论 2,C 可以被 ReduceRoleSet 中的概念替代,将其删除.若 C 的父概念中除了可选角色或必选角色概念的父概念之外只有一个父概念(第 8 ~ 10 行),根据据定理 3 和推论 1,可直接用该父概念替代 C.重复上述过程,直到找不到可约简或替代的角色概念.

在算法 2 中,MiniRoleSet 用于保存找到的最小概念角色集.RootSet 和 AttRootSet 分别用于保存对象概念和属性概念.CandidateRoles 用于保存角色概念集的替代.RoleSet 和 TempSet 用于保存角色替代过程中的临时最小概念角色集.

算法 2 第 1,2 行初始化 RootSet 和 AttRootSet;第 3,4 行计算并标记最小角色概念集的必选角色概念,并将其保存在 MiniRoleSet 中;第 5 行将对象概念中的非必选角色概念保存到 CandidateRoles 中,作为迭代求解最小角色概念集的起始概念集合;第 6 行调用算法 1 的 ReduceRoles 函数对 CandidateRoles 中的角色概念进行约简;第 9 ~ 18 行对 CandidateRoles 中的角色概念按推论 3 进行父概念替代,并调用 ReduceRoles 函数约简.如果找到的概念集合比原有保存在 RoleSet 中的概念个数更少,则继续以约简后的角色概念进行进一步迭代求解最小角色概念集.第 8,19 行的 UNTIL 循环不断将 CandidateRoles 中的集合进行替代和约简,直至找不到更小的角色概念集为止.最后在 20 行将找到的最小角色概念集存入 MiniRoleSet 中.

算法 2　SearchMiniRole 的算法描述

```
Function SearchMiniRole(L(K))
输入:概念格 L(K);
输出:找到的最小概念角色集 MiniRoleSet
BEGIN
1. RootSet: = {L(K)中的所有对象概念};
2. AttRootSet: = {L(K)中的所有属性概念};
3. MiniRoleSet: = RootSet ∩ AttRootSet;
4.   标记 MiniRoleSet 中的概念为必选角色概念;
5. CandidateRoles: = RootSet - MiniRoleSet;
6. RoleSet: = ReduceRoles(L(K),CandidateRoles);
7. DO
8.   CandidateRoles: = RoleSet;
9.   FOR each C∈CandidateRoles
10.    IF(C 不为属性概念) AND (父概念数 > 2) THEN
11.      将 C 的所有父概念加入 CandidateRoles;
12.      TempSet: = ReduceRoles(L(K),CandidateRoles);
13.      IF|TempSet| < = RoleSet THEN
14.        RoleSet: = TempSet;
15.      END IF;
16.      RoleSet: = TempSet;
17.    END IF;
18.  END FOR;
19.  UNTIL CandidateRoles≠RoleSet;
20. 将 CandidateRoles 中的所有概念移至 MiniRoleSet;
21. Return MiniRoleSet;
END
```

需要指出的是,在算法的第 9 ~ 18 行,由于只对 CandidateRoles 中的父概念约简后最小的角色概念集进行下一轮迭代,可能会将本轮结果不是最小、但后续迭代结果最小的角色概念集遗漏.因此,本算法不是全局最优解,而只是局部最优解,是一种贪婪算法.

设安全形式背景为 $K(U,P,IA)$.算法 2 的时间复

杂度主要依赖于第 8 ~ 20 行的 UNTIL 循环和第 9 ~ 18 行的 FOR 循环中对算法 1 的调用次数.其中,FOR 循环中 CandidateRoles 的元素个数小于对象概念的个数,根据定理 1 对象概念的个数小于用户数$|U|$.所以 FOR 循环的次数小于$|U|$.而 UNTIL 循环取决于角色概念由父概念向上迭代的次数,由于概念格的层数小于属性的个数,所以 UNTIL 循环的次数小于$|P|$.

算法 1 的时间复杂度与算法 2 类似,也取决于 UNTIL 循环和 FOR 循环的次数,同样的两者的循环次数分别小于$|U|$和$|P|$.

综上,本文算法的时间复杂度为 $O(|U|^2|P|^2)$.

4.2 算法示例

下面对图 1 所示的安全格进行最小角色的求解,来说明本文算法的求解过程.

步骤 1 得到对象概念和属性概念 RootSet = {11 #,4 #,10 #,9 #,8 #};AttRootSet = {2 #,4 #,3 #,7 #,6 #}.

步骤 2 找出必选角色概念 RootSet ⋂ AttRootSet = {4 #}.

步骤 3 角色概念替代初始集 CandidateRoles = {11 #,10 #,9 #,8 #}.

步骤 4 利用函数 ReduceRoles 对初始集进行角色概念约简.其中 11 # 概念的两个父概念 4 # 和 8 # 分别为必选和可选角色概念,故删除 11 # 概念.约简后 CandidateRoles = {10 #,9 #,8 #}.

步骤 5 对 CandidateRoles 进行第一轮第一个角色概念的替代.10 # 概念被它的父概念 6 # 和 7 # 替代:CandidateRoles = {6 #,7 #,9 #,8 #}.

步骤 6 利用函数 ReduceRoles 对 CandidateRoles 进行约简.其中,6 # 和 7 # 为属性概念;9 # 概念的父概念 5 # 为仅有的既不是必选角色概念又不是 CandidateRoles 中的可选角色概念的概念,故用 5 # 概念替代 9 # 概念;8 # 概念的两个父概念 5 # 和 6 # 分别为可选角色和必选角色,故删除.约简后 CandidateRoles = {6 #,7 #,5 #}.

步骤 7 约简后的 CandidateRoles 比当前最小角色概念集的概念数目少,故当前最小角色概念集 RoleSet = {6 #,7 #,5 #}.

步骤 8 对 CandidateRoles 进行第一轮第二个角色概念的替代.9 # 概念被它的父概念 5 # 和 7 # 替代:CandidateRoles = {10 #,5 #,7 #,8 #}.利用函数 ReduceRoles 对 CandidateRoles 进行约简.过程与步骤 6、步骤 7 类似,约简后当前最小角色概念集 RoleSet = {6 #,5 #,7 #}.

步骤 9 对 CandidateRoles 进行第一轮第三个角色概念的替代.8 # 概念被它的父概念 5 # 和 6 # 替代:CandidateRoles = {10 #,9 #,5 #,6 #}.利用函数 ReduceRoles 对 CandidateRoles 进行约简.过程与步骤 6、步骤 7 类似,约简后当前最小角色概念集 RoleSet = {7 #,5 #,6 #}.

步骤 10 将 CandidateRoles 重新赋值为当前最小角色概念集{7 #,5 #,6 #},并对 CandidateRoles 中的概念进行第二轮替代.过程与步骤 5 ~ 步骤 9 类似,不再赘述.

步骤 11 最后得到最小角色概念集 MiniRoleSet = {4 #,6 #,7 #,5 #}.

由示例可以发现,在自底向上进行角色概念替代时,会出现大量经由不同概念替代到达同一个概念替代的情形,如第一轮的角色概念替代中,步骤 7 ~ 步骤 9 计算出的替代为同一个角色概念集.如何利用概念格的数学性质来避免这种重复性的计算,是提高算法性能的一个方向.

在示例的求解过程中,能发现存在两个最小角色概念集{4 #,6 #,7 #,5 #}和{4 #,10 #,9 #,8 #}.前者是本文算法最终找到的最小角色概念集,后者是在步骤 4 的中间结果集.一个安全格中,可能存在多个最小角色概念集.本文算法优先考虑靠近属性概念的最小角色概念集.这是由于在实际的角色分配中,单个角色的权限越小,管理越方便.

5 实验及讨论

为了检验算法的性能与准确性,选取随机生成的两组安全形式背景作为测试数据.测试平台硬件为 2.3Ghz 的 CPU 和 3GB 内存,操作系统为 Windows XP.

在第 1 组安全形式背景集中,具有相同的权限数 30,用户数目从 100 到 500 以间隔 20 变化.测试的目的在于观察由用户数目增大导致候选角色概念增多的情况下算法的时间性能和准确度(真实的最小角色概念集与贪婪算法所找到的最小角色概念集的比例)的变化.算法的时间性能如图 2 所示,准确度如图 3 所示.可以看到随用户数目增大,算法的时间开销呈指数级增长,准确度则不断降低.这是由于用户数目的增加,导致了安全格更加庞大,搜索角色的替代所需要的时间越来越多.同时用户数目的增加导致定义 2 条件 1 的初始集增大,贪婪算法趋向于局部最优解的概率增加.

第 2 组安全形式背景集具有相同的用户数 200,权限数目从 10 到 150 以间隔 10 变化.测试的目的在于观察由权限数目对算法的时间性能和准确度的影响.时间性能如图 4 所示,准确度如图 5 所示.由图示可知随权限数目增大,算法所需时间呈指数级增长,准确度则不断增加.这是由于权限数目的增加,导致了安全格增大,算法搜索所需要的时间也随之增多.但是权限的增

多使得对满足定理3的角色替代的概率增大,这样使得贪婪算法由于角色替代的选择导致限于局部最优解的可能性降低,所以更容易找到全局最优解.

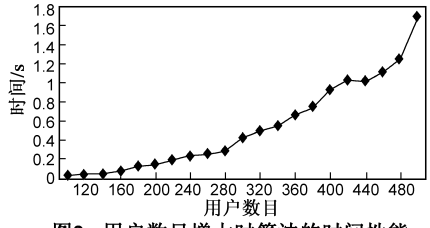


图2　用户数目增大时算法的时间性能

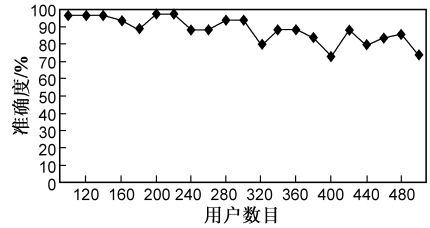


图3　用户数目增大时算法的准确度

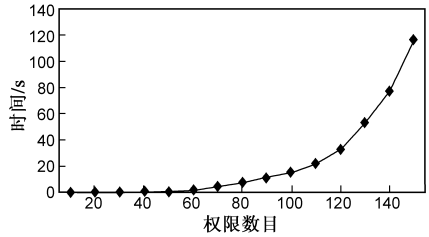


图4　权限数目增大时算法的时间性能

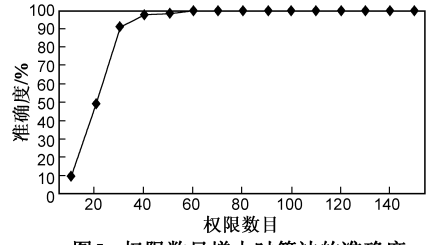


图5　权限数目增大时算法的准确度

由两组实验的结论可以发现,对于用户数目不太多而权限非常多的情况下,本文的贪婪算法具有较高的准确率.因此比较适用于待管理资源庞大、权限管理复杂、用户数目相对不多的复杂信息系统中的角色优化问题.同时,对用户数目庞大、但权限需求同质化较高的系统(也即大量用户具有相同的权限),本文算法也能取得很好的角色最小化效果.

6　结论及进一步的工作

本文研究了基于概念格的 RBAC 模型中的最小角色集问题,给出了最小角色集、角色替代和角色约简的定义,证明了角色替代、角色约简和最小角色的相关定理,初步建立了基于角色替代方式的最小角色集求解模型,并提出了一种基于替代和约简的最小角色集求解算法.这对降低基于概念格的 RBAC 模型的安全管理的复杂度有着积极的意义.实验和分析验证了本文相关理论和算法的有效性.

研究概念格的数学性质,继续寻找时间复杂度更低的算法是本文的进一步工作.另外,最小角色集的全局最优解与可接受的局部最优解之间的差异性度量研究也是一项需要进一步研究的有意义的工作.

参考文献

[1] 李凤华,苏锷,马建峰.访问控制模型研究进展及发展趋势[J].电子学报,2012,40(4):805－813.
Li Feng-hua, Su Mang, Shi Guo-zhen, Ma Jianfeng. Research status and development trends of access control model[J]. Acta Electronica Sinica, 2012, 40(4): 805－813. (in Chinese)

[2] Vaidya J, Atluri V, Guo Q. The role mining problem: finding a minimal descriptive set of roles[A]. Proceedings of the 12th ACM symposium on Access control models and technologies[C]. New York: ACM, 2007. 175－184.

[3] Vaidya J, Atluri V, Guo Q, et al. Edge-rmp: Minimizing administrative assignments for role-based access control[J]. Journal of Computer Security, 2009, 17(2): 211－235.

[4] Lu H, Vaidya J, Atluri V. Optimal boolean matrix decomposition: Application to role engineering[A]. IEEE 24th International Conference on Data Engineering[C]. Piscataway: IEEE, 2008. 297－306.

[5] Ene A, Horne W, Milosavljevic N, et al. Fast exact and heuristic methods for role minimization problems[A]. Proceedings of the 13th ACM symposium on Access control models and technologies[C]. New York: ACM, 2008. 1－10.

[6] Zhang D, Ramamohanarao K, Ebringer T. Role engineering using graph optimisation[A]. Proceedings of the 12th ACM symposium on Access control models and technologies[C]. New York: ACM, 2007. 139－144.

[7] Ganter B, Wille R. Formal Concept Analysis: Mathematical Foundations[M]. Berlin: Springer, 1999.

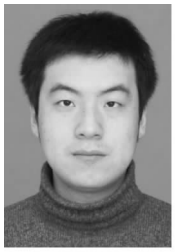
[8] 张磊,张宏莉,殷丽华,韩道军.概念格的属性渐减原理与算法研究[J].计算机研究与发展,2013,50(2):248－259.
Zhang Lei, Zhang Hong-li, Yin Li-hua, Han Dao-jun. Theory and algorithms of attribute decrement for concept lattice[J]. Journal of Computer Research and Development, 2013, 50(2): 248－259. (in Chinese)

[9] Sobieski Ś, Zieliński B. Modelling role hierarchy structure using the formal concept analysis[J]. Annales UMCS, Informatica, 2010, 10(2): 143－159.

[10] Wang Jian, Zeng Cheng, He Chuan, Hong Liang, et al. Context-aware role mining for mobile service recommendation

[A]. Proceedings of the 27th Annual ACM Symposium on Applied Computing[C]. New York: ACM, 2012. 173 – 178.
[11] Gauthier F, Merlo E. Investigation of access control models with formal concept analysis: A case study[A]. 2012 16th European Conference on Software Maintenance and Reengineering(CSMR)[C]. Piscataway: IEEE, 2012. 397 – 402.
[12] Han DaoJun, Zhuo Hankui, Xia Lanting, Li Lei. Permission and role automatic assigning of user in role-based access control[J]. Journal of Central South University of Technology, 2012, 19(4): 1049 – 1056.
[13] Molloy I, Li N, Li T, et al. Evaluating role mining algorithms [A]. Proceedings of the 14th ACM Symposium on Access Control Models and Technologies[C]. New York: ACM, 2009. 95 – 104.
[14] Molloy I, Chen H, Li T, et al. Mining roles with multiple objectives[J]. ACM Transactions on Information and System Security(TISSEC), 2010, 13(4): 36.

作者简介



张 磊 男, 1981 年出生, 河南博爱人. 博士研究生, 讲师, 主要研究方向为形式概念分析, 数据挖掘, 信息安全.
E-mail: zhanglei@henu.edu.cn

张宏莉 女, 1973 年出生, 吉林榆树人. 博士, 教授, 博士生导师, 主要研究领域为信息内容安全, 网络测量和建模, 网络计算, 并行处理.
E-mail: zhanghongli@hit.edu.cn

韩道军 男, 1979 年出生, 河南固始人. 博士, 副教授, 主要研究方向为形式概念分析, 信息安全.
E-mail: handj@henu.edu.cn

沈夏炯 男, 1963 年出生, 河南开封人. 博士, 教授, 主要研究方向为形式概念分析, 信息安全.
E-mail: shenxj@henu.edu.cn