# 角色工程中一种最小角色集的求解算法

韩道军
(河南大学数据与知识工程研究所 开封 475004)

**摘 要** 角色工程是基于角色访问控制(Role-Based Access Control,RBAC)中的一个重要研究方向,它主要研究角色的获取与优化。目前已有很多关于角色获取与优化的研究,但这些研究所提出的算法要么复杂度较高(NP完全的),要么不能保证优化的效果是最优的。因此,研究建立了一种新的最小角色集求解算法。该算法的时间复杂度是多项式的,而且可以保证优化效果是最优的。首先通过引入代数方法对数据进行预处理,使用极大线性无关组对角色集合进行化简;然后在分析了集合各个运算符特点的基础上,利用概念格模型建立等价类,并最终获得最小角色集。实验结果表明所提算法是有效的。

**关键词** RBAC,角色工程,最小角色集
**中图法分类号** TP301 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2017.08.021

## Acquiring Minimal Role Set Algorithm in Role Engineering

HAN Dao-jun
(Institute of Data and Knowledge Engineering, Henan University, Kaifeng 475004, China)

**Abstract** Role engineering, which focuses on role acquisition and optimization, is an important research area in role-based access control (RBAC). Recently, there have been many researches on role acquisition and optimization. However, these researches either have high time-complexity (NP complete), or cannot guarantee the acquired results optimal. In this paper, we proposed a new algorithm to acquire the optimal (or minimal) role set. Our algorithm has polynomial time complexity, and guarantees to acquire the optimal result. We first pretreated the role set using an algebra measure, and maximum linear independent group was introduced to simplify the role set. And after analyzing the characteristic of every set operator, we built equivalence classes using concept lattice model, and finally attained the optimal role set. The experiment results show that our algorithm is effective.

**Keywords** Role based access control, Role engineering, Minimal role set

## 1 引言

在各类复杂信息系统的开发过程中,需要对受控资源进行保护,因此离不开访问控制。基于角色的访问控制实现了用户与权限的逻辑分离,管理方便,被广泛应用于各类复杂信息系统。然而,在RBAC应用过程中,对角色的获取、设置、管理等较为困难。为了解决该问题,E. J. Coyne首次提出了角色工程[1]。他主要参考RBAC3,研究了系统中的角色名称、权限、约束和角色层次关系。在角色工程中,角色的提取与优化是一项重要的研究内容,其作用有以下两点。1)降低角色管理复杂度。通过角色化简,可以找出系统中需要管理的核心角色对象,减少管理角色的数量,降低角色管理复杂度。文献[2]指出引入角色的目的在于:①减少管理对象数目;②权限变化没有用户变化频繁。显然,减少管理角色的数量与引入角色的目的①相同。2)提高系统安全性。角色与系统安全紧密相关,无冗余的角色集合有利于管理员制定出合理且安全的资源访问策略。

针对角色的提取与优化,目前有基于图论、布尔矩阵分解和数据挖掘等算法[3-8]。然而,基于图论和矩阵分解的方法都是NP完全的;而基于数据挖掘的方法是统计得到的结果,不能保证是最优的。经研究发现,利用概念格表示方法和极大线性无关组求解方法,可以在多项式时间内对角色集合进行优化。与现有优化方法相比,本文算法具有多项式时间复杂度,并且可以保证优化效果。

根据这个思想,文中建立一种全新的算法——最小角色集算法(Minimal Role Set Algorithm,MRS)。该算法分为两个阶段。第一阶段,数据预处理及初步化简。首先将角色集合转换为向量组表示,使用求解极大线性无关组的方法对向量组进行化简;然后引入多重集,分析多重集中运算符与代数符号之间的关系,建立等价转换关系,将向量间的线性相关关

到稿日期:2016-07-15 返修日期:2016-12-26 本文受国家自然科学基金资助项目(61272545,61402149),河南省科技攻关计划基金资助项目(142102210390),河南省教育厅科技攻关计划基金资助项目(14A520026),河南省博士后科研项目(2015036)资助。
韩道军(1979—),男,博士,副教授,主要研究方向为形式概念分析、空间数据处理、信息安全,E-mail:hdj@henu.edu.cn。

系转换为多重集之间的集合运算，据此得到初步化简的角色集合以及被化简角色的表示关系。第二阶段，通过概念格模型进行角色优化。首先分析集合完备运算符中各个运算符的特点；然后利用概念格，对化简后的角色集合按照集合之间的运算关系构造一种单调的集合表达式表达方法。

本文第 2 节介绍相关工作及其与本文工作的区别；第 3 节介绍最小角色集的定义并描述待解决问题；第 4 节首先介绍整个算法框架，接着介绍算法的预处理步骤——基于向量的数据预处理方法，然后分析集合运算符特征，结合概念格模型给出最小角色集的求解算法及复杂度分析；第 5 节通过实验说明本文算法的优越性（包括效果和效率）和有效性；最后总结全文。

## 2 相关工作

自 E. J. Coyne 提出角色工程以来，众多研究人员提出了一系列算法来讨论 RBAC 中的角色提取、优化、约束和角色层次[3-19]，为 RBAC 的扩展模型及授权安全分析提供了重要支撑。按照传统方法，可以根据系统的组织结构，按照自顶向下或自底向上两种方法得到基本角色集合，但此时得到的角色集合是粗糙的，可能存在冗余。文献[1]指出了设计角色集合的 3 个目标：完备（complete）、正确（correct）、有效（efficient）。在实际应用中，设计者往往为了追求完备和正确而引入了过多的角色，从而给管理带来不便。显然，若引入的角色过多，将会使系统访问控制的复杂度大大增加。因此，角色的优化是一个值得研究的课题，具有重要的现实意义。

现有的角色提取算法存在时间复杂度过高（NP 完全）或结果不是最优的问题。文献[7]提出了以角色挖掘问题（Role Mining Problem，RMP）描述的最小角色集合；分析了几种类型（Basic RMP，$\delta$-approx RMP 和 MinNoise RMP）的复杂度，其时间复杂度是 NP 完全的。文献[3]使用图优化技术对分解后的矩阵进行处理，提取出具有层次关系的角色，其时间复杂度也是 NP 完全的。文献[4]介绍了一种可以使用领域专家知识进行聚类的角色提取方法；文献[5]提出了一种通过子集枚举的聚类方法提取角色；文献[9]通过给角色赋值权重，设定阈值，将角色挖掘过程限定为用户权限的识别过程，将用户和权限以二部图的形式表示，通过聚类的方法得到稳定的（stable）角色集合；文献[8]在文献[7]的基础上，利用图论中的二分团覆盖（biclique cover）方法对用户、角色和权限集合进行处理，并构造一种基于格论的启发式方法来降低时间复杂度，其在一些实际领域中得到了较好的实验结果；文献[16]提出了一种通过分析业务的意义来获取角色的方法。文献[4-5,7-8]存在的问题是不能保证结果最优。

本文建立了一种能够在多项式时间内得到最优结果的算法。上述方法集中于如何使用一些方法（图论、数据挖掘和布尔矩阵分解等）提取出角色，而本文关注的是：对于给定的角色集合 $R$，根据每个角色已经分配的权限关系，找出 $R$ 中能够被化简（可由其他角色构造的表达式表示）的角色，从而减少管理角色的数量。其中，$R$ 中角色的产生方式比较简单：具有完全相同权限的一类用户即为一个角色。我们认为，初始的角色及权限关系可在需求获取过程中采集到，本文方法具有普适性，能够直接利用现有需求获取无冗余的角色集合及相关知识。

从已有文献来看，国内对角色工程研究较多。文献[17]考虑到权限的特点，引入权重刻画权限，体现了不同权限在系统中的重要程度，并提出了一种基于权重的加权角色提取算法。文献[18]提出了一种基于活动和事件驱动的角色工程方法，其中，事件是常规任务，活动则由事件来触发，角色由多个事件的重叠部分来创建，三者之间是多对多关系。张磊等研究了在概念格的 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，并给出概念格模型上最小角色集、角色替代、角色约简的定义以及相关定理的证明；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[19]。蔚清琴、米秀明、叶威分别提出分层角色挖掘、基于进化算法的角色挖掘算法和基于 CSP 的角色挖掘等方法[20-22]。冯登国等指出，当前角色挖掘技术大多基于精确、封闭的数据集，在应用于大数据场景时还需要解决数据集动态变更以及质量不高等特殊问题[23]。方滨兴等指出，大数据下的角色工程需要从攻击和防护的角度综合考虑[24]。

## 3 问题描述

文献[7]中对最小角色集的定义为：给定一个 $m\times n$ 的矩阵 $A$ 来表示用户权限关系，将 $A$ 分解为两个矩阵 $B$ 和 $C$，其中 $m\times k$ 的矩阵 $B$ 表示用户角色的赋值关系，$k\times n$ 的矩阵 $C$ 表示角色的权限关系，并且满足 $k$ 是最小的，则 $k$ 为期望的最小角色数量，且可以由矩阵 $C$ 得到最小角色集。该文献同时指出，找出最小角色集的问题不存在多项式时间复杂度。根据文献[7]中矩阵分解的实际意义，本文将最小角色集的定义等价转换为基于集合的方式描述。

**定义 1**（最小角色集） 设角色集合为 $S=\{r_1,r_2,\cdots,r_n\}$，记 $Per(r_i)$ 为 $r_i$ 所具有的权限集合（$0<i\leqslant n$）。令 $S_m\subseteq S$，若 $S_m$ 满足以下两个条件，则称 $S_m$ 为 $S$ 的最小角色集。

（1）除 $S_m$ 之外的其他元素都可以由 $S_m$ 中的元素通过运算获得。具体地，记 $PER=\{Per(r_i)\mid r_i\in S_m\}$，$OP=\{\cup,\cap,-,+_M,-_M\}$，其中 $+_M$ 和 $-_M$ 为多重集中的求和与相对差运算符，则对于任意 $r_j\in S-S_m$，$Per(r_j)$ 是由 $PER$ 和 $OP$ 构成的系统 $G$ 的一个元素。

（2）$S_m$ 中的元素个数是最少的。具体地，对于任意 $r_k\in S_m$，$Per(r_k)$ 不是由 $PER$ 和 $OP$ 构成的系统 $G$ 的一个元素，其中 $PER=\{Per(r_i)\mid r_i\in S_m-\{r_k\}\}$。

在一个复杂信息系统 $A$ 中，若存在 $n$ 个权限需要控制，则最多可以定义 $2^n-1$ 个角色。显然，这些角色之间具有关联关系，其中一些角色可由其他角色表示。对于 $A$ 中用到的角色集合 $S_A$，使用 $S_A$ 的最小角色集对 $A$ 中的访问控制策略进行描述，可以简化对系统的描述，降低系统研发和安全管理成本。

在给出最小角色集定义的基础上，本文要解决的问题描述如下：寻找一个最小角色集求解算法，使得该算法能够处理

输入的角色集合 $S$,以找出 $S$ 的一个最小角色集 $S_m$,同时表示出集合 $S-S_m$ 中每个元素与 $S_m$ 的运算关系。

## 4 最小角色集求解算法

根据问题的描述,本文建立一种最小角色集求解算法 MRS。根据角色集合的实际意义,分析集合运算符"∪","∩","−"的特点,并通过引入概念格模型求出角色集合之间的完备交集运算结果;然后找出一种单调的集合表达式构造方法;最后判定待化简角色是否与集合表达式等值。$\{\cup,\cap,-,+_M,-_M\}$中的运算符$+_M$和$-_M$应用于算法的数据预处理阶段,详细介绍见 4.2.3 节。算法框架描述如下:

(1)运用代数方法对角色集合 $S$ 进行初步化简(即数据预处理),得到集合 $S_1$。

(2)分析角色集合 $S_1$ 的特点,找出包含私有权限的角色集合 $S_2$,则集合 $S_2$ 不能被约简。

(3)对于 $S_3=S_1-S_2$ 中的每个元素 $x$,判断 $Per(x)$是否是由 $PER$ 和 $OP$ 构成的系统 $G$ 的一个元素。若 $Per(x)$是系统 $G$ 的一个元素,则 $x$ 可以被约简,修改 $S_3=S_3-\{x\}$;否则 $x$ 不可被约简。其中 $PER=\{Per(r_i)\mid r_i\in S_3-\{x\}\}$。

下面对算法的各个步骤进行详细介绍。

### 4.1 数据预处理

如第 3 节所述,求最小角色集的过程就是判断某个角色的权限集合是否是系统 $G$ 中的一个元素。该过程可描述为:对集合 $R=\{r_1,r_2,\cdots,r_n\}$,要判断 $R$ 中是否存在元素 $r_i$ 可被化简,则只需找出一个表达式 $Y$ 使得 $Per(r_i)$与表达式 $Y$ 的值相等。其中,$Y$ 由 $R-\{r_i\}$中的元素在函数 $Per$ 的作用下应用运算符$\{\cup,\cap,-,+_M,-_M\}$中的任一元素构造。对于一个有限集合 $X$($X$ 中元素为集合),若不加任何约束,使用运算符集合$\{\cup,\cap,-,+_M,-_M\}$时,能够构造出的表达式的个数几乎是无穷的,空间巨大。显然,在构造表达式之前通过一种简单的方法来减少 $X$ 中的元素个数具有重要的意义。本文采用一种代数方法,该方法在将角色集合通过向量描述的基础上,通过寻找向量组的极大线性无关组的方法实现集合的初步化简。为了实现这个目标,需要用到两个概念,即极大线性无关组和多重集。

#### 4.1.1 极大线性无关组

按照向量的方式对 $R$ 中角色具有的权限进行描述,则 $R$ 中所有角色具有的权限构成一个向量组 $RV$,且 $RV$ 中每个维度分量的取值范围均为$\{0,1\}$。表 1 所列的角色及权限按照向量的方式描述为 $RV_1$:$r_{v1}(0,1,0,1,1)$,$r_{v2}(1,0,1,0,1)$,$r_{v3}(0,1,1,0,1)$,$r_{v4}(1,0,0,1,0)$,$r_{v5}(1,1,1,0,1)$,$r_{v6}(1,1,0,1,0)$。

表 1　角色及权限示意表

| | $p_1$ | $p_2$ | $p_3$ | $p_4$ | $p_5$ |
|---|---|---|---|---|---|
| $r_1$ | 0 | 1 | 0 | 1 | 1 |
| $r_2$ | 1 | 0 | 1 | 0 | 1 |
| $r_3$ | 0 | 1 | 1 | 0 | 1 |
| $r_4$ | 1 | 0 | 0 | 1 | 0 |
| $r_5$ | 1 | 1 | 1 | 0 | 1 |
| $r_6$ | 1 | 1 | 0 | 1 | 0 |

将 $R$ 转换为向量形式表示后,能够利用代数中线性相关特性,通过求解向量组的极大线性无关组对向量组进行化简,即对 $R$ 进行初步化简。下面介绍线性相关和极大线性无关组的概念及问题转换的等价性。

**定义 2**(线性相关)[25]　如果向量组 $a_1,a_2,\cdots,a_s(s\geqslant 2)$中有一向量可以经其余的向量线性表出,那么向量组 $a_1,a_2,\cdots,a_s$ 称为线性相关的。

**定义 3**(线性无关)[25]　向量组 $a_1,a_2,\cdots,a_s$ 不线性相关,即没有不全为零的数 $k_1,k_2,\cdots,k_s$ 使 $k_1a_1+k_2a_2+\cdots+k_sa_s=0$,则称为线性无关。

**定义 4**(极大线性无关组)[25]　向量组的一个部分组称为一个极大线性无关组,如果这个部分组本身是线性无关的,并且从这个向量组中任意添一个向量(在还有向量的前提下),所得的部分向量组都线性相关。

求解向量组的极大线性无关组的方法是使用行列初等变换,即通过改变向量系数及利用向量间的加、减操作,使用某些向量(极大线性无关组中的部分或全部)表示其他向量。例如:向量组 $RV_1$ 的极大线性无关组为 $r_{v1},r_{v2},r_{v3},r_{v4},r_{v5}$,$r_{v6}=r_{v4}+r_{v5}-r_{v2}$。

#### 4.1.2 多重集

在求解向量组的极大线性无关组的过程中,有两个对象需要指明:1)数乘运算。一个数与向量相乘,就等于这个数与向量的每个分量相乘,从而得到一个相同维度的新向量。2)加(减)运算。向量间的加(减)为向量间对应分类的值进行算术意义下的加(减)运算,并得到一个相同维度的新向量。

以上两个对象在代数意义下是平凡的,但对于本文中的具有集合意义且分量取值仅为 0 或 1 的特殊向量,则需要说明这种操作的合理性。在集合中,并运算与相对差运算均不能与线性表出中的加运算和减运算直接映射,否则将会出现错误,导致等值关系不再成立。例如,向量组 $RV_1$ 中存在线性表出 $r_{v6}=r_{v4}+r_{v6}-r_{v2}$,若使用并运算替换加运算,相对差替换减运算,向量转换为集合,则有:$Per(r_6)=\{p_1,p_2,p_4\}$,$Per(r_4)\cup Per(r_5)-Per(r_2)=\{p_1,p_2,p_3,p_5\}\cup\{p_1,p_4\}-\{p_1,p_3,p_5\}=\{p_2,p_4\}$,显然该表达式的值与 $Per(r_6)$不相等。为了解决这类问题,本文引入多重集(multiset)对集合进行扩充,使得多重集中的运算能够与普通的代数运算建立映射关系。

集合的 3 个基本性质为:确定性、互异性、无序性。如果把集合的互异性性质去掉,则集合扩充为多重集(multiset),即集合中允许重复的元素。在实际问题中,某些元素的重复出现体现了某种实际意义,如方程的解中有重根、数据库中出现相同记录等。文献[26]总结了多重集理论的基本元素及常见操作;与模糊集等理论结合,同时讨论了多重集的偏序关系及混合集(hybrid set)等。下面介绍相关定义。

**定义 5**(多重集)[26]　假定 $D$ 是一个集合。建立在 $D$ 上的多重集为序偶$\langle D,f\rangle$,其中函数 $f:D\to N$,$N$ 为自然数集合,且称 $D$ 为支持集,$f$ 为重复度函数,$f(d)$为元素 $d$ 的重复度。

对于有限集 $D$,建立在 $D$ 上的多重集为一个序偶,与集

合的表示差异较大。为了统一描述，可以使用两种形式较为简单的方法：1)罗列重复元素的集合描述方法；2)将重复度变为对应元素的系数，即定义多重集 $X=\{k_0a_0,k_1a_1,\cdots,k_na_n\}$。其中，对于 $i\neq j$，有 $a_i\neq a_j$，$0<i\leqslant n$，$0<j\leqslant n$；$k_i\in N$ 为第 $i$ 个元素 $a_i$ 在多重集 $X$ 的重复度，若 $k_i=0$，则元素 $a_i$ 不在多重集 $X$ 中。例如，一个多重集 $A=\{a,a,a,b,c,c\}$，则 $A$ 的支持集为 $\{a,b,c\}$。多重集 $A$ 的另外一种描述方式为 $A=\{3a,b,2c\}$。

多重集上的运算种类很多，本文介绍相关的 3 个二元操作符：和、相对差、倍乘。

**定义 6**(多重集求和)[26] 假定有两个多重集 $A=\langle D,f\rangle$ 和 $B=\langle D,g\rangle$，$A$ 和 $B$ 的和用 $A+_MB$ 表示，结果为 $C=\langle D,h\rangle$，其中，$\forall a\in D,h(a)=f(a)+g(a)$。

**定义 7**(多重集求相对差)[26] 假定有两个多重集 $A=\langle D,f\rangle$ 和 $B=\langle D,g\rangle$，$A$ 和 $B$ 的相对差用 $A-_MB$ 表示，即从多重集 $A$ 中去除包含在 $B$ 中的元素，结果为 $C=\langle D,h\rangle$，其中，$\forall a\in D,h(a)=\max((f(a)-g(a)),0)$。

**定义 8**(多重集倍乘) 假定多重集 $A=\langle D,f\rangle$，$A$ 的 $m$ 倍数用 $mA$ 表示，即将多重集 $A$ 中的每个元素 $d$ 的重复度变换为 $mf(d)$，结果为 $C=\langle D,h\rangle$，其中，$\forall a\in D,h(a)=mf(a)$。

显然，根据定义，对任意两个多重集执行“$+_M$”和“$-_M$”操作时，得到的结果(多重集)中元素的重复度均为非负整数；并且，集合可以视为多重集的一种特殊情况，即集合中各个元素的重复度均为 1，且集合满足多重集的和与相对差等几种二元运算。

#### 4.1.3 极大线性无关组与多重集之间的转换关系

根据多重集的定义和相关运算的介绍，本文可以将角色具有的权限集合视为一个特殊的多重集，则在使用极大线性无关组中的向量表示其他被化简的向量(可由极大线性无关组线性表出)时，其中的初等变换使用的两种算术运算符(加、减)可以映射到多重集中进行计算。

假设极大线性无关组为 $B=\{a_1,a_2,\cdots,a_i\}$，与 $B$ 线性相关的一个向量为 $a_r$，其中 $a_i$ 为向量。则

$$a_r=k_1a_1+k_2a_2+\cdots+k_ia_i \tag{1}$$

其中，$k_j\in Z$，$0<j\leqslant i$，$Z$ 为整数集。与常规的向量组不同，对于本文给定的向量组，向量中的每个分量的取值范围都为 $\{0,1\}$，且在使用极大线性无关组中的向量线性表出某一向量时，要求线性表出的表达式中的系数为整数，即式(1)中的 $k_j\in Z$。显然，只需要对传统的极大无关组求解算法施加约束“行列初等变换时的系数必须为整数”即可。

根据式(1)，在 $a_r$ 的计算过程中，可以根据向量间的运算关系得到结果。由最小角色集的求解意义可知，找出被约简角色与最小角色集中的角色之间的关系是不可缺少的内容，也即式(1)中的右侧表达式(设为 $S$)为本文算法要找出的目标，同时，在找出 $S$ 后，能够对 $S$ 求值。在将角色及权限关系变换为向量形式进行处理时，式(1)可以直接求解。考虑到等式(1)表示的意义，可将其转换到多重集中运算，使得表示更为自然，即在对 $S$ 求值时，将其中的 $a_i$ 转换为多重集，运算符分别使用多重集中的运算符。按照运算符的特性，可以将算术运算符“+”和“−”直接替换为多重集中的“$+_M$”和“$-_M$”操作符。下面分析对应运算符的特性及映射的正确性。

(1)“+”与“$+_M$”的对应分析

根据“$+_M$”的定义，算术意义下对应系数的加与多重集中对应元素的重复度相加等同，可以直接替换。

(2)“−”与“$-_M$”的对应分析

在将“−”与“$-_M$”对应替换时，需要满足一个前提条件，即“−”中的负值不会在“$-_M$”中出现，否则，根据“$-_M$”的定义，运算结果中不包含为负的情况。解决方法有两种：1)将多重集转换为混合集[27]，使得负数在混合集中有解释；2)分析式(1)的特点，将表达式进行变换，使得式(1)中右侧表达式前 $m$ 项的系数为正，后 $n$ 项的系数为负，且 $m,n\in N,m>0,n\geqslant 0,m+n=i$，即在表达式中先计算加，再计算减，则可以使得每一个向量中均不出现负数。显然，方法 2)更为简洁，本文选用方法 2)。

根据上述分析(1)、(2)及定义 6，式(1)可以变换为：

$$a_r=k_ja_j+_M\cdots+_Mk_{j+m-1}a_{j+m-1}-_M|k_l|a_l-_M\cdots-_M|k_{l+n-1}|a_{l+n-1} \tag{2}$$

其中，$a_i$ 为多重集，且前 $m$ 项的系数为正，后 $n$ 项的系数为负，$m,n\in N,m>0,n\geqslant 0,m+n=i$。

#### 4.1.4 数据预处理过程的描述

数据预处理过程的框架描述如下。

**算法 1** DataPretreat(R)(对角色集合 R 进行预处理)

输入：角色集 R，权限集合 P

输出：化简后的角色集 $R_t$，表达式集合 ER

```
Begin
1. RV:=∅
2. For all r∈R do {
3.   r':=GetVector(r,P)//根据 P 中元素的关系将集合 r 转换为向量 r'
4.   add r' to RV}
5. RV_t:=∅,ERV:=∅,ER:=∅;// RV_t 表示向量组；ERV 表示以向量表示的表达式集合
6. (RV_t,ERV):=MaximumLinearIndependent(RV)
7. For all e∈ERV do {
8.   e':=GetMultiSetExpression(e)//将 e 转换为多重集形式的集合表达式 e'
9.   add e' to ER}
10. R_t:=ConvertVectorToSet(RV_t,P)//将向量组转换为集合
11. return R_t,ER
End
```

其中，第 6 行的 MaximumLinearIndependent($RV$)表示对向量组 $RV$ 求解极大线性无关组 $RV_t$ 以及经由 $RV_t$ 线性表出的其他向量，并将线性表出关系输出至 $ERV$；第 8 行的 GetMultiSetExpression($e$)表示根据式(1)和式(2)的关系将代数形式下的线性表出 $e$ 转换为多重集运算下的 $e'$。

### 4.2 最小角色集求解算法的描述及分析

对于以向量形式表示的角色集合 $R_s$，在使用算法 1 进行预处理后，可以得到一个约简的角色集合 $R_t$，且 $|R_t|\leqslant|R_s|$，

但是 $R_t$ 不一定是 $R_s$ 的最小角色集。例如,3 个向量 $v_1=(1,1,0)$,$v_2=(1,0,1)$,$v_3=(1,1,1)$,极大线性无关组为 $v_1$,$v_2$,$v_3$。若把 $v_1$,$v_2$,$v_3$ 转换为等价的集合 $s_1$,$s_2$,$s_3$ 进行处理,显然有 $s_3=s_1\cup s_2$,其中,$s_1=\{a,b\}$,$s_2=\{a,c\}$,$s_3=\{a,b,c\}$。产生这种现象的原因是:1)在使用极大线性无关组方法进行角色约简时,仅仅使用了集合的 $+_M$ 和 $-_M$ 运算,这不是完备的运算符集合;2)$+_M$ 操作与集合并不完全等价。因此还需要对 $R_t$ 进行化简。

在对 $R_t$ 进行操作以得到一个最小角色集的过程中,使用集合完备的"∪","∩","−"3 个运算符。如果不考虑集合之间的关系,则对应的计算表达式可能有很多种,运算空间很大。在对 $R_t$ 进行化简之前,首先分析集合运算符的特征及 $R_t$ 中元素(角色)之间的关系,以减小运算空间。

4.2.1　集合运算符的特征分析

假设表达式 $S_3=S_1\triangle S_2$,运算符$\triangle\in\{\cup,\cap,-\}$。设 $S_1$ 为当前结果集合,$S_2$ 为待选集合,$S_3$ 为计算结果,$S_4$ 为目标结果(即待化简元素的值),且 $S_1$,$S_2$ 不为空且不相等。则根据△的取值,依次分析:

1)当△为∪时,$S_3=S_1\cup S_2$,$|S_1|<|S_3|$;

2)当△为∩时,$S_3=S_1\cap S_2$,$|S_1|>|S_3|$;

3)当△为一时,$S_3=S_1-S_2$,$|S_1|\geqslant|S_3|$。

以上表达式中,$S_1$ 为已知,△和 $S_2$ 为待确定元素,$S_3$ 由 $S_1$,$S_2$ 和△确定。其中,$S_2\cap S_4\neq\emptyset$,否则 $S_1$ 无意义,可以直接被替换掉。则可以根据 $S_1$ 与目标集合 $S_4$ 的关系选择△和 $S_2$,分析过程如下:

(1)运算符△的选择

1)当$|S_4|>|S_1|$时,表明此时需要向 $S_1$ 中增加元素,则△为∪;

2)当$|S_4|<|S_1|$时,表明此时需要减少 $S_1$ 中的元素,则△为∩或△为−。

(2)待选集合 $S_2$ 的分析

1)当$|S_4|>|S_1|$,△为∪时,$S_2$ 应满足 $S_2\cap S_4\neq\emptyset$;

2)当$|S_4|<|S_1|$,△为∩时,$S_2$ 应满足 $S_2\cap S_1\neq\emptyset$,$S_2\cap S_4\neq\emptyset$;

3)当$|S_4|<|S_1|$,△为−时,$S_2$ 应满足$S_2\cap S_1\neq\emptyset$。

4.2.2　角色集合中元素关系的分析

由于角色与权限密切相关,权限的私有特性会影响到对角色的处理。本文分析了具有私有权限的角色特点,并对这种情况进行特殊处理。

(1)具有私有权限的角色特点分析

文献[2]介绍了私有权限的概念,私有权限在角色化简中具有特殊的性质,详细描述如下。

**定义 9**(私有权限)　权限 $p$ 为私有权限,当且仅当 $p$ 为某一个角色 $r$ 所拥有。

在求解最小角色集的过程中,私有权限具有的性质可用定理 1 描述。

**定理 1**　如果一个角色 $r$ 具有私有权限,则:1)$r$ 不可被约简;2)在使用"∪"操作构造表达式时,不可独立使用 $r$。

证明:根据私有权限的定义,存在权限集合 $X$ 仅仅包含于 $Per(r)$,且其他角色都不具有。显然,由于 $X$ 的存在,$r$ 不可以由其他角色表示,1)得证。假设在使用"∪"操作构造表达式时独立使用了 $r$,则必将引入 $X$,且 $X$ 无法通过非 $r$ 以外的集合的操作消去,从而引入了无法被消去的集合,故 $r$ 不可独立使用"∪"操作,2)得证。1)和 2)得证,定理 1 便得证。

由于角色集合 $R$ 可能存在具有私有权限的角色,因此先判断 $R$ 中是否包含具有私有权限的角色。如果 $R$ 中存在具有私有权限的角色,则可以在找出后利用其特殊性质来减少 MRS 算法中待化简角色的数量,从而提高算法效率。对应的判断算法 GetPrivacyPermissionSet 的伪码描述如下。

**算法 2**　GetPrivacyPermissionSet(R)(获取 R 中具有私有权限的角色集合)

输入:角色集合 R,权限集合 P

输出:具有私有权限的角色集合

Begin

1. 设多重集 $R_m=\langle P,f\rangle$,$R_m$ 中元素的获取及对应的重复度计算方式如下:$\forall r\in R$,对于$\forall x\in Per(r)$,如果 $x\notin R_m$,则添加 x 至 $R_m$,$f(x)=1$;如果 $x\in R_m$,则 $f(x)=f(x)+1$。
2. $S_t:=\emptyset$,$S_r:=\emptyset$
3. 对于$\forall a\in R_m$,如果 $f(a)=1$,则 $a\in S_t$
4. $\forall R_i\in R$,如果 $Per(r)\cap S_t\neq\emptyset$,则 r 为具有私有权限的角色,$r\in S_r$
5. 返回 $S_r$

End

(2)概念格模型及与集合运算符的结合分析

概念格[28]是一种有力的数据分析工具,已经在众多领域中得到了应用。假设给定形式背景(context)为三元组 $K=(G,M,I)$,其中 $G$ 是实例(对象)集合,$M$ 是描述符(属性)集合,$I$ 是 $G$ 和 $M$ 之间的一个二元关系,则存在一个唯一的偏序集合与之对应,并且根据这个偏序集合产生一种格结构,这种由背景$(G,M,I)$所构造的格 $L$ 称为一个概念格。格 $L$ 中的每个节点是一个序偶(称为形式概念,或简称概念),记为 $(X,Y)$,其中 $X\in 2^G$ 称为概念的外延($2^G$ 表示 $G$ 的幂集);$Y\in 2^M$ 称为概念的内涵。每一个序偶关于关系 $I$ 是完备的,即有性质:1)$X=\{x\in G|y\in Y,(x,y)\in I\}$;2)$Y=\{y\in M|x\in X,(x,y)\in I\}$。

在形式背景 $K$ 中,可以定义两个映射 $f:2^G\to 2^M$ 和 $g:2^M\to 2^G$,满足:

$$f(G_i)=\{m|(x,m)\in I,\forall x\in G_i\}$$

$$g(M_i)=\{x|(x,m)\in I,\forall m\in M_i\}$$

$f$ 和 $g$ 被称为 $2^G$ 和 $2^M$ 之间的 Galois 连接。对于任意的二元组 $(G_1,M_1)\in 2^G\times 2^M$,如果满足 $G_1=g(M_1)$ 和 $M_1=f(G_1)$,则称该二元组是信息表 $K$ 的一个形式概念(formal concept)。对于给定的形式概念 $C=(G_1,M_1)$,称 $G_1$ 为形式概念 $C$ 的外延,记为 $Extension(C)$,称 $M_1$ 为形式概念 $C$ 的内涵,记为 $Intension(C)$,$K$ 的所有形式概念的集合标记为 $CS(K)$。表 2 列出了一个形式背景,$u$ 行 $m$ 列的交叉处为 1 表示$(u,m)\in I$,$u$ 为对象,$m$ 为属性。其中 $G=\{1,2,3,4\}$,$M=\{a,b,c,d,e\}$,$I$ 描述了 $G$ 中元素具有 $M$ 中的属性值集。图 1

是由 $K_1$ 生成的概念格，以 Hasse 图表示。

表 2 形式背景示例 $K_1$

| | $a$ | $b$ | $c$ | $d$ | $e$ |
|---|---|---|---|---|---|
| 1 | 0 | 1 | 0 | 1 | 1 |
| 2 | 1 | 1 | 0 | 0 | 1 |
| 3 | 1 | 0 | 1 | 0 | 1 |
| 4 | 1 | 1 | 1 | 1 | 1 |

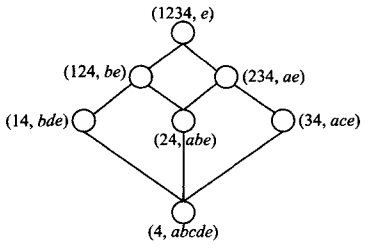


图 1 形式背景 $K_1$ 对应概念格的 Hasse 图

根据形式概念分析中函数 $f$ 和 $g$ 的定义可知，对于形式概念 $c$ 的外延和内涵，对应元素之间的关系可以按照如下方式描述：$Intension(c)=\bigcup_{i=1}^{k}x_i$，$Extension(c)=\bigcap_{j=1}^{l}y_j$，其中 $x_i\in Intension(c)$，$y_j\in Extension(c)$。

基于概念格的完备性，可以直接提供集合之间的交运算结果，因此在根据运算符进行选择时，可以根据操作符的情况，按照对应策略选取。显然，由于概念格中不但提供了不同集合间交集的所有信息，同时也提供了 $R_i$ 中单个集合的信息，因此可以在集合完备运算符中以概念格中节点的内涵为基本运算单位，以此作为并操作和相对差操作的单位。由对运算符的特征分析可知，在提取出交运算的结果作为基本运算单位后，并运算和相对差运算作为两个对集合 $S$ 作用不同（并运算使得集合 $S$ 有增大趋势，相对差运算使得集合 $S$ 有减小趋势）的运算符，可以根据当前结果与目标结果之间的关系直接确定运算符，使得表达式的构造具有单调特性。并且，在基于概念格的集合表示基础上，存在定理 2。

**定理 2** 在构造与 $R_i$ 相等的表达式的过程中，若节点 $c_x$ 作为并操作对象，设 $Intension(c_x)-R_i=X$，若 $X\neq\varnothing$，则 $X$ 需要被减掉，若 $c_x$ 为可以使用节点，则此时需要满足条件：$\forall x\in X$，$x$ 必须包含于概念格的其他某个或部分节点的内涵中；否则 $c_x$ 不能被选为可用节点。

证明：假设需要被减掉集合 $X$ 由概念 $C_1$ 引起，则可以将 $Intension(C_1)$ 分为两部分，$Intension(C_1)-X$ 包含于 $R_i$，$X$ 为多余的部分。显然，$X$ 中不含有私有权限，否则说明 $Extension(C_1)$ 为具有私有权限的角色，与定理 1 矛盾。假设 $\exists x\in X$，$x$ 不包含于概念格的其他某个节点的内涵中，此时说明概念格中不存在比 $C_1$ 大的节点，显然 $x$ 无法被消去，则 $c_x$ 不能被选为可用节点。证毕。

或者说，定理 2 表示在构造表达式的过程中，若需要对某个集合 $S$ 使用并操作，则 $S$ 引入的无效内容 $X$ 能够被现有集合表示，即 $X$ 能够被消去；否则，不能够使用 $S$。

可以将概念格中的节点 $c$ 根据其内涵与当前待化简角色 $r_1$ 的权限集合的交集情况分为 3 种：$A$：$Intension(c)\subseteq Per(r_1)$；$B$：$ntension(c)\cap Per(r_1)\neq\varnothing$，且 $Intension(c)\not\subset Per(r_1)$；$C$：$Intension(c)\cap Per(r_1)=\varnothing$。

如图 2 所示，对于以上 3 种节点类型，$A$ 和 $B$ 类型中的元素（形式概念）可以用来做并操作，$C$ 中的元素（形式概念）做相对差操作。在构造与 $Per(r_1)$ 相等的表达式时，$S_1:=\varnothing$，表达式 $E_1:=\varnothing$，具体操作步骤如下。

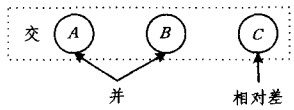


图 2 概念格中节点分类与操作符示意图

(1) $\forall x\in A$，如果 $S_1\neq Per(r_1)$ 且 $Intension(x)\not\subset S_1$，则 $S_1:=S_1\cup Intension(x)$，$E_1:=E_1\cup Extension(x)$。

(2) 若 $S_1\neq Per(r_1)$，则从 $B$ 中选择元素 $x$，且选择的元素 $x$ 满足以下两个条件：1) $(Per(r_1)-(S_1\cup Intension(x)))\subset(Per(r_1)-S_1)$；2) $(S_1\cup Intension(x))-Per(r_1)$ 能够在 $C$ 集合中消去。

若 $B$ 中找到满足以上两个条件的元素，则 $S_1:=S_1\cup Intension(x)$，选择 $Extension(x)$ 和消去相关的节点外延作为表达式 $E_1$ 的一部分。若找到与 $Per(r_1)$ 等值的集合表达式，则退出；否则在 $B$ 集合中遍历剩余元素。

4.2.3 算法框架描述

根据以上分析，可以将最小角色集的求解算法 MRS 描述如下。

**算法 3** MRS(R)（求解角色集合 $R$ 的最小角色集）

```
输入：角色集 R，权限集合 P
输出：最小角色集 Rs，表达式集合 Es
Begin
1. (Rt,Es):=DataPretreat(R,P)
2. PRt:= GetPrivacyPermissionSet(Rt)//找出 Rt 中的具有私有权限的角色集合 PRt
3. For all rx∈Rt−PRt do
4.    C0:= CreateContext(Rt−{rx})//创建形式背景 C0
5.    For all c∈CS(C0) do
6.        If Intension(c)⊆Per(rx) then add c to NSA EndIf
7.        If Intension(c)∩Per(rx)≠∅ and Intension(c)⊄Per(rx) then add c to NSB EndIf
8.        If Intension(c)∩Per(rx)=∅ then add c to NSC EndIf
9.    EndFor
10.   S1:=∅,S2:=∅,E1:=∅;// S1 表示当前集合，S2 表示待消去集合，E1 表示表达式集合
11.   For all xi∈NSA do
12.     If Intension(xi)⊄S1 and S1≠Per(rx) then S1:=S1∪Intension(xi),E1=E1∪Extension(xi) EndIf
13.       If S1=Per(rx) break EndIf
14.   EndFor
15.   For all yi∈NSB do
16.     S2:= Intension(yi)−S1
17.     If Intension(yi)∩S1≠S1 and CouldResolve(S2,NSC) then
18.         S1:=S1∪Intension(yi)
19.         E1=E1∪Extension(yi)−Y//Y 为消去集合
```

20.　EndIf
21.　If $S_1 \supset Per(r_x)$ break EndIf
22.　EndFor
23.　If $S_1 \supset Per(r_x)$ then add expression $Per(r_x)=E_1$ to $E_s$, $R_t := R_t - \{r_x\}$ EndIf
24.　NSA:=∅, NSB:=∅, NSC:=∅;
25.　EndFor
26.　Add ER to $E_s$
27. Return $R_t$, $E_s$
End

其中,第 6－8 行分别使用 *NSA*,*NSB*,*NSC* 依次代表概念格中的 *A*,*B*,*C* 3 种节点类型。第 17 行中 CouldResolve 方法表示多余的集合能够在 *NSC* 集合中消去。第 19 行中构造表达式时,根据外延中元素的交关系,可以直接将 *Extension*($y_i$)作为一个子表达式进行整体替换。

4.2.4　*算法分析*

在分析算法 MRS 的时间复杂度之前,首先对算法中调用的几个关键子算法的时间复杂度进行估计。*R* 为角色集合,*P* 为角色相关的权限集合,设 $|R|=n$, $|P|=m$。子算法 *DataPrement*(*R*,*P*)的时间复杂度为 $O((mn)^2)$;概念格构造算法的时间复杂度为 $O(m*n*|L|)$,其中 $|L|$ 为概念格中形式概念的个数,与 *m* 有关,可以视为常数。最坏情况下需要使用 *n* 次概念格构造算法,则算法中概念格构造的时间复杂度为 $O(m*n^2*|L|)$。综上分析,根据算法 MinimalRoleSet 的运行步骤,可得其时间复杂度为 $O((mn)^2)$。文献[7]给出了 RMP,$\delta$-approx RMP 和 MinNoise RMP 3 种算法在最坏情况下均不存在多项式时间的复杂度。文献[8]在文献[7]的基础上发现,对于一类实际问题,使用二分团覆盖(biclique cover)方法及基于格论的启发式方法,可以在多项式时间内得到结果。同时,文献[8]给出的时间复杂度为 $O(|E|^3 * V * \log|V|)$,其中 *E* 为二部图描述的用户权限关系中边的总数,*V* 为顶点数(用户数量)。由于本文算法的输入与文献[8]不同,有 $n \leqslant |V|$。按照普通的角色提取算法,即对于具有相同权限的用户,产生一个角色最多需要 $|V|^2$ 次操作,可以将该算法作为本文算法的一个附加步骤,则此时本文算法的复杂度仍然为 $O((mn)^2)$,小于文献[8]中的复杂度。另外,本文算法具有一定的普适性,可以对文献[8]中的结果进行进一步处理,从而得到优化的角色集合。

## 5　实验分析

角色工程中常用的数据集有两类:1)从实际系统中抽取的数据集[8];2)随机产生的数据集[29]。本文首先选用数据集 1)即文献[8]中的数据集作为实验数据集,数据集及实验对比结果如表 3 所列。该数据集包括 HP 公司用于管理外部业务连接的网络访问控制规则(americas_samll, americas_large, apj, emea)、US Veteran's Administration 中的访问控制数据集(healthcare)、Lotus Domino 服务器中的用户与访问控制配置信息(domino)以及 HP 公司的 IT 部门的访问控制系统(customer)等。其中,Users 指用户数;Permissions 指权限数;Edges 指系统中用户的权限分配情况,与权限的分布有关,影响角色数目;Role lower bound 指角色数目的下限;Real Rank 表示按照文献[8]中的方法得到的角色数目;Minimal Role Set 为本文算法的运行结果。

表 3　实验数据集及对应的最小角色集

| Data Set | Users | Permissions | Edges | Role lower bound | Real Rank | Minimal Role Set |
|---|---|---|---|---|---|---|
| americas large | 385 | 10127 | 185294 | 390 | 403 | 390 |
| americas small | 3477 | 1587 | 105205 | 172 | 203 | 172 |
| apj | 2044 | 1164 | 6841 | 453 | 455 | **449** |
| emea | 35 | 3046 | 7220 | 34 | 34 | 34 |
| healthcare | 46 | 46 | 1486 | 14 | 14 | **13** |
| domino | 79 | 231 | 730 | 20 | 20 | **19** |
| customer | 10021 | 277 | 45427 | 276 | 276 | 276 |
| firewall 1 | 365 | 709 | 31951 | 64 | 68 | **62** |
| firewall 2 | 325 | 590 | 36428 | 10 | 10 | 10 |

从表 3 可以看出,最小角色集的获取与测试数据集中的数据分布相关,即角色的提取与化简受到用户、权限及用户权限分布 3 个因素的影响。下面针对实验结果(角色化简情况)及相应数据集进行说明,其中,与本文进行对比的数据为表 3 中的 Role lower bound 列,且角色的数量越少,效果越好,越易于管理。

americas large:该数据集中权限数目较多,每个角色具有的权限均不相同,且角色包含的用户数少,权限数目多。每个角色中存在某些权限不可由其他角色的权限表示,经过分析,找不到被化简的角色。

americas small:与 americas large 数据集相比,该数据集中的权限数目大大减少,但是用户权限(Edges)总体数目较大,故不存在能够被化简的角色。

apj:用户权限分布较为均匀,存在一些角色可以被化简。具体地,4 个被化简的角色为 $r_9$, $r_{28}$, $r_{225}$, $r_{331}$,其表达式为:$Per(r_7) \cap Per(r_8) = Per(r_9)$,权限为{7};$Per(r_{26}) \cap Per(r_{27}) = Per(r_{28})$,权限为{69};$Per(r_{222}) \cap Per(r_{223}) = Per(r_{225})$,权限为{654};$Per(r_{330}) \cap Per(r_{327}) - Per(r_{328}) - Per(r_{329}) = Per(r_{331})$,权限为{889}。其中,$r_i$ 为数据集中的角色编号,*j*(*j* 为自然数)为数据集中的权限编号,以上编号均基于数据安全性考虑。下同。

emea:该数据集的用户数较少,权限数目及用户权限数目较多,提取到的角色中每个角色包含的权限数目较多,找不到能够被化简的角色。

healthcare:该数据集中存在某些用户集具有一些权限,且其他用户具有的权限数目较少。经过分析,存在一个角色

$r_7$ 可以被化简，其表达式为 $Per(r_1) \cap Per(r_6) = Per(r_7)$，权限为{2,29}。

domino：该数据集与 healchcare 类似。经过分析，存在一个角色 $r_{20}$ 可以被化简，其表达式为 $Per(r_{16}) \cap Per(r_8) \cup Per(r_1) = Per(r_{20})$，权限为{2,9}。

customer：该数据集的特征为用户具有的公共权限较少，不存在某些用户的权限完全相同，且每个角色包含的权限没有公共部分。经过分析，找不到被化简的角色。

firewall 1：该数据集中用户权限分布较为均匀，存在一些角色可以被化简。被化简的角色为 $r_{13}$ 和 $r_{15}$，其表达式为 $Per(r_{14}) \cap Per(r_{24}) = Per(r_{13})$，权限为{273,624}；$Per(r_{16}) - Per(r_{32}) = Per(r_{15})$，权限为{7,656}。

firewall 2：该数据集的特征与 customer 类似。经过分析，找不到被化简的角色。

为了说明本文算法的普适意义，利用文献[29]中提供的方法产生随机数据集进行测试。该数据集的产生方式如下：在满足用户、角色和权限数据间的约束关系（角色数量固定、用户及权限间数据满足高斯分布）下，通过设定每个角色具有的平均权限数量和数据偏差、每个角色包含的平均用户数及数据偏差，由相应的算法分别产生用户-权限数据（描述用户具有的权限情况）、角色-权限数据（描述用户具有的权限情况）、用户-角色数据（描述角色具有的角色情况）。本文产生的随机数据集（分别改变用户数、权限数以及角色数目）中，参数设置如下：角色包含的平均用户数为 5，偏差为 2；角色具有的平均权限数为 8，偏差为 2。实验对比结果如表 4 所列。

表 4 随机数据集下的实验结果

| Users | Permissions | Roles | Minimal Role Set |
|---|---|---|---|
| 30 | 30 | 5 | 3 |
| 30 | 30 | 10 | 10 |
| 30 | 30 | 15 | 14 |
| 30 | 30 | 20 | 19 |
| 30 | 30 | 25 | 25 |
| 50 | 50 | 5 | 5 |
| 50 | 50 | 10 | 10 |
| 50 | 50 | 15 | 15 |
| 50 | 50 | 20 | 19 |
| 50 | 50 | 25 | 25 |

表 4 中的实验数据随机产生，但是权限的概率密度满足高斯分布，这将导致某些数据集中的用户-权限数据分布不均匀，数据集中产生的角色无法被进一步化简或替换。本文使用概念格模型作为最小角色求解的工具，而概念格模型可以表示属性（文中为权限）的所有可能组合，故总能够找到与每个数据集对应的最小角色集合。

由以上分析可知，本文算法是实用和有效的，能够根据角色拥有的权限情况找出角色之间的关系，同时找出能够被化简的角色，降低访问控制策略的管理难度。

**结束语** 角色的提取与化简是角色工程中重要的研究内容。本文在给出最小角色集定义的基础上，介绍了一种快速的最小角色集求解算法，主要贡献有：1）在将角色集合转换为向量组的基础上，引入了代数中的极大线性无关组对角色向量进行化简，有效减小了待求解集合的大小，同时建立了向量间的线性表出与多重集中集合操作的等价转换关系；2）根据角色集合中的权限分布情况，在分析集合完备运算符特性的基础上，利用概念格模型，建立了一种快速的最小角色集求解算法。

本文的最小角色集定义建立在数学意义上，通过代数和集合的方法建立角色之间的关联关系，尚未考虑到角色的逻辑意义，进一步的研究工作将结合角色的逻辑意义，使得集合的化简更为自然，能够体现出被化简角色与最小角色集中的某些角色之间的内在联系。同时，我们在实践中发现，权限是建立在操作和资源的笛卡尔积基础之上的，与传统的存取访问控制矩阵相比，表示空间增大，可以对权限的表示进行进一步探索，找出一种建立在限定操作基础之上的权限表示方式。根据模式的包含关系，分别建立主体与操作、操作与资源间的联系，从而提高访问控制的动态性。

## 参 考 文 献


[1] COYNE E J. Role-engineering[C]// 1st ACM Workshop on Role-Based Access Control. 1996.

[2] SANDHU R, COYNE E J. Role based access control models [J]. IEEE Computer, 1996, 29(2): 38-47.

[3] ZHANG D, RAMAMOHANRAO K, EBRINGER T. Role Engineering using Graph Optimisation[C]// Symposium on Access Control Models and Technologies (SACMAT). 2007: 139-144.

[4] SCHLEGELMILCH J, STEENS U. Role mining with orca[C]// Symposium on Access Control Models and Technologies (SACMAT). ACM, June 2005.

[5] VAIDYA J, ATLURI V, WARNER J. Role Engineering via Prioritized Subset Enumeration[J]. IEEE Transactions on Dependable and Secure Computing, 2010, 7(3): 300-314.

[6] MOLLOY I, LI N, LI T, et al. Evaluating role mining algorithms [C]// Proceedings of the 14th ACM Symposium on Access Control Models and Technologies (SACMAT). 2009: 95-104.

[7] VAIDYA J, ATLURI V, GUO Q. The Role Mining Problem: Finding a Minimal Descriptive Set of Roles[C]// Symposium on Access Control Models and Technologies (SACMAT). 2007: 175-184.

[8] ENE A, HORNE W, MILOSAVLJEVIC N, et al. Fast exact and heuristic methods for role minimization problems[C]// The ACM Symposium on Access Control Models and Technologies. 2008.

[9] COLANTONIO A, DI PIETRO R, OCELLO A, et al. Taming role mining complexity in RBAC[J]. Computers & Security, 2010, 29(5): 548-564.

[10] MITRA B, SURAL S, VAIDYA J, et al. A Survey of Role Mining[J]. ACM Computing Surveys, 2016, 48(4): 1-37.

[11] JAFARIAN J H, TAKABI H, TOUATI H, et al. Towards a General Framework for Optimal Role Mining: A Constraint Satisfaction Approach[C]// The ACM Symposium. 2015: 211-220.

[12] WU J W, ZHANG Y, LI R X, et al. Role Updating for Assignments[C]// Symposium on Access Control Models and Technologies. 2010: 89-98.

[13] AHMED S, OSBORN S L. A system for risk awareness during role mining[C]// ACM Symposium on Access Control MODELS and Technologies. ACM, 2014: 181-184.

[14] LU H, VAIDYA J, ATLURI V. Optimal Boolean matrix decomposition: Application to role engineering[C]// IEEE Iternational Conference on Data Engineering. IEEE Computer Society, 2008: 297-306.

[15] LI R, LI H, WANG W, et al. RMiner: a tool set for role mining [C]// ACM Symposium on Access Control MODELS and Technologies. 2013: 193-196.

[16] COLANTONIO A, DI PIETRO R, OCELLO A, et al. A formal framework to elicit roles with business meaning in RBAC systems[C]// Proceedings of the 14th ACM Symposium on Access Control Models and Technologies. 2009: 85-94.

[17] MA X P, LI R X, LU Z D. Role Mining Based on Weights[C]// Symposium on Access Control Models and Technologies. Pittsburgh, Pennsylvania, USA, 2010: 65-74.

[18] WU M Y. Activities and Event-Driven-Based Role Engineering [C]// 2012 Sixth International Conference on Genetic and Evolutionary Computing (ICGEC). IEEE, 2012: 550-553.

[19] ZHANG L, ZHANG H L, HAN D J, et al. Theory and Algorithm for Roles Minimization Problem in RBAC Based on Concept Lattice[J]. Acta Electronica Sinica, 2014, 42(12): 2371-2378. (in Chinese)
张磊,张宏莉,韩道军,等. 基于概念格的 RBAC 模型中角色最小化问题的理论与算法[J]. 电子学报, 2014, 42(12): 2371-2378.

[20] YU Q Q. Role Hierarchy Mining For Role Based Access Control [D]. Harbin: Harbin Institute of Technology, 2013. (in Chinese)
蔚清琴. RBAC 中分层角色挖掘算法研究[D]. 哈尔滨:哈尔滨工业大学, 2013.

[21] MI X M. Role Mining Algorithm Based on Evoloutionary Algorithms[D]. Beijing: Beijing Jiaotong University, 2014. (in Chinese)
米秀明. 基于进化算法的角色挖掘算法[D]. 北京:北京交通大学, 2014.

[22] YE W. Role Mining and Logical Programming Model in Access Control[D]. Wuhan: Huazhong University of Science and Technology, 2014. (in Chinese)
叶威. 访问控制中的角色挖掘与逻辑编程模型研究[D]. 武汉:华中科技大学, 2014.

[23] FENG D G, ZHANG M, LI H. Big Data Security and Privacy Protection[J]. Chinese Journal of Computer, 2014, 37(1): 246-257. (in Chinese)
冯登国,张敏,李昊. 大数据安全与隐私保护[J]. 计算机学报, 2014, 37(1): 246-257.

[24] FANG B X, JIA Y, LI A P, et al. Privacy Preservation in Big Data: A Survey[J]. Big Data, 2016, 2(1): 1-18. (in Chinese)
方滨兴,贾焰,李爱平,等. 大数据隐私保护技术综述[J]. 大数据, 2016, 2(1): 1-18.

[25] 北京大学数学系几何与代数教研室代数小组. 高等代数(第二版)[M]. 北京:高等教育出版社. 1988: 6.

[26] SYROPOULOS A. Mathematics of Multisets[M]// Multiset Processing. Spinger Berlin Heidelberg, 2000: 347-358.

[27] LOEB D. Sets with a negative number of elements[J]. Advances in Mathematics, 1992, 91(91): 64-74.

[28] GANTER B, WILLE R. Formal concept analysis: Mathematical foundations[M]. Berlin: Springer-Verlag, 1999.

[29] ZHANG D, RAMAMOHANARAO K, ZHANG R. Synthetic data generation for study of role engineering[EB/OL]. http://www.cs.mu.oz.au/~zhangd/roledata.

(上接第 99 页)

[11] BARAKAT C, KALLA A, SAUCEZ D, et al. Minimizing bandwidth on peering links with deflection in named data networking [C]// Third International Conference on Communications and Information Technology. IEEE, 2013: 88-92.

[12] MA W, YAO Y, FAN H L, et al. A virtual network architecture for private cloud based on Openflow[J]. Journal of Beijing Jiaotong University (Natural Science Edition), 2015, 39(5): 15-21. (in Chinese)
马威,姚远,范慧莉,等. 基于 Openflow 的私有云虚拟网络结构设计[J]. 北京交通大学学报(自然科学版), 2015, 39(5): 15-21.

[13] DAI D, WEI J, WANG L. Wireless Mesh Network Channel Assignment Scheme Based on SIR Conflict Graph and Maximal Independent Set[J]. Natural Science Journal of Xiangtan University, 2016, 38(2): 109-113. (in Chinese)
戴冬,卫娟,王磊. 基于 SIR 冲突图和最大独立集的无线 Mesh 网络信道分配方案[J]. 湘潭大学自科学报, 2016, 38(2): 109-113.

[14] ZHANG T, LI T S, GE Z H. Research on Wireless Mesh Network QoS Based on M/M/n/m Model under Non-preemptive Limited-priority[J]. Computer Science, 2014, 41(8): 135-138. (in Chinese)
张挺,李陶深,葛志辉. 非强占有限优先权 M/M/n/m 模型的无线 Mesh 网络 QoS 研究[J]. 计算机科学, 2014, 41(8): 135-138.

[15] ZHAI H B, JIANG H, SUN Y, et al. A Node-Link Based Cache Deployment Algorithm for P2P Traffic in ISP Networks[J]. Journal of Computer Research and Development, 2013, 50(1): 122-135. (in Chinese)
翟海滨,蒋海,孙毅,等. 一种基于点路结合的骨干网 P2P 缓存部署方法[J]. 计算机研究与发展, 2013, 50(1): 122-135.

[16] HUANG C Y, RAMANATHAN P. Network Layer Support for Gigabit TCP Flows in Wireless Mesh Networks [J]. IEEE Transactions on Mobile Computing, 2015, 14(10): 2073-2085.

[17] ZHENG Y, HE S B, ZHANG X Y, et al. A Game-based Channel assignment for Wireless Mesh Networks [J]. Journal of Chongqing University of Technology (Natural Science), 2013, 27(4): 90-95. (in Chinese)
郑鹏宇,何世彪,张馨月,等. 一种基于博弈论的无线网状网络信道分配算法[J]. 重庆理工大学学报(自然科学), 2013, 27(4): 90-95.