访问控制策略中信息流的最优化去环方法

杨 智[1,2,3,4],段▮毅[1],金舒原[1,4],殷丽华[1,4],郭 莉[1,4]
(1.中国科学院计算技术研究所,北京 100190;2.解放军信息工程大学电子技术学院,河南郑州 450004;
3.中国科学院研究生院,北京 100039;4.信息内容安全技术国家工程实验室,北京 100190)

摘 要: 最优化去除访问控制中信息流的环路是许多重要信息系统向多级安全系统迁移时保证系统可用性的重要前提.证明了该问题是NP难题,提出了基于动态规划的最优解算法,利用遗传算法搜索近似最优解.复杂度分析和实验结果表明,对于小规模环境,最优解方法能较快地找出最优解;对于大规模环境,近似最优解算法能有效找出近似解.

关键词: 访问控制策略; 信息流; 环路; NP难题; 动态规划; 遗传算法
中图分类号: TP309 **文献标识码:** A **文章编号:** 0372-2112 (2011) 07-1530-08

Methods for Optimal Eliminating Cycles in Information Flow of Authorization Policies

YANG Zhi[1,2,3,4], DUAN Mi-yi[1], JIN Shu-yuan[1,4], YIN Li-hua[1,4], GUO Li[1,4]
(1. *Institute of Computing Technology, Chinese Academy of Sciences, Beijing* 100190, *China*;
2. *Institute of Electronic Technology, Information Engineering University, Zhengzhou, Henan* 450004, *China*;
3. *Graduate School of Chinese Academy of Sciences, Beijing* 100039, *China*;
4. *National Engineering Laboratory for Information Security Technologies, Beijing* 100190, *China*)

Abstract: Optimal eliminating cycles in Information flow of authorization policies is an important prerequisite to the success of migrating important information systems to multi-level security systems. This paper firstly proves it is a NP-hard problem, then presents optimization algorithm based on dynamic programming and approximate optimization algorithm based on genetic algorithm. Computational complexity analysis and experiments show that the former is fast to find the optimal solution in small-scale environment, the latter is effective to find the approximate optimal solution in large-scale environment.

Key words: access control policy; information flow; cycle; NP-hard; dynamic programming; genetic algorithm

1 引言

等级保护是我国信息安全保障工作的一项基本制度和根本方法.全国重要信息系统的评定等级工作已基本完成,目前的主要任务是系统安全整改建设,现实的情况是,很多重要电子政务和商务系统,由于其本身的重要程度被要求达到三级或以上级别,且三级以上信息系统要求必须实现强制访问控制,主要是要求信息单向流动以提供机密性或完整性保护.然而,这些已存在系统的访问控制由于历史等原因并不是强制访问控制(常见的是自主访问控制、基于角色的访问控制等),需要实现向强制访问控制系统的迁移.目前主要有两种迁移方法:一是通过分析信息系统原有工作流程来为主客体添加合理的敏感标记,根据主客体标记之间的支配关系决定访问权限,从而直接实现强制访问控制;二是用原有访问控制模型模拟和间接表达强制访问控制,主要是通过修改少量的具体的原有授权策略来完成.为了最大程度体现原有系统的访问控制逻辑以保证迁移后系统的可用性,从信息流图角度来看,两种方法的有效性的基础保证是能够最优化地去除访问控制策略中信息流的环路,依据或者参考得到的最优单向信息流图,理论上才有可能合理地为主客体分配标记或者合理地修改相关策略来实现强制访问控制.在实际系统中,去除环路的解可能有多个,为了最大程度优先维护原系统较重要的访问控制策略不被改变,可能需要考虑每条策略有权重情况下的最优化去环问题,修改既能消除环且权重和最小的授权策略集以实现信息的单向流动.

已知判定一个有向图是否存在环路是一个P类问

收稿日期:2010-08-13;修回日期:2010-11-05
基金项目:国家自然科学基金(No.61070186);国家863高技术研究发展计划(No.2009AA01Z438,No.2006AA01Z457);国家973重点基础研究发展计划(No.2007CB311100)

题.然而给定整数 k,判定一个有向图能否在删除不多于 k 条边后不存在环路是一个 NP 完全问题[1](参见定义 5),后一个问题与我们这里的问题相似.本文研究了访问控制策抢中信息流的最优化去环问题,分析了该问题的复杂度,提出了最优解算法和近似最优解算法,并通过实验验证了算法的性能.

2　相关工作

经典的 BLP 保密性模型[2]是许多信息系统安全评测标准的制定依据和理论基础.为了提高可用性,BLP 模型引入可信实体概念,并使可信主体可通过一定范围的安全级调整写访问低安全级客体[3].有些研究试图进一步解决可信主体安全隐患,例如,文献[4]提出一种基于可信状态的多级安全模型,运用可信计算理论和技术,引入可信度和可信状态测量函数,以提高模型的抗攻击能力.文献[5]提出一种具有可信度特征的多级安全模型,增加主客体的可信度标记和可信度评估函数,建立对可信主体的约束机制.概括来说,已有研究偏重于对 BLP 模型的研究和改进,很少见到如何从旧系统向强制访问控制系统迁移方法方面的研究.

也有不少文献研究了不同访问控制模型之间的表达方法.例如利用 RBAC 表达和构建 DAC[6,7]、MAC[6,8]、中国墙策略[9].也有研究使用 MAC 构建 RBAC[10];用 BLP 框架解释中国墙模型[11]等.总的说来,策略之间的互表达方法有了较广泛研究,然而缺乏对维护可用性的策略转换研究.

3　问题定义及复杂度分析

3.1　访问控制策略中信息流的最优化去环问题描述

访问控制矩阵是最常见的一种访问控制策略表示方法,本文主要针对访问控制矩阵来研究信息流的最优化去环问题.在访问控制矩阵中,行表示主体,列表示客体,矩阵中的元素表示相应主体访问相应客体的权限.权限集合 $P:\{r,a,w,e\}$,分别表示读权限(r),写权限(a),读/写权限(w)和空权限(e).为了保证在访问控制矩阵中信息流单向性,有时需要修改一些授权(将某些元素的值如 r,a,w 修改为 e),这里可以理解为删除授权策略.为了最大程度优先维护原系统较重要的访问控制策略不被改变,当去除环的解有多个时,需要考虑策略的重要性,使得既能消除环,又能使被修改的授权策略集的权重和最小化,为此引入加权访问控制矩阵概念.

定义 1　(加权访问控制矩阵)　对于一个有 m 个主体,n 个客体的授权系统,加权访问控制矩阵 $\boldsymbol{A}=(p_{ij},h_{ij})_{m\times n}$,其中 $p_{ij}\in P$,$h_{ij}\in Z^+$,分别表示主体 i 对客体 j 的权限和本条策略的权重.

策略的权重反映了该策略的重要程度或价值.从信息流角度来看,访问控制策略反映了信息在主体之间、客体之间以及主体和客体之间的信息流动规定,可以用有向图表示这种信息流动情况.

定义 2　(加权访问控制矩阵导出的加权有向图)　对于加权访问控制矩阵 $\boldsymbol{A}=(p_{ij},h_{ij})_{m\times n}$,主体集合为 $S=\{s_1,s_2,\cdots,s_m\}$,客体集合为 $O=\{o_1,o_2,\cdots,o_n\}$,其导出的加权有向图是 $G_A=(V,E)$,其中顶点集合 $V=S\cup O$,有向边集合 $E=\{(s_i,o_j)\mid p_{ij}=a,w\}\cup\{(o_j,s_i)\mid p_{ij}=r,w\}$,对于 $(s_i,o_j)\in E$ 和 $(o_j,s_i)\in E$,权值 $w(s_i,o_j)$和权值 $w(o_j,s_i)$为 h_{ij}.对于 $E'\subseteq E$,$w(E')=\sum_{e\in E'}w(e)$.

在上述两个定义基础上,我们给出如下的访问控制策略中信息流的最优化去环问题形式化定义:

定义 3　(访问控制策略中信息流的最优化去环问题 Optimal Eliminating Cycles in Authorization Policies,简称 OECAP)　给定一访问控制策略集 Γ,加权访问控制矩阵 $\boldsymbol{A}=(p_{ij},h_{ij})_{m\times n}$,$\boldsymbol{A}$ 导出的加权有向图 $G_A=(V,E)$,令 $\Psi=\{E'\mid E'\subseteq E$ 且 $G=(V,E\text{-}E')$不存在路径长大于 2 的环路$\}$,$\delta=\min_{E'\in\Psi}w(E')$,$\gamma=\operatorname{argmin}_{E'\in\Psi}w(E')$,则称访问控制策略集 Γ 是 δ 代价-单向的,γ 是 Γ 的信息流不存在环路的一个最优解.

如果 $\delta=0$,表示访问控制策略集 Γ 的信息流不存在环路;如果 $\delta>0$,表示为了实现信息流单向性,需要付出最小代价 δ,可以通过删除 γ 集合中边所对应的访问控制策略来实现.

3.2　OECAP 计算复杂度分析

为了研究 OECAP 的计算复杂性,我们将其表述为如下判定问题

定义 4　(OECAP 判定版本)　给定一个访问控制策略集 Γ,其加权访问控制矩阵 $\boldsymbol{A}=(p_{ij},h_{ij})_{m\times n}$,$\boldsymbol{A}$ 导出的加权有向图 $G(\boldsymbol{A})=(V,E)$,令 $\delta\geqslant 0$,判定是否存在 $E'\subseteq E$,使得 $G=(V,E\text{-}E')$不存在路径长大于 2 的环路且 $w(E')\leqslant\delta$,问题记作 $DOECAP(\boldsymbol{A},\delta)$.

下面,我们用归约的方法分析 $DOECAP(\boldsymbol{A},\delta)$的复杂性.先看一个 NP 完全问题[1],问题描述如下:

定义 5　(反馈有向边集合 Feedback Arc Set,简称 FAS)　给定一个有向图 $G=(V,E)$,正整数 $K\leqslant|E|$.判定是否存在一个 $E'\subseteq E$,$|E'|\leqslant K$,使得 $G=(V,E\text{-}E')$不存环路,问题记作 $FAS(G,K)$.

定理 1　$DOECAP$ 是 NP 难的.

证明:我们通过 $FAS(G,K)\propto_P DOECAP(\boldsymbol{A},\delta)$证明 $DOECAP$ 是 NP 难的.给定一个 FAS 问题实例 $FAS(G_{fas},K)$,$G_{fas}=(V_{fas},E_{fas})$.构造对应的 $DOECAP(\boldsymbol{A},\delta)$实例过程主要是构造 $G_A=(V_A,E_A)$的过程,初始化 $V_A=$

$\varnothing$, $E_A=\varnothing$, 对于 G_A 中每个顶点 $v_i\in V_{fas}$, 在 G_A 中建立两个与之相对应的顶点 s_i 和 o_i, 即 $V_A=\{s_i,o_i\}\cup V_A$, 同时令 $E_A=\{(o_i,s_i)\}\cup E_A$, $w(o_i,s_i)=+\infty$. 对于 G_{fas} 中每条边 $(v_i,v_j)\in E_{fas}$, 在 E_A 中添加一条边 (s_i,o_j), 即 $E_A=\{(s_i,o_j)\}\cup E_A$, 同时令 $w(s_i,o_j)=1$. 根据 G_A 和定义 2, 得出 $\boldsymbol{A}=(p_{ij},h_{ij})_{m\times n}$, 其中 $m=n=|V_{fas}|$,

$$(p_{ij},h_{ij})=\begin{cases}(a,w(s_i,o_j)), & (s_i,o_j)\in E\\ (r,w(o_i,s_i)), & (o_i,s_i)\in E\\ (e,0), & \text{其他}\end{cases}$$

令 $\delta=K$. 构造 $DOECAP(\boldsymbol{A},\delta)$ 实例显然可在多项式时间内完成. 图 1(a)是一个 FAS 问题中的 $G_{fas}=(V_{fas},E_{fas})$, 图 1($b$)是针对 G_{fas} 构造出的 $G_A=(V_A,E_A)$.

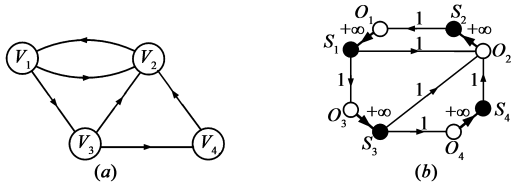

图1 FAS实例的G_{fas}及其对应的$DUOAP$实例的G_A

事实上, 若存在一个 $E_{fax}'\subseteq E_{fax}$, $|E_{fax}'|\leqslant K$, 使得 $G_{fas}=(V_{fas},E_{fas}-E_{fax}')$ 不存环路, 令 $E_A'=\{(s_i,o_j)|(s_i,o_j)\in E_A$ 且 $(v_i,v_j)\in E_{fas}'\}\subseteq E_A$, 则 $G_A=(V_A,E_A-E_A')$ 必定不存在环路, 同时由于 $w(E_A')\leqslant K$, $\delta=K$ 则 $w(E_A')\leqslant\delta$. 反之若存在 $E_A'\subseteq E_A$, $w(E')\leqslant\delta$, 使得 $G_A=(V_A,E\text{-}E_A')$ 不存在路径长大于 2 的环路, 由 G_A 构造方法知 G_A 中不可能出现路径长为 2 的环路, 则 $G_A=(V_A,E\text{-}E_A')$ 不存在任何环路. 由于 $w(o_i,s_i)=+\infty$, 考虑到 $w(E')\leqslant\delta$, 则有向边 $(o_i,s_i)\notin E_A'$, 令 $E_{fax}'=\{(v_i,v_j)|(v_i,v_j)\in E_{fax}$ 且 $(s_i,o_j)\in E_A'\}$, 则 $G_{fax}=(V_{fax},E_{fax}-E_{fax}')$ 必定不存在环路. 考虑到 E_{fax}' 中边和 E_A' 中边的一一对应且 $w(s_i,o_j)=1$, 则 $|E_{fax}'|\leqslant\delta$, 即 $|E_{fax}'|\leqslant K$. 因此实例 $FAS(G_{fas},K)$ 存在一个解当且仅当对应的 $DOECAP(\boldsymbol{A},\delta)$ 存在一个解, 即 $FAS(G,K)\propto_P DOECAP(\boldsymbol{A},\delta)$, 定理 1 得证.

OECAP 是一个最优化问题, 由于 DOECAP 是 NP 难的, 不存在多项式时间算法, 针对小规模环境, 我们仍然考虑最优解算法; 针对大规模环境, 考虑该问题的近似最优解算法.

4 算法实现

4.1 基于动态规划的最优解算法

最优解算法搜索整个解空间, 选出权重和最小的访问控制策略子集, 删除该子集可以保证整个访问控制策略集的单向性. 算法输入是加权访问控制矩阵 $\boldsymbol{A}$, 算法输出最小代价 δ 和需要消除的边集合 γ. 算法包括两个过程: 过程一是找出加权访问控制矩阵导出的加权有向图中所有路径长度大于 2 的环路; 过程二是找出权重和最小的要删除边集合, 边集合包含过程一得到的每个有向环的至少一条边.

4.1.1 枚举所有长大于 2 的环

我们用文献[12]给出的经典算法来寻找简单有向环, 算法预先指定各个顶点的序号, 执行时首先求出有向图的各个强连通分量, 然后按序号从小到大的顺序选择根顶点, 从每个根结点开始在根结点所在的连通分量中进行深度优先搜索得到环路.

算法通过在路径扩展中不加入当前路径已出现的结点来保证得到简单路径; 通过在路径扩展中只加入比根顶点序号值大的顶点保证每个环路只考虑一次; 通过持续阻塞一些顶点来保证不是环的简单路径只被考虑一次, 在第一次遍历这些顶点时, 若从这些顶点到根顶点存在路径, 则该路径和当前路径必定有交叉.

算法的时间复杂度是 $O((n+e)(c+1))$, 其中 n 是图中顶点个数, e 是边的个数, c 是简单环的个数.

4.1.2 最优化方法找出删除边集合

对于过程二, 假设给定加权有向图是 $G_A=(V,E)$, 通过 4.1.1 中算法得到环集合 $C=\{c_i|c_i=(s_{i1}o_{i1}s_{i2}o_{i2}\cdots s_{ip}o_{ip})$, 其中 $p\geqslant 2$, $(o_{ip},s_{i1})\in E$, 且 $(s_{ij},o_{ij})\in E$, $(o_{ij},s_{i(j+1)})\in E$, $1\leqslant j\leqslant p\}$, 我们用 $E_C=\{e_1,e_2,\cdots,e_m\}$ 表示环集合 C 所包含的边集合, 用 $e\in c$ 表示边 e 是环 c 包含的边, 用 C_e 表示包含边 e 的环集合. 子过程二求解的问题变为: 求 m 元 0-1 向量 $(\lambda_1,\lambda_2,\cdots,\lambda_m)$, 其中 $m=|E_C|$, 使得 $\sum_{i=1}^{m}\lambda_i w(e_i)$ 达到最小值, 且对任意 $c\in C$, 存在 $e_j\in E_C$, 满足 $e_j\in c$ 和 $\lambda_j=1$.

对于该问题我们用备忘录方法求解. 备忘录方法是动态规划方法的一种, 基本思想是将待求解问题分解成若干个子问题, 先求解子问题, 然后从这些子问题的解得到原问题的解. 求解过程中保存已解决的子问题的答案, 可以避免大量重复计算. 备忘录方法采用自顶向下的方法求解子问题, 只求解递归过程中遇到的子问题, 而不是将子问题空间中所有子问题都求解一次.

过程二求解问题具有最优子结构性质, 即问题的最优解包含了其子问题的最优解, 证明过程如下.

设 $(\lambda_1,\lambda_2,\cdots,\lambda_m)$ 是所给子过程二求解问题的一个最优解, 则 $(\lambda_2,\cdots,\lambda_m)$ 是下面相应子问题的一个最优解. min: $\sum_{i=2}^{m}\lambda_i w(e_i)$, subject to: 对任意 $c\in C^-$, 存在 $e_j\in E_C-\{e_1\}$, 满足 $e_j\in c$ 和 $\lambda_j'=1$, 其中 $C^-=C-\lambda_1 C_{e_1}$, 定义 $0C_{e_1}=\varnothing$, $1C_{e_1}=C_{e_1}$.

若不然, 设 $(\lambda_2',\cdots,\lambda_m')$ 是上述子问题的一个最优

解,而 $(\lambda_2, \cdots, \lambda_m)$ 不是它的最优解,由此可知 $\sum_{i=2}^{m} \lambda_i' w(e_i) < \sum_{i=2}^{m} \lambda_i w(e_i)$,则 $(\lambda_1, \lambda_2', \cdots, \lambda_m')$ 是子过程二求解问题的更优解.一方面 $w(\lambda_1) + \sum_{i=2}^{m} \lambda_i' w(e_i) < w(\lambda_1) + \sum_{i=2}^{m} \lambda_i w(e_i)$.另一方面,若 $\lambda_1 = 0$,则 $C^- = C$,由子问题的约束条件知对任意 $c \in C$,存在 $e_j \in E_C - \{e_1\}$,满足 $e_j \in c$ 和 $\lambda_j' = 1$;若 $\lambda_1 = 1$,则对任意 $c \in C - C^-$,有 $e_1 \in E_C$,满足 $e_1 \in C - C^-$ 和 $\lambda_1 = 1$.所以 $(\lambda_1, \lambda_2', \cdots, \lambda_m')$ 是问题的最优解,此为矛盾.

考虑过程二的子问题:min: $\sum_{i=k}^{m} \lambda_i w(e_i)$, subject to:对任意 $c \in C', C' \subseteq C$,存在 $e_j \in E_C - \{e_1, e_2, \cdots, e_k\}$,满足 $e_j \in c$ 和 $\lambda_j = 1$,记子问题的最优值为 $sub(k, C')$,建立递归式如下:$sub(k, C') =$

$$\begin{cases} \min\{sub(k+1, C'), sub(k+1, C' - C_{e_k}) + w(e_k)\}, & C_{e_k} \cap C' \neq \varnothing \\ sub(k+1, C'), & C_{e_k} \cap C' = \varnothing, C' \neq \varnothing \\ 0, & C' = \varnothing \end{cases}$$

而 $sub(m, C') = \begin{cases} w(e_m), & C' \subseteq C_{e_m}' \text{ 且 } C' \neq \varnothing \\ +\infty, & C' \not\subset C_{e_m} \text{ 且 } C' \neq \varnothing \\ 0, & C' = \varnothing \end{cases}$

根据递归式,算法流程如下.

算法 1　备忘录方法求最优的删除边集合

输入:加权图 G_A,环集合 C

输出:单向性代价 δ,删除边集合 γ

全局变量:C 所包含的边的集合 E_C,E_C 的大小 m,存储子问题最优值的二维整型数组 sub

```
Main(G_A, C, δ, γ)
{E_C = GetEdegeSet(G_A, C); m = |E_C|;
    Initialize(sub);   – 初始化 sub 各元素值为 – 1;
    δ = Recur(1, C);   – 求单向性代价;
    γ = Solution();}   – 求删除边集合;
Recur(k, C')
{if ((C' = ∅))  return 0;
if ((C' – (C_{e_k} ∪ C_{e_{k+1}} ... ∪ C_{e_m}) ≠ ∅)) return +∞; – 子问题无解;
if (k = m)  return w(e_m); – 已递归到最后一条边 e_m;
if (sub(k, C') ≠ – 1)  return sub(k, C'); – 子问题已求出;
if (C_{e_k} ∩ C' ≠ ∅) – 考虑选和不选两种情况;
return sub(k, C') = min{sub(k + 1, C'), sub(k + 1, C' – C_{e_k}) +
w(e_k)}
else return sub(k, C') = sub(k + 1, C');}
Solution()
{C' = C; γ = ∅;
    for (int k = 1; k < m to; k ++)
    if (sub(k, C') ≠ sub(k + 1, C')) γ = r ∪ {e_k}; C' = C' – C_{e_k};
    – 若 sub(k, C') 和 sub(k + 1, C') 不等,删除边集合中有 e_k
    if (sub(m, C') = w(e_m)) γ = r ∪ {e_m};}
```

算法 1 中二维整型数组 sub 用 E_C 中边的索引 k 和环集 C' 的编码来索引,C' 用二进制位串编码,串长等于环的个数,串中每位对应一个环,位值等于 1 表示 C' 包含相应的环,位值等于 0 表示不包含相应的环.

设 m 是所有的环用到的不同边的个数,c 是环的个数,在最坏情况下,所有 $sub(k, C')$ 至少求解一次,则至多需要 $O(m2^c)$ 计算时间,同时由递归过程看出子问题的计算次数又不会超过 $O(2^m)$,所以最坏情况下算法时间复杂性是 $O(\min\{m2^c, 2^m\})$.算法在递归中通过判断子问题所能消除的环集是否包含环集 C' 要求的边集来剪枝不可能有解的子问题,并且不求解 C_{e_k} 和 C' 交集为空的子问题 $sub(k, C')$,可以有效提高算法效率.容易看出最好情况下算法计算时间复杂性为 $O(m)$.

由于要存储 $sub(k, C')$,算法的空间复杂度是 $O(m2^c)$,这里用哈希表存储 sub 数组的部分内容以降低空间复杂度,即只保存部分已求子问题,设哈希表 M_H 长为 ub,哈希函数 $H(k, C') = b(k \bmod u) + H(C')$,其中 u 是边的模数,b 是同余边的哈希地址块大小,$b(k \bmod u)$ 部分可以在一定程度上抑制哈希地址冲突,哈希函数 $H(C')$ 用折叠法构造,即将 C' 的编码分割成位数均为 $c/\log_2 b$ 的几部分(最后一部分的位数可以不同),然后取这几部分的 1 位叠加和(舍去进位)作为哈希地址,它的特点是计算简单快速,哈希冲突的办法是链地址法,即将发生哈希地址冲突的记录存储在同一线性链表中,哈希地址指向的位置存放链表的头指针,再建一个具体存储子问题解的区域 M_R,链表的分配 M_R 中进行,M_H 和 M_R 的大小可根据实际情况设置.

4.2　基于遗传算法的近似解求解

对于较大规模的授权访问控制系统,我们通过为主客体添加合适标记来进行求解,试图在较合理时间内求出近似解甚至最优解.因为 BLP 模型要求信息向高安全级单向流动,可寻找一个 BLP 授权系统去逼近和模仿访问控制策略集 Γ 的授权系统,逼近程度可作为判断 Γ 单向性的参考.因此给定一个访问控制策略集 Γ,我们寻找一种对主客体的标记,使得由主客体的标记导出的访问控制矩阵与 Γ 的加权访问控制矩阵的距离最小(参见定义 6,7).如果距离为 0,则 Γ 必定是单向的;否则,距离就是 Γ 实现单向性的最小代价.

算法包括加权有向图化简和用遗传算法搜索主客体的范畴集两个过程.

4.2.1　加权有向图的化简

化简包括顶点删除和路径压缩两个办法.设加权有向图 $G(\boldsymbol{A}) = (V, E)$,化简如下:一是反复删除出度或入度为 0 的顶点及其关联边,这些点不会是图中任何环上的点,实际上若图中没有环,则化简图是空图,删

除 $G(\boldsymbol{A})$中顶点时相应地删除 $\boldsymbol{A}$ 中对应的行或列;二是反复压缩没有分叉的路径,如果 $a,b,c,d\in V, a\neq c, b\neq d,(a,b),(b,c),(c,d)\in E, b,c$ 的出度和入度都为1,则令 $V=V-\{b,c\}, E=E\cup\{(a,d)\}-\{(a,b),(b,c),(c,d)\}, w(a,d)=\min\{w(a,b),w(b,c),w(c,d)\}$,也就是说对于一条没有分叉的路径,如果它处于一个环中,这段路径消除环能所需的最小代价是 $w(a,d)$,即删除路径上权值最小的边.相应的对 $\boldsymbol{A}$ 进行修改.在顶点删除过程中,用邻接表存储 $G(\boldsymbol{A})$,用一数组记录所有顶点当前的出入度,先反复删除入度为0的顶点及边,然后将化简后的图用逆邻接表存储,反复删除出度为0的顶点及边,其计算时间为 $O(|V|+|E|)$.在路径压缩过程中,先用十字链表存储 $G(\boldsymbol{A})$,用一数组记录所有顶点当前的出入度,顺序查找出入度为1的顶点并合并符合条件的路径,若考虑初始化所有顶点出入度时间,其计算时间为 $O(|V|+|E|)$.

我们没有在最优解算法中删除出度或入度为0的顶点,因为文献[12]的算法执行时首先求出有向图的各个强连通分量,这必然会删除这些顶点.也没有在最优解算法中反复进行边的合并,因为文献[12]的算法搜索每条路径只有一次,路径压缩没有显出优势.而在这里可以减少搜索的状态数目.

4.2.2 基于标记的遗传算法搜索

遗传算法可用于很多复杂的搜索优化问题.算法首先产生候选解决方案的种群,然后通过自然选择使这些解决方案进化,从而使得不好的解决方案趋于淘汰,好的解决方案存活并继续繁殖,不断重复这个过程,算法就得到了最优的解.遗传算法求解该问题的描述如下.

(1)**用染色体表示标记分配** 遗传算法中对问题的解以编码形式呈现,一个解对应一条染色体.这里将每个实体分配的范畴集看成染色体上的一个相应位置的基因,基因用定长二进制位串表示,串长是整个范畴集包含的范畴个数,串上每一位对应一个范畴,取值1/0表示该实体是否属于该范畴.

(2)**适应度函数和选择方法** 适应度函数反映了个体的适应能力.其值大小决定某些个体是繁殖还是消亡.为了更好的描述问题和函数,先给出如下定义:

定义6 (标记系统导出的访问控制矩阵) 设 $P=\{r,a,w,e\}$代表权限集合,对于一个有 m 个主体和 n 个客体的安全系统,标记集合记为 $K=\{k_1,k_2,\cdots,k_s\}$, $\forall k_i,k_j\in K, k_i\cap k_j=\varnothing$;所有主体的标记记为向量 $\boldsymbol{p}=(ls_1,ls_2,\cdots,ls_m)^{\mathrm{T}}$,其中 $ls_i\subseteq K$,它是为第 i 个主体分配的标记元组;所有客体的标记记为向量 $\boldsymbol{q}=(lo_1,lo_2,\cdots,lo_n)^{\mathrm{T}}$,其中 $lo_j\subseteq K$,它是为第 j 个客体分配的标记;则标记系统$(\boldsymbol{L},\boldsymbol{p},\boldsymbol{q})$导出的访问控制矩阵 $\boldsymbol{M}$ 定义为 $\boldsymbol{M}=\boldsymbol{p}\boldsymbol{q}^{\mathrm{T}}=(x_{ij})_{m\times n}$,其中若 $ls_i=lo_j$,有 $x_{ij}=w$;若 $ls_i\subset lo_j$,有 $x_{ij}=a$;若 $ls_i\supset lo_j$,有 $x_{ij}=r$;其他情况,有 $x_{ij}=e$.

范畴集合 K 的子集的支配关系可以表达读/写/读写/空权限,密级不能表达空权限,为了降低问题搜索复杂度,在标记主客体时只考虑为其添加范畴集.

定义7 (加权访问控制矩阵到信息单向流动的访问控制矩阵的距离) 加权访问控制矩阵 $\boldsymbol{A}=(p_{ij},h_{ij})_{m\times n}$,其中 $p_{ij}\in P, h_{ij}\in Z^+$,信息单向流动的访问控制矩阵 $\boldsymbol{B}=(\hat{p}_{ij})_{m\times n}$,其中 $\hat{p}_{ij}\in\boldsymbol{P}$,那么 $\|\boldsymbol{A}-\boldsymbol{B}\|=\sum_{i=1}^{m}\sum_{j=1}^{n}\ell((p_{ij},h_{ij}),\hat{p}_{ij})$称为 $\boldsymbol{A}$ 到 $\boldsymbol{B}$ 的距离.距离函数

$$\ell((p_{ij},h_{ij}),\hat{p}_{ij})=\begin{cases}h_{ij}, & p_{ij}\neq\hat{p}_{ij}\text{且}p_{ij}\neq e\text{且}\hat{p}_{ij}\neq w\\ 0, & \text{其他}\end{cases}$$

距离函数反映了若 $\boldsymbol{A}$ 实现单向性,在参照 $\boldsymbol{B}$ 的基础上去调整而需要的代价.如果 $\boldsymbol{B}$ 中某个元素值为读写,则 $\boldsymbol{A}$ 中对应元素可为任意值,这不影响 $\boldsymbol{A}$ 单向性;如果 $\boldsymbol{A}$ 中某个元素值为空,则该元素肯定不会造成 $\boldsymbol{A}$ 中信息流出现环路;如果 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 对应元素的权限不同,$\boldsymbol{A}$ 调整需要付出的代价就是其权值,即置该元素值为空.

给定 $\boldsymbol{A}=(p_{ij},h_{ij})_{m\times n}$,染色体 x 表示对全部主客体的一种标记分配方法,由定义7得到相应的 $\boldsymbol{B}$,则染色体 x 适应度函数定义为 $f(x)=1-\|\boldsymbol{A}-\boldsymbol{B}\|/mn$.

采用轮盘赌选择结合最优个体保存方法选择染色体,文献[13]证明二者结合的方法可使进化收敛到全局最优解.在搜索过程中,从当代种群$\{b_1,b_2,\cdots,b_c\}$中轮盘赌选择当代个体 b_i 成为下一代成员父代的概率 $p(b_i)=f(b_i)/\sum_{j=1}^{c}f(b_j)$,得到新一代种群后,将老一代中最优个体也加入其中,淘汰适应度值最小的个体.

(3)**种群初始化** 初始种群通过随机方式产生.为了保证初始种群的多样性,定义第 i 个基因的基因熵[14] $Entropy(i)=-\frac{1}{k}\sum_{s=0}^{k-1}\sum_{t-0}^{1}p(s,t)\log_2 p(s,t)$,其中 $p(s,t)$是初始种群中第 i 个基因的第 s 位取值为 t 的情况占该位所有取值的比例,k 是串长即范畴集大小,对应分配的范畴集空间.设定一阈值 θ,若不能满足 $\log_2(2)-Entropy(i)<\theta$,重新初始化种群中该基因,直至其基因熵满足上述不等式.初始种群规模可针对实例规模通过实验获得.若初始种群数目太少,容易陷入局部最优解;若太大则计算复杂度又较高.

(4)**交叉** 交叉是指两个父代个体的部分结构加以替换重组而生成新个体的操作.在范畴集分配问题中,主客体对应到染色体的位置并无前后次序要求,因此采用均匀交叉方法,即两个相同配对个体的每个基

因都以相同的概率进行交换,从而形成两个新个体.

(5)**变异**　变异用于对个体的编码串产生随机的小变化,即以很小概率从群体中选出一些染色体,随机选择某些基因上某些位,并改变其值.变异概率太大,会导致搜索产生振荡;变异概率太小容易得到局部最优解.

(6)**终止条件**　在指定遗传代数后中止遗传算法,并检查种群中的最优的染色体,如果没有得到满意的解决方案,遗传算法重新启动.

算法复杂度分析:假设访问控制矩阵是 $\boldsymbol{A}_{m\times n}$,范畴个数是 k,群体规模为 l,迭代次数为 t,则上述过程中,初始化种群的时间为 $O(lk(m+n))$,每一轮迭代中,计算个体适应度的时间为 $O(lkmn)$,计算交叉和变异的时间为 $O(lk(m+n))$,则遗传算法计算时间 $O(lktmn)$.若 e 是 $\boldsymbol{A}_{m\times n}$ 中权限值不是空的元素个数,整个基于标记的近似解算法复杂度是 $O(lktmn+e)$.

5　实验与性能分析

5.1　实验方法和评价指标

我们通过实验来分析和验证本文的算法性能.实验数据源采用两种人工合成方法产生,我们称为随机关联方法和噪声方法.设系统有 m 个主体,n 个客体,在随机关联方法中,初始化 $\boldsymbol{A}_{m\times n}$ 中每个元素值为空,权重为[0, $w_{\max}$]内的随机整数值,然后随机选择 $\alpha\times m\times n$ 个不同元素,基于集合{读,写,读/写}随机生成这些元素的新值,α 称为关联因子;在噪声方法中,指定范畴个数 k',然后基于集合{0,1}随机生成主体与范畴的关系和范畴与客体的关系,从而得到主客体的标记,根据定义 6,可生成 $\boldsymbol{A}_{m\times n}$ 中所有元素的值,然后基于[0, $w_{\max}$]生成元素中的权重,最后在 $\boldsymbol{A}_{m\times n}$ 中随机选择 $\beta\times m\times n$ 个不同元素,基于集合{空,读,写,读/写}随机生成这些元素的新值,β 称为噪声因子.随机关联方法容易控制访问控制矩阵中非空权限元素的个数,主体对客体的权限是直接生成的,随机性较好;噪声方法试图在不存在环路的访问控制矩阵中加入噪音,通过噪声因子容易控制访问控制矩阵中环的个数,可控性较好.算法实现形式是 VC6.0 程序,包括产生数据、枚举所有环、备忘录求解、图化简、遗传算法求解和性能评估过程,时间计量精度为 ms.实验环境是 OS/Windows2003,CPU/P4 1.73G,RAM/2G.

我们给出的评价指标如下:设 $\boldsymbol{A}=(p_{ij},h_{ij})_{m\times n}$,经过去环处理,最终调整为 $\boldsymbol{A}'=(p'_{ij},h_{ij})_{m\times n}$,则算法代价比 $\mathrm{CR}=(\sum_{i=1}^{m}\sum_{j=1}^{n}\ell(p_{ij},p'_{ij}))/(\sum_{i=1}^{m}\sum_{j=1}^{n}h_{ij})$,其中 $\ell(p_{ij},p'_{ij})=\begin{cases}h_{ij}, & p_{ij}\neq p_{ij}'\\ 0, & p_{ij}=p_{ij}'\end{cases}$.CR 指出为了消除环路,需要修改的策略的权重和占整个策略集的权重和的比例,CR 在一定程度上代表问题的解.备忘录算法缓存命中率 CH 是备忘录算法从存储 *sub* 数组的哈希表中查找子问题解时成功次数占查找总次数的比例.CH 一定程度上反映了算法的执行效率.其他指标有枚举环时间 ET、备忘录算法时间 MT、化简图时间 ST 和遗传算法时间 GT.

5.2　实验结果及分析算法评测

实验过程中逐步增大 m 和 n,此时最优解算法和近似解算法的性能及变化结果如表 1 所示.c 表示环的个数;e 表示所有的环包含的不同边的个数;m' 和 n' 表示对图化简后仍保留的主体和客体个数."/"表示算法未执行,"N/A"表示在 3 小时内算法未能获得该值.表 1 的结果在以下参数下获得:$w_{\max}=10$,$k'=6$,备忘录算法 $u=128$,$b=1$K,M_R 大小为 512K,遗传算法设置的范畴个数 $k=6$,群体规模 $l=100$,遗传代数 $t=1000$,交叉概率 $p_c=0.8$,突变概率 $p_m=0.05$.评价指标 ET,MT,ST,GT 以秒表示,CH,CRM,CRG,CRI 均以百分比表示.由于 ST 的实验时间均超过 1s,表 1 中未显示 ST.

表 1　各个算法的性能比较

m	n	α/β	最优解算法						近似解算法				
			枚举环			备忘录			化简图		遗传算法		近似解
			ET	c	e	MT	CH	$\mathrm{CR_M}$	m'	n'	$GT(s)$	CR_G	CR_I
10	10	0	0.001	0	0	/	/	/	0	0	/	/	0
10	15	0.2	0.001	1	4	0.001	33.3	0.667	2	2	0.175	13.100	0.667
10	15	0.2	0.001	4	13	0.002	57.5	0.246	8	7	1.067	3.126	1.369
50	50	0.2	0.002	7	29	0.010	43.7	0.209	24	25	8.848	3.096	0.761
50	50	0.2	0.005	22	54	0.029	32.5	0.298	34	31	14.905	4.037	1.689
100	120	0.2	0.049	42	49	0.517	26.1	0.510	40	42	23.577	4.798	0.691
100	120	0.2	0.053	38	68	1.023	23.3	0.551	59	60	48.007	5.313	1.571
150	150	0.2	0.171	77	76	223	24.2	0.525	72	64	63.09	6.248	1.279
150	150	0.2	0.213	100	96	871	29.7	0.441	88	78	92.404	8.215	2.498
200	200	0.2	0.554	136	116	N/A	N/A	N/A	95	103	130.69	8.977	2.105
200	200	0.2	0.857	182	152	4207	11.3	0.394	122	118	191.316	9.602	3.467
800	10^3	0.1	9811	1.2×10^5	573	N/A	N/A	N/A	400	489	2577	12.710	3.209
10^3	10^3	0.1	N/A	N/A	N/A	N/A	N/A	N/A	571	588	4417	15.692	5.271

表 1 结果说明,固定 α 或 β,随着主体和客体数目增多,枚举时间、遗传算法运行时间迅速增长,枚举时间与乘积 $mn(c+1)$大致呈正比例关系,而遗传算法运行时间与环数无关,与 $m'n'$成正比例关系;备忘录算法运行时间主要受到 c 和 e 影响,有时呈指数关系,当 c 和 e 较大时,出现未能在有效时间内完成情况,由于 M_R 能够保存子问题的能力随着问题规模的变大而降低,备忘录算法的缓存命中率 CH 也有所下降;备忘录算法如果能在有效时间内完成,可获得最优解 CR_{MB};近似解算法中图化简过程可以显著减少后续遗传算法求解问题的规模,从而大幅度提高解的准确率;随着问题的规模的变大,遗传算法得到的解与精确解 CR_{MB}之间差值变大,这可能是解空间变大,如果仍保持相同的进化代数和种群大小,则难以充分搜索解空间所致.

图 2 分析了调整遗传算法参数对算法的影响,数据通过噪声方法产生,指定的范畴个数 $k'=6$,噪声因子 $\beta=0$,因此不存在环路,CR 应等于 0,设置遗传算法范畴个数分别为 $k=4,6,8$.结果表明当 $k=4$ 时,由于算法设置的范畴个数范围未能包含正确的范畴个数,算法未收敛到正确解;当 k 不小于 k'时,算法最终都收敛到了完全正确的解 CR = 0.

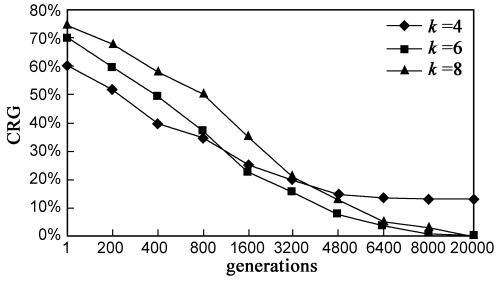

图2 参数调整对遗传算法性能影响

实验结果显示,最优解算法适合主客体数目不大的小规模环境,也可用于存在环路数少的大规模环境,而近似解算法执行时间与环数无关,适合大规模环境.

6 结束语

策略转换是目前我国等级保护推进工作的一个重要内容.访问控制策略中信息流的最优化去环问题是保证策略转换的可用性的一个重要问题.本文形式化描述了该问题.证明了它是 NP 难问题.在此基础上,提出了最优解算法和近似解算法.前者包括枚举环和用备忘录方法找出删除边集合两个子过程;后者包括图化简和基于标记的遗传算法求解两个子过程,算法复杂度分析和实验结果表明对于小规模环境,最优解方法能较快地找出最优解;对于大规模环境,近似最优解算法能有效找出近似解.

下一步研究包括对近似解算法的改进以及用其他近似解求解方法如蚁群算法求解,以获得更准确的解.

参考文献

[1] Garey M R, Johnson D S. Computers and Intractability: A Guide to the Theory of NP-Completeness[M]. New York: W. H. Freeman & Co., 1979. 192 – 193.

[2] Bell D, LaPadual L J. Secure Computer System: Unified Exposition and Multics Interpretation[R]. Bedford: MITRE Corporation, 1976.

[3] Bell D E. Security policy modeling for the next-generation packet switch[A]. Proceeding of the IEEE Symposium. on Security and Privacy[C]. Oakland : IEEE Press, 1988. 212 – 216.

[4] 张晓菲,许访,沈昌祥.基于可信状态的多级安全模型及其应用研究[J].电子学报,2007,35(8):1511 – 1515.
Zhang Xiao-fei, Xu Fang, Shen Chang-xiang. Research on multilevel security model based on trustworthy state and its application[J]. Acta Electronica Sinica, 2007, 35(8): 1511 – 1515. (in Chinese)

[5] 谭智勇,刘铎,司天歌,戴一奇.一种具有可信度特征的多级安全模型[J].电子学报,2008,36(8):1637 – 164.
Tan Zhi-yong, Liu Duo, Si Tian-ge, Dai Yi-qi. A multilevel security model with credibility characteristics[J]. Acta Electronica Sinica, 2008, 36(8): 1637 – 1641. (in Chinese)

[6] Osborn S, Sandhu R S, Munawer Q. Configuring role-based access control to enforce mandatory and discretionary access control policies[J]. ACM Trans on Information and System Security, 2000, 3(2): 85 – 106.

[7] Ravi Sandhu, Ravi S, Qamar Munawer. How to do discretionary access control using roles[A]. Proceedings of the 3rd ACM Symposium on Access Control Models and Technologies[C]. New York: ACM Press, 1998. 47 – 54.

[8] 李斓,冯登国,徐震. . RBAC 与 MAC 在多级关系数据库中的综合模型[J].电子学报,2004,32(10):1635 – 1339.
Li Lan, Feng Deng-guo, et al. A Integrated modeI of RBAC and MAC in muItiIeveI reIation database system[J]. Acta Electronica Sinica, 2004, 32(10): 1635 – 1339. (in Chinese)

[9] 何永忠,李晓峰,冯登国. RBAC 实施中国墙策略及其变种的研究[J].计算机研究与发展,2007,44(4):615 – 622.
He Yongzhong, Li Xiaofeng, Feng Dengguo. Implementing Chinese wall policies on RBAC[J]. Journal of Computer Research and Development, 2007, 44(4): 615 – 622. (in Chinese)

[10] Kuhn D R. Role based access control on MLS systems without kernel changes[A]. Proceedings of the Third ACM Workshop on Role Based Access Control[C]. New York: ACM Press, 1998. 25 – 32.

[11] Sandhu R S. A lattice interpretation of the Chinese wall policy [A]. Proceedings of the 15th NIST-NCSC National Computer Security Conference[C]. Baltimore, MaryLand: NIST-NCSC,

1992,329 – 339.
[12] Johnson D B. Finding all the elementary circuits of a directed graph[J]. SIAM Journal on Computing. 1975,4(1):77 – 84.
[13] Rudolph G. Convergence properties of canonical genetic algorithms[J]. IEEE Trans on Neural Networks,1994,5(1):96 – 101.
[14] Maekawa K,Mori N,et al. A genetic solution for the traveling salesman problem by means of a thermodynamical selection rule[A]. IEEE Conference on Evolutionary Computation[C]. New York:IEEE Press,1996. 529 – 534.

作者简介

杨　智　男,1975 年出生于河南开封,现为中国科学院计算技术研究所信息安全研究中心博士研究生,解放军信息工程大学电子技术学院讲师,主要研究方向为信息安全,系统软件等.
E-mail:yangzhi@software.ict.ac.cn

段▮毅　男,1953 年出生于北京,中国科学院计算技术研究所研究员,博士生导师,中国计算机学会理事,中国电子学会遥感遥测遥控分会常务委员,中国科技情报学会常务理事.主要研究方向为网络与信息安全、人工智能等.

金舒原(通信作者)　女,1974 年出生于吉林白城,中国科学院计算技术研究所副研究员,2006 年获香港理工大学电子计算学博士学位,主要研究方向为网络与信息安全.
E-mail:jinshuyuan@software.ict.ac.cn

殷丽华　女,1973 年出生于辽宁朝阳,中国科学院计算技术研究所副研究员,2007 年获哈尔滨工业大学信息安全博士学位,主要研究方向为网络与信息安全.

郭　莉　女,1969 年出生于湖南株洲,中国科学院计算技术研究所正研级高级工程师,信息内容安全技术国家工程实验室常务副主任,主要研究方向为网络与信息安全,数据流处理.