Journal of Computer Applications
计算机应用,2013,33(5):1367 - 1370,1390
ISSN 1001-9081
CODEN JYIIDU
2013-05-01
http://www.joca.cn

文章编号:1001 - 9081(2013)05 - 1367 - 04
doi:10.3724/SP.J.1087.2013.01367

基于 Partial MAX-SAT 求解法的 RBAC 授权查询方法

孙 伟*,李艳灵,鲁 骏
(信阳师范学院 计算机与信息技术学院,河南 信阳 464000)
(*通信作者电子邮箱 sw_810715@163.com)

摘 要:为保证系统的安全性并体现授权的有效性,结合部分最大可满足性问题(Partial MAX-SAT)的研究,提出一种基于 Partial MAX-SAT 求解法的授权查询方法。使用转换规则将静态授权逻辑和动态互斥角色约束转化为严格子句,采用子句更新算法将满足不同匹配的请求权限转化为松弛子句,并利用子句编码及递归算法寻求真值指派,以满足所有严格子句和尽可能多的松弛子句。实验结果表明,该方法搜索的角色组合能够保证系统的安全性,并满足最小权限分配要求,且最大、精确匹配请求的查询效率优于 MAX-SAT 求解法。

关键词:基于角色的访问控制;部分最大可满足性问题;用户授权查询问题;严格子句;松弛子句
中图分类号:TP309.2 **文献标志码**:A

Authorization query method for RBAC based on partial MAX-SAT solver

SUN Wei*, LI Yanling, LU Jun
(*School of Computer and Information Technology, Xinyang Normal University, Xinyang Henan* 464000, *China*)

Abstract: In order to ensure system security and reflect availability in authorization management, a method for querying authorization was proposed based on solvers for partial maximal satisfiability problem. Static authorization descriptions and dynamic mutually exclusive constraints were translated into hard clauses. The algorithm was adopted to update hard clauses and translate requested permissions into soft clauses. Soft clauses were effectively encoded, and the recursive algorithm was utilized to satisfy all hard clauses and as many soft clauses as possible. The experimental results show that the method can ensure system security, it follows the least privilege principle, and the query efficiency outperforms solvers for maximal satisfiability problem.

Key words: Role-Based Access Control (RBAC); Partial Maximal Satisfiability problem (Partial MAX-SAT); User Authorization Query (UAQ) problem; hard clause; soft clause

0 引言

在基于角色的访问控制(Role-Based Access Control, RBAC)系统中,授权管理通过角色实现用户与权限的逻辑分离,这已被证明是一种灵活、方便的访问控制技术。随着信息技术的不断发展和大规模应用,互斥角色约束[1-2]和最小权限分配[3]对于检测系统安全性与授权有效性至关重要。用户临时请求的多个权限可以由不同角色组合通过角色-权限映射得到,而为用户授权角色时存在以下问题:一方面,不同的角色组合可能存在冲突,即违反职责分离原则;另一方面,不同角色组合可能包含请求之外的其他权限,即违反最小特权原则。由于访问请求的随意性及系统设置的模糊性,寻求恰好覆盖请求权限的理想角色组合往往是不可行的。在严格遵循职责分离的前提下,寻求覆盖尽可能多权限的授权查询问题却是可满足的。命题逻辑公式的合取范式的最大可满足性问题(Maximal Satisfiability problem,MAX-SAT)是理论计算机科学的核心问题之一,在人工智能、模型检测等研究领域有着广泛的应用[4-5]。作为 MAX-SAT 的分支领域,部分最大可满足性问题(Partial Maximal Satisfiability problem,Partial MAX-SAT)以满足所有严格子句和尽可能多松弛子句为求解目标[6],描述问题能力更丰富。因此,如何将授权查询问题转化为 Partial MAX-SAT,并寻求合适的角色组合很具挑战性,值得研究。

文献[7]针对授权管理中权限查询难以计算,提出基于衍生规则和可达矩阵的两种查询方法,但不满足最小权限分配要求,且未对角色授权作进一步研究。文献[8]通过静态互斥角色约束对权限查询进行有效约束,支持灵活的角色授权。文献[9]针对云计算环境中授权管理繁重,提出最小唯一角色集及其求解方法,缩短为用户授权角色的时间。然而两文献均不满足动态互斥角色约束[10]要求。Zhang 等[11]定义给出了基于 RBAC 的用户授权查询问题(User Authorization Query problem,UAQ),采用贪心算法进行搜索,并运用动态互斥角色约束进行检测,但在精确匹配请求下一旦检测到不符合要求的角色组合便停止搜索,权限请求即被拒绝。针对该问题,文献[12]采用 DNA 链描述 UAQ 中的角色和权限,建立角色组合解空间,并使用基本生物操作实现 UAQ 的求解过程,但 DNA 计算解决规模问题的效率不高,存在"指数爆炸"问题。Wickramaarachchi 等[13]引入请求权限上界和下界,将带最大、最小匹配请求的 UAQ 转化为 MAX-SAT,寻求真值指派以满足尽可能多 MAX-SAT 子句,并使用现有 SAT 解法器分析方案的运行效率。该方案能够满足授权有效性要求,但不适用于带严格约束限制的 UAQ。Argelich 等[14]通过对至多约束和至少约束进行规则编码,将约束满足性问题(Constraint Satisfaction Problem, CSP)转化为 Partial MAX-

收稿日期:2012-11-15;**修回日期**:2013-01-10。 **基金项目**:国家自然科学基金资助项目(61202194);河南省教育厅科学技术研究重点项目(13A520765);河南省信息技术教育研究项目(ITE12192)。

作者简介:孙伟(1981 -),男,河南信阳人,讲师,硕士,主要研究方向:访问控制、模型检测; 李艳灵(1975 -),女,河南新乡人,副教授,博士,主要研究方向:目标检测与识别、软件工程; 鲁骏(1981 -),男,河南信阳人,讲师,硕士,主要研究方向:计算机网络、软件工程。

SAT。Koshimura 等[15]提出基于 Partial MAX-SAT 的 QMAX-SAT 查询方法,使用粘贴变量对松弛子句重新编码,并通过现有 SAT 解法器验证了方案的高效性。

针对现有 RBAC 授权管理在安全性、有效性等方面存在的问题,结合部分最大可满足性问题的研究,提出一种基于 Partial MAX-SAT 求解法的授权查询方法(Partial MAX-SAT-based Authorization Query method, PMSAQ)。PMSAQ 将 UAQ 转化为 Partial MAX-SAT,以寻求合适的角色组合,能够严格保证系统的安全性,并满足授权的有效性要求。

1 预备知识及问题描述

1.1 部分最大可满足性问题

Partial MAX-SAT 是 MAX-SAT 的一种扩展形式,它将问题中的实例分为严格实例与松弛实例。

定义 1[6] Partial MAX-SAT 的布尔逻辑描述。

1)文字:用布尔变量 $x_1,x_2,\cdots,x_i,\cdots$ 表示实例中的若干个体元素。对于任意 x_i,称 x_i 或 $\neg x_i$(x_i 的逻辑非)为文字。

2)子句:由 n 个文字通过析取运算符($\vee$)连接成形如"$x_1 \vee x_2 \vee \cdots \vee x_n$"的实例,称为子句 c。

3)真值指派:用映射函数($t:\{x_1,x_2,\cdots,x_i,\cdots\} \to \{0,1\}$)为子句中每一文字赋以真(用 1 表示)或假(用 0 表示)。

4)合取范式:由 m 个子句通过合取运算符($\wedge$)连接成"$c_1 \wedge c_2 \wedge \cdots \wedge c_m$"的形式,称为 Partial MAX-SAT 的合取范式,用 $cnf(c_1,c_2,\cdots,c_m)$ 表示。$cnf(c_1,c_2,\cdots,c_m)$ 为 1,当且仅当每个 c_i 在同一真值指派下均为 1。

5)严格子句:称子句 $c_1,c_2,\cdots,c_m$ 为 Partial MAX-SAT 的严格子句,是可满足的,当且仅当存在真值指派,使得 $cnf(c_1,c_2,\cdots,c_m)$ 为 1。所有严格子句的集合用 HC 表示。

6)松弛子句:PartialMAX-SAT中不满足严格子句要求的,称为松弛子句。所有松弛子句的集合用 SC 表示。

$$match = \begin{cases} \min, .\ \text{iff} \forall s' \in S\backslash\{s\}, \exists (r,r') \in R, \bigcup_{r \in roles\_session(s)} perms\_session(r) \subseteq \bigcup_{r' \in roles\_session(s')} perms\_session(r') \subseteq P_{lb} \\ \max, .\ \text{iff} \forall s' \in S\backslash\{s\}, \exists (r,r') \in R, P_{ub} \supseteq \bigcup_{r \in roles\_session(s)} perms\_session(r) \supseteq \bigcup_{r' \in roles\_session(s')} perms\_session(r') \\ \text{exact}, .\ \text{iff } P_{lb} = \bigcup_{r \in roles\_session(s)} perms\_session(r) = P_{ub} \end{cases}$$

2 基于 Partial MAX-SAT 求解法的授权查询

用 $\bar{p}$、$\bar{r}$ 分别表示权限 p、角色 r 对应的 Partial MAX-SAT 文字,对 $\bar{p}$,$\bar{r}$ 及其逻辑非作以下描述:

1) $\bar{p}$: $\forall p \in P, \exists r \in R, \exists s \in S,$

$$p \in \bigcup_{r \in roles\_session(s)} perms\_session(r);$$

2) $\neg \bar{p}$: $\forall p \in P, \exists r \in R, \exists s \in S,$

$$p \notin \bigcup_{r \in roles\_session(s)} perms\_session(r);$$

3) $\bar{r}$: $\forall r \in R, \exists s \in S, r \in roles\_session(s)$;

4) $\neg \bar{r}$: $\forall r \in R, \exists s \in S, r \notin roles\_session(s)$。

2.1 预处理阶段

为了严格保证系统的安全性,PMSAQ 将静态授权逻辑及动态互斥角色约束转换成严格子句,并置 SC 为 $\varnothing$,具体规则如下:

1) $\forall p \in P, \exists (r,r') \in R: (r,r') \in RH \wedge (p,r') \in PA \Rightarrow \neg \bar{r} \vee \bar{p}$;

2) $\forall p \in P, \exists (r',r_1,r_2,\cdots) \in R: (p,r') \in PA \wedge (r_1,r') \in RH \wedge (r_2,r') \in RH \wedge \cdots \Rightarrow \neg \bar{p} \vee \bar{r}_1 \vee \bar{r}_2 \vee \cdots$;

3) $\forall r \in R, \exists r' \in R, \exists (p_1,p_2,\cdots) \in P: (r,r') \in RH \wedge (p_1,r') \in PA \wedge (p_2,r') \in PA \wedge \cdots \Rightarrow \bar{r} \vee \neg \bar{p}_1 \vee \neg \bar{p}_2 \vee \cdots$;

4) $\forall r \in R, \exists s \in S: (user\_session(s),r) \notin UA \Rightarrow \neg \bar{r}$;

1.2 用户授权查询问题

UAQ 是指寻求在单个会话中激活能覆盖用户请求权限的角色组合,同时遵循动态互斥角色约束并满足最小权限分配要求。

定义 2[10] 动态互斥角色约束(Dynamic Mutually Exclusive Role, DMER)。任意用户在同一会话 s 中都不能激活 $\{r_1,r_2,\cdots,r_m\}$ 的 t 个或更多角色,表示为 $dmer\langle\{r_1,r_2,\cdots,r_m\},t,s\rangle$,其中 $1 < t \leqslant m$。所有动态约束的集合用 DC 表示。

定义 3 RBAC 授权系统。沿用传统 RBAC 的用户集(U)、角色集(R)、权限集(P)、会话集(S)、用户 - 角色关联(UA)、权限 - 角色关联(PA)及角色层次关系(RH),授权系统是一个五元组$\langle UA,PA,RH,S,DC\rangle$。若 $user\_session(s)$ 表示 S 到 U 一对一的映射函数,则 $roles\_session(s)$ 为 S 到 2^R 一对多的映射函数,$perms\_session(r)$ 为 R 到 2^P 一对多的映射函数,且可形式化描述为:

$$\begin{aligned} roles\_session(s) = \{&r_1',r_2',\cdots,r_n' \mid \forall dmer\langle\{r_1,r_2,\cdots,r_m\},t,s\rangle \in DC, (user\_session(s),r_1') \in UA \wedge \\ &(user\_session(s),r_2') \in UA \wedge \cdots \wedge (user\_session(s),r_n') \in UA \wedge \mid \{r_1',r_2',\cdots,r_n'\} \cap \{r_1,r_2,\cdots,r_m\} \mid < t\}; \end{aligned}$$

$$perms\_session(r) = \{p \in P \mid \exists s \in S, \exists r' \in R, (r,r') \in RH \wedge r' \in roles\_session(s) \wedge (p,r') \in PA\}$$

定义 4[13] 权限请求。权限请求是一个四元组$\langle s,P_{lb},P_{ub},match\rangle$。其中:$s \in S$,表示单个会话;$P_{lb}$、$P_{ub}$ 分别表示请求权限下界、上界,且 $P_{lb} \subseteq P_{ub} \subseteq P$;$match \in \{\min(\text{最小匹配}), \max(\text{最大匹配}), \text{exact}(\text{精确匹配})\}$,表示匹配类型,规定并形式化描述如下:

5) $\forall r' \in \{r_1,r_2,\cdots,r_m\} \subseteq R, \exists s \in S, \exists t \geqslant 2: dmer\langle\{r_1,r_2,\cdots,r_m\},t\rangle \in DC \wedge \mid \{r_1,r_2,\cdots,r_m\} \cap roles\_session(s) \mid < t \Rightarrow \bar{r}'$。

2.2 子句更新阶段

为了体现授权的有效性及最小权限分配要求,分析给出 PMSAQ 的两个基本性质。

基本性质 1 有效性要求。请求权限下界内的所有权限在单个会话中都是可满足的、有效的。

$$\forall p \in P, \exists s \in S, \exists r \in R: \quad p \in P_{lb} \Rightarrow p \in \bigcup_{r \in roles\_session(s)} perms\_session(r)$$

基本性质 2 最小权限分配要求。权限上界之外的其他权限在单个会话中都是不可满足的、无效的。

$$\forall p \in P, \exists s \in S, \exists r \in R: \quad p \in P\backslash P_{ub} \Rightarrow p \notin \bigcup_{r \in roles\_session(s)} perms\_session(r)$$

结合两个基本性质,对于给定的权限请求,给出子句更新算法(Clause_Update Algorithm),描述如下。

算法 1 Clause_Update Algorithm。

输入:权限请求$\langle s,P_{lb},P_{ub},match\rangle$及预处理结果 HC,SC。

输出:HC,SC。

```
Begin
  for (P_lb 中的任意权限 p)
    HC = HC ∪ {p̄};                          //由性质 1 可知
  for (P\P_ub 中的任意权限 p)
    HC = HC ∪ {¬ p̄};                        //由性质 2 可知
  return (HC);
  if (match == min) then
    for (P_ub\P_lb 中的任意权限 p)
      SC = SC ∪ {¬ p̄};                      //满足最小匹配请求的松弛子句
  else if (match == max) then
    for (P_ub\P_lb 中的任意权限 p)
      SC = SC ∪ {p̄};                        //满足最大匹配请求的松弛子句
  else
    SC = SC ∪ ∅;                            //满足精确匹配请求的松弛子句
  return (SC);
End
```

2.3　子句编码及查询阶段

在满足 HC 中所有严格子句的前提下,对 SC 中松弛子句进行编码及查询,步骤如下:

1) 对于给定的 $SC = \{sc_1, sc_2, \cdots, sc_n\}$,引入 n 个粘贴变量 $b_i (1 \leqslant i \leqslant n)$;

2) 构造新的子句集 $SC^b = \{sc_1 \vee b_1, sc_2 \vee b_2, \cdots, sc_n \vee b_n\}$,并将子句 $sc_i \vee b_i$ 作为 sc_i 的 b_i 个副本;

3) 寻求最小整数 k,使得 SC^b 也为真,并满足 $\sum_{i=1}^{n} b_i \leqslant k$;

4) 将 $HC \cup SC^b$ 作为参数,使用 QMAX-SAT 方法[15] 进行查询,返回 SC 中值为真的所有子句。

3　PMSAQ 合理性与有效性

3.1　关于动态约束的进一步简化

针对预处理阶段给出的 $dmer\langle\{r_1, r_2, \cdots, r_m\}, t, s\rangle$ 转换规则,需要将 m 个单元子句 $\bar{r}_1, \bar{r}_2, \cdots, \bar{r}_m$ 分别添加到 PMSAQ 的 HC。如果动态约束包含的角色数庞大,那么 HC 存在规模扩张问题,必将加大求解规模问题的难度。因此,有必要对动态约束作进一步简化,寻求更加合理的转换规则。

定义 5　强约束关系 $\succ$。如果任意两动态约束 dc_1 和 dc_2 都可实施系统策略,且 dc_1 的约束性不弱于 dc_2,并比 dc_2 约束的范围更大,那么称 dc_1 比 dc_2 具有更强的约束性,表示为 $dc_1 \succ dc_2$。

当 dc_1 和 dc_2 同时实施系统策略时,采用 dc_1 会产生更多的冗余约束,增加系统开销。考虑采用约束性弱、约束范围小的 dc_2,产生的冗余约束就越少,实施就越准确。因此,当存在多个动态约束可同时实施系统策略时,应采用约束性更弱的实施方案[8]。$dmer\langle\{r_1, r_2, \cdots\}, 2, s\rangle$ 比 $dmer\langle\{r_1, r_2, \cdots\}, t, s\rangle (t > 2)$ 更准确,可通过以下转换规则降低 HC 规模,并替换预处理阶段的规则 5)。

$$\forall s \in S, \exists (r_1, r_2, \cdots) \subseteq R: dmer\langle\{r_1, r_2, \cdots\}, 2, s\rangle \in DC \Leftrightarrow \neg \bar{r}_1 \vee \neg \bar{r}_2 \vee \cdots$$

3.2　关于严格子句的真值指派

为满足 HC 的严格子句,必须寻求真值指派使所有子句均为真。通过以下递归算法(True_HC_Assign Algorithm)实现,描述如下。

算法 2　True_HC_Assign Algorithm。

输入:$length(c)$。其中 c 为子句变量,$length(c)$ 表示该子句的长度。

输出:true|false。

```
Begin
  取 HC 的任一子句 c;
  if( length(c) ==1) then // c 为单元子句,即只包含单个文字
  {
    c =1;
    HC = HC\{c};
    for (HC 中含 c 的任意子句 c' ∨ c)
      HC = HC\{c' ∨ c};
    for (HC 中含 ¬ c 的任意子句 c'' ∨ ¬ c)
      HC = HC\{c'' ∨ ¬ c} ∪ {c''};
    return( true);
  }
  else True_HC_Assign (length(c) - 1);      //c 为非单元子句
  return( false);
End
```

定理　算法 2 对 HC 的真值指派过程是正确、有效的。

证明可通过数学归纳法,并结合预处理及子句更新过程,限于篇幅不再详述。

3.3　应用实例

给定 $P = \{p_0, p_1, p_2, p_3, p_4, p_5, p_6, p_7\}$, $R = \{r_0, r_1, r_2\}$, $RH = \{r_0 \geqslant r_2\}$, $PA = \{(r_0, p_0), (r_0, p_1), (r_0, p_2), (r_0, p_4), (r_0, p_5), (r_0, p_6), (r_1, p_3), (r_1, p_7), (r_2, p_2), (r_2, p_6)\}$, $DC = \{dmer\langle\{r_0, r_1\}, 2, s\rangle\}$,在预处理中通过转换规则构造严格子句,结果如表 1 所示。

针对不同权限请求:1) $P_{lb} = \varnothing, P_{ub} = \{p_2, p_3, p_6\}$, $match$ = max; 2) $P_{lb} = \{p_2, p_3, p_6\}, P_{ub} = P, match$ = min; 3) $P_{lb} = \{p_2, p_3, p_6\}, P_{ub} = \{p_2, p_3, p_6\}, match$ = exact,在子句更新中调用算法 1 修改 HC、SC,结果如表 2 所示。

通过调用算法 2,能够满足 HC 所有子句及 SC 尽可能多子句,同时允许激活的角色组合如表 3 所示。

表 1　授权逻辑、动态约束与严格子句对应关系

授权逻辑及动态约束	严格子句
$\forall p \in P, \exists (r, r') \in R:$ $(r, r') \in RH \wedge (p, r') \in PA$	$\neg \bar{r}_0 \vee \bar{p}_0, \neg \bar{r}_0 \vee \bar{p}_1, \neg \bar{r}_0 \vee \bar{p}_2, \neg \bar{r}_0 \vee \bar{p}_4, \neg \bar{r}_0 \vee \bar{p}_5,$ $\neg \bar{r}_0 \vee \bar{p}_6, \neg \bar{r}_1 \vee \bar{p}_3, \neg \bar{r}_1 \vee \bar{p}_7, \neg \bar{r}_2 \vee \bar{p}_2, \neg \bar{r}_2 \vee \bar{p}_6$
$\forall p \in P, \exists (r', r_1, r_2, \cdots) \in R:$ $(p, r') \in PA \wedge (r_1, r') \in RH \wedge (r_2, r') \in RH \wedge \cdots$	$\neg \bar{p}_0 \vee \bar{r}_0, \neg \bar{p}_1 \vee \bar{r}_0, \neg \bar{p}_2 \vee \bar{r}_0 \vee \bar{r}_2, \neg \bar{p}_3 \vee \bar{r}_1, \neg \bar{p}_4 \vee \bar{r}_0,$ $\neg \bar{p}_5 \vee \bar{r}_0, \neg \bar{p}_6 \vee \bar{r}_0 \vee \bar{r}_2, \neg \bar{p}_7 \vee \bar{r}_1$
$\forall r \in R, \exists r' \in R, \exists (p_1, p_2, \cdots) \in P:$ $(r, r') \in RH \wedge (p_1, r') \in PA \wedge (p_2, r') \in PA \wedge \cdots$	$\bar{r}_0 \vee \neg \bar{p}_0 \vee \neg \bar{p}_1 \vee \neg \bar{p}_2 \vee \neg \bar{p}_4 \vee \neg \bar{p}_5 \vee \neg \bar{p}_6,$ $\bar{r}_1 \vee \neg \bar{p}_3 \vee \neg \bar{p}_7, \bar{r}_2 \vee \neg \bar{p}_2 \vee \neg \bar{p}_6$
$\forall s \in S, \exists (r_1, r_2, \cdots) \subseteq R:$ $dmer\langle\{r_1, r_2, \cdots\}, 2, s\rangle \in DC$	$\neg \bar{r}_0 \vee \neg \bar{r}_1$

表 2 权限请求与 *HC*、*SC* 更新值对应关系

权限请求	*HC* 更新值	*SC* 更新值
$\langle s,\varnothing,\{p_2,p_3,p_6\},\max\rangle$	$HC \cup \{\neg \bar{p}_0,\neg \bar{p}_1,\neg \bar{p}_4,\neg \bar{p}_5,\neg \bar{p}_7\}$	$\{\bar{p}_2,\bar{p}_3,\bar{p}_6\}$
$\langle s,\{p_2,p_3,p_6\},P,\min\rangle$	$HC\{\bar{p}_2,\bar{p}_3,\bar{p}_6\}$	$\{\neg \bar{p}_0,\neg \bar{p}_1,\neg \bar{p}_4,\neg \bar{p}_5,\neg \bar{p}_7\}$
$\langle s,\{p_2,p_3,p_6\},\{p_2,p_3,p_6\},\text{exact}\rangle$	$HC\{\bar{p}_2,\bar{p}_3,\bar{p}_6,\neg \bar{p}_0,\neg \bar{p}_1,\neg \bar{p}_4,\neg \bar{p}_5,\neg \bar{p}_7\}$	$\varnothing$

表 3 权限请求与激活角色对应关系

权限请求	角色文字	激活角色
$\langle s,\varnothing,\{p_2,p_3,p_6\},\max\rangle$	$\neg \bar{r}_0 \neg \bar{r}_1 \quad \bar{r}_2$	$\{r_2\}$
$\langle s,\{p_2,p_3,p_6\},P,\min\rangle$	$\neg \bar{r}_0 \quad \bar{r}_1 \quad \bar{r}_2$	$\{r_1,r_2\}$
$\langle s,\{p_2,p_3,p_6\},\{p_2,p_3,p_6\},\text{exact}\rangle$	$\neg \bar{r}_0 \neg \bar{r}_1 \neg \bar{r}_2$	$\varnothing$

4 实验分析

4.1 实验设置及测试环境

对于权限请求$\langle s,P_{lb},P_{ub},match\rangle$，假定 s 为 S 的某一确定会话，若用 Pu 表示系统原始分配给用户 u 的权限，且 $Prequest \subseteq Pu$，则关于 P_{lb}，P_{ub} 及 $match$ 分别作以下设置：

1）$P_{lb} = Prequest, P_{ub} = Pu, match = \min$；

2）$P_{lb} = \varnothing, P_{ub} = Prequest, match = \max$；

3）$P_{lb} = P_{ub} = Prequest, match = \text{exact}$。

实验对角色数$|R|$变化在 40 至 300 之间且增量为 20 的不同授权系统进行测试，相关的测试环境包括：奔腾双核 E5400CPU 2.70 GHz，2 GB 内存，160 GB 硬盘，Window XP 操作系统。在 Java 中实现子句更新及真值指派，并使用在求解 MAX-SAT 问题中表现优越的 QMAX-SAT 方法[15]执行 *SC*。

4.2 结果分析

图 1 给出了 PMSAQ 执行时间的比较，其中 x 轴表示角色数($|R|$)，y 轴表示执行时间，包括 *HC* 的真值指派、*SC* 的编码及使用 QMAX-SAT 的查询时间。图 1 表明随着角色数的不断增大，对于精确和最大匹配的权限请求，授权查询时间均呈平缓提高趋势，当$|R|$达到 300 时，查询时间还不超过10 ms；对于最小匹配请求，随着角色数的不断增大，授权查询时间呈不规则跳变趋势。其实在预处理及 *HC*、*SC* 更新阶段，不同匹配请求花费的时间可以认为是一样的。不同的是使用 QMAX-SAT 查询时，最小匹配请求要比精确、最大匹配花费的时间要长，效率更低些。然而从图 1 可以看出，三种匹配请求各自的查询时间均不超过 1 s。

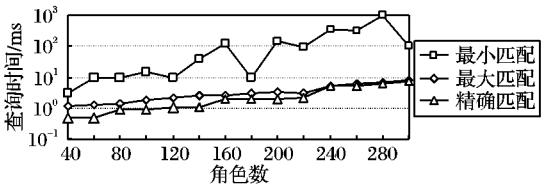

图 1 不同匹配查询比较

4.3 与其他方案比较

为了进一步验证 PMSAQ 的查询效率，实验选用文献[13]的数据集：$30 \leqslant |R| < 300$，$|U| = 50$，$|P| = 80$，$|DC| = 5$，s 仍表示 S 中确定的会话。任意选取 10 对不同的(P_{lb},P_{ub})测试，并与文献[13]中基于 MAX-SAT 求解法的查询方案对不同匹配请求进行了比较。图 2 表示 $match = \text{exact}$ 的查询时间比较，图 3 表示 $match = \max$ 的查询时间比较，图 4 表示 $match = \min$ 的查询时间比较。

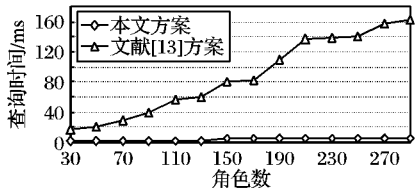

图 2 精确匹配查询比较

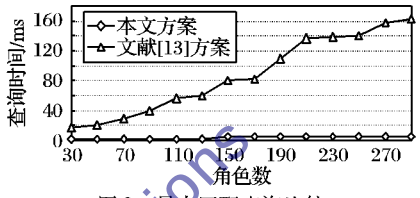

图 3 最大匹配查询比较

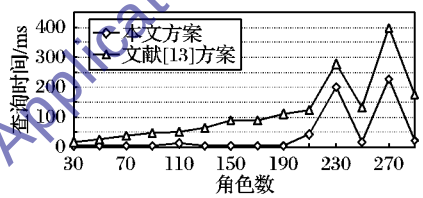

图 4 最小匹配查询比较

从图 2、图 3 的实验结果可以看出，随着$|R|$的增长，PMSAQ 的查询时间变化并不明显，当$|R|$达到 290 时还不超过 20 ms；基于 MAX-SAT 求解法的查询时间呈线性增长趋势，当$|R|$达到 290 时超过 160 ms。这是因为 MAX-SAT 求解法在查询阶段包含对所有子句进行编码，而 PMSAQ 将 MAX-SAT 子句分为严格子句与松弛子句，所有子句的编码都是在预处理阶段完成的，总查询时间低于 MAX-SAT 求解法。因此在精确匹配和最大匹配请求下，PMSAQ 的查询效率均优于 MAX-SAT 求解法。

从图 4 的实验结果可以看出，当$30 \leqslant |R| \leqslant 190$，随着$|R|$的增长，PMSAQ 的查询时间变化并不明显，而 MAX-SAT 求解法的查询时间呈线性增长趋势；当$190 < |R| \leqslant 290$，两方案查询时间均出现不规则跳变，最高值分别逼近 240 ms 和 400 ms。因此在最小匹配请求下，PMSAQ 的查询效率并不优于 MAX-SAT 求解法。

5 结语

本文提出的 PMSAQ 将 RBAC 的授权查询问题转化为部分最大可满足性问题，并运用转换规则和更新算法分别构造严格子句与松弛子句，通过子句编码及递归算法寻求真值指派，能够满足所有严格子句和尽可能多的松弛子句。应用实例及实验分析结果表明，该方法搜索的角色组合保证了系统的安全性并体现了授权的有效性，且最大、精确匹配请求的查询时间明显低于 MAX-SAT 求解法，对于大规模系统的授权管理具有一定的理论参考价值。结合本文的研究方法，如何从具有不同信任度的多个用户中选取合适的对象实施角色授权或委托操作，是下一步需要研究的问题。

（下转第 1390 页）

发给 B,否则 B 不会发货或者把数字商品内容的解密密钥发给 A。所以 A 对 B 在 *keystone* 上的优势也就不存在了。

4.5 高效性

高效性指的是方案的运行需要更少的运算开销。同样是基于签密的并发签名,但本方案比文献[11 - 12]中的方案具有更高的效率。各方案中用到的各种运算的个数统计如表 1 所示。

表 1 本文方案与文献[11 - 12]方案的效率比较

运算	文献[11 - 12]方案			本文方案		
	签密	解密	验证	签密	解密	验证
指数	1	0	0	1	0	0
双线性对	1	1	4	1	1	2
Hash	4	1	0	2	0	2
乘法	6	3	4	3	0	2
求逆	1	0	0	0	0	0

从表 1 中可以看出本文的方案不管是耗时比较多的双线性对运算还是简单的 Hash、乘法以及求逆运算都有不同程度地减少。此外,本文的方案是基于身份的,避免了复杂的证书管理。所以在效率上要高于文献[11 - 12]中的方案。

5 结语

本文将基于身份的密码体制与签密技术进行有效的结合,设计了一个用基于身份的环签密构造的并发签名方案,大大提高了并发签名的效率,降低了通信成本,并基于该方案设计了一个公平交换协议。该协议既可用于交易实物又可用于交易数字商品,并将解密密文与绑定签名分开来处理,获得了更好的公平性,使得该协议在电子支付、合同签署、邮件认证等电子商务活动中有着比较广泛的应用前景。

参考文献:

[1] CHEN L, KUDLA C, PATERSON K G. Concurrent signatures [C]// Advances in Cryptology: EUROCRYPT 2004, LNCS 3027. Berlin: Springer-Verlag, 2004: 287 - 305.

[2] SUSIL O W, MU Y, ZHANG F. Perfect concurrent signature schemes [C] // ICICS'04: Proceedings of Information and Communications Security, LNCS 3269. Berlin: Spriger-Verlag, 2004: 14 - 26.

[3] WANG G L, BAO F, ZHOU J Y. The fairness of perfect concurrent signatures[C] // ICICS'06: Proceedings of the 8th International Conference on Information and Communications Security, LNCS 4307. Berlin: Springer-Verlag, 2006: 435 - 451.

[4] CHOW S S M, SUSILO W. Generic construction of (identity-based) perfect concurrent signatures [C] // ICICS'05: Proceedings of the 7th International Conference on Information and Communications Security, LNCS 3783. Berlin: Springer-Verlag, 2005: 194 - 206.

[5] HUANG Z J, CHEN K F, WANG Y M. Analysis and improvements of two identity-based perfect concurrent signature schemes[J]. Informatica, 2007, 18(3): 375 - 394.

[6] HUANG X F, WANG L C. A fair concurrent signature scheme based on identity [C] // HPCA'09: Proceedings of the 2nd International Conference on High-performance Computing and Applications, LNCS 5938. Berlin: Springer-Verlag, 2010: 198 - 205

[7] QIN W, ZHOU N R. New concurrent digital signature scheme based on the computational Diffie-Hellman problem[J]. The Journal of China Universities of Posts and Telecommunications, 2010, 17(6): 89 - 94.

[8] 刘景伟, 孙蓉, 郭庆燮. 公平交换签名方案[J]. 中国科学: 信息科学, 2010, 40(6): 786 - 795.

[9] 陈光辉, 卿斯汉, 齐志峰, 等. 新颖的基于并发签名的公平交换协议[J]. 通信学报, 2008, 29(7): 39 - 43.

[10] 李云峰, 何大可, 路献辉. 无需可信第三方的防滥用公平交换协议[J]. 计算机应用研究, 2009, 26(8): 3053 - 3055.

[11] 罗铭, 邹春华, 胡军, 等. 基于签密的公平交易协议[J]. 通信学报, 2010, 31(8A): 146 - 150.

[12] 孙艳宾, 孙燕, 赵辰, 等. 基于签密的公平交换协议的安全性分析与改进[J]. 北京邮电大学学报, 2011, 34(6): 38 - 41.

[13] 祁正华. 基于身份的签密方案研究[D]. 南京: 南京邮电大学, 2011.

(上接第 1370 页)

参考文献:

[1] 杨勇. 基于多约束关系的安全授权分析与验证[D]. 济南: 山东大学, 2009.

[2] 鲁剑峰. 访问控制策略的安全与效用优化方法研究[D]. 武汉: 华中科技大学, 2010.

[3] MA X P, LI R X, LU Z D, *et al*. Specifying and enforcing the principle of least privilege in role-based access control[J]. Concurrency and Computation: Practice and Experience, 2011, 23(12): 1313 - 1331.

[4] 赵同昇, 朱文兴. MAX-SAT 问题一种改进的局部搜索算法[J]. 计算机工程与科学, 2008, 30(11): 50 - 52, 79.

[5] 孙如祥, 唐天兵, 李炳慧. 并行蚁群算法求解加权 MAX-SAT[J]. 计算机应用研究, 2012, 29(1): 49 - 51.

[6] FU Z H, MALIK S. On solving the partial MAX-SAT problem[C]// SAT 2006: Proceedings of the 9th International Conference on the Theory and Application of Satisfiability Testing. Seattle: IEEE Press, 2006: 252 - 265.

[7] 王婷, 陈性元, 任志宇. 授权管理中的权限衍生计算方法[J]. 计算机应用, 2011, 31(5): 1291 - 1294.

[8] 王婷, 陈性元, 张斌, 等. 基于互斥角色约束的静态职责分离策略[J]. 计算机应用, 2011, 31(7): 1884 - 1886, 1890.

[9] 杨柳, 唐卓, 李仁发, 等. 云计算环境中基于用户访问需求的角色查找算法[J]. 通信学报, 2011, 32(7): 169 - 175.

[10] LI N H, TRIPUNITARA M V, BIZRI Z. On mutually-exclusive roles and separation of duty[J]. ACM Transactions on Information and System Security, 2007, 10(2): 42 - 51.

[11] ZHANG Y, JOSHI J B D. UAQ: a framework for user authorization query processing in RBAC extended with hybrid hierarchy and constraints[C]// Proceedings of the 13th ACM Symposium on Access Control Models and Technologies. New York: ACM Press, 2008: 83 - 92.

[12] 管蓉. DNA 计算在基于角色的访问控制系统中的应用研究[D]. 长沙: 湖南大学, 2010.

[13] WICKRAMAARACHCHI G T, QARDAJI W H, LI N H. An efficient framework for user authorization queries in RBAC systems [C]// Proceedings of the 14th ACM Symposium on Access Control Models and Technologies. New York: ACM Press, 2009: 23 - 32.

[14] ARGELICH J, CABISCOL A, LYNCE I, *et al*. Regular encodings from MAX-CSP into partial MAX-SAT[C]// Proceedings of the 39th International Symposium on Multiple-Valued Logics. Piscataway: IEEE Press, 2009: 196 - 202.

[15] KOSHIMURA M, ZHANG T, FUJITA H, *et al*. QMaxSAT: a partial MAX-SAT solver[J]. Journal on Satisfiability, Boolean Modeling and Computation, 2012, 8(2): 95 - 100.