doi:10.3969/j.issn.1000-436x.2012.z1.032

面向社会网络的个性化隐私策略定义与实施

王媛[1]，孙宇清[1]，马乐乐[1,2]
（1. 山东大学 计算机科学与技术学院，山东 济南 250101；2. 中国科学院，北京 100190）

摘 要：为了实现社会网络中个性化隐私保护，提出了支持个性化隐私偏好授权模型，采用基于一阶逻辑隐私偏好描述语言，表达用户个性化隐私需求；引入基于主体属性的访问者－角色授权规则和基于客体标签的角色－权限指派规则，解决了动态用户授权和细粒度访问权限指派问题；分析了隐私策略冲突各种情况，实现了基于 Prolog 逻辑编程的策略一致性自动验证；设计了面向社会网络个性化隐私策略管理和实施中间件，将个性化隐私策略管理有效地集成到对既有资源的访问控制系统中，实验表明策略冲突分析具有良好的执行效率。

关键词：社会网络；隐私策略；个性化；冲突解决；Prolog
中图分类号：TP311 **文献标识码**：A **文章编号**：1000-436X(2012)Z1-0239-11

Specification and enforcement of personalized privacy policy for social network

WANG Yuan[1], SUN Yu-qing[1], MA Le-le[1,2]
(1. School of Computer Science and Technology, Shandong University, Jinan 250101, China;
2. Chinese Academy of Sciences, Beijing 100190, China)

Abstract: An authorization model was proposed to support personalized privacy preferences in the first-logic privacy preference language. The language allowed users to express personalized privacy preferences. Visitor-role authorization rules based on the attributes of visitors and role-permission assignment rules based on the tags of objects were introduced, which resolve dynamic authorization and fine-grained rights assignment problems. Analyzed privacy policy conflict cases and realized policy consistency verification by Prolog logic programming. Designed a personalized privacy policy management and implementation middleware for social network. In the middleware, it integrated the management of personalized policy into access control system on the existing resources. The experiments show that the policy conflict analysis has good efficiency in the implementation.

Key words: social network; privacy policy; personalized; conflict resolution; Prolog

1 引言

在线社会网络服务为用户提供了信息互动、共享、交友的平台。通过社会网络服务，用户可以创建个人主页、上传照片、发表日志、评论等，实现与朋友、家人信息分享；甚至帮助用户寻找志趣相同的新朋友、提供商务资讯分享、发现潜在合作对象等。随着社会网络的普及和发展，社交网站存储

收稿日期：2012-08-06
基金项目：国家自然科学基金资助项目（61173140）；山东大学自主创新基金资助项目（交叉基金 2010JC010）；山东省自然科学基金资助项目（Y2008G28）；中国科学院计算机系统结构重点实验室开放课题基金资助项目（ICT-ARCH200904）
Foundation Items: The National Natural Science Foundation of China(61173140); The Independent Innovation Foundation of Shandong University (2010JC010); The Natural Science Foundation of Shandong Province (Y2008G28); The Open Funding of Key Laboratory of Computer System and Architecture of Chinese Academy of Sciences(ICT-ARCH200904)

了大量用户个人数据，如 Facebook 中存储了约有 9 亿注册用户的属性信息，包括：姓名、年龄、工作经历等；同时每日产生约 30 亿的个人数据包括网络链接、日志、博客、照片等[1]。社会网络中用户的隐私保护主要是对用户不愿意完全公开的个人数据的保密，如敏感的属性信息，涉及个人隐私的数据资源以及用户间关系等。在线社会网络中的隐私保护是当前数据安全研究领域的一个热点问题。

在开放性的社会网络中，用户既是自己数据的资源拥有者，同时又是其他用户资源的访问请求者，而服务提供方则是资源管理者。社会网络的这些特点使得用户存放在社会网络上的个人数据具有可扩展性和动态性，资源访问者具有不确定性和灵活性，因此，面向社会网络隐私保护方案需要满足以下要求。

1) 个性化隐私策略：支持用户定义满足其隐私偏好的数据访问规则；

2) 一致性授权：由于隐私策略定义的灵活性容易出现逻辑上的不一致，需要对隐私策略进行一致性验证，保证隐私策略正确、有效执行；

3) 策略可实施性：能够依据用户的个性化隐私策略实施有效的、细粒度的访问控制；

4) 自动推理：通过规则的逻辑推理，解决大量未知用户的访问请求问题以及动态资源的授权管理问题，实现自动、灵活的访问控制；由于数据资源的动态性、多样性，引起用户隐私偏好是变化、复杂的，需要制订策略一致性推理规则，实现策略冲突的自动化检测。

在社会网络中，普遍采用的隐私保护方法是基于策略的授权管理，根据用户参与策略制定程度的不同，可以分为 3 类：默认的隐私设置，是由社交网站预先提供给用户的隐私设置，它能够保证用户基本的隐私安全，但不能满足不同用户多样式、个性化的隐私保护需求；自适应的隐私设置[2,3]，是社交网站通过分析抽取的用户输入、特征信息以及上下文环境等信息，自动推理出用户的隐私设置，其隐私设置的准确性取决于抽取信息的精确性以及与用户的直接交互，与用户交互越多，推理的用户隐私意愿越准确，但是对大量未知、可变的隐私数据频繁交互，会造成用户的负担，不能满足用户自动、灵活的访问控制需求；自定义的隐私设置，是基于社交网站设计的访问控制模型，用户根据自身需求完成的隐私配置。目前，主要的访问控制模型有：基于角色的访问控制模型(RBAC)[4]，通过预先设定角色（基本角色、好友角色、群组角色）以及对应的访问权限（对社会网络内公开的内容，对好友公开的内容、对群组内公开的内容）实现资源的访问控制，这种方法主要针对确定用户群体的环境，不能解决社会网络中未知用户和动态资源的访问授权问题。基于属性的访问控制模型(ABAC)[5,6]，较好地解决了这一问题，授权表示为基于属性的规则集合，通过主体属性、客体属性及环境属性约束实现开放式环境中动态访问控制。如面向电子图书馆的授权模型实施基于资源内容和用户属性的动态访问控制[7]，改变传统的基于用户身份的权限分配，适合于开放环境中匿名用户访问请求和动态资源的授权管理。但是这种模型仅适用于资源拥有者和资源管理者是一体的情况，由管理者制定访问控制策略，不适用于社会网络中资源拥有者和资源管理者分离的情况，不能满足社会网络用户个性化隐私偏好的需求。基于规则的访问控制模型[8]，规则定义访问者与资源拥有者之间的关系类型，最大拓扑距离以及最小信任度等限制条件，只有访问者证明自己满足规则约束条件才能获得授权，实现了基于规则推理的自动、灵活的访问控制，提高了资源拥有者对资源传播的控制能力，但是由于规则数量众多，容易产生策略冲突，该模型缺乏策略一致性验证，不能保证一致性授权和策略的有效实施。基于授权规则的 RBAC 模型(RB-RBAC)[9]，在 RBAC 模型的基础上增加了用户属性和权限分配规则 2 个概念，实现了动态角色—权限分配，但该模型不满足社会网络中用户自定义隐私策略的需求，以及缺少授权规则的管理和规则冲突的解决。

在隐私策略分析与验证方面，现有研究工作主要针对封闭环境中策略分析，如基于模型检测方法针对角色访问控制的管理系统(ARBAC)中的策略冲突，自动验证是否存在不满足安全属性约束（如职责分离约束）的可达系统状态[10,11]。这些方法不适用于开放的动态隐私策略的社会网络环境。另一种可视化隐私策略评估系统[12]，分析社会网络中的隐私策略，从访问者角度进行隐私评估，通过可视化技术帮助用户理解隐私策略的含义以及作用效果，但这种方法仅验证隐私策略的执行效果，并没有分析可能出现的隐私策略冲突。综上，现有研究尚缺乏从策略定义、分析与验证，到策略动态实施的完整过程。

针对上述不足，本文提出了支持个性化隐私偏好的授权模型，采用基于一阶逻辑的隐私偏好描述方法，支持用户自定义个性化的动态隐私策略，借助逻辑编程方法进行自动化的策略一致性分析，并实施基于推理规则的访问授权，开发了面向社会网络的个性化隐私策略管理和实施中间件 Privacy Holder，实验验证了模型的可行性，并分析自动化策略冲突检测的执行效率。本文其余部分组织如下：第 2 节给出个性化隐私策略描述和授权模型；第 3 节讨论隐私策略冲突形式；第 4 节实现隐私策略一致性验证；第 5 节介绍系统实现以及实验分析；第 6 节总结全文和提出未来工作。

2　支持个性化隐私偏好的授权模型

2.1　基本概念

隐私策略主要是由主体、客体、动作以及权限授权需要满足的约束条件等组成。主体是指发起对资源访问请求操作动作的用户，即访问者，所有的主体集合表示为 *US*。客体是指访问者试图访问的秘密资源，所有的客体集合表示为 *RS*。根据资源类型的不同，分为属性信息和数据资源，属性信息是指用户个人资料如姓名、年龄、性别、身份证号码、家庭地址和职业等，数据资源是指用户发布的个人日记、相册、视频和音乐等。动作是指主体在客体上执行的操作，如访问、阅读，评论，分享等，所有的动作集合表示为 *ActS*。权限是指在某个客体上操作动作,表示为<*r*,*a*>，其中 $r \in RS$，$a \in ActS$，所有权限的集合表示为 *PES*。

定义 1　主体属性 由属性名和属性值组成，是主体基本的属性信息。

主体属性包含访问者的姓名、性别、地址、教育程度、年龄、工作/学校、爱好等。

定义 2　客体标签 由标签名和标签值组成，是对数据资源进行标识的方法。

由于用户拥有的数据资源众多，为实现灵活的数据资源的分组和细分，用户可以为数据资源添加“类型”、“时间”、“地点”、“重要程度”等标识。

定义 3　角色 (简记为 *role*) 是针对主体属性需求划分的用户分组，通过角色与权限进行关联，隔离主体与权限直接的逻辑关系，所有的角色集合表示为 *RoleS*。

定义 4　角色层次(简记为 *RH*) 角色集上定义的一种偏序关系，定义 *N* 组角色（$role_1,\cdots,role_n$），$RH \subseteq role_i \times role_j$，$role_i \in RoleS$，当满足 $role_i \geqslant role_j$ 时，$role_i$ 称为高级角色，$role_j$ 为低级角色，高级角色继承低级角色的权限，低级角色继承高级角色的用户。

定义 5　谓词(简记为 *prd*)　描述某个实体具有某种属性或者多个实体之间存在某种关系，由谓词名和参数两部分组成，表示为 $Predicate(x_1, x_2,\cdots,x_k)$，其中 *Predicate* 是谓词名，$x_i$ 可以是变量、常量或者是一阶谓词，$i=1,\cdots, k$，所有的谓词集合表示为 *PS*。

例如：*Is*(*x.role*, '*classmate*')，表示主体 *x* 的角色是同学；*Is*(*y.tag*, '*red* ')，表示客体 *y* 的标签是红色。

定义 6　约束　描述访问授权需要满足的基本条件和限制，可以为谓词或谓词的逻辑表达式，表示为：$prd_1 \Theta\ prd_2 \cdots \Theta\ prd_n$，其中Θ表示逻辑与(∧)、或(∨)操作，$prd_j \in PS$，$j=1,\cdots,n$，表示谓词。

根据约束内容的不同，约束主要分为以下 3 类。

1) 主体属性约束 是指主体的年龄、性别、地址、爱好等约束，所有主体约束集合表示为 *SAttrC*。例如：*Larger*(*x.age*, '18') ∧ *Is*(*x.city*, '*jinan*')，要求访问者 *x* 的年龄大于 18 岁且所在城市为济南；

2) 客体标签约束　是指数据资源权限授权所需的客体标签条件，所有客体约束集合表示为 *RtagC*。例如：*Is*(*y.type*, '*photo*') ∧ *Is*(*y.tag*, '*party*')，表示访问授权客体仅为标记聚会照片的数据资源；

3) 环境约束　是指系统状态、上下文环境等约束，所有环境约束集合表示为 *EC*。例如：*TimeWithin* ('8:00*AM*','6:00*PM*')表示时间约束[8:00AM, 6:00PM]；*particiated*(*x*, '*party*')，表示主体 *x* 参加过聚会的历史事件约束；本文将上下文环境约束用于隐私策略的制定中，使得访问控制策略具有实时性以及良好的交互性，进一步提高了用户隐私资源的安全性。

2.2　支持个性化隐私偏好的授权模型

本文提出支持个性化隐私偏好的授权模型，对基于角色的访问控制模型进行扩展，增加了基于主体属性的访问者－角色授权规则和基于客体标签的角色—权限指派规则，如图 1 所示。在模型中，根据访问者—角色授权规则，满足主体属性约束的访问者获得角色授权；根据角色—权限授权规则，满足客体标签约束的权限指派给相应角色，同时也包含由角色的层次关系引起的权限继承；最终，用户通过角色以及角色层次关系获得权限授权。其相关规则定义如下。

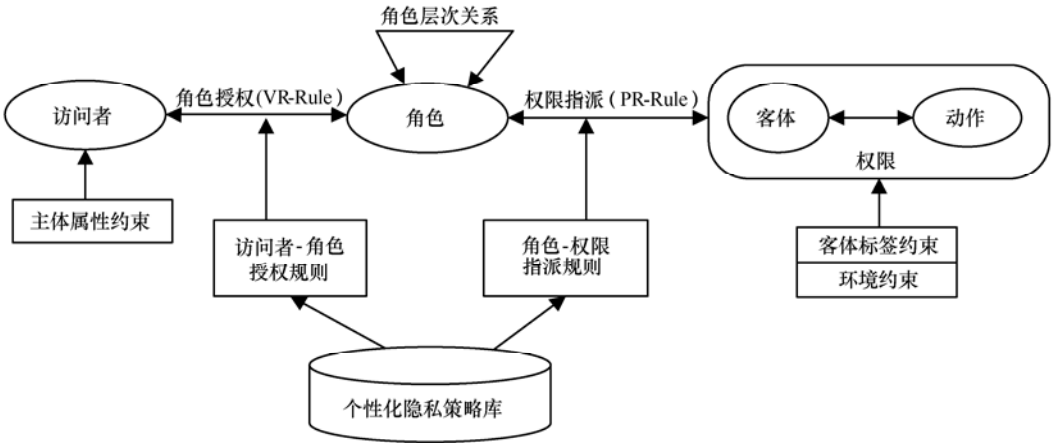


图 1　支持个性化隐私偏好的授权模型

定义 7　访问者—角色授权规则(简记为 *VR-Rule*)：*assign_role*(*u*,*role*) ← $Q_1x_1 \cdots Q_mx_m(sc_1 \Theta sc_2 \cdots \Theta sc_n)$，其中 $u \in US$，$role \in RoleS$，Θ为逻辑与(∧)、或(∨)操作，$sc_i \in SAttrC$，$i=1,\cdots,n$，x_j 为实体变量，$j=1,\cdots,m$，$Q_i \in \{\exists;\forall\}$，∃为存在量词，∀为全称量词，表示若访问者 *u* 满足所有主体属性约束，获得角色 *role*。

实例 1　*assign_role*(*x*, '*friend*') ← ∀ *x Larger*(*x.age*, '25') ∧ *Is*(*x.city*, '*Jinan*') ∧ *Is*(*x.hobby*, '*swimming*') 表示年龄超过 25 岁且爱好游泳的济南人为好友。

定义 8　角色—权限指派规则(简记为 *PR-Rule*)：*P_assign*[*D_assign*] (*role*,*r*,*a*) ← $Q_1x_1 \cdots Q_mx_m(re_1 \Theta re_2 \cdots \Theta re_n)$，其中 $role \in RoleS$，$r \in RS$，$a \in ActS$，Θ为逻辑与(∧)、或(∨)操作，$re_i \in \{RtagC; EC\}$，$i=1,\cdots,n$，x_j 为实体变量，$j=1,\cdots,m$，$Q_i \in \{\exists;\forall\}$，∃为存在量词，∀为全称量词，*P_assign* 为正授权，*D_assign* 为负授权，表示在环境约束 *EC* 下，满足所有的客体标签约束的权限<*r*,*a*>指派[禁止指派]给角色 *role*。

实例 2　*P_assign* ('*friend* ',*y*, '*comment*') ← ∃ *y Is*(*y.type*, '*photo*') ∧ *Is*(*y.tag*, '*party*')，表示好友可以评论聚会照片。

定义 9　隐私策略　同一用户定义的访问者—角色授权规则和角色—权限指派规则的规则集合。

实例 3　隐私策略={$VR\text{-}Rule_1$, $PR\text{-}Rule_1$},其中 $VR\text{-}Rule_1$：*assign_role*(*x*,'*friend*') ← ∀ *x Larger*(*x.age*, '25') ∧ *Is*(*x.city*, '*Jinan*') ∧ *Is*(*x.hobby*, '*swimming*')；$PR\text{-}Rule_1$：*P_assign* ('*friend*',*y*, '*comment*') ← ∃ *y Is*(*y.type*, '*photo*') ∧ *Is*(*y.tag*, '*party*'). 系统中存储 *Alice* 用户信息为：*name*=*Alice*, *age*=35, *city*= *Jinan*，*hobby*={*swimming*, *music*},客体 *photo*1 的标签为：*type*= *photo*, *tag* ={*party*, *red*}，根据 $VR\text{-}Rule_1$ 用户 *Alice* 授予好友角色，同时资源 *photo*1 满足客体标签约束，则 *Alice* 通过好友角色获得评论照片 *photo*1 的权限。

支持个性化隐私偏好的授权模型的优势体现在 2 个方面：一方面采用基于一阶逻辑的隐私策略描述语言，满足用户的个性化隐私需求，如对资源保护的细粒度授权需求、能够明确表达用户的隐私意愿的明确语义需求、以及支持未明确描述的策略推理授权等；另一方面实现了社会网络中未知用户和大量、动态数据资源的访问控制。通过基于主体属性约束的角色授权推理，实现了自动、动态的访问者—角色授权，解决了社会网络中未知用户的访问请求问题；由于社会网络中用户拥有的数据资源众多，且经常添加、修改，传统的针对某个具体资源定义授权变得难以维护，提出基于客体标签约束的角色—权限指派规则，实现了对大量、动态资源的权限指派。

3　隐私策略冲突分析

由于规则的主体属性、资源属性以及动作属性之间存在重叠或者层次关系，在制定隐私策略时可能出现逻辑不一致的情况，如在不同的策略中，对相同主体、客体既有正向授权，又有拒绝授权（负授权），造成隐私策略冲突。根据策略冲突发生的原因是否与具体数据相关，可以分为逻辑冲突和实例冲突。

1) 逻辑冲突指策略定义过程中所出现的逻辑

上的不一致，如角色矛盾授权，是指不同策略对同一角色既有正授权又有负授权的逻辑冲突。

实例 4　隐私策略={*VR-Rule*$_1$, *PR-Rule*$_1$, *PR-Rule*$_2$}，其中 *VR-Rule*$_1$:*assign_role*(*x*,'*groupmember*') ← ∀ *x Is*(*x.project*,'*mobile Application*')；*PR-Rule*$_1$: *P_assign* ('*groupmember*',*y*, '*read*') ← ∃ *y Is*(*y.type*, '*log*') ∧ *Is*(*y.tag*, '*work*') ∧ *TimeWithin*('8:00*AM*', '6:00 *PM*')；*PR-Rule*$_2$: *D_assign*('*groupmember*',*y*, '*read*')← ∃*y Is*(*y.type*, '*log*') ∧ *Is*(*y.tag*, '*work*') ∧ *DayWithin* ('*Saturday*', '*Sunday*')。*VR-Rule*$_1$ 表示参与相同项目的为小组成员；*PR-Rule*$_1$ 表示小组成员可以在 8：00 到 18：00 查看工作日志；*PR-Rule*$_2$ 表示小组成员不能在周末查看工作日志。

另一种典型的逻辑冲突为权限继承冲突，是指由角色层次关系引起蕴含授权与显式授权的矛盾，如图 2 所示，其中圆形表示角色，方形表示权限，+P 和-P 分别表示对相同资源的正、负授权，箭头表示角色层次关系，实线表示已经存在的角色—权限指派关系，虚线表示新增加的角色—权限指派关系。根据权限在角色层次中的继承关系[13]，当低级角色被指派正授权时，按照正授权由低级角色到高级角色正向传播，高级角色继承低级角色的正授权，如果新增加高级角色的负授权，则与低级角色正授权矛盾，引起策略冲突，如图 2(a)所示；当低级角色被指派负授权时，新增加高级角色的正授权，不会引起策略冲突。当高级角色被指派负授权时，按照负授权由高级到低级反向传播，高级角色对资源的负授权一定蕴含着低级角色的负授权，如果新增加低级角色的正授权，则与高级角色正授权矛盾，引起策略冲突，如图 2(b)所示；当高级角色存在多个低级角色，并且低级角色之间存在互斥的权限，如果新增加高级角色的负授权，则与低级角色的负授权矛盾，引起策略冲突，如图 2(c)所示。

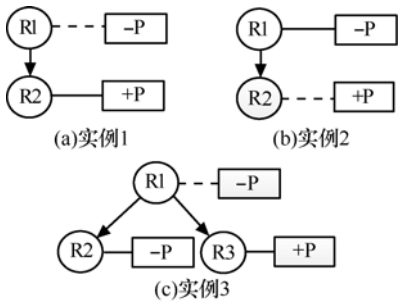


图 2　权限继承冲突实例

实例 5　隐私策略={*VR-Rule*$_1$,*VR-Rule*$_2$, *PR-Rule*$_1$, *PR-Rule*$_2$}，其中 *VR-Rule*$_1$:*assign_role*(*x*, '*schoolmate*') ← ∀ *x Is*(*x.graducate*, '*Shandong University*')；*VR-Rule*$_2$：*assign_role*(*x*,'*classmate*')← ∀ *x Is*(*x. graducate*, ' *Shandong University* ') ∧ *Is*(*x.class_name*, '0122-41')；*PR-Rule*$_1$: *P_assign*('*schoolmate*',*y*,'*tag*') ← ∃ *y Is*(*y.type*, '*log*') ∧ *Is*(*y.tag*, '*personal*'); *D_assign* ('*classmate*',*y*, '*tag*') ← ∃ *y Is*(*y.type*, '*log*') ∧ *Is*(*y.tag*, '*personal*'). *VR-Rule*$_1$ 表示毕业于山东大学为校友; *VR-Rule*$_2$ 表示毕业于山东大学且班级名称为 0122-41 为同班同学；*PR-Rule*$_1$ 表示校友可以标记个人日志；*PR-Rule*$_2$ 同班同学不能标记个人日志。根据 *VR-Rule*$_1$，*VR-Rule*$_2$ 可知，同班同学≥校友，由于角色层次关系，同班同学继承好友角色的正授权可以标记个人日志，但 *PR-Rule*$_2$ 显式定义同班同学不能标记个人日志，造成策略冲突。

2) 实例冲突指就策略定义本身不存在逻辑冲突，但是由于数据库中存在实例，引起策略冲突条件触发而造成的冲突。在支持个性化隐私偏好的授权模型中，用户通过 *VR-Rule* 和 *PR-Rule* 共同作用获得访问授权，在两种规则定义过程中，可能存在某个用户实例同时满足两种角色约束，导致同时适用于两条相反的策略，造成策略冲突。

实例 6　隐私策略={*VR-Rule*$_1$,*VR-Rule*$_2$, *PR-Rule*$_1$, *PR-Rule*$_2$}，其中 *VR-Rule*$_1$：*assign_role*(*x*, '*friend*') ← ∀ *x Larger*(*x.age*,'25') ∧ *Is*(*x.city*, '*Jinan*') ∧ *Is*(*x. hobby*, '*swimming*')；*VR-Rule*$_2$：*assign_role* (*x*,' *groupmember*') ← ∀ *x Is*(*x.project*, '*mobileApplication*')；*PR-Rule*$_1$：*P_assign*('*friend*',*y*,'*comment*') ← ∃ *y Is*(*y.type*, '*photo*') *Is*(*y.tag*, '*party*')；*PR-Rule*$_2$：*D_assign*('*groupmember*', *y*, '*read*') ← ∃ *y Is*(*y.type*, '*photo*') *Is*(*y.tag*, '*red*')。*VR-Rule*$_1$ 表示年龄超过 25 岁且有相同爱好的同城人均为好友；*VR-Rule*$_2$ 表示参与相同项目的为小组成员；*PR-Rule*$_1$ 表示好友可以评论聚会照片；*PR-Rule*$_2$ 表示小组成员不能查看标记为红色的照片。系统中存储 *Anny* 用户信息为 *name*= *Anny*, *age*=28,*city*= *Jinan*, *hobby*= {*swimming*, *music*}, *profession*=*computer*, *project*= *mobileApplication*，*Anny* 满足好友和小组成员 2 个角色的主体约束条件，同时拥有 2 个角色；客体 *photo*$_1$ 的标签为：*type*= *photo*, *tag* ={*party*, *red*}，根据 *PR-Rule*$_1$ 规定 *Anny*

可以评论照片 $photo_1$，但 $PR\text{-}Rule_2$ 规定 *Anny* 不能查看照片 $photo_1$，引起策略冲突。

4 隐私策略一致性验证

为了有效地分析隐私策略的矛盾，采用逻辑编程的验证方法[14]，使用有较强的逻辑表达能力和推理能力的 Prolog 语言作为程序设计语言，将用户定义的隐私策略转化为逻辑形式，通过规则推理，实现隐私策略冲突的自动化检测。其具体过程如图 3 所示，分为以下几步：a. 用户定义个性化隐私策略；b. 根据隐私策略，设计 Prolog 形式的访问授权推理规则和策略冲突规则；c. 借助 Prolog API 接口实现用户对策略授权指派和策略冲突的查询；d. 根据冲突查询请求，调用逻辑转化程序将关系数据库中存放的数据和隐私策略转化为 Prolog 事实；e. 推理引擎依据已有的事实和推理规则，完成用户授权和策略冲突的自动化推理；f. 策略冲突结果显示，采用与用户交互的方式，完成冲突策略的修正。

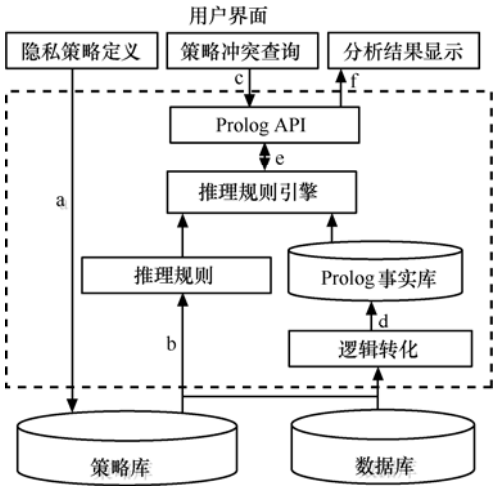


图 3 隐私策略一致性验证流程

4.1 构建事实库

事实是指已经存在的实体和实体间的关系，由谓词名及变量组成。通过逻辑转化程序将数据库和策略库中存储的关系数据转化为Prolog可识别的事实语句，作为逻辑推理的基础。其转化过程为：首先，根据用户查询请求，按照不同表格内容，从数据库如用户表、客体标签表、客体表、权限表等和策略库如用户—角色分配表、角色—权限分配表等提取数据。其次，调用不同表格逻辑转化方法，将提取的数据转化为 Prolog 事实语句。例如：用户表(*User_*)存储 001 号用户信息为：*u_id*=001, *name*=*David*, *age*= 23,*city*= 济 南 ,*online_time*=100, *graudate*=山东大学，调用格式转化算法将用户主体属性解析为 Prolog 事实，表示为 *user_*('001', '*u_id*', '001').user_('001','*name*','*David*').user_('001','*age*',23). user_('001','*city*','济南').user_('001', '*online_time*',100). user_('001','*graudate*',山东大学)。最后，调用文件读写方法将转化后的逻辑语句写入 Prolog 文件中，为推理引擎提供事实库。

4.2 设计推理规则

推理规则描述事实之间的依赖关系，形式为：h:-b_1,b_2,…,b_n，其中 h 为规则头，表示规则的结论，b_1,b_2,…,b_n 为规则体，表示规则成立的条件。根据支持个性化隐私偏好的授权模型，设计 Prolog 形式推理规则如下：

1) 访问者—角色授权规则：表示 Prolog 事实库中存在角色 *Role* 和用户名为 *UserName* 的访问者 *U*, 同时访问者 *U* 主体属性满足角色 *Role* 所有主体属性约束条件，则访问者 *U* 被授权角色 *Role*。

assign_role(*UserName*,*Role*):-*role*(*_*,*Role*),*user_*(*User*,'*name*',*UserName*,*_*),*role_on_attr*(*Role*,*U_attr*, *U_attr_range*),*user_*(*User*,*U_attr*,*U_attr_value*,*U_attr_ type*),*match*(*U_attr_type*,*U_attr_value*,*U_attr_range*).

2) 角色—权限指派规则：角色获得权限不仅包含直接的权限指派，还包含由于角色层次关系引起的权限继承，因此本规则主要包含 3 个部分：①直接的权限指派，*permission_limit* 定义给定标签的客体访问权限，*object_tag* 和 *match* 查询标签取值匹配的客体，*assign_PER* 和 *permission* 将符合条件的客体访问权限赋予制定角色 *Role*；②低级角色 *Child* 被指派正授权，根据正授权正向传播，则高级角色 *Ancestor* 继承低级角色正授权；③如果高级角色 *Ancestor* 被指派正授权，根据负授权反向传播，低级角色 *Child* 继承高级角色负授权。具体转换规则如下。

role_has_pe(*PermissionLimit*,*Role*):-*permission_ limit*(*PermissionLimit*,*O_tag_name*,*O_tag_range*,*Actions*,*Flag*),*object_tag*(*Object*,*O_tag_name*, *O_tag_type*, *O_tag_value*),*match*('*char*',*A_id*,*Actions*),*match*(*O_tag_type*,*O_tag_value*,*O_tag_range*),*permission*(*Object*, *A_id*), *assign_PER*(*Role*, *PermissionLimit*).

role_has_pe(*PermissionLimit*,*Role*):- *is_ancestor* (*Child*,*Ancestor*), *role_has_pe*(*PermissionLimit*,*Child*) *permission_limit* (*PermissionLimit*,*O_tag_name*, *O_*

tag_ range, *Actions*, *'grant'*).

role_has_pe(*PermissionLimit*,*Role*):- *is_ancestor* (*Child*, *Ancestor*), *role_has_pe*(*PermissionLimit*, *Ancestor*), *permission_limit* (*PermissionLimit*, *O_tag_ name*, *O_tag_range*, *Actions*, *'deny'*).

3) 用户—权限授权规则 表示通过访问者—角色授权和角色—权限指派推理实现用户—权限授权。

user_has_permission(*User*,*Role*,*PermissionLimit*): -*assign_role*(*User*,*Role*),*role_has_pe*(*Permission Limit*, *Role*).

4)匹配规则 自定义的内部函数 *match* 用来判别数值 *Value* 是否在给定的范围 *Range* 之内。主要分为以下 3 种情况进行。

情况 1 不考虑数据类型，数据值 *Value* 与范围 *Range* 完全相同, *match* 成立。

match(*Type*,*Value*,*Range*):-*constant*(*Range*),*const ant*(*Value*),*be_equal*(*Value*,*Range*).

be_equal(*Value*,*Range*):-*Value*=*Range*.

情况 2 判断数字类型数据匹配关系

match(*Type*,*Value*,[*M*,*N*]):-*Type*='*int*',*constant*(*M*), *constant*(*N*),*contain_num*(*Value*,[*M*,*N*]).

contain_num(*Value*,[*M*,*N*]):-*constant*(*Value*), (*M*\='*null*',*N*\='*null*', *M*≤*N*,*Value*≥*M*,*Value*≤*N*).

contain_num(*Value*,[*M*,*N*]):-*constant*(*Value*), (*N*='*null*',*M*\='*null*',*Value*≥*M*).

contain_num(*Value*,[*M*,*N*]):-*constant*(*Value*), (*M*='*null*',*N*\='*null*',*Value*≤*N*).

contain_num([*P*,*Q*],[*M*,*N*]):-*contain_num*(*P*,[*M*, *N*]), *contain_num*(*Q*,[*M*,*N*]).

情况 3 判断字符类型数据匹配关系

match(*Type*,*Value*,[*M*|*N*]):-*Type*='*char*',*constant*(*M*),*contain_string*(*Value*,[*M*|*N*]).

contain_string(*Value*,[*M*|*N*]):-*constant*(*Value*), *member*(*Value*,[*M*|*N*]).

contain_string([*P*],[*M*|*N*]):-*constant*(*P*),*member* (*P*,[*M*,*N*]).

contain_string([],_).空列表包含于任何的取值范围

contain_string([*P*|*Q*], *R*): -*contain_string*(*P*, *R*), *contain_string*(*Q*,*R*).

5) 逻辑冲突规则 通过定义互斥权限 *mute PermLimit* 即在相同客体上同时拥有*'grant'*和*'deny'* 访问授权以及角色—权限指派规则 *role_has_pe*，判断角色是否拥有互斥权限。

role_pe_conflict(*Role*,*PermLimit*1,*PermLimit*2):- *role_has_pe*(*PermLimit*1,*Role*),*role_has_pe*(*PermLimi t*2,*Role*),*mutePermLimit*(*PermLimit*1,*PermLimit*2).

mutePermLimit(*PermLimit*1,*PermLimit*2): -*permission_limit*(*PermLimit*1,*O_tag_name*,*O_tag_ra nge*,*Actions*,*'grant'*),*permission_limit*(*PermLimit*2, *O_tag_name*,*O_tag_range*,*Actions*,*'deny'*).

6) 实例冲突规则 通过用户-权限授权规则 *user_has_permission* 和给定标签的客体访问权限 *permission_limit* 定义正、负授权，判断用户是否拥有互斥权限。

conflict(*User*,*Object*,*Action*):- *grant_user_access* (*User*,_,_,*Object*,*Action*),*deny_user_access*(*User*,_,_, *Object*,*Action*).

grant_user_access(*User*,*Role*,*PermissionLimit*,*O bject*,*Action*):-*user_has_permission*(*User*,*Role*,*Permis sionLimit*),*permission_limit*(*PermLimit*,*O_tag_name*, *O_tag_range*,*Actions*,*'grant'*).

deny_user_access(*User*,*Role*,*PermissionLimit*,*Ob ject*,*Action*):-*user_has_permission*(*User*,*Role*,*Permissi onLimit*),*permission_limit*(*PermLimit*,*O_tag_name*,*O_ tag_range*,*Actions*,*'deny'*).

4.3 策略冲突查询

通过对 Prolog 策略冲突规则的查询请求分析，完成策略一致性验证，主要包括：直接冲突查询、个性化定制查询。直接冲突查询是指不设定查询限制范围，按照策略冲突推理规则(5)，(6)，直接进行策略冲突的查询。该查询只提供了策略冲突检测结果，并没有列举出策略冲突的原因，为帮助用户查找策略冲突原因，实现策略冲突的修正，提供以下 2 种规则的查询。

1) 逻辑冲突路径规则：给出用户—角色授权冲突发生的完整授权路径，便于用户获得逻辑冲突发生的授权路径。

role_pe_detail(*Role*,*PermLimit*,*O_tag_name*,*O_t ag_range*,*Actions*,*Flag*):-*role_has_pe*(*PermLimit*,*Role*), *permission_limit*(*PermLimit*,*O_tag_name*,*O_tag_range*, *Actions*,*Flag*).

2) 实例冲突路径规则：给出用户—权限授权冲突发生的完整授权路径, 便于用户获得实例冲突发生的授权路径。

query_conflict_uoa_trace(*User*,*Role*1,*Role*2,*Per*

*missionLimit*1,*PermissionLimit*2,*Object*,*Action*):-*user_has_permission*(*User*, *Role*, *PermissionLimit*), *permission_limit* (*PermissionLimit*1, *O_tag_name*, *O_tag_ range*, *Actions*, *'grant'*), *permission_limit* (*PermissionLimit*2, *O_tag_name*,*O_tag_range*, *Actions*, *'deny'*).

由于直接冲突查询采用枚举的方式，当规则数量众多时执行效率低，为此，增加个性化定制查询，用户可以自定义查询限制范围，实现了快速验证和准确定位策略冲突原因。

用户授权路径规则：给出访问者授权的完整路径，用户可以根据自身需要限制某些变量，实现个性化定制查询，查找整个授权中可能存在的策略冲突。

query_policy_trace(*User*,*Role*,*PermissionLimit*, *Object*, *Action*): *-user_has_permission* (*User*, *Role*, *PermissionLimit*),*permission_limit*(*PermissionLimit*, *O_tag_name*, *O_tag_ range*, *Actions*,*Flag*), *permission*(*Object*, *A_id*), *object_tag* (*Object*, *O_tag_name*, *O_tag_type*, *O_tag_value*),*match*(*'char'*,*A_id*,*Actions*), *match*(*O_tag_type*,*O_tag_value*,*O_tag_range*).

根据策略冲突类型和策略冲突原因，设置个性化定制查询主要包括：

给定用户的角色查询 *query_policy_trace* (*User*,? *Role*,_,_,_)查询用户 *User* 承担的所有角色，判断是否包含用户 *User* 不满足主体属性约束条件的角色 *Role*，从而断定用户—角色授权规则是否完备。

给定角色的权限查询 *query_policy_trace*(_, *Role*,? *PermissionLimit*,_,_) 查询角色 *Role* 授权的所有权限，包括 2 个方面：1)直接的权限授权；2)由于角色层次关系引起的权限继承，判断是否存在逻辑冲突。

给定客体的角色查询 *query_policy_trace*(_,? *Role*,_,*Object*,*Action*)查询对于资源 *Object* 可以执行动作 *Action* 的所有角色，判断是否包含不满足客体标签约束条件的权限<*Object*, *Action*>被指派给角色 *Role*，从而断定角色-权限指派规则是否完备。

给定用户的权限查询 query_policy_trace (User,_,? PermissionLimit,_,_)查询用户 User 授权的所有权限，实现对用户授权结果的分析，判断是否存在实例冲突。

5　系统实现与数据分析

5.1　面向社会网络的个性化隐私策略管理和实施中间件

为了将支持个性化隐私偏好的授权模型有效地集成到现有的社会网络系统中，本文设计实现了个性化隐私策略管理和实施中间件 Privacy Holder，允许用户定义个性化的细粒度隐私策略、进行隐私策略一致性验证和实施基于隐私策略的访问控制，其系统结构图如图 4 所示。主要组成部分：用户属性库、资源属性库、环境属性库、Prolog 知识库、隐私策略库、策略管理模块、策略分析模块、策略评估模块等，其中用户属性库存在用户主体属性信息，资源属性库存放资源属性信息以及资源标签，环境属性库存放上下文环境信息，Prolog 知识库存放 Prolog 形式的访问授权规则、策略冲突规则以及转化后的 Prolog 事实，隐私策略库存放一致性的隐私策略。

1) 策略管理模块：实现个性化的隐私策略定义。主要功能包括：通过创建角色，新建角色以及角色层次关系；通过创建资源标签，添加数据资源标签；通过用户—角色授权，其定义界面如图 5(a)所示，定义角色主体约束条件的取值范围完成角色指派；通过角色—权限指派，其定义界面如图 5(b)所示，定

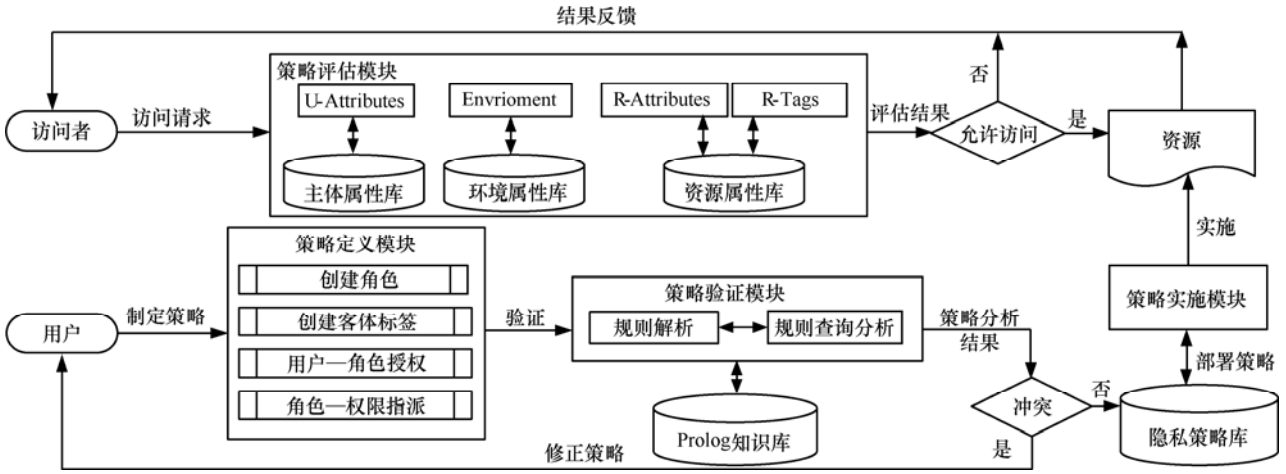
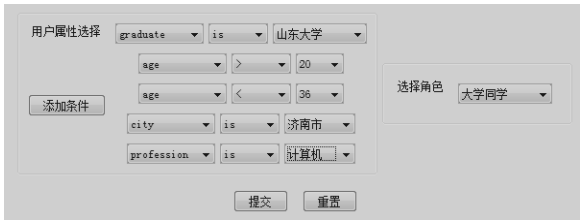


图 4　面向社会网络的个性化隐私策略管理和实施中间件系统结构

义客体约束条件、授权动作以及授权角色完成权限指派；同时提供以上定义的修改、删除功能。



(a) 用户—角色授权界面

(b) 角色—权限指派界面

(c) 策略冲突检测界面

图 5　隐私策略管理与冲突检测系统界面

2) 策略分析模块：实现自动化的策略冲突检测，其检测界面如图 5(c)所示。主要功能包括：规则解析部分，将用户隐私策略、数据库数据自动解析为 Prolog 事实，并将结果存储到 Prolog 文件中。规则查询分析部分，用户根据预先定义的策略冲突规则进行查询分析请求，根据查询结果判断策略是否满足一致性，并对冲突的策略提示修正，保证策略的正确执行。

3) 策略评估模块：实现基于策略的访问控制。当访问者发出对某资源访问请求时，查询主体属性、资源属性、环境属性等信息，将访问者主体属性与主体属性约束进行匹配得到访问者角色，遍历角色集合的授权权限，提取数据库中有关访问资源的信息，与权限集合中客体属性、标签进行匹配，如果匹配成功则资源对访问者开放，并进行指定的操作，否则拒绝访问者的请求，并将访问评估结果反馈给访问者。

系统数据库主要由用户信息表(User)、角色表(Role)、客体标签表(Object_tag)、客体表(Object)、动作表(Action)、权限表(Permission_Limit)以及用户—角色分配表(U-R)、角色-权限分配表(R-P)组成。通过用户—角色分配表映射出用户表与角色表一对一、一对多以及多对多的关系，同理，角色—权限分配表也映射出角色表与权限表一对一、一对多以及多对多的关系，其数据库 E-R 图如图 6 所示。

5.2　数据分析

Privacy Holder 系统实验环境为 CPU：IntelPentium(R) 1.86GHZ，内存：512M，软件环境：Windows XP，开发语言：java，开发工具：MyEclipse+Sql2000，推理工具：tuProlog，引用 java-Prolog 接口：tuPrologIDE。

针对冲突查询的不同方式，首先测试用户数量对查询性能的影响。假设用户信息表设置 10 项主体属性，按照每增加 10 名用户为一组进行实验，每组查询进行 50 次测试，计算其查询时间的平均值，共进行 10 轮，得到实验结果如图 7(a)所示。其中直接冲突查询是指不设定查询限制范围直接对实例冲突规则进行查询，即 *conflict*(*User*,*Object*, *Aciton*)，用虚线表示，个性化定制查询是指限制某些变量对用户授权路径规则进行查询，即 *conflict*

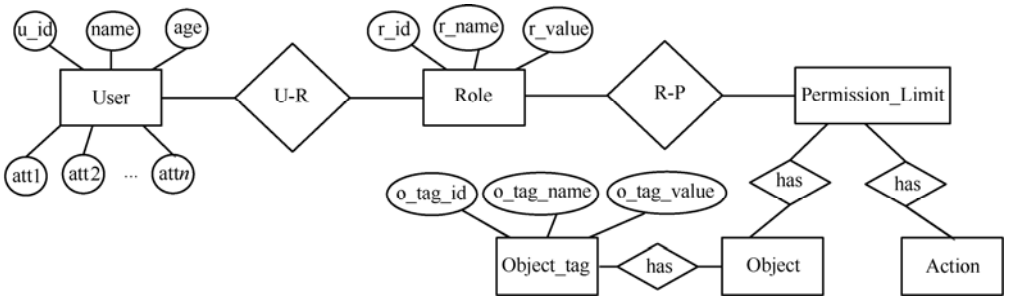
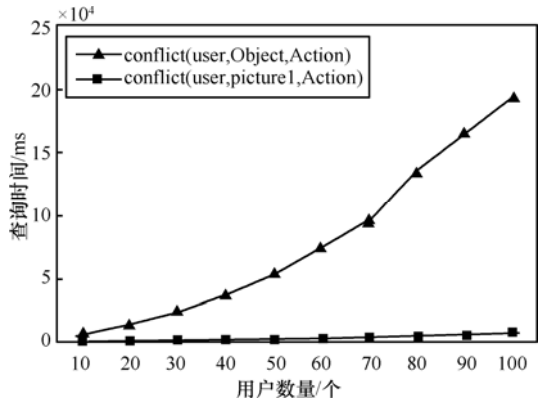
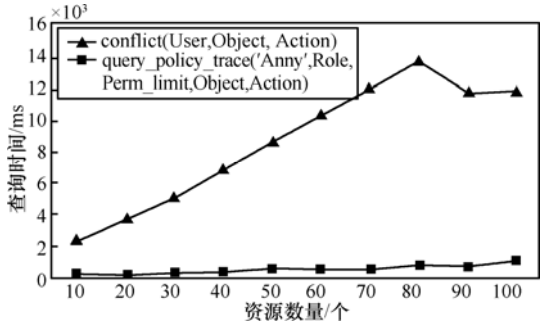


图 6　系统数据库 E-R 图

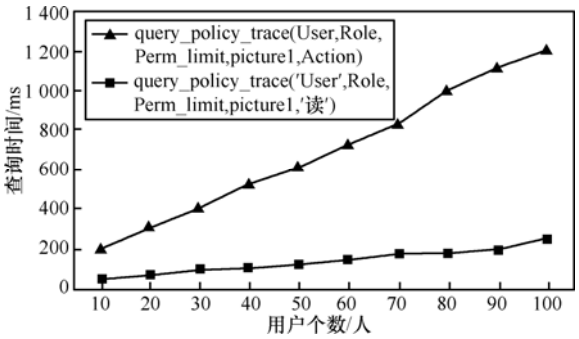
(*User*, *picture*1,*Aciton*)，限制 *Object*= *picture*1，用直线表示。实验结果表明：随着用户数量的增加，直接冲突查询时间呈线性增长，因为直接冲突查询采用枚举方式进行检测，用户数量增加，枚举查询的数量也相应增加，导致其查询时间快速增长；个性化定制查询相比直接冲突查询执行效率高，因为个性化定制查询限制某些变量，能够快速定位策略冲突原因，且受用户数量因素影响较小。



(a) 用户数量对查询性能的影响



(b) 资源数量对查询性能的影响



(c) 不同限定条件的情况下，查询性能的比较

图 7　系统查询性能分析

其次，测试资源数量对查询性能的影响。选定用户数量为 100，实验结果如图 7(b)所示，其中直接冲突查询为 *conflict*(*User*,*Object*, *Aciton*)，用虚线表示，个性化定制查询 *query_policy_trace*(*User*, *Role*, *PermissionLimit*, *Object*,*Action*)，限制 *User*= *Anny*，用直线表示。实验结果表明：随着资源数量的增加，直接冲突查询时间先线性增长后逐渐平稳，因为该模型的访问授权是针对于满足客体标签所有资源的授权，不是针对于某个资源的授权，虽然资源数量增加，但满足客体标签约束的资源确定时，其策略冲突查询时间会相对稳定；个性化定制查询相比直接冲突查询执行效率高，且受资源数量因素影响小。

最后，测试在个性化定制查询中不同限定条件的情况下，用户数量对查询性能的影响，实验结果如图 7(c)所示。其中限制 3 个变量的查询，即 *query_policy_trace*('*Anny*',*Role*,*PermissionLimit*, '*picture*1', '读')，表示限制 3 个查询条件：*User*= *Anny*, *Object*= *picture*1, *Action*=读，其查询时间用实线表示，限制一个查询条件，即 *query_policy_trace* (*User*, *Role*, *Perm_ limit*, '*picture*1',*Action*)，表示限制一个查询条件 *Object*=*diary*1，其查询时间用虚线表示。实验结果表明：个性化定制查询执行效率高，且限制查询条件越多，查询性能越好。

6　结束语

本文通过支持个性化隐私偏好的授权模型实现灵活的、实用的隐私策略定义，满足用户个性化隐私策略需求；针对隐私策略可能出现的冲突进行分析，采用 Prolog 逻辑编程方法实现自动化的策略一致性分析，为了克服规则数量大而造成的分析效率低下，采用个性化定制查询方式，实现快速验证和准确定位策略冲突原因；开发了面向社会网络的个性化隐私策略管理和实施中间件 Privacy Holder，通过可视化的界面，方便非专业用户实现策略定义与一致性分析。下一步工作将综合考虑资源层次关系带来的策略冲突，进一步扩展隐私策略一致性规则，实现资源多层次的隐私策略冲突检测。

参考文献：

[1] Facebook fact sheet and statistics[EB/OL]. https://www.facebook.com/press/info.php?statistics, 2011.

[2] SQUICCIARINI A, PACI F, SUNDARESWARAS N. PriMa: an effective privacy protection mechanism for social networks[A]. Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security[C]. Beijing, China, 2010. 320-323.

[3] WANG T, SRIVATSA M, LIU L. Fine-grained Access control of personal data[A]. ACM Symposium on Access Control Models and Technologies[C].

Newark, New Jersey, USA, 2012. 145-156.

[4] LI J, TANG Y, MAO C, *et al*. Role based access control for social network sites[A]. Proceedings of Joint Conferences on Pervasive Computing[C]. Taiwan, China, 2009. 389-394.

[5] CIRIO L, CRUZ I F, TAMASSIA R. A role and attribute based access control system using semantic web technologies[A]. Proceedings of International Federation for Information Processing Workshop on Semantic Web and Web Semantics[C]. Santiago, Chile, 2007. 1256-1266.

[6] YUAN E, TONG J. Attributed based access control (ABAC) for web services[A]. Proceedings of the IEEE International Conference on Web Services[C]. Orlando, Florida, 2005.561-569.

[7] ADAM N, ATLURI V, BERTINO E, *et al*. A Content-Based Authorization Model for Digital Libraries[R]. Computer Science Department, Rutgers University, 2001.

[8] CARMINATI B, FERRARI E, PEREGO A. Rule-based access control for social networks[A]. On the Move to Meaningful Internet Systems: OTM'06 Workshops[C]. Montpellier, France, 2006. 1734-1744.

[9] JAYARAMAN K, RINARD M C, TRIPUNITARA M, *et al*. Automatic error finding in access-control policies[A]. Proceedings of 18th ACM Conference on Computer and Communications Security[C]. Chicago,USA, 2011. 17-21.

[10] ALBERTI F, ARMANDO A, RANISE S. Efficient symbolic automated analysis of administrative attribute-based RBAC-policies[A]. Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security[C]. Hong Kong, China, 2011.165-175.

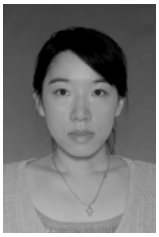
[11] ANWAR M, FANG P, YANG X D, *et al*. Visualizing privacy implications of access control policies in social network systems[A]. Proceedings of the 27th Annual ACM Symposium on Applied Computing[C]. Trento, Italy,2012.1443-1450.

[12] 周晓军, 蒋兴浩, 孙锬锋. RB-RBAC 模型的研究与改进. 信息安全与通信保密[J], 2010, (4): 100-102.
ZHOU X J, JIANG X H, SUN T F. Research and Improvement of RB-RBAC[J]. 2010,(4): 100-102.

[13] 努尔买买提 • 黑力力, 开依沙尔 • 热合曼. 带负授权 RBAC 模型的 OWL 表示及冲突检测[J]. 计算机工程与应用,2010, 46 (30): 82-85.
Nurmamat Helil, Kaysar Rahman. Representaion of RBAC model with negative authorization in OWL and conflict detection[J]. Computer Engineering and Applications, 2010,46(30):82-85.

[14] 李鼎. 基于逻辑编程的安全策略分析系统设计及其关键技术研究[D]. 郑州: 解放军信息工程大学, 2009.
LI D. The Design and Research of Policy Analysis System Based on Logic Programming[D]. Zhengzhou: The PLA Information Engineering University, 2009.

作者简介：



王媛（1982-），女，山东文登人，山东大学硕士生，主要研究方向系统安全与隐私保护。

通讯作者**孙宇清**（1967-），女，山东即墨人，博士，山东大学教授，主要研究方向为系统安全与隐私保护。E-mail: sun_yuqing@sdu.edu.cn

马乐乐（1989-），男，山东莱芜人，中国科学院研究生，主要研究方向为社会网络与隐私保护。