基于属性的访问控制模型

李晓峰[1-4]，冯登国[1,2]，陈朝武[3,4]，房子河[4]

（1. 中国科学院 软件研究所 信息安全国家重点实验室，北京 100080；
2. 中国科学院 研究生院 信息安全国家重点实验室，北京 100039；
3. 北京中盾安全技术开发公司，北京 100044；4. 公安部第一研究所，北京 100044）

摘 要：利用受限数据库为理论对访问请求、属性权威、策略和判定过程的抽象描述，给出了一个基于属性的访问控制模型，讨论了模型中访问请求、属性权威、策略和判定过程之间的关系，给出了一个访问控制判定过程可终止的一种特定条件。

关键词：访问控制；基于属性的访问控制；属性；受限数据库

中图分类号：TN918.91 **文献标识码**：A **文章编号**：1000-436X(2008)04-0090-09

Model for attribute based access control

LI Xiao-feng[1-4], FENG Deng-guo[1,2], CHEN Zhao-wu[3,4], FANG Zi-he[4]

(1. State Key Laboratory of Information Security, Institute of Software of Chinese Academy of Sciences, Beijing 100080, China;
2. State Key Laboratory of Information Security, Graduate School of Chinese Academy of Sciences, Beijing 100039, China;
3. Beijing Zhongdun Security Technology Development Co., Beijing 100044, China
4. The First Research Institute of Ministry of Public Security of P.R.C, Beijing 100044, China)

Abstract: Constrained database was used to abstractly describe access request, attribute authority, policies and decision procedure. An attribute based access control model was proposed. The relationships among access request, attribute authority, policies and decision procedure were discussed. A special condition on which the decision procedure is terminable is proposed.

Key words: access control; attribute based access control; attribute; constrained database

1 引言

在开放环境下(如互联网)不同的客户端和服务器频繁交互，这些交互方有时处于不同的安全域之内，相互只能知道对方部分信息，传统的基于身份的访问控制(IBAC)已不能适用于这种环境，基于属性的访问控制(ABAC)能够很好地适应这种开放的网络环境。在基于属性的访问控制中，访问判定是基于请求者和资源具有的属性，请求者和资源在ABAC中通过特性来标识，而不像IBAC那样只通过ID来标识，这使得ABAC具有足够的灵活性和可扩展性，同时使得安全的匿名访问成为可能，这

收稿日期：2006-11-16；修回日期：2008-02-20
基金项目：国家自然科学基金资助项目（60603017）；国家高技术研究发展计划（“863”计划）基金资助项目（2006AA01Z454）；国家科技支撑计划基金资助项目（2006BAH02A02, 2006BAK08B06, 2006BAK08B03）
Foundation Items: The National Natural Science Foundation of China (60603017); The National High Technology Research and Development Program of China (863 Program)(2006AA01Z454); The National Key Technology R&D Program of China (2006BAH02A02 , 2006BAK08B06 , 2006BAK08B03)

在大型分布式环境下是十分重要的。

ABAC 与 IBAC 显著不同之处在于其对请求者、被请求资源通过属性来描述，而一些限制条件同样也使用环境属性来描述，这就是说在 ABAC 中所有实体的描述都统一采用同一种方式——属性来进行描述，不同的是不同实体的属性权威可能不同，这使得访问控制判定功能在判定时，对访问控制判定依据能够采取统一处理。同时，基于属性的策略描述也摆脱了基于身份的策略描述的限制，其能够利用请求者所具有的一些属性来决定是否赋予其访问权限，在开放的环境下，访问控制判定功能并不关心访问者是谁（有时也可能根本无法获取这类信息）。在系统运行过程中，属性是一个易变量，而策略比较稳定，基于属性的策略描述方式可以很好地将属性管理和访问判定相分离。基于角色的访问控制（RBAC）通过引入角色中间元素，使得权限先经过角色进行聚合，然后再将权限分配给主体，通过这种方式可以简化授权，可将角色信息看成是一种属性，这样 RBAC 就成为了 ABAC 的一种单属性特例。XACML 是一个基于 XML 的访问控制标记语言，其采用访问者、被请求资源、被请求行为和环境属性来描述策略，是一个典型的在 ABAC 环境下的策略描述语言。

目前针对 ABAC 的研究大多集中在应用方面，而对其理论模型的研究较少，这使得 ABAC 中很多概念没有一个规范的定义。本文以受限 Datalog 和 CDB 理论为基础，描述了 ABAC 整体框架，以及各个组件的理论含义，同时给出 ABAC 中一些基本概念的规范定义，建立进一步的分析基础。

2 ABAM 与 XACML 简介

ABAM[1]用属性值元组来描述访问矩阵中行和列对应的主体和客体，并用关于属性的谓词来描述指令执行条件，通过指令来修改系统状态，然后在指令和属性满足某个特定条件下，ABAM 的安全问题是可判定的。Lingyu Wang 等人[2]提出基于属性的访问控制的逻辑框架（LABAC），LABAC 使用 CLP(SET)中的集合来描述属性和服务，讨论了逻辑程序的语义，由于逻辑程序在实际执行时很耗时，LABAC 详细介绍了优化策略的 2 种方式，提高逻辑程序执行性能。Barker[3]讨论了使用 CLP 来描述基于身份的访问控制策略，并用 CLP 对 RBAC 系统进行描述。本文在这些相关工作的基础上，以 XACML 为主要参考，从基于属性的访问控制的角度，模型化了访问控制系统中的各个组件，并指出相关之间的关系。

2003 年 OASIS（organization for the advancement of structured information standards）制定了基于 XML（extensible markup language）的访问控制策略和访问控制请求/响应描述语言规范——XACML（extensible access control markup language）。作为一种基于属性的通用访问控制策略和请求描述语言，其能适应多种应用环境，使访问控制中间件的开发和使用成为可能，同时其良好的扩展性。XACML 定义了访问请求和访问策略的语法，并且描述了访问请求判定的基本流程。在 XACML 中访问请求描述的自然语义为“在当前条件下，主体以某种方式访问资源”，其中条件、主体、方式、资源是通过属性值来描述，具体在 XACML 中是通过 <xacml:Environment>来描述条件，<xacml:Subject> 来描述主体，<xacml:Resource>来描述资源，<xacml：Action>来描述访问方式。策略执行点（PEP）接受到一个访问请求后，构造一个使用 XACML 描述的访问请求上下文，并发送给策略判定点（PDP），PDP 根据访问请求中给定的各个元素的属性值，查找适用策略和策略中适用的规则，根据规则的判定值、策略中的规则合并算法和策略集中的策略合并算法计算出此访问请求的判定结果。

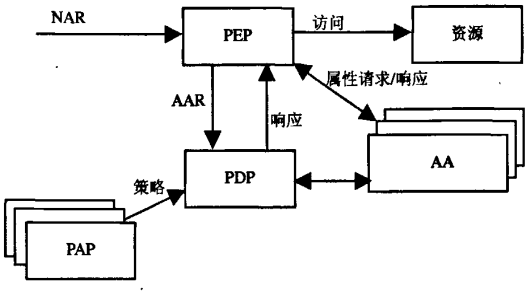
3 ABAC 概述

在访问控制中，请求者(用户或代理)、被访问客体(资源)及访问方法是在一个访问请求中必须明确的 3 个基本元素。在 IBAC 中，请求者、被访问资源和访问方法使用各自的标识来惟一确定，这使得 IBAC 授权不灵活，不能很好适应分布式协作环境。虽然在 IBAC 中通过引入角色改进了授权方式，但其仍然不能满足开放式计算环境的要求，如网格和组织间的协作。为了解决开放式环境下的访问控制要求，基于属性的访问控制（ABAC）被提出。

ABAC 中的基本元素仍然包括请求者，被访问资源，访问方法和条件，所不同的是这些元素统一使用属性来描述，而且各个元素所关联的属性可以根据系统需要定义，属性概念将访问控制中对所有元素的描述统一起来，同时摆脱了基于身份的限制，如果将标识看成一个属性的话（标识在认证系统中有着很重要的作用，但是在基于属性的访问控制中，并不赋予其特殊功能，其与其他属性具有同

等的地位)，ABAC 包含 IBAC。在 ABAC 中，策略中的访问者是通过访问者属性来描述，同样，被访问资源，方法也是通过资源和方法的属性来描述，而条件用环境属性来描述。环境属性通常是一类不属于主体，资源和方法的动态属性，如访问时间，历史信息等。条件有时也会用来描述不同类型属主具有的属性之间的关系，如访问者的某一属性与资源的某一属性之间的关系。

ABAC 框架示意图如图 1 所示。



NAR:原始访问请求 AA: 属性权威 AAR:基于属性访问请求
PEP:策略执行点 PDP:策略判定点 PAP:策略管理点

图 1 ABAC 框架示意图

在 ABAC 中，PEP 接收原始访问请求(NAR)，然后根据 NAR，利用不同的属性权威(AA)中存储的属性信息构建一个基于属性的访问请求(AAR)，AAR 描述了请求者、资源、方法和环境属性，PEP 将 AAR 传递给 PDP， PDP 根据从 PAP 处获取的策略对 AAR 进行判定，并将判定结果传给 PEP，PEP 执行此访问判定结果。为了使得讨论的问题更加突出，本文假设构造的访问请求中包含当前 PEP 能够获取的请求主体，被请求资源，请求操作和当前环境的所有属性（在 PDP 进行访问请求判定时，如果基于属性的访问请求中不包含其所需的相关属性，PDP 可以直接去获取，也可以通过与 PEP 交互获取，不论采取那种方式，从 PDP 角度看均形成了一个新的基于属性的访问请求，省去中间交互过程，可以直接假定基于属性的访问请求获取了所有所需的能获取的属性，如果仍然有需要的属性不包括在请求中，那么表示这个属性在此环境下不能获取)，获取的每一个属性值都是真实，可信的（系统可通过一些密码机制来保证属性是真实可信的，如利用基于公钥加密技术的属性证书)，同时 PDP 在判定时已获取了所有相关策略（可以假定 PDP 中存有所有可获取策略的一个副本，事实上这也是通常 PDP 实现的一种方式)。本文在此框架下，讨论基于属性的访问请求，访问策略，以及访问请求判定的语义，并在此框架下讨论策略冲突的语义和解决方法。

4 基于属性的访问控制模型

在 ABAC 中，并不关心或者必须知道访问者是谁，而只需知道访问者具有那些属性，这就是使得一个系统中的 2 个不同的主体，在另外一个系统中可能映射为具有相同属性的 2 个主体，也就是在另外一个访问控制系统成为同一个主体，这就打破了基于身份的访问控制的限制。比如在一个系统中规定“销售部门的人员可以读销售计划”，在 ABAC 中，只要请求者“部门”属性的取值是“销售部”，此请求者就可以读取销售计划，ABAC 并不关心请求者是谁，只要请求者的属性能够满足此策略的限制，访问就被允许。在 ABAC 中，PDP 判定依据的组件有 AAR、访问策略和 AA。

在 ABAC 中，AAR 是对请求者、被请求资源、被请求行为和当前环境属性的描述，AAR 中的属性赋值来源于 AA，PEP 构造 AAR 的过程可以看成一个映射 AAR : $NAR, AAs \rightarrow AAR$，这个映射和 NAR 具体语义有关系，在此不做讨论。

在 ABAC 中，属性是一个重要的概念。系统中属性用属性名来惟一标识，每个属性都有确定的类型表示属性的定义域，用关系 Domain(*att*)表示属性 *att* 的类型，属性在 ABAC 有明确的含义。为了讨论方便在每个属性名前加一个前缀对属性进行区别，前缀为 s 表示为主体属性，前缀为 r 表示是资源属性，前缀为 a 表示是动作属性，前缀为 e 表示是环境属性，这样命名后不同实体间的属性名没有重复，可以用属性名下标来表示属性类型，但是当属性类型在上下文明确时，可以省略下标，如主体的一个整型 Id 属性 sID_{int} 可以简写为 sID。属性名在 AAR、PDP 和 AA 之间共用。将所有属性名组成的集合记为 **ATT**。

定义 1 一个属性 *attname* 的取值记为 *attname* =value，其中 *attname*∈ATT，value∈Domain (*attname*)，给属性赋值的过程也称为一个属性变量指派或属性变量替换。

定义 2 一组主体、资源、动作和环境属性取值组成一个 AAR。在一个 AAR 中必须至少包含一个主体、一个资源和一个动作属性值，可形式化地表示为 att_1= constant$_1$ ∧ att_2= constant$_2$ ⋯ ∧ att_m=

constant$_m$，也可用集合的形式表示为 AAR={att_1= constant$_1$,att_2= constant$_2$,…,att_m= constant$_m$}。

在 ABAC 中存在多个 AA，AA 中存储主体、资源、动作和环境属性的取值列表、属性值之间的关系、属性之间的关系、属性和关系之间的关系，以及关系之间的关系。比如在一个 AA 有属性取值列表：sID、sRole、rID、aID，属性之间的关系有：sID 和 sRole 之间关系 RoleAssign，rID 和 aID 之间关系 Permission，属性值之间的关系：RoleHierarchy(用>来表示)，属性和关系之间的关系：sRole 和 Permission 之间的关系 RolePermission，关系之间的关系：DRolePermission，这些关系使用 Datalog 中的事实和规则来表示(如图 2 所示)，用 Datalog 来表示 AA 中内容，其可分为两部分，一部分使用事实定义称为外延数据库（EDB），如属性取值列表、RoleAssign、Permission、RolePermission、>，一部分使用规则定义称为内涵数据库（IDB），如 DRolePermission。AA 中的规则必须是安全规则，AA 中的规则可以是递归规则，如果允许在规则体中包含否定谓词，那么要求 AA 中的规则组成的程序必须是可局部分层的，在这种条件下，AA 可以确定地计算出所有由规则定义的内涵关系，也就是说 AA 中的属性描述程序具有确定的语义（在一个基于属性证书的 ABAC 中，AA 使用属性证书来描述属性，这时属性之间的关系直接通过关系实例来描述，这可以看成是直接给出了 AA 的语义）。这一点对于 PDP 来说非常重要，因为 ABAC 在定义访问策略时会使用 AA 中的这些属性关系，PDP 在做访问判定时，需要知道某个属性关系实例是否为真，而 AA 必需要能够返回给 PDP 这样的判定信息。

```
sID('allice')←
sID('bob')←
sID('tom')←
sRole('manager')←
sRole('sales')←
rID('plan')←
rID('contact')←
aID('read')←
aID('create)←
aID('delete)←
RoleAssign('allice', 'manager')←
RoleAssign('bob', 'sales')←
Permission('plan', 'create')←
Permission('plan', 'read')←
Permission('plan', 'delete')←
Permission('contact', 'read')←
Permission('contact', 'create')←
Permission('contact', 'delete')←
RolePermission('sales', 'plan', 'read')←
RolePermission('manager', 'plan', 'create')←
RolePermission('manager', 'plan', 'delete')←
RolePermission('sales', 'contact', 'read')←
RolePermission('sales', 'contact', 'create')←
RolePermission('manager', 'contact', 'delete')←
'manager'> 'sales'←
DRolePermission(X,Y,Z)←X>R,RolePermission(R,Y,Z)
```

图 2　AA 中属性列表和相关关系示例

在 ABAC 中，PDP 中存储相关的访问策略，为了丰富访问策略表达能力，在策略描述上引入限制，限制类型采用文献[4]中的方式来表达。

定义 3[4]　限制域是一个三元组(Σ,D,L)，Σ是一个词汇表，D 是 Σ上的一个结构，L 是允许出现的 Σ上的无量词一阶原子公式的形式，称为基本限制或原子限制。一组基本限制的合取称为限制。

定义 4　访问策略的形式为 $R(pid)\leftarrow G_1,\cdots,G_n,\psi\ (n\geq 1)$，其中，$G_i$（$i$=1,…,$n$）是由 AA 定义的谓词，$\psi$ 是限制域上的限制，对于 ψ、G_i中的任一变量 var，有 $var\in ATT$，称为属性变量，$G_1,\cdots,G_n$，ψ 称为策略体，策略体内所有谓词组成集合记为 $BP(pid)$，ψ 中所有原子限制组成集合 BC。$R\in$ {$permit$，$deny$}，pid 是一个常量，标识当前策略，$R(pid)$称为策略头。

$G_1,G_2,\cdots,G_n$ 中所有的属性变量组成策略相关属性变量组 X，在不关心策略标识时，通常可以将 R 简写为 permit 或 deny。例如，有一条访问策略“销售部（sales）的任何人员都可以阅读（read）销售计划（salesplan）”，可表示为 permit←$sDepartment$= 'sales'，$rCategory$= 'salesplan'，aID= 'read'。

在策略中出现什么类型的限制，与属性的类型相关，属性类型决定了限制域中的结构，属性基本类型有自然数 N、整数 Z、有理数 Q、实数 R、布尔 B、字符串、集合、列表，以及这些基本类型的子集，在 CDB 中研究的限制在文献[5]中有详细的总结，而在基于逻辑的访问控制中用到的限制类型在文献[3,4,6,7]中有相关总结。

5　访问策略的评估

在 ABAC 中，访问策略定义了 Permit 和 Deny 2 个关系，在一个 AAR 下，如果某个访问策略的策略体中的限制和谓词均满足，那么此策略头为真，permit/deny(pid)分别表示此策略允许/拒绝此 AAR。而策略体不满足分为两种情况，一种是 AAR 提供的属性信息不足，无法做出判断，一种是策略体在此 AAR 下不满足，第一种情况用关系 Unknown 来

表示，第二种情况用关系 Unsatisfy 表示。所以在 ABAC 中存在 4 个访问策略评估值 unknown、unsatisfy、permit 和 deny，PDP 依据 AAR 和 AA 对访问策略进行评估过程也是计算 Unknown、Unsatisfy、Permit 和 Deny（也可以简化定义为 3 个关系 Permit、Deny 和 Unknown，为了区分 Unknown 中两种不同的情况，使用 4 个关系来描述评估结果）关系的过程，这 4 个关系的最简单定义是 4 个单元关系（可以扩展此关系定义，使其能够用在元策略的描述中，本文只考虑这种简单的情况），其惟一的列对应访问策略标识，这 4 个关系两两不相交。

定义 5 AAR 中的所有属性组成的集合称为请求的相关属性集，记为 $\overline{X}_R$。一个访问策略 P 策略体内的所有属性变量组成的集合称为策略属性集，记为 $\overline{X}_P$。

给定一个访问策略 P，访问策略 P 的评估是在给定 AAR 和 AA 下的一个映射 PolicyEva：$P\rightarrow${unknown, unsatisfy, permit, deny}。

定义 6 在给定访问策略评估环境下（给定 AAR 和 AA（根据定义 5，$\overline{X}_R$ 是 AAR 中的所有属性组成的集合，所以 AAR 对 PolicyEva 的影响体现在 $\overline{X}_R$ 上，同样 AAR 是通过 AA 和 NAR 来构造的，所以 AA 对 PolicyEva 的影响依然体现在 $\overline{X}_R$ 上)），将 PolicyEva 定义为

① 如果 $\overline{X}_R \not\supseteq \overline{X}_P$，PolicyEva($p$)=unknown；

② 如果 $\overline{X}_R \supseteq \overline{X}_P$，在 $\overline{X}_R$ 属性取值情况下，p 策略体中的元素均为真，则 PolicyEva(p)=R，$R\in$ {permit,deny}；

③ 如果 $\overline{X}_R \supseteq \overline{X}_P$，在 $\overline{X}_R$ 属性取值情况下，p 策略体中的元素有一个为假，则 PolicyEva(p)= unsatisfy。

在 PolicyEva 中，p 策略体元素评估可以分为 2 种情况：原子和限制。在 ABAC 中 AA 看成一个 Datalog 数据库，策略体中包含的原子谓词对应 AA 一个关系。对于 AA 中使用 IDB 定义的关系，如何 AA 中描述的程序语义确定，能够找到一个不动点算法来计算 AA 中的内涵谓词，此算法是可终止的、完备的和可靠的[8,9]。在给定 AAR 时，就是给定了一组变量赋值，在此变量赋值下，策略体中的原子表达式变为一个基原子，判断原子实例是否为真转化为判断此基原子是否包含在对应关系中。在实际实现中，这可以转化为一个关系查询，通过返回结果的个数来判断此基原子为真或假。而对策略体中的限制 c，在给定 AAR 下 c 的可满足性，就是判断一组变量赋值是否满足 c，及判断一组无变量公式是否满足，通常这是很容易判断的[5]。

在 ABAC 中，计算 Unknown、Unsatisfy、Permit 和 Deny 这 4 个关系的过程如图 3 所示。对一组策略进行评估后，如果 Permit、Deny 2 个关系中只有一个关系非空，那么判定结果无二意地被确定，如果这两个关系均为非空，这时就产生了策略冲突，需要引入元策略来解决冲突问题。

```
PolicySetEvaluation(AAR,P)
Input:AAR,P
Output:Unknown,Unsatisfy,Permit,Deny
设置 Unknown 包含所有的访问策略
for each p in P
  result=PolicyEva(p)
if result=permit then
从 Unknown 关系中删除对应的策略记录
    将策略 p 的规则头添加到相应的关系 Permit
elseif result=deny then
从 Unknown 关系中删除对应的策略记录
将策略 p 的规则头添加到相应的关系 Deny 中
  elseif result=unsatisfy then
    从 Unknown 关系中删除对应的策略记录
将策略 p 添加到 Unsatisfy 关系中
fi
rof
```

图 3 访问策略集评估过程

6 元策略

定义 7 在一个 AAR 和一组 AA 下评估一组策略集，评估完成后，如果 Permit 和 Deny 关系均为非空称评估结果冲突，如果只有一个为非空获得了惟一评估结果，如果两个关系均为空，称此 AAR 在当前系统状态下无法进行判断。

当 PDP 获取的结果并不只有一个时，需要定义结果的合并方法，在系统中有两种策略合并方式，一种是定义一个全局策略合并方式（GPCMP），另外一种是定义多个局部策略合并方式（LPCMP）。

首先讨论全局策略合并方式。常用的全局策略合并方式为：

如果评估结果是“无法进行判断”，也要给出一个结果，在实际中，通常有 2 种选取方法存在，一种是开策略、一种是闭策略，开策略是指如果评估结果是无定义，那么最终结果为允许，闭策略是指如果评估结构无定义，那么最终结果为拒绝，使用中只能选择这两种策略其中之一。

如果评估结果为冲突，使用全局策略冲突解决办法，通常有以下几种解决办法：

Permit 优先（GPCMP0）：是指如果出现冲突，最终结果为 Permit；

Deny 优先（GPCMP1）：是指如果出现冲突，最终结果为 Deny；

Undefine（GPCMP2）：是指如果出现冲突，最终结果为无定义；

在实际使用中，一个 PDP 只能选择一种全局策略冲突解决办法。

如果评估结果惟一，那么最终结果就是此结果。比如只有 Permit 非空，那么最终结果就是 Permit。

GPCMP 可以看成是一个映射 GPCMP：Unknown × Unsatisfy × Permit × Deny → {permit, deny,undefine}，其中×表示关系的笛卡尔积。

定义 8　对于任意 2 个访问策略 $p1$ 和 $p2$，给定 AAR 和 AA，Permit($p1$)和 Deny($p2$)为真，称 $p1$ 和 $p2$ 在当前评估环境下冲突。对于一组访问策略 P，如果有 2 个策略 $p1,p2 \in P$ 在当前评估环境下冲突，称 P 在当前环境下是一组冲突策略集。

局部策略合并方式根据独立的评估结果定义合并方式，将所有 LPCMP 策略组成的集合记为 **LPCMP**。LPCMP 策略也有不同的策略类型，一个策略类型对应一个映射 $RMap_i$：$(\{permit,deny, unknown, unsatisfy\})^n \rightarrow \{permit,deny, undefine\}$，在 ABAC 中给每一个 LPCMP 一个惟一标识，LPCMP 策略记为 $RMap_i(p_1,\cdots, p_n, pid)$，其中 pid 是一个常量（RMap(…)中最后一个参数是此 LPCMP 的标识），是此 LPCMP 的标识，在 LPCMP 中，允许引用 LPCMP 策略的评估结果，$p_i \in \{p_1,\cdots, p_n\}$ 可以是 LPCMP 标识或者访问策略标识，表示对此策略评估结果的引用，Rma p_i 是此 LPCMP 策略的类型。系统可以根据实际情况定义自己的 LPCMP 映射类型，常见的 LPCMP 映射类型有 Permit 优先(LPCMP0)和 Deny 优先(LPCMP1)：

LPCMP0：如果 Permit(p)或 CPermit(p)为真,其中 $p \in \{p_1, \cdots, p_n\}$，那么 $LPCMP0(p_1, \cdots ,p_n, pid)$=permit；如果对于$\{p_1,\cdots,p_n\}$中任一元素 p，Permit(p)、CPermit(p)、Deny(p)或 CDeny(p)均不为真，那么 $LPCMP0(p_1,\cdots,p_n,pid)$=undefine；否则 $LPCMP0(p_1,..\cdots,p_n, pid)$=deny。

LPCMP1：如果 Deny(p)或 CDeny(p)为真，其中 $p \in \{p_1, \cdots ,p_n\}$，那么 $LPCMP1(p_1, \cdots, p_n, pid)$=deny；如果对于$\{p_1,\cdots,p_n\}$中任一元素 p，Permit(p)、CPermit(p)、Deny(p)或 CDeny(p)均不为真，那么 $LPCMP1(p_1,\cdots,p_n,pid)$=undefine；否则 $LPCMP1(p_1,\cdots,p_n,pid)$=permit。

在 ABAC 中，对一组 LPCMP 策略评估的结果生成 3 个以 LPCMP 略标识为惟一列的单元关系 CUndefine、CPermit 和 CDeny，LPCMP 策略集评估算法（LPCMPEvaluation）如图 4 所示。

```
LPCMPEvaluation
Input:Permit,Deny,Unknown,Unsatisfy,LPCMP
Output:CPermit,CDeny,CUndefine
设 CPermit ,CDeny ()为空
CUndefine 包含所有 LPCMP
changed=true
loop while changed=true
  for each lp in LPCMP
    result=执行 lp 对应的映射
    if result=permit then
      if lp ∈CDeny 或 lp ∈CUndefine then
        从对应关系中中删除 lp
        将 lp 加入 CPermit
      fi
    fi
    if result=deny then
      if lp ∈CPermit 或 lp ∈CUndefine then
       从对应关系中中删除 lp
       将 lp 加入 CDeny
      fi
    fi
    if result=undefine then
      if lp ∈CDeny 或 lp ∈CPermit then
        从对应关系中中删除 lp
        将 lp 加入 CUndefine
      fi
    fi
  rof
  if CPermit 和 CDeny 没有变化 then
    changed=false
  fi
pool
```

图 4　LPCMP 策略评估过程

为了保证在不同 LPCMP 类型下 LPCMP 评估过程能够终止，需要对 LPCMP 做出限制。

定义 9　对于一个 LPCMP 策略 $RMap_i(p_1,\cdots,p_n, lp)$，将 p_m ($m \in [1,n]$)称为策略 lp 的相关策略，集合$\{p_1,\cdots, p_n\}$称为策略 lp 的相关策略集，记为 $RP(lp)$。

定义 10　LPCMP 策略依赖图 $DG(N,V)$，其节点集合 N 是所有 LPCMP 标识，V 是节点间的有向连线，$n1$ 到 $n2$ 的连线记为$(n1,n2) \in V$。对于 2 个 LPCMP 策略 $lp1$: $RMap_i(p_{i,1}, \cdots ,p_{i,n}, lp1)$ 和 $lp2$:$RMap_j(p_{j,1},\cdots,p_{j,n}, lp2)$，如果 $lp1 \in RP(lp2)$，那么有$(lp1,lp2) \in V$。

定义 11　对于一个 LPCMP 策略 p，如果 RP(p)

中所有的元素均是访问策略（$RP(p)\subseteq P$），那么 p 称为基本 LPCMP。对于一 LPCMP 集合，如果其包含基本 LPCMP 策略，称此 LPCMP 集合为有基 LPCMP 策略集。

定理 1 LPCMP 策略是基本 LPCMP 策略的充要条件是 DG 中对应节点的入度为零的。

证明 首先证明充分性，如果一个节点入度为零，其为基本 LPCMP 策略，一个节点入度为零表示对应的策略 p 的相关策略集中只有访问策略，可见其是一个基本 LPCMP。再证明其必要性，一个 LPCMP 是基本 LPCMP，说明此策略的相关策略集中只有访问策略，其不依赖于任何 LPCMP，在 DG 中没有从其他节点到此节点的有向线段，所以此节点入度为零。

定义 12 对于一个 $DG(N,V)$，如果存在一个节点的分割 $N1,N2,\cdots,Nm$，使得对于任何 $(n1,n2)\in V$，$n1\in N_i$，$n2\in N_j$，有 $i<j$，那么称 DG 对应的 LPCMP 策略组可层次化。

先给出 LPCMP 策略的分层算法 LPCMPStratify（如图 5 所示），可以证明对于任何有基 LPCMP 策略组，算法 LPCMPStratify 是可终止的，同时是完备和可靠的。

```
LPCMPStratify()
Input: LPCMP
Output: DG(N,V)
N1,N2,…是每一个层次包含的 LPCMP 策略集合
N 和 N1,N2,…初始为空
将所有 LPCMP 策略标识添加到 N 中
for each n in N
  if RP(n) ⊆ P then
    将 n 添加到 N1 中
  fi
rof
i=1, changed=true
loop while changed=true
  i=i+1,changed=false
for each n in N / ⋃_{j=1}^{i-1} Nj
  if RP(n) ⊆ ⋃_{j=1}^{i-1} Nj then
将 n 添加到 Ni 中
changed=true
  fi
rof
pool
```

图 5 有基 LPCMP 策略集的分层算法

定理 2 对于一组可分层的有基 LPCMP 策略，算法 LPCMPEvaluation 是可终止的。

证明 首先对于一组可分层的有基 LPCMP 策略，调用 LPCMPStratify 算法对其进行分层，结果是将策略分为 m 层：$N1,\cdots,Nm$，并且 $N1$ 非空。使用算法 LPCMPEvaluation 对其进行评估。在第一次循环中，不论 $N2,\cdots,Nm$ 层策略评估结果如何，$N1$ 层策略的评估结果在后续递归过程中均不会发生变化，在第二轮循环中，不论 $N3,\cdots,Nm$ 层策略评估结果如何，$N2$ 层策略的评估结果在后续递归过程中均不会发生变化，依次类推，在第 m 轮循环中，Nm 层策略评估结果也会最终确定，在 $m+1$ 轮循环中，所有策略的评估结果将不会发生变化，CPermit 和 CDeny 关系也不会发生变化，LPCMPEvaluation 算法终止。

在对 ABAC 中 LPCMP 策略集评估完成后，会获取所有 LPCMP 策略的评估结果（用关系 CPermit、CDeny 和 CUndefine 来表示），ABAC 最终评估结果要根据 LPCMP 评估结果来确定，并不是所有的 LPCMP 评估结果都是 ABAC 最终的评估结果。

定义 13 DG 中出度为零的节点称为结果节点，结果节点对应的 LPCMP 策略称为结果 LPCMP 策略。

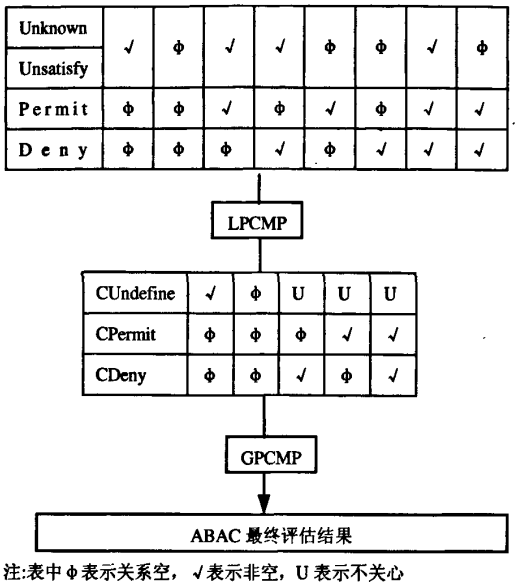
定理 3 调用 LPCMPStratify 对一个可分层 LPCMP 集进行分层后，最后一层中的所有节点均是结果节点。

证明 假设最后一层中有一个非结果节点 b，b 出度不为零，那么必然有一个节点 a 依赖 b，根据可分层定义有 a 的层次大于 b，这与 b 在最后一层相矛盾，所以有最后一层均为结果节点。

定理 4 对于一个可分层 LPCMP 策略，存在一个分层使最后一层对应所有的结果节点。

证明 按照 LPCMPStratify 算法进行分层，然后将不在最后一层的所有出度为零的节点提高到最后一层，根据分层定义，这仍然是一个有效的分层，然后此时的最后一层包含所有出度为零的节点，及最后一层包含所有结果节点。

在定理 4 的证明中将 LPCMPStratify 分层结果进行简单修整后就得到一个最后一层包含所有结果节点的分层方法，称此方法为“扩展的 LPCMPStratify”（E-LPCMPStratify），那么使用 LPCMP 策略后 ABAC 的评估结果就是最后一层策略对应的结果，整个 ABAC 评估流程示意图如图 6 所示。



注:表中 Φ 表示关系空，√表示非空，U 表示不关心

图 6　ABAC 评估流程

7　XACML 的模型表示

XACML 作为一种基于属性的访问控制策略和请求描述语言，得到广泛关注，XACML 和 ABAC 模型均以 ISO/IEC 10181-3[10]中提出的框架为基础，下面使用本文的模型来表示 XACML 中的概念。

一个 XACML 请求上下文描述了本次请求的请求者、被请求资源、动作和环境的属性，其对应 ABAC 中的 AAR。XACML 请求中请求者可以有不同的类别，如请求发起用户、请求发起程序等，这些不同种类的主体属性，在策略中使用一致的方法进行引用，在 AAR 这都可以看成是请求者的属性。一个 XACML 请求可以选择包含多个被请求资源，这种情况下，一个 XACML 请求可以对应多个 AAR，对每个 AAR 可进行分别评估（在 Multiple resource profile of XACML v2.0 中采用的就是这种方法）。在 XACML 请求中对属性也是用取值来描述，这与 AAR 中对属性描述是相同的。

XACML 策略集（策略体、规则）中对主体属性满足的条件用由 SubjectMatch 为基本元素的一个析取范式来表达，而 SubjectMatch 可以对应 ABAC 中的一个谓词或限制，同样资源、动作、环境也有同样的对应。在 XACML 规则中的 Condtion 可以同时包含主体、资源、动作和环境属性，所以 Condition 可以描述不同实体之间属性的相互限制，Condition 也可对应为一个谓词或限制（XACML 中 Condition 是可以进行多层嵌套的，对于这种情况可以采用适当的描述方式使其在具体描述时不存在嵌套）。XACML 中层次化的策略描述方式，暗示了其对主体、资源、动作和环境的限制也是逐渐严格的，根据文献[11]中的讨论，策略树中叶子节点（Rule）实际对应的对主体、资源、动作和环境的描述，是从策略树的根到此节点路径上所有限制性描述的合取，可其所对应的析取范式，析取范式中的每一个合取项，对应一个策略体，而策略头是 Rule 中的 Effect，这就将策略树中的一个 Rule 节点分解为多个 ABAC 中的访问策略，并且给每个策略一个标识。

在 XACML 策略树中，除了叶子节点，每个中间节点都赋予了一个规则/策略合并算法，合并算法描述了如何将多个不同的评估结果映射到一个评估结果上，这对应 ABAC 中 LPCMP 策略，LPCMP 策略参数可以包含另外一个 LPCMP 策略，所以每一个 Policy 或 PolicySet 的策略合并可对应为一个 LPCMP 策略。对于 XACML 策略树中的一个叶子节点，其父节点的规则合并对应一个 LPCMP 策略，此策略的相关策略集包含其下叶子节点对应的所有访问策略，而 LPCMP 策略类型根据合并算法选取；对于其他任何一个中间节点，其对应的 LPCMP 策略的相关策略集包含其下子节点对应的所有 LPCMP 策略。通过这种方法产生的 LPCMP 策略集，由于其是一个树状结构，可以很容易证明其是可分层有基 LPCMP 集。利用 LPCMPStratify 算法对 LPCMP 策略集分层后，最后一层将只包含一个 LPCMP 策略，或者说 XACML 只有一个结果 LPCMP 策略，所以 ABAC 评估完 LPCMP 后，CPermit、CDeny、Cundefine 3 个关系中只有一个非空关系，此时评估结果可以惟一确定，不需要 GPCMP 的参与。

8　结束语

本文以 XACML 为主要参考，给出了一个 ABAC 模型。在 ABAC 中，AAR 抽象为一组属性赋值，AA 抽象为属性数据库，而将策略分解为访问策略和元策略，访问策略抽象为带限制的 Datalog 规则，由于元策略在实际使用中会出现很复杂的情况，本文将其抽象为映射，并描述了在什么情况下，ABAC 的评估过程是可终止的，并给出了 XACML 在此模型中的解释。此模型的访问策

略虽然是一组组逻辑规则，但由于其只包含属性变量，所以在实际评估时每个谓词都实例化了，并不会出现完全的基于逻辑的系统中出现的程序子句不断向后追溯而引起的性能问题，从 XACML 实际使用情况看，这种描述方式能够满足实际系统的需求。

本文给出了 ABAC 模型，但只讨论了访问请求判定问题，关于基于属性的委托和管理问题没有进行讨论，这部分将在后续文章中继续进行讨论。但是本文提出的基于属性的访问控制模型，可以作为进一步研究基于属性访问控制中委托等管理问题的基础。

参考文献：

[1] ZHANG X, LI Y, NALLA D. An attribute-based access matrix model[A]. Proceedings of the 2005 ACM Symposium on Applied Computing[C].2005. 359-363.

[2] WANG L Y, WIJESEKERA D, JAJODIA S. A logic-based framework for attribute based access control[A]. Proceedings of the 2004 ACM Workshop on Formal Methods in Security Engineering[C].2004.45-55.

[3] BARKER S, STUCKEY P J. Flexible access control policy specification with constraint logic programming[J]. ACM Trans Inf Syst Secur, 2003,6(4): 501-546.

[4] LI N H, MITCHELL J C. Datalog with constraints: a foundation for trust-management languages[A]. Proceedings of the Fifth International Symposium on Practical Aspects of Declarative Languages (PADL 2003)[C]. New Orleans, Louisiana, 2003.28-73.

[5] REVESZ P. Introduction to Constraint Databases[M]. New York: Springer-Verlag, 2002.

[6] 钟勇, 秦小麟, 郑吉平等. 一种灵活的使用控制授权语言框架研究[J]. 计算机学报, 2006,8: 1408-1418.
ZHONG Y, QIN X L, ZHENG J P, *et al*. A flexible usage control authorization language framework[J]. Chinese Journal of Computers, 2006,8:1408-1418.

[7] ZHANG X W, PARISI-PRESICCE F, SANDHU R, *et al*. Formal model and policy specification of usage control[J]. ACM Transactions on Information and System Security (TISSEC), 2005, 8(4): 351-387.

[8] DAHR M. Deductive Databases: Theory and Applications[M]. USA: International Thomson Computer Press,1997.

[9] ULLMAN J. Principles of Database and Knowledge-Base Systems(volume I and volume II)[M]. Rockville MA:Computer Science Press, 1989.

[10] ISO/IEC 10181-3:1996 Information Technology-Open Systems Interconnection-Security Frameworks for Open Systems: Access Control Framework (Corresponding to Chinese Standard GB/T 18794. 3-2003)[S].

[11] LI X F, FENG D G. Composing administrative scope of delegation policies based on extended XACML[A]. Proceedings of the Tenth IEEE International EDOC Enterprise Computing Conference[C]. Hong Kong, China, 2006. 467-470.

[12] PARK J, ZHANG X W, SANDHU R. Attribute mutability in usage control[A]. IFIP WG 11.3[C]. 2004. 15-29.

[13] IRWIN K, YU T. Preventing attribute information leakage in automated trust negotiation[A]. Proceedings of the 12th ACM Conference on Computer and Communications Security[C]. 2005. 36-45.

[14] YUAN E, TONG J. Attribute based access control (ABAC): a new access control approach for service oriented architectures[A]. Ottawa New Challenges for Access Control Workshop[C]. 2005. 359-363.

[15] 徐洁磐, 马玉书, 范明. 知识库系统导论[M]. 北京: 科学出版社,2000.
XU J P, MA Y S, FAN M. Introduction of Knowledge Base[M]. Beijing: Science Press,2000.

作者简介：

李晓峰（1973-），男，陕西延安人，博士，公安部第一研究所和中盾安全技术开发公司技术部经理，主要研究方向为网络与系统安全、安全评估、流媒体技术。

冯登国（1965-），男，陕西靖边人，博士，信息安全国家重点实验室主任、研究员、博士生导师，主要研究方向为密码学与信息安全。

陈朝武（1956-），女，安徽巢湖人，硕士，公安部第一研究所副所长、研究员，主要研究方向为综合安全技术、信息系统集成、流媒体等。

房子河（1968-），男，河北保定人，硕士，公安部第一研究所安防与警用信息集成技术研发部主任、副研究员，主要研究方向为多媒体网络通信、信息系统集成、信息安全。

基于属性的访问控制模型

作者： 李晓峰， 冯登国， 陈朝武， 房子河， LI Xiao-feng， FENG Deng-guo， CHEN Zhao-wu， FANG Zi-he

作者单位： 李晓峰,LI Xiao-feng(中国科学院,软件研究所,信息安全国家重点实验室,北京,100080;中国科学院,研究生院,信息安全国家重点实验室,北京,100039;北京中盾安全技术开发公司,北京,100044;公安部第一研究所,北京,100044)， 冯登国,FENG Deng-guo(中国科学院,软件研究所,信息安全国家重点实验室,北京,100080;中国科学院,研究生院,信息安全国家重点实验室,北京,100039)， 陈朝武,CHEN Zhao-wu(北京中盾安全技术开发公司,北京,100044;公安部第一研究所,北京,100044)， 房子河,FANG Zi-he(公安部第一研究所,北京,100044)

刊名： 通信学报 ISTIC EI PKU

英文刊名： JOURNAL ON COMMUNICATIONS

年，卷(期)： 2008,29(4)

被引用次数： 30次

参考文献(15条)

1. ZHANG X;LI Y;NALLA D An attribute-based access matrix model 2005

2. WANG L Y;WUESEKERA D;JAJODIA S A logic-based framework for attribute based access control 2004

3. BARKER S;STUCKEY P L Flexible access control policy specification with constraint logic programming[外文期刊] 2003(04)

4. LI N H;MITCHELL l C Datalog with constraints:a foundation for trust-management languages 2003

5. REVESZ P Introduction to Constraint Databases 2002

6. 钟勇;秦小麟;郑吉平 一种灵活的使用控制授权语言框架研究[期刊论文]-计算机学报 2006(08)

7. ZHANG X W;PARISI-PRESICCE F;SANDHU R Formal model and policy specification of usage control[外文期刊] 2005(04)

8. DAHR M Deductive Databases:Theory and Applications 1997

9. ULLMAN J Principles of Database and Knowledgu-Base Systems(volume Ⅰ and volume Ⅱ) 1989

10. ISO/IEC 10181-3-1996. Information Technology-Open Systems Interconnection-Security Frameworks for Open Systems:Access Control Framework (Corresponding to Chinese Standard GB/T 18794.3-2003)

11. LI X F;FENG D G Composing administrative scope of delegation policies based on extended XACML 2006

12. PARK J;ZHANG X W;SANDHU R Attribute mutability in usage control 2004

13. IRWIN K;YU T Preventing attribute information leakage in automated trust negotiation 2005

14. YUAN E;TONG J Attribute based access control (ABAC):a new access control approach for service oriented architectures 2005

15. 徐洁磐;马玉书;范明 知识库系统导论 2000

本文读者也读过(4条)

1. 王小明. 付红. 张立臣. WANG Xiao-ming. FU Hong. ZHANG Li-chen 基于属性的访问控制研究进展[期刊论文]-电子学报2010,38(7)

2. 李晓峰. 冯登国. 徐震. Li Xiaofeng. Feng Dengguo. Xu Zhen 一种通用访问控制管理模型[期刊论文]-计算机研究与发展2007,44(6)

3. 程相然. 陈性元. 张斌. 杨艳. CHENG Xiang-ran. CHEN Xing-yuan. ZHANG Bin. YANG Yan 基于属性的访问控制策略模型[期刊论文]-计算机工程2010,36(15)

4. 林莉.怀进鹏.李先贤.LIN Li.HUAI Jin-Peng.LI Xian-Xian 基于属性的访问控制策略合成代数[期刊论文]-软件学报2009,20(2)

引证文献(30条)

1. 韩道军.黄泽龙.翟浩良.李磊 基于逻辑合一的访问控制规则描述[期刊论文]-计算机科学 2011(10)

2. 韩道军.黄泽龙.翟浩良.李磊 基于逻辑合一的访问控制规则描述[期刊论文]-计算机科学与探索 2011(10)

3. 周加根.叶春晓.罗娟 ABAC策略语义表示和决策方法[期刊论文]-计算机工程与应用 2013(23)

4. 韩道军.贾培艳.马宇翔 基于属性的访问控制模型及其在企业信息系统中的应用[期刊论文]-计算机时代 2012(5)

5. 程相然.陈性元.张斌.杨艳 基于属性的访问控制策略模型[期刊论文]-计算机工程 2010(15)

6. 张毅.高东怀.许卫中.许浩 基于用户属性的终端安全防护系统研究与实现[期刊论文]-科学技术与工程 2009(18)

7. 刘飞.常朝稳 基于多维度量和上下文的访问控制模型[期刊论文]-计算机工程 2011(24)

8. 王小明.付红.张立臣 基于属性的访问控制研究进展[期刊论文]-电子学报 2010(7)

9. 霍远国.马殿富.刘建.李竹青 面向Web服务资源的两层访问控制方法[期刊论文]-计算机科学 2010(7)

10. 马赛.田飞.靳婷 服务集成总线中基于属性的策略服务的研究[期刊论文]-计算机工程与设计 2013(2)

11. 汪海玲.郝玉洁.白敬培 内容管理信息系统中访问控制方案的切换[期刊论文]-电子科技大学学报 2013(5)

12. 谢学智.李春晓 属性加密机制在电子政务中的应用[期刊论文]-中国科技论文 2013(1)

13. 单云江 基于属性证书的跨域访问技术设计与实现[期刊论文]-数字通信 2013(4)

14. 滕震方 基于多属性的移动终端安全接入网络认证协议[期刊论文]-计算机应用与软件 2013(8)

15. 王立.万世昌.张珍 基于互信属性调配机制的访问控制模型[期刊论文]-计算机技术与发展 2009(12)

16. 苏凡.柴获 基于属性的RFID中间件访问控制模型研究[期刊论文]-兰州交通大学学报 2012(3)

17. 张斌.张宇 基于属性和角色的访问控制模型[期刊论文]-计算机工程与设计 2012(10)

18. 李迎涛.马春光.李增鹏 一种涉密系统中增强的角色访问控制方案[期刊论文]-信息网络安全 2013(11)

19. 熊厚仁.张斌 支持动态授权的网格授权机制[期刊论文]-计算机工程与设计 2011(9)

20. 孙翠翠.张永胜 基于角色和属性的Web Services安全模型研究[期刊论文]-微计算机信息 2011(2)

21. 孙翠翠.张永胜 基于角色和属性的Web Services安全模型研究[期刊论文]-微计算机信息 2011(5)

22. 张伟.王立.李岩 数字版权管理中的访问控制研究[期刊论文]-计算机技术与发展 2011(7)

23. 钟将.侯素娟 开放网络环境中基于属性的通用访问控制框架[期刊论文]-计算机应用 2010(10)

24. 辛艳.罗长远.刘辉.应一舟 基于上下文的普适计算角色访问控制模型[期刊论文]-计算机工程与设计 2010(8)

25. 韩道军.夏兰亭.卓汉逵.李磊 基于强化学习的业务流程中的柔性约束研究[期刊论文]-计算机科学 2011(3)

26. 翟浩良.韩道军.李磊 基于情景演算的动态访问控制模型[期刊论文]-计算机科学 2012(6)

27. 韩道军.高洁.翟浩良.李磊 访问控制模型研究进展[期刊论文]-计算机科学 2010(11)

28. 王婷.陈性元.张斌.张红旗 授权与访问控制中的资源管理技术研究综述[期刊论文]-小型微型计算机系统 2011(4)

29. 王婷.陈性元.张斌.张红旗 授权与访问控制中的资源管理技术研究综述[期刊论文]-小型微型计算机系统 2011(4)

30. 李凤华.苏铓.史国振.马建峰 访问控制模型研究进展及发展趋势[期刊论文]-电子学报 2012(4)