ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.20, No.2, February 2009, pp.403−414
doi: 10.3724/SP.J.1001.2009.03279

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

基于属性的访问控制策略合成代数*

林 莉+, 怀进鹏, 李先贤

(北京航空航天大学 计算机学院,北京 100191)

Attribute-Based Access Control Policies Composition Algebra

LIN Li+, HUAI Jin-Peng, LI Xian-Xian

(School of Computer Science and Engineering, BeiHang University, Beijing 100191, China)

+ Corresponding author: E-mail: linli@act.buaa.edu.cn

Lin L, Huai JP, Li XX. Attribute-Based access control policies composition algebra. *Journal of Software*, 2009, 20(2):403−414. http://www.jos.org.cn/1000-9825/3279.htm

Abstract: The composition of access control policies is the key to determine access control policies for distributed aggregated resource. To regulate policy composition and guarantee its correctness, an algebraic model called APoCA (attribute-based access control policy composition algebra) is proposed for composing access control policy. In APoCA, an authorization relation between entities is described at the attribute level. APoCA fertilizes the existing formal frameworks by taking into account the computation of attribute values. Several examples are given to demonstrate the expressiveness of ApoCA. ApoCA can be used for more complex applications. In addition, access control policies of aggregated resources can be formulated as expressions of the algebra. Several algebraic properties of policy expressions are discussed. It shows that the algebraic properties of policy expressions can be used to verify whether policy composition results meet the protection needs of each party. Furthermore, a translator is devised to convert the policy expressions into logic programs, which provides the basis for the evaluation and application of access control policies for aggregated resources.

Key words: aggregated resource; access control; attribute; policy composition algebra; logic program

摘 要: 访问控制策略合成是确定分布式聚合资源访问控制策略的关键.为了规范策略合成和保障策略合成正确性,基于属性刻画了实体间的授权关系,通过属性值的计算结构扩展了现有的策略合成形式化框架,建立了新的基于属性的策略合成代数模型 APoCA(attribute-base access control policy composition algebra).通过示例分析说明 APoCA 具有更强的策略合成描述能力和普适性,适应于更为复杂的应用场景.用代数表达式形式化地描述聚合资源的访问控制策略,讨论了策略表达式的若干代数性质,说明可借助策略表达式的代数性质去验证策

* Supported by the National High-Tech Research and Development Plan of China under Grant No.2007AA01Z426 (国家高技术研究发展计划(863)); the National Basic Research Program of China under Grant No.2005CB321803 (国家重点基础研究发展计划(973)); the National Natural Science Funds for Distinguished Young Scholar of China under Grant No.60525209 (国家杰出青年基金); the Funds for the International Cooperation and Exchange of the National Natural Science Foundation of China under Grant No.60731160632 (国家自然科学基金和国际(地区)合作项目); the Program for New Century Excellent Talents in University of China under Grant No.NCET-05-0186 (新世纪优秀人才计划)

Received 2007-01-04; Accepted 2008-02-27

略合成结果是否符合各方对聚合资源的保护性需求.给出了将代数表达式翻译成逻辑程序的翻译器,为聚合资源的访问控制策略评估和应用提供基础.

关键词: 聚合资源;访问控制;属性;策略合成代数;逻辑程序

中图法分类号: TP393 **文献标识码**: A

随着大规模分布式技术与应用的迅猛发展,跨自治域的资源协同需要各个自治域共同协调并建立一致的访问控制策略,实现对聚合资源统一的访问控制.例如,在 Crown(China research and development environment over wider-area network)环境的远程热部署应用中,控制服务访问的策略需要服务部署方与服务容器方共同确定[1];对联合科学协作研究产生的实验数据,其访问控制策略由参与研究的各方协商制定[2].无论采取何种手段制定聚合资源的访问控制策略(如通过协定、协商等[2−4]),抽象来看,最后确定的策略均可视为对各方策略以某种方式进行合成.例如,一个 Web 组合服务的访问控制策略一般通过对其所有组件服务的访问控制策略进行合成而得到(如文献[5]中取交集的方式).因此,访问控制策略合成是确定聚合资源的访问控制策略的一个关键问题.

策略合成将生成新的策略,其方式和过程都非常复杂,涉及多方的安全策略约束,使用数学方法刻画策略的合成性质有助于明确策略的结构、验证策略合成结果的正确性和安全性(例如,是否满足各方协商出的对聚合资源的保护需求).代数[6]是建立离散结构模型的一种自然的形式化工具,可清晰地描述策略的合成结构,并具有坚实的数学基础.Bonatti 等人[7]提出了经典的访问控制策略合成代数模型.其基本思想是将访问控制策略定义为由主体、客体和动作构成的三元组(授权项)集合,用并(addition)、交(conjunction)、差(subtraction)等代数算子抽象地刻画不同的策略合成方式,具有一定的普适性,能够支持其他传统的策略合成模型[8−10],为后续研究[5]等奠定了基础.

然而,随着 XACML(extensible access control markup language)[11]和 SAML(security assertion markup language)[12]技术的出现,基于属性的授权成为目前分布式环境中广泛应用的一种新兴的访问控制手段[13−15].基于属性的访问控制实施细粒度的授权,支持传统的粗粒度访问控制模型,如访问控制列表、强制访问控制模型、基于角色的访问控制模型等.近年来,基于属性的策略合成开始受到关注[16,17],但这些研究提出的合成方式本质上就是文献[7]中的并、交、差算子.

但是,由于文献[7]提出的策略合成代数模型未刻画访问实体属性,不能支持一些涉及属性协商的策略合成场景.例如,有两个医疗机构 H 和 G,过去一直独立地提供在线医疗服务.由于业务扩展等需求,H,G 决定建立联盟,共同为用户提供新的在线医疗服务,自然地,这些新服务的访问控制策略由 H,G 共同确定.若对于一级护理服务,机构 H 原有的策略为“付费金额大于 7 美元的用户可获得一级护理”,G 原有的策略为“付费金额大于 5 美元的用户可获得一级护理”,若双方协商确定按“平均原则”制定策略,即需要产生一条“付费金额大于 6 美元的用户可获得一级护理”作为新的访问控制策略,按照文献[7]提供的合成算子就无法刻画这条新策略.在这个例子中,由于协商出的合成策略涉及属性值计算(取平均数),因此,无法用集合的交、并、差等运算来刻画.事实上,在商业协作应用中还有大量涉及属性值计算的策略合成例子[18].

针对此问题,我们首先基于属性刻画了实体间的授权关系,通过属性值的计算结构扩展了现有的策略合成代数,提出了基于属性的策略合成代数模型(attribute-based access control policy composition algebra,简称 APoCA);其次,通过示例分析说明 APoCA 既支持现有的策略合成代数,又适用于更为复杂的应用场景,具有更强的策略合成方式描述能力和普适性;然后,用策略表达式形式化地描述聚合资源的访问控制策略,讨论策略表达式的若干代数性质,说明可借助策略表达式的代数性质去验证策略合成结果是否符合各方对聚合资源的保护性需求;最后,给出将代数表达式翻译成逻辑程序的翻译器,为聚合资源的访问控制策略评估和应用提供基础.

本文第 1 节给出基于属性的访问控制策略合成代数模型(APoCA).第 2 节给出应用实例分析.第 3 节讨论 APoCA 的表达式及其若干性质.第 4 节讨论访问控制策略评估.第 5 节进行相关工作分析与比较.第 6 节得出结论并提出下一步的工作.

1 基于属性的访问控制策略合成代数模型

一般地,访问控制策略规定访问主体(资源请求者)能否对客体(资源)进行操作.基于属性访问控制的基本观点是不直接在主体和客体之间定义授权,而是利用与其关联的属性作为授权决策的基础.主体和客体是访问控制的基本实体,每个实体均有可描述其特征的一些关联属性,如与主体关联的属性,包括年龄、信任度、安全级别和付费金额等.由于资源请求者可能同时担当被访问的对象(如一个请求调用服务的服务同时被另一个服务调用),故与主体关联的属性一般也是客体的关联属性;同时,随着应用对非功能属性需求的日益增长,为支持对具有不同非功能属性的服务资源实施细粒度访问控制,将体现服务质量的诸如资源的成功使用率、可用率、响应时间等[18]也视为与客体关联的属性.此外,操作与客体经常是密切相关的,例如,基于角色访问控制模型中用权限的概念统一刻画了客体和对客体的操作,这里我们把操作作为客体不可缺少的关联属性.

本文假设制定聚合资源策略的各方均采用基于属性的访问控制,并且各方对实体所有的关联属性有一个共同的认识.事实上,随着基于属性的访问控制的发展,这是在分布式跨域协作应用中常见和典型的环境假设.虽然环境的开放性和安全需求的多样性可能导致需要考虑的属性很多,但对常见的公共属性,可以建立一个通用的属性本体(ontology),然而这不是本文研究的重点,相关研究见文献[19].值得注意的是,这并不意味着各方访问控制都考虑相同的公共属性,允许各方策略具有不同的属性(即异构环境).另外,本文的工作主要解决策略合成方式的刻画问题,我们假定协作各方已经具有相应的访问控制策略.

为便于描述,本文采用大写 $SATTR$,$OATTR$ 加下标分别表示主体、客体的关联属性,S,O 加对应下标分别表示它们的值域.小写 s,o 加上/下标分别表示主体、客体的关联属性值,大写 P 表示访问控制策略的集合,小写 p 和 p 加上\下标表示访问控制策略.

下面,我们先定义属性元组,用属性元组值去标识具体的访问主体和客体,给出刻画具体授权关系的属性授权项的定义;然后以此为基础定义访问控制策略,通过讨论属性授权项的计算,建立策略合成的代数框架.

定义 1(属性元组). 我们称$\langle SATTR_1,SATTR_2,\ldots,SATTR_m\rangle$和$\langle OATTR_1,OATTR_2,\ldots,OATTR_n\rangle$分别为主体和客体的属性元组,其中 $OATTR_n$ 为操作属性.对任意 $s_i\in S_i(i=1,2,\ldots,m)$,$o_j\in O_j(j=1,2,\ldots,n)$,属性元组值$\langle s_1,s_2,\ldots,s_m\rangle$表示各属性值分别为 $s_1,s_2,\ldots,s_m$ 的资源请求者,称为主体$\langle s_1,s_2,\ldots,s_m\rangle$;$\langle o_1,o_2,\ldots,o_n\rangle$表示各属性值分别为 $o_1,o_2,\ldots,o_n$ 的资源,称为客体$\langle o_1,o_2,\ldots,o_n\rangle$.

定义 2(属性授权项). 形如$[\langle s_1,s_2,\ldots,s_m\rangle,\langle o_1,o_2,\ldots,o_n\rangle]$的元组称为属性授权项,表示主体$\langle s_1,s_2,\ldots,s_m\rangle$能够访问客体$\langle o_1,o_2,\ldots,o_n\rangle$[**].

直观上,属性授权项规定了一个具体被允许的访问,而访问控制策略一般刻画实体之间的授权关系,故有:

定义 3(访问控制策略). 访问控制策略是若干属性授权项的集合,记作 p.

注意到各方策略考虑的实体属性可能不尽相同.例如,一方考虑年龄 60、信任度为 0.7 的主体,另一方考虑年龄 60、安全级别为 3 的主体.而在确定聚合资源策略时,必须比较各方策略.因此,为便于比较,我们引入一个特殊的符号Δ,其值为空,表示未考虑该属性,用符号Δ扩展属性元组值.如上例中表示双方主体的属性元组值可分别扩展为$\langle 60,0.7,\Delta\rangle$和$\langle 60,\Delta,3\rangle$.由定义 2 易知,对任意一个属性授权项,扩展其主体或客体的属性元组值就等于扩展了这个属性授权项.这样,利用属性授权项的扩展就可以对异构策略进行比较.

定义 4(相容属性授权项). 设 a,a' 为任意两个属性授权项,令$[\langle s_1,s_2,\ldots,s_k\rangle,\langle o_1,o_2,\ldots,o_l\rangle]$和$[\langle s'_1,s'_2,...,s'_k\rangle,\langle o'_1,o'_2,...,o'_l\rangle]$分别为它们的扩展,如果满足下面两个条件,则称 a 和 a'为相容的:1) 对 $i=1,2,\ldots,k$,$s_i=\Delta$当且仅当 $s'_i=\Delta$;2) 对 $j=1,2,\ldots,l$,$o_j=\Delta$当且仅当 $o'_j=\Delta$.

直观上,若两个属性授权项的主体和客体的属性元组相同,那么它们一定相容.易见,属性授权项之间的相容关系是等价关系,即满足自反性、对称性和传递性.进一步地,我们有:

定义 5(相容策略). 设 p_1,p_2 为任意两个访问控制策略,若 p_1 中的属性授权项与 p_2 中的属性授权项均相容,

---

** 这里,我们把操作看作与客体关联的属性,o_n 为一个具体的操作.

则称策略 p_1 和 p_2 是相容的.

策略相容关系为等价关系可由属性授权项相容关系等价得到.根据上述定义,各方策略不一定相容.甚至一方的策略也可能由不相容的属性授权项构成.因此,一个好的模型应该能够刻画这些策略的合成方式.

先对策略的合成方式作如下分析:当两方策略相容时,直接合并其属性授权项或从它们的策略中取出相同的属性授权项等都是策略合成的方式,具有集合算子语义[7],但这些尚不足以支持所有相容策略的合成场景.正如文章开始部分所给出的例子,机构 H,G 均以〈付费金额〉作为主体的属性元组,以〈护理级别〉作为客体的属性元组,并均对护理级别为"一"的服务进行访问控制,即 H,G 策略是相容的,但"付费金额大于 6 美元的用户可获得一级护理"这样的合成结果无法由集合运算得到.实际上,该合成策略中的每个属性授权项由 H 策略的一个属性授权项和 G 策略的一个属性授权项共同确定,这体现了属性授权项的计算.更进一步可以发现,属性授权项的计算实际上是由各对应属性值的计算确定,如此例中付费金额属性值的计算.属性值之间的计算可由该属性值域上的算子刻画.例如,不少研究用信任度属性度量对用户信任的可能性,其值域为[0,1],那么[0,1]上的任意一个有限维算子如均值、最小值算子等都可用来计算出一个新的信任度值.此外,当两方策略不相容时,即至少存在 1 对不相容的属性授权项时,具有集合运算语义的算子显然无法刻画它们的合成方式,需要新算子的支持.

下面,我们把Δ加到各属性值域中,通过定义属性值域上算子的扩张,给出属性授权项计算的定义,为后面统一刻画策略的合成方式提供基础.

定义 6(算子的扩张). 设 D 为一个属性的值域,对 $D\cup\{\Delta\}$上任意一个二维算子 $f^\wedge$,若存在 D 上的二维算子 f,使 $f^\wedge|_D=f$ 成立,我们称 $f^\wedge$为 D 中二维算子的扩张.这里的 $f^\wedge\big|_D=f$ 是指,对任意的 $d,d'\in D$,$f^\wedge(d,d')=f(d,d')$.

为定义属性授权项的计算,先介绍前面属性元组值扩展的逆过程——非Δ投射,直观地看,属性元组值扩展经过非Δ投射就转化为原来的属性元组值.如属性元组值扩展〈60,0.7,Δ〉和〈60,Δ,3〉的非Δ投射结果分别为〈60,0.7〉和〈60,3〉.

定义 7(属性授权项的计算). 设 a,a'为任意两个属性授权项,其中$[\langle s_1,s_2,\ldots,s_k\rangle,\langle o_1,o_2,\ldots,o_l\rangle]$和$[\langle s_1',s_2',\ldots,s_k'\rangle,\langle o_1',o_2',\ldots,o_l'\rangle]$分别为它们的扩展,且 $f_i^\wedge$和 $g_j^\wedge$ 分别为 $S_i(i=1,2,\ldots,k)$和 $O_j(j=1,2,\ldots,l)$上二维算子的扩张.记 $w=(f_1^\wedge,\ldots,f_k^\wedge,g_1^\wedge,\ldots,g_l^\wedge)$,$s_i''=f_i^\wedge(s_i,s_i')(i=1,2,\ldots,k)$,$o_j''=g_j^\wedge(o_j,o_j')(j=1,2,\ldots,l)$且$\langle s_{a1}'',s_{a2}'',\ldots,s_{ab}''\rangle$为$\langle s_1'',s_2'',\ldots,s_k''\rangle$的非Δ投射结果,$\langle o_{c1}'',o_{c2}'',\ldots,o_{cd}''\rangle$为$\langle o_1'',o_2'',\ldots,o_l''\rangle$的非Δ投射结果,那么我们称 $a\ w\ a'=[\langle s_{a1}'',s_{a2}'',\ldots,s_{ab}''\rangle,\langle o_{c1}'',o_{c2}'',\ldots,o_{cd}''\rangle]$ 为 a,a'在 w 下的计算结果.

例 1:设 a,a'为属性授权项,其中 a=[〈60,0.7〉,〈2,读〉],表示年龄 60、信任度为 0.7 的主体可访问安全级别为 2、操作属性为读的客体;a'=[〈60,3〉,〈读〉],表示年龄 60、安全级别为 3 的主体可访问操作属性为读的客体.比较 a,a'有[〈60,0.7,Δ〉,〈2,读〉]和[〈60,Δ,3〉,〈Δ,读〉]分别为 a 和 a'的扩展,取 $w=(f_1,f_2^\wedge,f_3^\wedge,g_1^\wedge,g_2)$,其中 $f_1(60,60)=60$,$f_2^\wedge(0.7,\Delta)=0.7$,$f_3^\wedge(\Delta,3)=\Delta$,$g_1^\wedge(2,\Delta)=\Delta$,$g_2$(读,读)=读.取〈60,0.7,Δ〉的非Δ投射〈60,0.7〉,〈Δ,读〉的非Δ投射〈读〉,则 a,a'在此 w 计算下的结果为属性授权项[〈60,0.7〉,〈读〉],表示年龄 60、信任度为 0.7 的主体可访问操作属性为读的客体.

特别地,当 a,a'本身为相容属性授权项时,则有 $w=(f_1,\ldots,f_k,g_1,\ldots,g_l)$,其中 f_i 和 g_j 分别为 $S_i(i=1,2,\ldots,k)$和 O_j $(j=1,2,\ldots,l)$上的二维算子.用 A 表示所有属性授权项的集合,F_A 表示所有 w 的集合,则有下面的定理:

定理 1. (A,F_A)构成一个代数系统.

证明:由定义 7 中属性授权项关于 F_A 中的 w 运算封闭可证. □

有了上述基本概念,下面我们建立刻画策略合成结构的代数模型.首先定义只支持相容策略的合成算子.对任意相容的策略 p_1,p_2,有:

定义 8(⊕算子). $p_1\oplus p_2=\left\{\left[\langle s_1,\ldots,s_m\rangle,\langle o_1,\ldots,o_n\rangle\right]\middle|\left[\langle s_1,\ldots,s_m\rangle,\langle o_1,\ldots,o_n\rangle\right]\in p_1\text{或}\left[\langle s_1,\ldots,s_m\rangle,\langle o_1,\ldots,o_n\rangle\right]\in p_2\right\}$. $p_1\oplus p_2$ 表示一条新的访问控制策略,规定如果一个访问授权(属性授权项)被 p_1 允许或被 p_2 允许,那么它被 $p_1\oplus p_2$ 允许.直观上,此算子的作用就是合并 p_1 和 p_2 的属性授权项.

定义 9(⊙算子). $p_1\odot p_2=\left\{\left[\langle s_1,\ldots,s_m\rangle,\langle o_1,\ldots,o_n\rangle\right]\middle|\left[\langle s_1,\ldots,s_m\rangle,\langle o_1,\ldots,o_n\rangle\right]\in p_1\text{且}\left[\langle s_1,\ldots,s_m\rangle,\langle o_1,\ldots,o_n\rangle\right]\in p_2\right\}$. $p_1\odot p_2$ 表示一条新的访问控制策略,规定如果一个访问授权同时被 p_1 和 p_2 允许,那么它被 $p_1\odot p_2$ 允许.直观上,此算子

的作用就是提取相同的属性授权项.

定义 10(−算子). $p_1-p_2=\{[\langle s_1,...,s_m\rangle,\langle o_1,...,o_n\rangle]\mid[\langle s_1,...,s_m\rangle,\langle o_1,...,o_n\rangle]\in p_1$但$[\langle s_1,...,s_m\rangle,\langle o_1,...,o_n\rangle]\notin p_2\}$. p_1-p_2表示一条新的访问控制策略,规定如果一个访问授权在 p_1 中被允许且在 p_2 中不被允许,那么它在 p_1-p_2 中被允许.直观上,就是从 p_1 中删除在 p_2 中的属性授权项.显然,可以用这个算子来定义否定授权.

容易看出,上述 3 个算子具有集合运算并、交、差的语义[7],可用来解决传统的策略冲突,支持否定优先或肯定优先原则.例如,在运行时刻,访问请求"信任度为 0.5 的用户想读医疗数据库的所有文件"遇到下面两条策略:p_1={信任度大于 0.3 的用户可读医疗数据库中的所有文件};p_2={信任度大于 0.7 的用户可读医疗数据库中的所有文件}.若使用 p_1 则该访问请求被允许,若使用 p_2 则被否定,故这两条策略是冲突的.若采用否定优先原则,则此时对"读医疗数据库的所有文件"的策略应该为 $p_2\odot p_1=p_2$;若采用肯定优先原则,则为 $p_1\oplus p_2=p_1$.

下面的算子同时可刻画相容策略和不相容策略的合成方式.对任意的策略 p_1,p_2,p,有如下定义:

定义 11($\hat{}_c$算子). $p_c^{\wedge}=\{[\langle s_1,...,s_m\rangle,\langle o_1,...,o_n\rangle]\mid[\langle s_1,...,s_m\rangle,\langle o_1,...,o_n\rangle]\in p$且$[\langle s_1,...,s_m\rangle,\langle o_1,...,o_n\rangle]$满足约束$c\}$. $p^{\wedge}_c$表示一条新的访问控制策略,规定如果一个访问授权在 p 中被允许且满足约束 c,那么它在 $p^{\wedge}_c$ 中也被允许.直观上,$\hat{}_c$算子实际是给 p 加了一个限制条件 c,删除了不满足 c 的属性授权项,缩小了 p 的范围.$\hat{}_c$算子可看作是一个含参数 c 的高阶"算子",$\hat{}_c$算子依赖于具体的约束 c,约束 c 一般由谓词或关系式构成,这些谓词或关系式取自专门的授权约束语言,这不是本文的研究重点,我们将另文讨论.具体实例详见下一节.

定义 12. 对 F_A 中任意一个 w,有"$\otimes_w$算子",其中, $p_1\otimes_w p_2=\{a_1\ w\ a_2\mid a_1\in p_1,a_2\in p_2\}$.$p_1\otimes_w p_2$表示一条新的访问控制策略,其包含的属性授权项是 p_1 和 p_2 中的属性授权项在 w 计算下的结果.

$\otimes_w$算子同样可看作是含参数 w 的高阶算子,其依赖于代数系统(A,F_A)的算子 w.由定义 7 可知,w 实际上是不同属性值域上二维算子扩张的联合作用.这些扩张可选择的灵活性,使得$\otimes_w$ 具有灵活性,这也使得我们的合成算子可以刻画现有工作无法支持的合成方式.下一节,我们将举例说明此算子的表现力.

定理 2. 设 P 是访问控制策略的集合,$O=\{\hat{}_c,\oplus,\odot,-,\otimes_{w\mid w\in F_A}\}$ 为策略合成算子的集合,则(P,O)可构成一个代数系统,称为基于属性的访问控制策略合成代数(APoCA).

证明:由定义 6~定义 12 中 P 关于 O 中算子的封闭性可得. □

公理 1(策略合成算子性质).

1) $\oplus,\odot,\otimes_{w\mid w\in F_A}$满足结合律;

2) $\oplus,\odot$满足交换律;

3) $\oplus,\odot$满足分配律; $\oplus,\odot,\otimes_{w\mid w\in F_A}$ 满足分配律; $\odot,\otimes_{w\mid w\in F_A}$ 满足分配律.

2 应用分析

这一节,我们通过具体例子来说明 APoCA 模型既支持现有工作,也支持现有工作失效的策略合成场景.

在医疗应用中,海量且高度分散的临床医疗信息被不同医疗组织记录在各自的医疗信息库中,医疗信息可能涉及个人或群体隐私,故各组织采用一定的访问控制策略去开发信息系统.许多临床案例经常需要组织之间的信息共享,医疗人员可通过网络以付费等方式获取其他组织的电子医疗信息.而不同信息系统的导入和运行易使某些信息(如病人的敏感病历)泄露,或无法确定某些医疗工作者是否具有访问权限.因此,必须统一确定各组织共享信息(聚合资源)的访问控制策略.下面我们以组织 A,B,C,D 为例.假设它们的策略分别为:

- 组织 A:"安全级别大于 2 且付费金额不低于 5 美元的用户能够读安全级别小于 2 的医疗信息".
- 组织 B:"安全级别大于 2 且付费金额不低于 7 美元的用户能够读安全级别小于 2 的医疗信息".
- 组织 C:"付费金额不低于 7 美元的用户能够读安全级别小于 2 的医疗信息".
- 组织 D:"安全级别大于 2 且付费金额不低于 5 美元的用户都能够读医疗信息".

基于本文的APoCA模型,这里组织A,B的主体属性元组均为⟨安全级别,付费金额⟩,客体属性元组为⟨安全级别,操作⟩;组织C的主体属性元组为⟨付费金额⟩,客体属性元组为⟨安全级别,操作⟩;组织D的主体属性元组为⟨安全

级别,付费金额⟩,客体属性元组为⟨操作⟩.安全级别属性值域均为$\mathbb{N}$,付费金额属性值域为$\mathbb{R}^+$,操作属性值域为{读,写,…}.各组织的策略可形式化如下:

- 组织 A:p_A={[⟨s_{1A},s_{2A}⟩,⟨o_{1A},o_{2A}⟩]|s_{1A}>2,s_{2A}≥5,o_{1A}<2,o_{2A}=读}.
- 组织 B:p_B={[⟨s_{1B},s_{2B}⟩,⟨o_{1B},o_{2B}⟩]|s_{1B}>2,s_{2B}≥7,o_{1B}<2,o_{2B}=读}.
- 组织 C:p_C={[⟨s_{2C}⟩,⟨o_{1C},o_{2C}⟩]|s_{2C}≥7,o_{1C}<2,o_{2C}=读}.
- 组织 D:p_D={[⟨s_{1D},s_{2D}⟩,⟨o_{2D}⟩]|s_{1D}>2,s_{2D}≥5,o_{2D}=读}.

下面,我们分析 4 个组织的策略两两合成的可能结果.

(1) 组织 A,B 策略可能的合成结果,注意到 A,B 策略是相容策略.

1) 如果 A 同意“B 不允许的访问不能被访问”,则合成结果为 $p_A \odot p_B = p_B$;

2) 如果 B 同意“A 允许的访问”,则合成结果为 $p_A \oplus p_B = p_A$;

3) 如果A,B都各退一步,则合成结果可能为“安全级别大于 2 的用户且付费金额不低于 6 美元的用户能够读安全级别小于 2 的医疗信息”,即$p_A \otimes_{w0} p_B$={[⟨s_1,s_2⟩,⟨o_1,o_2⟩]|s_1>2,s_2≥6,o_1<2,o_2=读},其中w_0=(f_{01},f_{02},g_{01},g_{02}),f_{02} 为$\mathbb{R}^+$上的均值算子,f_{01},g_{01} 为$\mathbb{N}$上最大值算子,g_{02} 为{读,写,…}上二元算子且满足对任意o_{21},o_{22}∈{读,写,…},若o_{21}=o_{22},则有g_{02}(o_{21},o_{22})=o_{22}.

以上合成结果说明,由于算子⊙和⊕具备集合运算的语义,故分别支持 P.Bonatt 模型[7]的 Addition, Conjunction 算子,但 P.Bonatt 模型不能刻画场景 3)的合成结果.

(2) 组织 A,D 策略可能的合成结果,注意到 A,D 策略不相容(不相容原因在于 D 未考虑客体安全级别属性).

1) 如果 D 实际规定“安全级别大于 2 且付费金额不低于 5 美元的用户都能够读安全级别为任何值的医疗信息”,那么对“读安全级别小于 2 的医疗信息”的控制与 A 一样,若 A 同意“D 允许的访问”,则合成结果为 $p_A \oplus p_D{}^{\wedge}_{c_1} = p_D$,其中 c_1 为约束“客体安全级别不小于 2”,即 o_{1D}≥2;

2) 若 D 同意“A 不允许的访问不能被访问”,则合成结果为 $p_A \odot p_D{}^{\wedge}_{c_2} = p_A$,$c_2$ 为约束“客体安全级别小于 2”,即 o_{1D}<2.

以上合成结果说明,APoCA 中的“$^{\wedge}_{c}$”算子可把不相容策略的合成转化为相容策略的合成,支持 P.Bonatt 模型[7]提出的范围限制(scoping restriction)算子.

(3) 组织 A,C 策略可能的合成结果,注意到 A,C 策略不相容(不相容原因在于 C 未考虑主体安全级别属性).此外,容易看出 A,C 关注具有不同付费金额属性值的主体.

1) 若 C 实际规定“无论用户的安全级别为多少只要付费金额不低于 7 美元就能够读安全级别小于 2 的医疗信息”,那么当 C 同意“A 允许的访问”,A 同意“C 允许的访问”时,则合成结果为 $p_A \oplus p_C{}^{\wedge}_{c_3}$,其中 c_3 为约束“主体安全级别不大于 2”;

2) 若 A 同意“C 不允许的不能被访问”,C 同意“A 不允许的不能被访问”,则合成结果为 $p_A \odot p_C{}^{\wedge}_{c_4}$,其中 c_4 为约束“主体安全级别大于 2”;

3) 若A在主体安全级别上退一步,C在主体付费金额上退一步,则可能的合成结果为“付费金额不低于 5 美元能够读安全级别小于 2 的医疗信息”,即$p_A \otimes_{w1} p_C$={[⟨s_2⟩,⟨o_1,o_2⟩]|s_2≥5,o_1<2,o_2=读},其中w_1=($f_{11}^{\wedge}$,f_{12},g_{11},g_{12}),$f_{11}^{\wedge}$为$\mathbb{N}$上任二元算子的扩张且满足对任意s_{11}∈$\mathbb{N}$,$f_{11}^{\wedge}$(Δ,s_{11})=$f_{11}^{\wedge}$(s_{11},Δ)=Δ,f_{12} 为R^+上的最小值算子,g_{11} 为$\mathbb{N}$上最大值算子,g_{12} 如上g_{02};

4) 进一步来说,当A,C在主体付费金额上各退一步,而在主体安全级别上C同意A,则合成结果为“安全级别大于 2 且付费金额不低于 6 美元能够读安全级别小于 2 的医疗信息”,即$p_A \otimes_{w1} p_C$={[⟨s_1,s_2⟩,⟨o_1,o_2⟩]|s_1>2,s_2≥6,o_1<2,o_2=读},其中w_2=($f_{21}^{\wedge}$,f_{22},g_{21},g_{22}),$f_{21}^{\wedge}$为$\mathbb{N}$上任意一个二元算子的扩张且满足对任意s_{11}∈$\mathbb{N}$,$f_{21}^{\wedge}$(Δ,s_{11})=$f_{21}^{\wedge}$(s_{11},Δ)=s_{11},f_{22} 为R^+上的均值算子,g_{21},g_{22} 分别如上g_{11},g_{12}.

以上合成结果说明,APoCA 模型能够刻画现有基于属性的策略合成研究[16,17]支持的场景 1)和场景 2),但是现有工作无法刻画场景 3),4)中的合成结果.

对于组织 B,C,组织 B,D 和组织 C,D 的策略合成情况,类似于上面(2)、(3)的讨论.

这个医疗应用的例子说明,APoCA 模型不仅能够刻画现有工作支持的合成方式,如(1)~(3)中的场景 1)和 2),同时能刻画现有工作不支持的合成方式,如(1)中的场景 3)及(3)中的场景 3),4).限于篇幅,为使本文紧凑且具有良好的可读性,这里我们未进行繁琐的理论证明.

必须指出,虽然 APoCA 提供了更为丰富的刻画策略合成方式的"算子",但算子的选择依赖具体的应用场景,APoCA 并不能解决合成算子自动选择问题.要使聚合资源策略的产生过程能够完全自动化,除了刻画策略的合成方式,即定义"合成算子"以外,还必须提供选择合成算子的机制,并给出评价合成结果的方法.合成算子的选择涉及到多方的安全策略约束,与具体的应用场景相关.正如本节所述,各组织对合成算子的选择是不一样的,如(1)的场景 1)中组织 A 选择$\odot$算子,场景 2)组织 B 选择$\oplus$算子;(2)中组织 D 还可以选择不同的合成算子.由于各方可能选择不同的合成算子,可能得到不同的合成结果,因此,对各方合成结果的评价有助于最后合成算子的选择.合成结果的好坏一般根据协商结果(即各方达成一致的、对聚合资源共同的保护性需求)来评价.下一节,我们将讨论 APoCA 的代数表达式,提出一种评价合成结果的方法.关于合成算子的选择算法,需要根据具体的安全策略约束和应用问题来确定,然而这不是本文解决的重点,相关工作可见文献[1,2].

3 策略表达式

本节研究 APoCA 的表达式,它可形式化描述策略的合成结果.首先给出策略表达式的定义,然后讨论了若干表达式的代数性质,最后例示说明可通过代数表达式的性质去验证合成结果是否符合各方的协商结果,即是否满足各方达成一致的、对聚合资源共同的保护性需求.继上一节的应用例子,假设有:

- 组织 A,B 对聚合资源的保护性需求 1:除非能够得到组织 A,B 同时允许,否则,没有用户能够读安全级别小于 2 的医疗信息.
- 组织 D,B,A 对聚合资源的保护性需求 2:一个用户若能读安全级别小于 2 的医疗信息,则他必须得到组织 A 的同意.

我们先给出代数系统(P,O)表达式的递归定义.

定义 13(访问控制策略表达式). 设 P 是访问控制策略的集合,V 是访问控制策略变量的集合,对任意 $p\in P$, $v\in V$,可如下递归定义访问控制策略表达式(简称策略表达式):

(1) v 和 p 为访问控制策略表达式;

(2) 如果 E_1 和 E_2 均为访问控制策略表达式,那么 E_1 和 E_2 经过 O 中算子的有限次运算后仍是访问控制策略表达式.

由定义容易看出,策略表达式可刻画包含未知策略的合成结果.文献[7]中定义的模板算子实际上就是一个含有变量的策略表达式.下面,我们给出几个策略表达式的代数性质.

命题 1. 设 $E(x,y)=(x-\hat{x}_c)\oplus(\hat{x}_c\odot y)$为一个策略表达式,对任意访问控制策略常量 p_1,p_2 和满足 c 的属性授权项 a,有 $a\in E(p_1,p_2)\Leftrightarrow a\in p_1$ 且 $a\in p_2$.

证明:"$\Rightarrow$"对 $a\in E(p_1,p_2)=(p_1-\hat{p}_{1c})\oplus(\hat{p}_{1c}\odot p_2)$,有 $a\in p_1-\hat{p}_{1c}$ 或 $a\in\hat{p}_{1c}\odot p_2$.由于 a 满足 c,故 $a\notin p_1-\hat{p}_{1c}$,则 $a\in\hat{p}_{1c}\odot p_2$,而 $\hat{p}_{1c}\subseteq p_1$,因此 $a\in p_1$ 且 $a\in p_2$.

"$\Leftarrow$"因为 $a\in p_1$ 且 $a\in p_2$,所以 $a\in p_1\odot p_2$,因此 $a\in(p_1-\hat{p}_{1c})\oplus(\hat{p}_{1c}\odot p_2)=E(p_1,p_2)$. □

如果令 c 表示约束"客体安全级别小于 2",x,y 分别为表示组织 A,B 的策略变量,那么无论组织 A,B 的策略是什么,通过此性质就可验证组织 A,B 策略的合成结果是否满足需求 1.

命题 2. 设 $E(x,y,z)=\hat{x}_{c_1}\oplus(\hat{y}_{c_2}\odot z)$为一个策略表达式,$c_1$ 和 c_2 不能同时满足,那么对任意访问控制策略常量 p_1,p_2,p_3 和满足 c_2 的属性授权项 a,若 $a\in E(p_1,p_2,p_3)$,则有 $a\in p_3$.

证明:因为 $a\in E(p_1,p_2,p_3)=\hat{p}_{1c_1}\oplus(\hat{p}_{2c_2}\odot p_3)$且 a 满足 c_2 以及 c_1 和 c_2 不能同时满足,所以 $a\notin\hat{p}_{1c_1}$,因此 $a\in\hat{p}_{2c_2}\odot p_3\subseteq p_3$. □

若令 c_1 表示约束"客体安全级别不小于 2",c_2 表示"客体安全级别小于 2",x,y,z 分别为表示组织 D,B,A 策略的变量,则无论组织 D,B,A 的策略是什么,均可根据此性质验证组织 D,B,A 的策略合成结果是否满足需求 2.

此外,表达式的代数性质还可以用来判断两相容策略是否存在不一致的属性授权项.

命题 3. 设 $E(x,y)=(x\odot y-y)\oplus((x-y)\odot y)$为一个策略表达式,那么对任意两相容策略 p_1,p_2,有 $E(p_1,p_2)=\varnothing$.

证明:假设存在属性授权项 $a\in E(p_1,p_2)=(p_1\odot p_2-p_2)\oplus((p_1-p_2)\odot p_2)$,则有 $a\in(p_1\odot p_2-p_2)$或 $a\in((p_1-p_2)\odot p_2)$,故有 $\begin{cases}a\in p_1\odot p_2\subseteq p_2\\ a\notin p_2\end{cases}$ 或 $\begin{cases}a\in p_1-p_2\Rightarrow a\notin p_2\\ a\in p_2\end{cases}$. 无论哪种情况都推出矛盾.故 $E(p_1,p_2)=\varnothing$. □

综上所述,使用表达式的代数性质去验证策略合成结果是否满足各方对聚合资源的保护性需求是一种评价合成结果的可行办法.

4 访问控制策略评估

访问控制策略评估是指判断一个访问请求是否违背访问控制策略.由上一节的讨论可知,聚合资源策略可用一个策略表达式形式化地描述(这里称为聚合资源的策略表达式).在聚合资源的策略表达式给定的条件下,对聚合资源的策略评估有两种常见的方法,一是先计算聚合资源的策略表达式的值,确定聚合资源策略包含的所有属性授权项(即先编译聚合资源策略),当有请求时,则可直接判断;另一种是当用户提出访问请求时,才实时地计算聚合资源的策略表达式的值,然后再作判断.显然,当聚合资源的策略表达式时包含未知量,即不可知策略或部分可知策略(例如信任协商场景中,带有敏感信息的策略是逐步披露的),第 1 种方法失效.如果某方策略经常变化时,则使用第 1 种方法需要反复进行表达式值的计算,效率较低;而第 2 种方法由于没有事先明确允许的访问(即属性授权项),额外的计算时间显然增加了授权判断的开销.

局部评估方法可以结合前面两种方法的优点:先计算策略表达式中仅包含常量(即已知、不易变化的策略)的子表达式值,求出相应的属性授权项;把剩下的那些易变化、未知或部分可知的策略留到运行时刻再进行变量替换.例如,若策略表达式 $x\otimes_w(p_2\odot p_3)$刻画了一个 Web 组合服务 s 的策略,其中 p_2,p_3 分别为已知的组件服务 s_2,s_3 的策略,x 为未知的组件服务 s_1 的策略.局部评估方法允许先计算 $p_2\odot p_3$ 的值,当有用户请求访问服务 s 时,再根据获得的组件服务 s_1 的策略,计算出 s 的策略,最后进行授权判断.

当前,对逻辑程序的局部评估(partial evaluation)技术已较为成熟(如 Datalog 自底向上引擎,Prolog 自顶而下评估[20]等).类似于文献[7],我们希望 APoCA 能够支持对聚合资源访问控制策略的局部评估.因此,本节给出翻译器 pe2lp,其功能是完成 APoCA 的策略代数表达式到逻辑程序的翻译.

语法分析是翻译的必经阶段,要使 pe2lp 能够对策略表达式进行语法分析,则必须提供表达式语法树中非叶点的标识方法,这可以通过使用不同正整数对各算子作标号来实现.形式化地,带标号表达式定义如下.

定义 14(带标号的表达式). 对访问控制策略表达式 E 中的每一个算子,从左到右,由 0 开始编号,我们称经过编号的表达式为原表达式的带标号表达式,记作 E^l,其中 l 为 E 中最后进行计算的算子标号.称为 E^l 的主标号.

例 2:一个不含变量的策略表达式 $p_1\oplus(p_2-p_3)$的带标号表达式为 $p_1\overset{0}{\oplus}(p_2\overset{1}{-}p_3)$,其主标号为 0.

翻译器 pe2lp 的输入为一个带标号表达式 E^l,输出为与其等价的逻辑程序,记作 $pe2lp(E^l)$.这里我们基于 RT(role-based trust-management framework)[21]中的参数化角色把访问控制策略翻译为逻辑规则.RT 中的参数化角色 $A.r(r_1,\ldots,r_t)$表示 A 断言的同时充当角色 $r_1,\ldots,r_t$ 的实体集合.*ismember* 是一个谓词,

$$ismember(u,A.r(r_1,...,r_t))=\begin{cases}1, & u\in A.r(r_1,...,r_t)\\ 0, & \text{否则}\end{cases}.$$

当A表示属性权威[***]时,$A.r(r_1,\ldots,r_t)$就表示A断言的同时具有属性$r_1,\ldots,r_t$的实体集合.当r=s时,$A.s(s_1,\ldots,s_m)$是关于主体的;当r=o时,$A.o(o_1,\ldots,o_n)$是关于客体的.必须指出,属性授权项应基于相同的属性权威.

另外,用$(\cdot)_l$表示括号内公式是策略表达式中编号为 l 的算子对应的逻辑规则,特别地,用$(\cdot)_F$表示括号内公式

[***] 属性权威是策略制定者信任的对象,一般是 PMI 中的(attribute authority,简称 AA).本文假定参与策略合成的各方对彼此的属性权威都是信任的.信任如何建立属于信任管理的研究范畴,超出本文的研究范围.

是策略表达式 F 的主标号算子对应的逻辑规则.那么,对带标号表达式 E^l,可如表 1 递归定义的翻译器 pe2lp.

Table 1 Translator pe2lp: From policy algebraic expressions to logic programs

表 1 翻译器 pe2lp:从策略代数表达式到逻辑程序

E^l	$pe2lp(E^l)$
p	$\begin{cases}\{(ismember(?z,A.o(o_1,...,o_n)):-ismember(?z,A.s(s_1,...,s_m)))_p \mid (s_1,...,s_m,o_1,...,o_n)\in p, & \text{if } p\neq\varnothing\\ \varnothing, & \text{otherwise}\end{cases}$
$F\overset{i}{\oplus}G$	$\{(ismember(?z,A.o(y_1,...,y_n)):-ismember(?z,A.s(x_1,...,x_m)))_i \leftarrow$ $(ismember(?z,B_1.o(y_1,...,y_n)):-ismember(?z,B_1.s(x_1,...,x_m)))_F \vee$ $(ismember(?z,B_2.o(y_1,...,y_n)):-ismember(?z,B_2.s(x_1,...,x_m)))_G\}$ $\cup pe2lp(F)\cup pe2lp(G)$
$F\overset{i}{\otimes}G$	$\{(ismember(?z,A.o(y_1,...,y_n)):-ismember(?z,A.s(x_1,...,x_m)))_i \leftarrow$ $(ismember(?z,B_1.o(y_1,...,y_n)):-ismember(?z,B_1.s(x_1,...,x_m)))_F \wedge$ $(ismember(?z,B_2.o(y_1,...,y_n)):-ismember(?z,B_2.s(x_1,...,x_m)))_G\}$ $\cup pe2lp(F)\cup pe2lp(G)$
$F\overset{i}{-}G$	$\{(ismember(?z,A.o(y_1,...,y_n)):-ismember(?z,A.s(x_1,...,x_m)))_i \leftarrow$ $(ismember(?z,B_1.o(y_1,...,y_n)):-ismember(?z,B_1.s(x_1,...,x_m)))_F \wedge \neg$ $(ismember(?z,B_2.o(y_1,...,y_n)):-ismember(?z,B_2.s(x_1,...,x_m)))_G\}$ $\cup pe2lp(F)\cup pe2lp(G)$
$F\overset{i}{\underset{c}{\wedge}}$	$\{(ismember(?z,A.o(y_1,...,y_n)):-ismember(?z,A.s(x_1,...,x_m)))_i \leftarrow$ $(ismember(?z,B.o(y_1,...,y_n)):-ismember(?z,B.s(x_1,...,x_m)))_F \wedge tran(c,i)\}$ $\cup pe2lp(F)$
$F\overset{i}{\otimes}_w G$ $w=(f_1,...,f_m,g_1,...,g_n)$	$\{\left(ismember(?z,A.o(g_1(y_1,y_1'),...,g_n(y_n,y_n')):-ismember(?z,A.s(f_1(x_1,x_1')...,f_m(x_m,x_m'))\right)_i \leftarrow$ $(ismember(?z,B_1.o(y_1,...,y_n)):-ismember(?z,B_1.s(x_1,...,x_m)))_F \wedge$ $\left(ismember(?z,B_2.o(y_1',...,y_n')):-ismember(?z,B_2.s(x_1',...,x_m'))\right)_G\}$ $\cup pe2lp(F)\cup pe2lp(G)$
$(s_1,...,s_m,o_1,...,o_n)$	$ismember(?z,A.o(o_1,...,o_n)):-ismember(?z,A.s(s_1,...,s_m))$
c	$Tran(c,i)$

下面,我们给出不含变量的策略表达式翻译成逻辑程序的例子.

例 3:令 $E=p_1\otimes_w(p_2-p_3)$,$w=(f_1,\ldots f_m,g_1,\ldots,g_n)$其中,$p_1=\{[\langle s_1,...,s_m\rangle,\langle o_1,...,o_n\rangle],[\langle s_1',...,s_m'\rangle,\langle o_1',...,o_n'\rangle]\}$,其属性权威为 B_1;$p_2=\{[\langle s_1,\ldots,s_m\rangle,\langle o_1,\ldots,o_n\rangle],\ [\langle s_1',...,s_m'\rangle,\langle o_1',...,o_n'\rangle]\}$, $p_3=\{[\langle s_1,\ldots,s_m\rangle,\langle o_1,\ldots,o_n\rangle]\}$, 属性权威分别为 B_2,B_3. $p_1\overset{0}{\otimes}_w(p_2\overset{1}{-}p_3)$对应的逻辑程序如下:

p_1 规则: $(ismember(?z,B_1.o(o_1,...,o_n)):-ismember(?z,B_1.s(s_1,...,s_m)))_{p_1}$.

p_2 规则: $(ismember(?z,B_2.o(o_1,...,o_n)):-ismember(?z,B_2.s(s_1,...,s_m)))_{p_2}$.

p_3 规则: $(ismember(?z,B_3.o(o_1,...,o_n)):-ismember(?z,B_3.s(s_1,...,s_m)))_{p_3}$.

合成规则 1: $(ismember(?z,B.o(y_1,...,y_n)):-ismember(?z,B.s(x_1,...,x_m)))_1 \leftarrow$
$(ismember(?z,B_2.o(y_1,...,y_n)):-ismember(?z,B_2.s(x_1,...,x_m)))_{p_2} \wedge .$
$\neg(ismember(?z,B_3.o(y_1,...,y_n)):-ismember(?z,B_3.s(x_1,...,x_m)))_{p_3}$

合成规则 2: $(ismember(?z,A.o(g_1(y_1,y_1'),...,g_n(y_n,y_n')):-ismember(?z,A.s(f_1(x_1,x_1'),...,f_m(x_m,x_m'))))_0 \leftarrow$
$(ismember(?z,B.o(y_1',...,y_n')):-ismember(?z,B.s(x_1',...,x_m')))_1 \wedge$
$(ismember(?z,B_1.o(y_1,...,y_n)):-ismember(?z,B_1.s(x_1,...,x_m)))_{p_1}$

值得注意的是,逻辑程序中的逻辑规则实际上包括策略规则和策略合成规则两种.

这样,在聚合资源的策略表达式给定的条件下,通过翻译器 pe2lp,我们可以利用现有技术对聚合资源的访问控制策略实施局部评估.策略表达式的由来问题与合成算子的选择算法相关,正如前面第 2 节已述,合成算子的选择算法涉及到多方的安全策略约束,依赖于具体的应用场景,本文没有详细讨论策略表达式的由来,相关研究可见文献[1,2],翻译器 pe2lp 的作用是为聚合资源的访问控制策略局部评估和应用提供基础.

5 相关工作分析与比较

与本文相关的工作主要包括以下 4 个方面:

1) 现有的策略合成代数.Mclean[8]最早采用代数方法刻画了强制访问控制策略的合成结构,提出基于格的策略合成框架,但仅关注实体的安全级别,不足以适应商业协作环境.Bonatti 等人[7]提出了经典的策略合成代数模型,其基本思想是将访问控制策略定义为由主体、客体和动作构成的三元组(授权项)的集合,用并、交、差等代数算子抽象地刻画不同的策略合成方式.Jajodiae 等人[9]也提出了访问控制策略合成的命题代数,他们将策略形式化为主体、客体、动作三元组之间的非确定性变换(non-deterministic transformer),以命题条件为参数,根据命题条件确定具体的授权,主要支持动作(action)的合成.文献[7]提出的模型具有一定的普适性,能够支持文献[8,9]的合成代数,为后续研究[5]等奠定了基础,但正如第 2 节所述,它不能刻画涉及属性值计算的策略合成场景.

2) 基于属性的策略合成研究.基于属性的访问控制是目前分布式授权的主流[13−17,21].Ninghui Li 等人[21]提出的信任管理框架族(RT)是分布式访问控制研究领域具有代表性的研究成果,主要关注分布式环境中的权限委托及信任链发现等问题.该框架给出一种信任管理策略语言,支持参数化角色(角色与属性存在对应关系)的成员指定.尽管 RT 未直接讨论策略合成问题,但提出的 intersection roles,product roles 概念实际上可实现 APoCA 中“⊕算子”和“⊙算子”的功能.然而,由于 RT 以 Horn 子句为基础,受限于单调(monotonic)性假设,无法支持显式否定(explict negation),故在构造 difference roles 上受限,而显式否定对于访问控制策略中潜在冲突的消解是必要的.本文的 APoCA 不需要单调性假设,“−”和“$\hat{}_c$”算子能够更为方便、自然地定义否定授权策略,例如表达黑名单策略.此外,APoCA 还可刻画涉及属性值计算等更为复杂的策略合成方式,支持商业应用中基于属性协商的场景,更具普适性.Ferraiolo 等人[16]提出一种命名为策略机(policy machine)的访问控制机制,其主要功能是配置和实施基于属性的访问控制策略.该机制考虑隶属不同策略类(policy class)的策略合成,但正如摘自该文的例子,一条 RBAC 策略“医生角色可以读、写医疗记录”和一条 MLS 策略“安全级别为 M 的用户可读、写安全级别为 M 的医疗记录”的合成结果为策略“那些是医生且安全级别为 M 的用户可以读安全级别为 M 的医疗记录”.这是一种简单的策略合成场景,本质上是取各方策略的公共部分.2007 年,他们基于策略机研究了网格环境下的策略合成问题[17],本文提出的合成算子⊕,⊙,−,$\hat{}_c$ 可分别刻画他们新提出的 intersection,union,elimination 和 exclusion 这 4 种合成方式,但正如第 2 节所述,$\oplus_w$ 算子使得本文的工作更具普适性.

3) 策略协商.与策略合成密切相关的研究是策略协商[2].策略协商是指以各方策略为基础,协商确定聚合资源策略的过程.协商过程中各方每次的提案都是对已有策略的合成,因此策略协商可看作由多次策略合成构成的过程.由于基于属性的访问控制增加了灵活性和可扩展性,使得协商过程中可能产生更丰富的提案,因此,需要提供能够刻画更多策略合成方式的模型.本文的研究为策略协商提供了基础.如果缺乏协商过程,则一般意味着不需要协商或者各方不愿意协商.对于不需要协商的场景,聚合资源策略一般是取各方策略的公共部分,合成方式相对特殊,合成结果也没有多样性,参见文献[16];如果各方不愿意协商,则意味着有一方不愿意提供资源,因此无法进行资源共享,就没有合成策略的必要.激励资源共享是另一个热门的研究课题,我们做过相关研究[22].

4) 安全互操作.Gong 等人[23,24]研究了不同安全系统之间的安全互操作问题,提出了安全互操作应遵循的原则,通过实体之间的受限映射实现了不同安全系统之间的安全互操作,避免了互操作导致的安全冲突,提供了最大安全互操作方案.本质上,该方案解决了如何由多方策略合成出整个系统(多安全系统构成的整体)策略的问题,但需要各方提供自己的安全需求(显式否定).该研究基于粗粒度的访问控制模型,无法解决基于属性的策略合成问题.本文提供了刻画更丰富和细粒度策略合成方式的模型.

与现有工作相比,本文提出的策略合成代数主要突出了以下新特性:

(1) 具有更强的策略合成表达能力,适用于分布式环境下细粒度的访问控制策略合成刻画.本文提出的APoCA 模型既支持现有工作,又能刻画现有工作不支持的策略合成方式.支持商业协作应用中的策略协商,同时支持广泛应用于分布式环境的基于属性的访问控制机制,具有更为广泛的适用性.

(2) 支持策略合成的正确性分析.本文讨论了 APoCA 中算子的运算法则和表达式的代数性质,可用于分析策略合成结果(策略表达式)是否满足各方对聚合资源保护的需求.

(3) 更好地支持访问控制的灵活实现.本文以 APoCA 为基础,提供了将策略表达式翻译成逻辑程序的翻译器,从而可以利用逻辑程序的局部评估技术[20]实现对聚合资源访问控制策略的灵活而有效的评估.

6 结论和未来工作

本文研究了基于属性的访问控制策略合成的问题,建立了一个新的基于属性的策略合成代数模型 ApoCA.它既支持已有的策略合成框架,也能刻画涉及属性值计算等更为复杂的策略合成方式,为商业协作应用中的策略协商提供基础;通过讨论 APoCA 中策略表达式的代数性质,说明可以使用表达式的代数性质去验证策略合成结果是否满足各方对聚合资源的保护性需求;给出了将策略表达式翻译成逻辑程序的翻译器,为聚合资源的访问控制策略局部评估提供基础.

此外,为使聚合资源访问控制策略的确定过程能够完全自动化,除了刻画策略合成方式,还必须提供合成算子的选择机制以及合成结果的评价方法.根据策略表达式的代数性质去验证合成的策略是否满足各方对聚合资源保护的需求是评价合成结果的一种可行方法,这促使我们进一步挖掘更多策略表达式的代数性质.另外,针对不同安全约束及应用,研究相应的算子选择算法也是一项有意义的工作.

致谢 本文的工作得到项目组李建欣老师、邓婷同学的建议和帮助,在此表示感谢!

References:

[1] Huai JP, Hu CM, Li JX, Sun HL, Wo TY. CROWN: A service grid middleware with trust management. Science in China Series F: Information Sciences 2006,49(6):731−758.

[2] Gligor VD, Khurana H, Koleva RK, Bharadwaj VG, Baras JS. On the negotiation of access control policies. In: Christianson B, Crispo B, Malcolm JA, Roe M, eds. Proc. of the 9th Int'l Workshop on Security Protocols. LNCS 2467. Berlin: Springer-Verlag, 2001. 188−201.

[3] McDaniel P. Prakash A. Methods and limitations of security policy reconciliation. ACM Trans. on Information and System Security, 2006,9(3):259−291.

[4] Wang H, Jha S, Livny M, McDaniel PD. Security policy reconciliation in distributed computing environments. In: Chadha R, ed. Proc. of the 5th IEEE Int'l Workshop on Policies for Distributed Systems and Networks (POLICY 2004). Washington: IEEE Computer Society, 2004. 137−146.

[5] Agarwal S, Sprick B. Access control for semantic Web services. In: Zhang LJ, ed. Proc. of the IEEE Int'l Conf. on Web Services (ICWS 2004). Washington: IEEE Computer Society, 2004. 770−773.

[6] Burris S, Sankappanavar HP. A Course in Universal Algebra. New York: Springer-Verlag, 1981.

[7] Bonatti P, di Milano U, de Capitani di Vimercati S, Samarati P. An algebra for composing access control policies. ACM Trans. on Information and System Security, 2002,5(1):1−35.

[8] Mclean J. The algebra of security. In: Proc. of the 1988 IEEE Symp. on Security and Privacy. Washington: IEEE Computer Society, 1988. 2−7.

[9] Wijesekera D, Jajodia S. A propositional policy algebras for access control. ACM Trans. on Information and System Security, 2003,6(2):286−325.

[10] Backes M, Durmuth M, Steinwandt R. An algebra for composing enterprise privacy policies. In: Samarati P, Gollmann D, Molva R, eds. Proc. of the 9th European Symp. on Research in Computer Security (ESORICS 2004). LNCS 3193, Berlin: Springer-Verlag,

2004. 33−52.

[11] Moses T. QASIS extensible access control markup language (XACML) version 2.0 [EB/OL]. OASIS Specification, OASIS, 2005. http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf

[12] Maler E, Mishra P, Philpott R. QASIS Security assertion markup language (SAML) version 2.0 [EB/OL]. 2005. http://www.oasis-open.org/committees/download.php/3406/oasis-sstc-saml-core-1.1.pdf

[13] Zhang XW, Li YL, Nalla D. An attribute-based access matrix model. In: Haddad H, Liebrock LM, Omicini A, Wainwright RL, eds. Proc. of the 2005 ACM Symp. on Applied Computing. New York: ACM Press, 2005. 359−363.

[14] Wang LY, Wijesekera D, Jajodia S. A logic-based framework for attribute based access control. In: Atluri V, Backes M, Basin DA, Waidner M, eds. Proc. of the 2004 ACM Workshop Formal Methods in Security Engineering (FMSE 2004). New York, ACM Press, 2004. 45−55.

[15] Bertino E, Squicciarini AC, Mevi D. A fine-grained access control model for Web service. In: Zhang LJ, ed. Proc. of the 2004 IEEE Int'l Conf. on Services Computing (SCC 2004). 2004. 33−40.

[16] Ferraiolo DF, Gavrila S, Hu V, Kuhn DR. Composing and combining policies under the policies under the policy machine. In: Ferrari E, Ahn GJ, eds. Proc. of the 10th ACM Symp. on Access Control Models and Technologies (SACMAT 2005). New York, ACM Press, 2005. 11−20.

[17] Hu VC, Ferraiolo DF, Scarfone K. Access control policy combinations for the grid using the policy machine. In: Schulze B, Buyya R, Navaux P, Cirne W, Rebello V, eds. Proc. of the 7th IEEE Int'l Symp. on Cluster Computing and the Grid (CCGRID 2007). Washington: IEEE Computer Society, 2007. 225−232.

[18] Zeng LZ, Benatallah B, Ngu AHH, Dumas M, Kalagnanam J, Chang H. QoS-Aware middleware for Web services composition. IEEE Trans. on Software Engineering, 2004,30(5):311−327.

[19] Su LY, Chadwick DW, Basden A, Cunningham JA. Automated decomposition of access control policies. In: Winsborough W, Sahai A, eds. Proc. of the 6th IEEE Int'l Workshop on Policies for Distributed Systems and Networks (POLICY 2005). Washington: IEEE Computer Society, 2005. 3−13.

[20] Lloyd JW. Foundations of Logic Programming. New York: Springer-Verlag, 1984.

[21] Li NH, Mitchell JC, Winsborough WH. Design of a role-based trust-management framework. In: Heather H, ed. Proc. of the 2002 IEEE Symp. on Security and Privacy. Washington: IEEE Computer Society, 2002. 114−130.

[22] Lin L, Zhang Y, Huai JP. Sustaining incentive in grid resource allocation: A reinforcement learning approach. In: Schulze B, Buyya R, Navaux P, Cirne W, Rebello V, eds. Proc. of the 7th IEEE Int'l Symp. on Cluster Computing and the Grid (CCGRID 2007). IEEE Computer Society, 2007. 145−154.

[23] Gong L, Qian XL. The complexity and composability of secure interoperation. In: Rushby J, Meadows C, eds. Proc. of the IEEE Symp. on Security and Privacy. Washington: IEEE Computer Society, 1994. 190−200.

[24] Gong L, Qian XL. Computational issues in secure interoperation. IEEE Trans. on Software and Engineering, 1996,22(1):43−52.

林莉(1979－),女,广西南宁人,博士生,主要研究领域为信息安全,计算机科学理论.

李先贤(1969－),男,博士,教授,主要研究领域为信息安全,计算机科学理论.

怀进鹏(1962－),男,博士,教授,博士生导师,CCF 高级会员,主要研究领域为计算机理论与软件,网络安全.