一种基于多本体体系的语义Web服务访问控制方法

胡罗凯[1,2] 陈 旭[3] 柴 新[1] 应 时[3]
(湖北第二师范学院计算机学院 武汉 430205)[1] (华中科技大学计算机科学与技术学院 武汉 430074)[2]
(武汉大学软件工程国家重点实验室 武汉 430072)[3]

摘 要 提出一种基于多本体体系的语义Web服务访问控制方法。首先,基于分布式描述逻辑DDL,刻画了一种基于桥接本体的跨域多本体体系,它为语义Web服务的访问控制提供了知识库;其次,在基于语义的访问控制方法基础上,给出了适用于语义Web服务的访问控制模型;最后,设计了基于多本体体系的语义Web服务访问控制方法及其体系结构,并给出了该方法的案例应用。在语义Web服务的访问控制方法中,基于桥接本体的跨域多本体体系既为各安全域的语义模型提供了语义关联,又保证了各安全域中语义表示的隐私性。

关键词 语义Web服务,多本体体系,访问控制
中图法分类号 TP309 **文献标识码** A

Multi-ontology System Based Approach of Access Control for Semantic Web Services

HU Luo-kai[1,2] CHEN Xu[3] CHAI Xin[1] YING Shi[3]
(School of Computer, Hubei University of Education, Wuhan 430205, China)[1]
(School of Computer Science & Technology, Huazhong University of Science and Technology, Wuhan 430074, China)[2]
(State Key Lab of Software Engineering, Wuhan University, Wuhan 430072, China)[3]

Abstract A multi-ontology system based access control approach for semantic Web services was proposed. First, a bridge ontology based cross-domain multi-ontology system (CDMOS), which provides a semantic model for access control of Semantic Web Service, was presented based on the distributed description logic (DDL). Secondly, on the basis of semantic access control technology, the access control model for semantic Web service was given. Finally, this paper gave the architecture of multi-ontology system based access control approach for semantic Web service and the case study of this approach. In the access control for semantic Web service, CDMOS not only provides the semantic mapping for the semantic model of security domains, but also ensures the semantic independence among the security domains.

Keywords Semantic Web service, Multi-ontology system, Access control

近年来,语义Web服务的研究工作主要围绕语义Web服务组合、发现和执行等方面展开,随着语义Web服务研究的逐渐深入,其安全问题日益成为限制其大规模应用的关键性问题之一,引起了学术界和产业界的广泛关注。在语义Web服务的安全性问题中,访问控制是一个基础和核心的问题。在Web服务环境下,访问控制主要面临两个方面的挑战,一是Web服务环境的跨域特征导致访问控制需要在一个开放的、分布式的环境下进行,而传统的访问控制方法没有解决跨域访问控制中的异构问题;二是Web服务的动态性特征导致访问控制需要在环境动态变化的条件下进行,而传统访问控制的授权所需要的策略及决策信息在访问之前就已经确定,很少涉及到由于对客体的访问而带来的决策信息的变化。在Web服务封装成语义Web服务后,一方面,通过在语义Web服务间共享丰富的语义信息,在语义层实现服务搜索和组合,在一定程度上消除了由于语法异构所带来的问题,但是在对服务进行访问控制时,语义信息的共享与隐私保护成为一对矛盾;另一方面,来自不同安全域的具有相同语义而语法异构的语义Web服务在对决策信息的改变效果上是相同的,且对访问控制决策的影响也是等价的。针对这些问题,需要一种新的适合于语义Web服务的访问控制方法。

对于Web服务的安全问题,目前的加密、数字签名、证书、公钥基础设施等技术为基于Web服务的交互在较低抽象层次上提供了一个很好的安全基础,但是,在更加复杂、动态和灵活的语义Web服务环境下,需要提供更高抽象层次的授权机制,然而处于不同安全域的语义Web服务之间所使用的知识库的异构问题为其访问控制带来了挑战。

1 相关工作

在传统的访问控制方法中,基于格的访问控制(LBAC)[1]方法属于强制访问控制(MAC),是一种粗粒度的访问控制方

到稿日期:2012-02-13 返修日期:2012-07-21 本文受国家自然科学基金项目(61070012),湖北省教育厅重点科研项目(D20103004),湖北省科学技术研究计划优秀中青年人才项目(Q20113001),武汉大学青年教师资助项目(3101004),湖北第二师范学院优秀教师团队建设项目资助。
胡罗凯(1981-),男,博士后,讲师,主要研究领域为语义Web和SOA安全性,E-mail:luokaihu@gmail.com;陈 旭(1982-),男,博士,讲师,主要研究领域为语义Web和地理信息系统,E-mail:cxsklse@gmail.com(通信作者)。

法，不适合作为语义 Web 服务的访问控制方法。为适应自主访问控制(DAC)的需要，基于身份的访问控制(IBAC)和基于角色的访问控制(RBAC)[2]方法目前被广泛应用于各种互联网和数据库系统。然而，在 IBAC 模型中，几乎每一个主体都需要一个身份标识符和若干个访问控制权限，很难维护访问控制列表中越来越多的身份标识符。在 RBAC 模型中，随着系统规模和复杂性的提高，大型的组织仍然需要建立和维护大量的角色，即仍然需要在角色和权限间、角色和用户间分配相当多的工作。近几年，基于属性的访问控制(ABAC)[3]方法和 XACML[4]语言被越来越多地使用在 Web 服务的访问控制中，ABAC 方法并不直接对主体和客体赋予授权关系，而是通过策略对相关属性的描述来进行访问控制决策，如果把角色、格和身份都作为属性来看待的话，ABAC 完全兼容 LBAC、IBAC 和 RBAC 方法。但是，在语义 Web 服务环境下，服务请求者可能来自另一个安全域，ABAC 策略的制定者事先可能并不知道资源请求者的属性定义方式，针对这一问题，出现了基于语义的访问控制(SBAC)方法。

Yagüe 等人最早将语义 Web 技术引入到访问控制策略的决策和实施中[5]，KAoS 最初使用 DAML，后来使用 OWL 作为 Web 服务、网格计算和多 Agent 系统中策略表达和推理的基础[6]。L. Kagal 提出了一种基于 RDFS 和逻辑编程语言的策略定义语言及其策略引擎 Rei[7]，Rei 依赖于应用无关的本体来表示权限、禁止、义务、免除义务和策略规则等概念。OREL 为 XObjects 研究小组提出的一种基于本体的数字权限表达语言，主要用于数字权限管理和推理[8]，目前使用 OWL DL 作为其本体描述语言。T. Priebe 基于本体推理，通过对属性管理机制的扩展，提出了一种基于属性的访问控制策略描述和维护方法[9]。E. Demiani 通过对 XACML 的扩展，提出了一种策略实施的体系结构[10]。他们的研究工作都为基于语义的访问控制和语义 Web 服务的访问控制作出了贡献，然而，他们都没有涉及分布式语义表达的问题，即在分布式环境下，本体可能在不同的语义节点上。如何既能有效地共享这些语义信息，又能较好地保护这些信息的隐私是目前语义 Web 服务的访问控制面临的一个问题，本文提出的桥接本体是解决这个问题的方法之一。

此外，在本体映射的研究工作上，国内外也取得了一定的研究成果[11-13]，主要是从本体融合、本体集成和本体结盟 3 个方面解决本体的异构问题[14]，本体融合方法是将有关同一领域的本体合并成一个本体，本体集成方法是使用已存在的本体组成范围更广、体系更好的本体，本体结盟方法是在本体之间建立联系，它们各自仍然相对独立，本文所述本体体系属于这一类，但与直接建立本体之间的映射关系不同，本文提出使用桥接本体来间接建立本体之间的映射关系。

2 跨域多本体体系

语义 Web 服务分布在不同的安全域中，各安全域可能使用相同或完全不同的本体对原有的 Web 服务进行语义描述或封装，同时，为了使分布式环境下的访问控制策略能够得到正确的理解和解释，并对访问控制策略的分析和推理提供支撑机制，需要使用含有丰富语义信息的方式来描述跨域访问控制中的各类信息(如各安全域中主体、客体、动作和环境的属性信息等)。在跨越不同的安全域进行知识重用和共享时，为了消除参与跨域交互的多方在概念模型上的差异以及由此引起的歧义性，需要为其建立可共同理解的知识库，为跨域进行互操作提供支持。本文采用本体结盟的方法来解决各安全域间本体异构的问题，与现有本体结盟方法不同，本文提出了一种基于桥接本体的跨域多本体体系(Cross Domain Multi Ontology System，CDMOS)，该本体体系根据本体作用的不同，可以分为两类，即安全域本体和桥接本体。

安全域本体(Security Domain Ontology)。每个安全域中用于标注本安全域内的 Web 服务及访问控制策略的本体，是各安全域的知识库，简称域本体。

桥接本体(Bridge Ontology)。在各安全域间，用于映射域本体的概念或关系的本体，即各域本体的“翻译”本体。通过建立桥接本体，并不直接建立两个域本体之间的语义关系，而是通过分别建立域本体与桥接本体之间的语义关系，来间接地实现域本体之间的语义关联。

有多个安全域和多个桥接本体参与的 CDMOS 体系结构如图 1 所示，安全域本体 SDO1、SDO2、SDO3 之间的语义桥接使用了桥接本体 BO1，而安全域本体 SDO3 和 SDO4 之间的语义桥接使用了桥接本体 BO2。在本体桥接时，一般使用两种方法进行语义关联。对于简单语法异构的情况，只需要使用本体概念之间的语义扩展和语义紧缩关系即可；而对于复杂的语法异构情况，则需要使用 SWRL DL-Safe 规则进行语义关联。

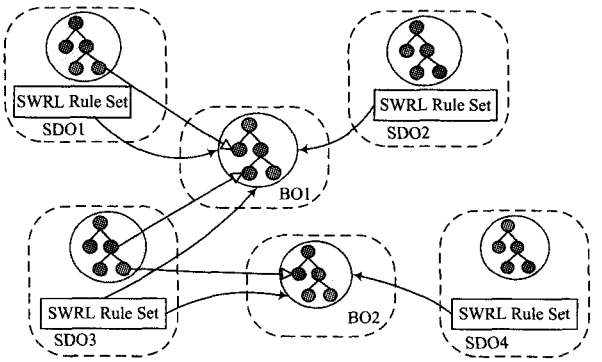

图 1 CDMOS 的结构

下面介绍建立 CDMOS 的一般方法。

(1) 虚线框中的安全域本体相对于规则集(如 SWRL 规则集)来说是独立的，可以在完全不依赖规则集的情况下，用现有的本体构建方法和本体构建工具(如 Protege)先构建安全域本体，即在构建安全域本体的时候不用关心如何实现与其它本体之间的语义关联关系。

(2) 确定要建立语义关联关系的域本体，也就是要确定哪些域之间需要进行跨域访问控制，则这些域的安全域本体之间需要建立语义关联关系，也即需要在这些域本体之间建立桥接本体，本节后半部分介绍建立桥接本体的具体方法。需要注意的是，并不是每两个域之间都需要建立一个桥接本体，如图 1 所示，安全域 1，2，3 之间需要进行跨域访问控制，所以在对应的 3 个域本体之间需要建立一个桥接本体，域 3 和域 4 之间需要进行跨域访问控制，故需要在对应的两个域本体之间需要建立一个桥接本体。此外，如果两个域使用同一个域本体，则不需要建立桥接本体，直接进行跨域访问控制。

(3) 本体之间的对应关系存在于描述的规则或概念的关

系之间，在每个安全域本体和桥接本体都构建完成之后，本体的维护人员可以撰写规则或概念的对应关系，在语义上实现本体之间的对应连接。

使用分布式描述逻辑 DDL[15]，CDMOS 可以形式定义为一个二元组〈T,A〉。

CDMOS 的 T-Box 可定义为：$T=\langle\{\mathbf{T}_{di}\}_{i\in I},\{\mathbf{T}_{bj}\}_{j\in I},B\rangle$ 其中，$\{\mathbf{T}_{di}\}_{i\in I}$表示使用普通描述逻辑描述的域 T-Box 集合，$\mathbf{T}_{bj}=\sqcup\{\mathbf{T}_{di[k]}\}_{i[k]\in I}$表示使用普通描述逻辑描述的桥接 T-Box，用于桥接若干域 T-Box$\{\mathbf{T}_{di[k]}\}_{i[k]\in I}$，$B=\{\mathbf{B}_{ij}\}_{i,j\in I}$表示从 di 到 bj 的概念桥接关系和规则桥接关系。如两个域的 T-Box 为 $\mathbf{T}_{d1}$ 和 $\mathbf{T}_{d2}$，它们的桥接本体为 $\mathbf{T}_{b1}=\sqcup\{\mathbf{T}_{d1},\mathbf{T}_{d2}\}$。

CDMOS 的 A-Box 可以定义为：$A=\langle\{\mathbf{A}_{di}\}_{i\in I},\{\mathbf{A}_{bj}\}_{j\in I},A\rangle$

其中，$\{\mathbf{A}_{di}\}_{i\in I}$表示使用普通描述逻辑描述的域 A-Box 集合，$\mathbf{A}_{bj}=\sqcup\{\mathbf{A}_{di[k]}\}_{i[k]\in I}$表示使用普通描述逻辑描述的桥接 A-Box，用于桥接若干域 A-Box$\{\mathbf{A}_{di[k]}\}_{i[k]\in I}$，$A=\{\mathbf{A}_{ij}\}_{i,j\in I}$表示从 di 到 bj 的实例桥接关系（包括部分实例关系和完全实例关系）。

域本体与桥接本体之间的语义关联，可以分为两类，即简单异构和复杂异构的语义关联。

(1)简单异构的语义关联

首先给出概念桥接关系和实例桥接关系的 DDL 定义。概念桥接关系可以定义为语义插入(into)关系和语义包含(onto)关系，给定 A 和 B 分别是域本体 di 和桥接本体 bj 的概念，从域本体到桥接本体的两种概念桥接关系可以表示为：

(into 关系) $di:A\xrightarrow{\sqsubseteq}bj:B$

(onto 关系) $di:A\xrightarrow{\sqsupseteq}bj:B$

而语义等价关系 $di:A\xrightarrow{\equiv}bj:B$ 可以理解为 $di:A\xrightarrow{\sqsubseteq}bj:B$ 且 $di:A\xrightarrow{\sqsupseteq}bj:B$。

实例桥接关系可以定义为语义部分(partial)关系和语义完全(complete)关系，给定 x 和 y 是域本体 di 和桥接本体 bj 的个体，从域本体到桥接本体的两种实例桥接关系可以表示为：

(partial 关系) $di:x\overset{=}{\mapsto}bj:y$

(complete 关系) $di:\{x\}\overset{=}{\mapsto}bj:\{y1,y2,\cdots\}$

简单异构是指概念或实例的名称上的异构关系，如在安全域 d1 中存在概念 FamilyName，而在安全域 d2 中对应的概念为 LastName，两者表达了同一含义的概念，却使用了不同时的名字，这显然给跨域进行访问控制带来了不便，可以定义两个域的桥接本体 b1，b1 中定义概念 FamilyName，则有

$d1:\text{FamilyName}\xrightarrow{\equiv}b1:\text{FamilyName}$，

$d2:\text{LastName}\xrightarrow{\equiv}b1:\text{FamilyName}$。

(2)复杂异构的语义关联

复杂异构是指概念或实例在结构上的异构关系，如图 2 所示，在安全域 d1 中存在概念 Fullage 表示成年人的年龄，在安全域 d2 中没有相对应的概念，d2 使用概念 Age 大于等于 18 岁来表示成年人年龄，可以使用以下两条 SWRL 规则来表示 Fullage 和 Age 的这种结构上的异构关系。d1:Person(?x) ∧ b1:Age(?x,?a) ∧ greaterThanOrEqual(?a,18)→d1:FullAge(?x,true) d1:Person(?x) ∧ d1:FullAge(?x,true) ∧ b1:Age(?x,?a)→greaterThanOrEqual(?a,18)。

但是由于 OWL DL 采用 SWRL 规则扩展后是不可判定的，本文使用 DL-Safe 规则，DL-Safe 规则是 SWRL 规则的一个受限子集。DL-Safe 限制了 SWRL 的表达能力，可能得到不完备的结果，也就是说，不一定能够得到所有正确的结果。但是另一点对于安全领域很重要，即得到的所有结果都是正确的。已有的支持 SWRL 的 OWL 推理机就只实现了 DL-Safe 的 SWRL 规则，如 Pellet。

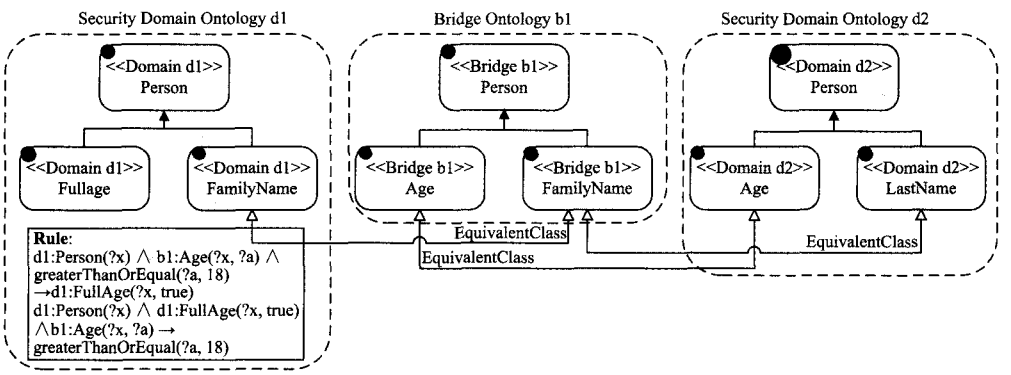

图 2　桥接本体示例

3　用于语义 Web 服务的访问控制模型

在基于语义的访问控制(SBAC)方法的基础上，结合语义 Web 服务环境的动态、异构、分布式等特点，给出一种用于语义 Web 服务的访问控制方法(Access Control for Semantic Web Service，AC4SWS)及其形式化模型。在 AC4SWS 模型中，需要描述在语义 Web 环境下，访问控制主体和客体(语义 Web 服务)的属性、访问控制请求和访问控制策略的定义方式，并给出语义 Web 服务效果和 Web 服务参数两个概念。本节所述的本体标注均使用安全域本体 SDO 进行标注。

定义 1(主体属性)　主体属性是一个四元组〈AS，AS_Domain，AS_Ontoref，AS_Value_Ontoref〉，其中，AS 是主体属性名，AS_Domain 是 AS 的定义域，AS_Ontoref 是 AS 的本体标注，当 AS_Domain 定义为非数值属性时，AS_Value_Ontoref 是 AS 属性值的本体标注，当 AS_Domain 定义为数值属性时，AS_Value_Ontoref 为 NULL。

定义 2(语义 Web 服务属性)　语义 Web 服务属性是一个四元组〈AW，AW_Domain，AW_Ontoref，AW_Value_Ontoref〉，其中，AW 是语义 Web 服务的属性名，AW_Domain 是 AW 的定义域，AW_Ontoref 是 AW 的本体标注，当 AW_Domain 定义为非数值属性时，AW_Value_Ontoref 是 AW 属性值的本体标注，当 AW_Domain 定义为数值属性时，AW_Va-

lue_Ontoref 为 NULL。语义 Web 服务的常见属性包括输入、输出、发布者、所属组织、分类和各种上下文约束等。

定义 3(Web 服务参数) Web 服务参数被定义为一个二元组〈P,P_Domain〉,其中,P 是参数名,P_Domain 是 P 的定义域。

上述 3 个定义中的 AS_Domain、AW_Domain 和 P_Domain 均可采用两种表示方式,即枚举法(表示出若干合法的属性值$\{v_1, v_2, \cdots, v_n\}$)和区间法(表示出属性值的范围$[v_{begin}, v_{end}]$),枚举法既适合数值属性值,也适合非数值属性值,而区间法一般适合于数值属性值的刻画。

定义 4(属性条件) 属性条件表示为一个逻辑表达式 LEP,LEP 递归定义为由若干个逻辑表达式通过逻辑运算符 lop 连接组成,形如 LEP_i lop LEP_j,lop={AND,OR},AND,OR 分别表示逻辑与和逻辑或。在逻辑表达式前加非运算符 NOT 仍然是逻辑表达式。比较表达式 CEP 是一种逻辑表达式,CEP 表示为形如 A cop A_Value 的表达式,其中,A 是属性名的语义标注,cop 是比较运算符,cop={GT,EQ,LT,HAS},A_Value 是属性 A 定义域规定的数值或非数值属性值常量或常量集的语义标注。当 A_Value 是数值属性时,GT,EQ 和 LT 分别表示为数值大于、等于和小于。当 A_Value 是非数值属性时,GT,EQ 和 LT 分别表示为"语义"大于、等于和小于。HAS 表示 A 具有 A_Value 属性值或集合。

定义 5(语义 Web 服务效果) 语义 Web 服务效果被定义为一个三元组〈SWSI,Con,Post〉,其中,SWSI 是 Web 服务标识。Con 是属性条件,见定义 4。Post 是 ABox 断言的集合,Post 表示调用语义 Web 服务 SWSI 后,根据情况 Con 的不同知识库改变的情况。根据服务效果的不同,语义 Web 服务可分为两类:(1)仅提供信息或计算的服务,调用此类服务后,知识库不发生改变,如调用查询考试成绩服务后,考试成绩并未改变。(2)改变"世界"状态的服务,调用此类服务后,知识库可能发生改变,如某学生 wang 调用参加 C 语言考试服务后,如果成绩及格,则本体中,该学生就已经修过了 C 语言这门课程,如果成绩不及格,该学生未修过 C 语言这门课程,表示为:

Con={(Score GT 60) OR (Score EQ 60)},

Post={hasStudy(wang,c programming)};

Con={Score LT 60},Post={NULL}。

定义 6(访问请求) 访问请求是一个二元组 Acq=〈SI,S〉,其中,SI 是 Web 服务标识,S 是主体属性集,S={(AS_1,a_1,AS_1_Ontoref,a_1_Ontoref),…,(AS_n,a_n,AS_n_Ontoref,a_n_Ontoref)},AS_i 是主体属性名,a_i 是和属性定义域相兼容的取值,AS_i_Ontoref 是属性名的本体标注,当 a_i 是数值属性时,a_i_Ontoref 是属性值的本体标注,否则 a_i_Ontoref 为空。

定义 7(访问控制策略) 访问控制策略可以定义为一个四元组 ACP=〈SWSI,W,C,P,ACR〉,其中,SWSI 是语义 Web 服务标识,同定义 5 中 SWSI 的定义。W 是语义 Web 服务属性集,W={(AW_1,a_1,AW_1_Ontoref,a_1_Ontoref),…,(AW_n,a_n,AW_n_Ontoref,a_n_Ontoref)},AW_i 是语义 Web 服务属性名,a_i 是和属性定义域相兼容的取值,AW_i_Ontoref 是属性名的本体标注,当 a_i 是数值属性时,a_i_Ontoref 是属性值的本体标注,否则 a_i_Ontoref 为空。Con 是属性条件,见定义 4。Par 是 Web 服务的参数集,Par={(SI_1,P_1,P_Va-lue$_1$),…,(SI_n,P_n,P_Value$_n$)},其中,SI_i 是 Web 服务标识,P_i 是 Web 服务参数名,P_Value$_i$ 是 Web 服务参数定义域的子集,表示可取的参数值的集合。ACR 是访问控制结果,ACR={DENY,PERMIT},DENY 表示同意访问,PERMIT 表示拒绝访问。

AC4SWS 方法是一种基于语义的访问控制方法,结合了基于语义属性的访问控制和基于策略的访问控制方法的特点,AC4SWS 方法适合于基于桥接本体或集中本体的语义 Web 服务访问控制。当访问控制策略中 ACP_1. W 满足访问请求 Acq_1. SI 所标识的 Web 服务的语义属性时,该访问控制策略适用于该请求,当没有任何控制策略适用于该请求时,使用本域的安全设置,即默认拒绝或同意该请求。当 Acq_1. S 满足 ACP_1. Con 中对属性条件的约束时,应用访问控制决策为 ACP_1. ACR,即拒绝或同意该访问请求。

4 基于 CDMOS 的语义 Web 服务访问控制体系结构

现有的策略描述语言(如 XACML)只能表达语法层的访问控制策略,为了适应语义访问控制的需要,提出了 SACPL (Semantic Access Control Policy Language)[16,17] 作为策略描述语言,SACPL 语言是在已有 XACML 语言的基础上添加语义标注和相应的辅助机制设计而成的语义策略描述语言。本文使用 SACPL 作为语义 Web 服务访问控制的策略描述语言,能够在开源的 XACML 实现项目(如 Sun's XACML)的基础上进行扩充,快速开发 SACPL 的实现项目,SACPL 不是本文所述的主要内容,更多细节见文献[16,17]。因为 SACPL 是 XACML 的扩展语言,在对其体系结构进行描述时,借用了 XACML 中的一些术语,图 3 以两个安全域互操作时进行访问控制为例,详述了基于 CDMOS 的语义 Web 服务访问控制体系结构中各模块的交互过程。

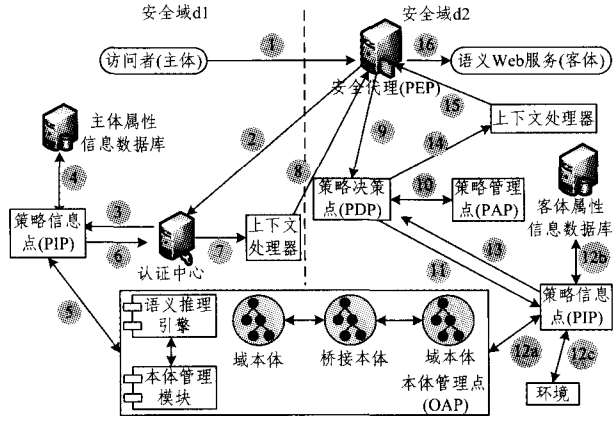

图 3 基于 CDMOS 的语义 Web 服务访问控制体系结构

(1)安全域 d1 中的主体 S_{d1} 希望能访问安全域 d2 中的某 Web 服务(客体)WS_{d2},S_{d1} 带着安全域 d1 的密钥 K_{sd1} 向 WS_{d2} 发出访问 SOAP 消息;

(2)该 SOAP 消息被 Web 服务 WS_{d2} 的安全代理 PEP_{d2} 截获,PEP_{d2} 向安全域 d1 的认证中心 AC_{d1} 发出询问,询问主体 S_{d1} 的相关属性;

(3)认证中心 AC_{d1} 首先确认密钥 K_{sd1} 的有效性,如果有

效，AC_{d1} 向策略信息点 PIP_{d1} 发出属性查询的请求，否则拒绝访问；

(4) 策略信息点 PIP_{d1} 从属性数据库 DB_{d1} 中或相关属性查询 Web 服务处查询出主体 S_{d1} 的所有已知属性 A_{sd1}；

(5)策略信息点 PIP_{d1} 将 A_{sd1} 作为输入进行语义推理，推理出主体 S_{d1} 的其它相关属性信息 A'_{sd1}，所有属性均使用桥接本体作为词汇表；

(6)策略信息点 PIP_{d1} 将查询结果($A_{sd1}+A'_{sd1}$)返回认证中心 AC_{d1}；

(7)认证中心 AC_{d1} 将查询结果发送给上下文处理器 CH_{d1}；

(8)上下文处理器 CH_{d1} 将查询结果封装成 SACPL Request 格式后发送给安全代理 PEP_{d2}；

(9)安全代理 PEP_{b1} 将该 SACPL Request 发送给安全域 b2 的策略决策点 PDP_{b2}；

(10)策略决策点 PDP_{b2} 解析该 SACPL Request，将 Target 作为索引向策略管理点 PAP_{b1} 查询 Web 服务 WS_{d2} 所对应的语义 Web 服务 SWS_{d2} 所涉及的策略；

(11)策略决策点 PDP_{b2} 向策略信息点 PIP_{b2} 发出属性查询请求；

(12a)策略信息点 PIP_{b2} 通过语义推理引擎，将($A_{sd1}+A'_{sd1}$)转换为安全域 S_{d2} 可理解的属性 A_{sd2}，即使用 S_{d2} 的域本体作为词汇表；

(12b)策略信息点 PIP_{b2} 从属性数据库中查询出 S_{d2} 需要访问的语义 Web 服务(客体)SWS_{d2} 的所有属性 A_{Od2}；

(12c)策略信息点 PIP_{b2} 获得所有环境属性 A_{ed2}；

(13)策略信息点 PIP_{b2} 将所有得到的属性信息($A_{sd2}+A_{Od2}+A_{ed2}$)返回给策略决策点 PDP_{b2}；

(14)策略决策点 PDP_{b2} 生成访问决策并将该决策发送给上下文处理器 CH_{b2}；

(15)上下文处理器 CH_{b2} 将访问控制决策封装成 SACPL Response 格式并发送给安全代理 PEP_{b2}；

(16)安全代理 PEP_{b2} 根据 SACPL Response 的 Result 进行相应的动作，即允许或拒绝这次访问。

5 案例应用

随着 SOA 的成熟与应用，A 大学与 B 大学都已经将自己原有的网上教学系统成功服务化。随着校际间合作的不断扩大，为解决搜索和组合 Web 服务时遇到的语法异构问题，又拟将原有的 Web 服务语义化，计划分别使用 OWL-DL 语言来建立各自的域本体，使用 WSDL4S[18] 语言进行服务标注，采用基于桥接本体的访问控制方法来解决两个学校间语义 Web 服务的访问授权问题，使得两个学校现有的基于 Web 服务的考试系统可以充分共享，并互相承认学分。

A 大学的计算机学院开设有 C 语言程序设计(C programming language) 和 Unix 环境编程(programming in Unix)两门课程，但规定只有修完所有必修课才能选修 C 语言网络编程。

B 大学实行学分制管理，该校计算机学院开设了程序设计基础(basic programming)课程，实际讲授的是 C 语言的内容，通过考试的学生将会获得 3 个学分，该校计算机学院和软件学院都开设有 Unix 高级编程(advanced programming in Unix)课程，为防止学生偏科，学校规定只有修满 30 个学分，并且修过程序设计基础的学生才可以选修 Unix 高级编程。上述两校之 4 门课程的考试均已发布为 Web 服务，并使用各自开发的本体和基于 WSDL4S 语言成功实现了语义化。且一门课程如果考试通过后，不得重复考试。

给定命名空间：

d1="http://www.university-a.edu/Onto"；

d2="http://www.university-b.edu/Onto"；xsd="http://www.w3.org/2001/XMLSchema"。

A 大学的 T-Box $\mathcal{T}_{d1}$ 及其部分解释 $\mathcal{I}_{d1}$：

$Course^{\mathcal{I}d1}=\{c\_prog\_lang, prog\_unix\}$

$Student^{\mathcal{I}d1}=\{wangMing\}$

$Required\_Course \sqsubseteq Course$

$Student \sqsubseteq \forall hasStudy.Course \sqcap \forall hasNotStudy.Course$

B 大学的 T-Box $\mathcal{T}_{d2}$ 及其部分解释 $\mathcal{I}_{d2}$：

$Course^{\mathcal{I}d2}=\{basic\_prog, adv\_prog\_unix\}$

$Student \sqsubseteq \forall hasStudy.Course \sqsubset \forall hasNotStudy.Courseu \forall hasCredit.Credit$

$Credit \sqsubseteq \exists creditFrom.Course \sqcap \forall creditFrom.Course \sqcap \forall hasValue.xsd:int$

桥接本体的 T-Box $\mathcal{T}_{b12}$ 及其部分解释 $\mathcal{I}_{b12}$：

$Course^{\mathcal{I}b12}=\{c\_programming, c\_advanced\_programming\}$

$Senior\_Student \sqsubseteq Student$

$Student \sqsubseteq \forall hasStudy.Course \sqcap \forall hasNotStudy.Course$

在 A 大学的 $d1$ 本体和桥接本体之间建立的语义关系使用 DDL 表示为：

$d1:Student \sqcap \rightharpoonup \exists d1:hasNotStudy.d1:Required\_Course \xrightarrow{\equiv} b12:Senior\_Student$

$d1:\{c\_prog\_lang\} \overset{=}{\mapsto} b12:\{c\_programming\}$

$d1:prog\_unix \mapsto b12:\{c\_programming\_unix\}$

在 B 大学的 $d2$ 本体和桥接本体之间建立的语义关系使用 DDL 表示为：

$d2:Student(?a) \wedge d2:Credit(?x) \wedge d2:Course(?c) \wedge d2:creditFrom(?x,?c) \wedge d2:hasCredit(?a,?x) \wedge d2:hasValue(?x,?y) \wedge greatThan(?y,30)) \rightarrow b12:Senior\_Student(?a)$

$b12:Senior\_Student \xrightarrow{\equiv} d2:Student \sqcap \exists d2:hasCredit.(\exists d2:hasValue.\{\geqslant 30\})$

$d2:\{basic\_prog\} \overset{=}{\mapsto} b12:\{c\_programming\}$

$d2:adv\_prog\_unix \mapsto b12:\{c\_programming\_unix\}$

B 大学程序设计基础课程考试服务的效果表示为：

〈Basic_Prog,{(SCORE GT 60) OR (SCORE EQ 60)},{d2:Student(?a),d2:hasStudy(?a,basic_prog)}〉

〈Basic_Prog,{(SCORE GT 60) OR (SCORE EQ 60)},{d2:Student(?a),d2:Credit(?c),xsd:int(?i),d2:hasValue(?c,?i+3),d2:hasCredit(?a,?c)}〉

B 大学 Unix 高级编程课程考试服务的效果表示为：

〈Adv_Prog_Unix,{(SCORE GT 60) OR (SCORE EQ 60)},{d2:Student(?a),d2:hasStudy(?a,adv_prog_unix)}〉

B大学Unix高级编程课程考试服务的一个访问控制策略为：

ACP_1=〈Adv_Prog_Unix,{(output,SCORE,d2:Output,d2:adv_prog_unix_score),(input,PAPER,d2:input,d2:adv_prog_unix_paper)},(d2:StudiedCourse HAS d2:basic_prog) AND (NOT d2:StudiedCourse HAS d2:adv_prog_unix) AND ((d2:Credit GT 30) OR (d2:Credit EQ 30))},{(Adv_Prog_Unix_CS,Discount,{80}),(Adv_Prog_Unix_SS,Discount,{90})},PERMIT〉

表示在B大学中只有学分超过30并修过程序设计基础的学生才能访问Unix高级编程课程考试服务，并且访问计算机学院的Unix高级编程课程考试服务可以打8折，而访问软件学院的同样服务时只能打9折，其它学院如果还有该课程则不打折。

A大学的wang同学已经选修过了本校的C语言程序设计课程，并且已经修完了所有必修课，他希望可以通过B大学的Unix高级编程课程考试并获得相应的学分。wang同学向B大学计算机学院的Unix高级编程课程考试服务发出了如下的访问请求：

〈Adv_Prog_Unix_CS,{(name,wang,d1:StudentName,d1:wangMing),(BasicCredit,finished,d1:Required_Course,d1:finished),(hasStudy,c programming language,d1:hasSudy,d1:c_prog_lang))}〉

B大学的PEP截获该请求，向A大学继续请求其它相关属性，经OAP语义推理，得到以桥接本体作为语义标注的访问请求：

〈Adv_Prog_Unix_CS,{(name,wang,b12:StudentName,d1:wangMing),(b12:seniorStudent,true,b12:Senior_Student,null),(hasStudy,c programming,b12:hasSudy,b12:c_programming)…}〉

B大学的PDP对该请求进行决策，适用策略ACP_1，经桥接本体语义推理，得到以B大学的域本体作为语义标注的访问请求：

〈Adv_Prog_Unix_CS,{(name,wang,d2:StudentName,d1:wangMing),(Credit,{≥30},d1:Credit,null),(hasStudy,basic programming,d2:hasSudy,d2:basic_prog)…}〉

B大学的PDP作出的访问控制决策结果为PERMIT，调用该服务后，根据服务效果，如果通过考试，则该同学不能再次访问该服务。

结束语 随着网构软件的发展，分布于Internet的各个节点的安全信息通常难以收集，因此，如何在安全信息相对缺乏的前提下进行授权以保障软件系统的安全性成为重要的研究问题[19,20]。基于Web服务或语义Web服务的应用系统是网构软件的重要形式，适合于语义Web服务环境的访问控制方法成为影响其可信性的重要问题之一。本文提出了一种基于多本体体系的访问控制方法，其通过桥接本体来“串联”各安全域本体，既解决了各安全域本体中语义信息的私密性，又解决了跨安全域本体中语义信息的共享问题；另外，语义Web服务通过把Web服务中的语法属性标注为语义属性后，语义相同而语法异构的服务在主体或客体的属性改变的效果上应该是一样的。在语义Web服务访问控制中，本文定义了语义Web服务效果，解决了语义Web服务调用前后知识库的变化对访问决策带来的影响。本文提出的基于多本体体系的语义Web服务访问控制方法为在授权过程中获得各安全域的安全信息(包括主体语义信息和语义Web服务属性信息)提供了一种新的思路，是一种适合于语义Web服务环境的访问控制方法。基于桥接本体的跨域多本体体系中的语义不一致性传播问题可能是影响分布式本体可靠性的关键问题之一，是下一步研究的重点。

参考文献

[1] Sandhu R S. Lattice based Access Control Models [J]. IEEE Computer,1993,27(11):9-19

[2] Sandhu R S,Coyne E,Feinstein H. Role-based Access Control Models[J]. IEEE Computer,1996,29(2):38-47

[3] Yong E,Tong J. Attributed Based Access Control (ABAC) for Web Services[C]// Proc. of IEEE International Conference on Web Services (ICWS'05). Florida USA,2005:561-569

[4] OASIS. Extensible Access Control Markup Language (XACML) Version 2.0[EB/OL]. http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf,2005-03-09

[5] Yagüe M,Mana A,Lopez L,et al. Applying the Semantic Web Layers to Access Control[C]// Proc. of DEXA 03 Workshop on Web Semantics. Prague Czech Republic,2003:622-626

[6] Johnson M,Chang P,Jeffers R,et al. KAoS Semantic Policy and Domain Services: An Application of DAML to Web-Services-Based Grid Architectures[C]// Proc. of AAMAS 03 Workshop on Web Services and Agent-Based Engineering. Melbourne Australia,2003:835-842

[7] Kagal L,Berners-Lee T,Connolly D,et al. Using Semantic Web Technologies for Open Policy Management on the Web[C]// Proc. of 21st National Conference on Artificial Intelligence (AAAI 06). 2006:34-50

[8] Qu Yu-zhong,Zhang Xiang,Li Hui-ying. OREL: An Ontology-based Rights Expression Language[C]// Proc. of the Poster Track of 13th World Wide Web Conference. New York City,USA,2004:324-325

[9] Priebe T,Dobmeier W,Kamprath N. Supporting Attribute-Based Access Control with Ontologies[C]// Proc. of 1st International Conference on Availability,Reliability and Security. Vienna Austria,2006:465-472

[10] Damiani E,De Capitani di Vimercati S,Fugazza C,et al. Extending Policy Language to the Semantic Web[C]// Proc. of 4th International Conference on Web Engineering. Munich Germany,July 2004:330-343

[11] Madhavan J,Bemstein P A,Rahm E. Generic Schema Matching with Cupid[C]// Proc. of 27th International Conference on Very Large Databases(VLDB). Rome Italy,2001:49-58

[12] Do H,Rahm E. Coma: A System for Flexible Combination of Schema Matching Approaches[C]// Proc. of 28th International Conference on Very Large Databases(VLDB). Hongkong,China,2002:26-39

[13] Kalfoglou Y,Schoflemmer M. If-Map: An Ontology-Mapping Method Based On Information-Flow Theory[J]. Journal on Data

Semantics,2003,1(1):98-127

[14] Kalfoglou Y, Schodemmer M. Ontology Mapping: The State of the Art[J]. The Knowledge Engineering Review, 2003, 18(1): 1-31

[15] Borgida A, Serafini L. Distributed description logics: assimilating information from peer sources[J]. Journal of Data Semantics, 2003, 1(1): 153-184

[16] Hu Luo-kai, Ying Shi, Jia Xiang-yang. Towards an Approach of Semantic Access Control for Cloud Computing[C]// Proc. of 1st International Conference on Could Computing. Beijing, China: Springer, 2009: 145-156

[17] Hu Luo-kai, Ying Shi, Jia Xiang-yang. A Semantics Based Approach for Cross Domain Access Control[J]. Journal of Internet Technology, 2010, 11(1): 279-288

[18] Hu Luo-kai, Ying Shi, Chen Rui. A Semantic Web Service Description Language[C]// Proc. of 4th International Conference on Information Engineering. Taiyuan, China: IEEE Computer Society, 2009: 449-452

[19] 吕建,马晓星,陶先平,等. 网构软件的研究与进展[J]. 中国科学E辑:信息科学,2006,36(10):1037-1080

[20] 林闯,封富君,李俊山. 新型网络环境下的访问控制技术[J]. 软件学报,2007,18(4):955-966

(上接第 97 页)

运算。在存储占用量方面,用户端需要为每个文件存储一个预计算的校验值,服务器端只需存储数据文件即可。在网络通信量方面,挑战值和应答值都是固定常数值比特。

在配置为 Intel Pentium(R) Dual-Core CPU 2.7GHz,2G RAM 的机器上,使用 C++语言和 GMP 大数运算库编写了(l=128,n=51200)参数的算法模拟程序。在程序运行中,服务端计算应答值花费的时间为 443165us,用户端验证应答值的时间为 23us。

下面选取云存储完整性检测方案中比较典型的 3 种方案,即 PDP 方法原创者 Ateniese 等的方案[4]、支持完全动态更新的 Erway 等的方案[9]和 Wang Qian 等的方案[10],分别从计算量、网络通信数据量和存储占用量来对比分析算法的性能,具体参数如表 1 所列。从表 1 可以得出,我们的方案在计算量、网络通信量和存储量方面都是常数量级别,具有比较好的实用性。

表 1 部分性能参数对比(n 为数据划分的块数,t 为数据检测时的抽样块数)

方案	计算量		网络通信量	存储占用量
	用户	服务器		
我们算法	O(1)	n	O(1)	O(1)
文献[4]	O(n)	O(n)	O(1)	O(1)
文献[9]	O(tlogn)	O(tlogn)	O(logn)	O(1)
文献[10]	O(tlogn)	O(tlogn)	O(logn)	O(1)

结束语 本文针对云存储数据完整性验证问题,在哈希树结构的基础上,结合大数模运算,提出一种新的树形完整性检测结构——IC-树(Integrity Checking tree)。基于 IC-树和 Seny 等提出的云存储框架[13],设计了一种新的云存储数据完整性检测概率解决算法。分析结果表明,对于远端云中的文件,该算法在同步存储其相应的常量级的校验信息的前提下,用户能够以常量的网络通信量和计算量极大概率地正确检测数据的完整性,且支持文件的数据动态更新。下一步研究工作包括如何减少服务器端的计算量以及如何实施公开验证、提供隐私保护等。

参 考 文 献

[1] Kaufman L M. Data Security in the World of cloud computing [J]. Security & Privacy, 2009, 7: 61-64

[2] 冯登国,张敏,张妍,等. 云计算安全研究[J]. 软件学报,2011,22(1):71-83

[3] Mykletun E, Narasimha M, Tsudik G. Authentication and integrity in outsourced databases [J]. ACM Transactions on Storage, 2006, 2(2): 107-138

[4] Ateniese G, Burns R, Curtmola R, et al. Provable data possession at untrusted stores[C]// Proceedings of the 2007 ACM Conference on Computer and Communications Security. New York: ACM, 2007: 598-609

[5] Juels A, Kaliski B S. Pors: proofs of retrievability for large files [C]// Proceedings of the 2007 ACM Conference on Computer and Communications Security. New York: ACM, 2007: 584-597

[6] Shacham H, Waters B. Compact proofs of retrievability[C]// Proceedings of Asiacrypt 2008. Berlin: Springer-Verlag, 2008: 90-107

[7] Ateniese G, Pietro R D, Mancini L V, et al. Scalable and efficient provable data possession [C]// Proceedings of the 4th international conference on security and privacy in Communication networks. New York: ACM, 2008: 9: 1-10

[8] Wang C, Wang Q, Ren K, et al. Ensuring data storage security in cloud computing[C]// Proceedings of International Workshop on Quality of Service 2009. New York, USA: IEEE, 2009: 1-9

[9] Erway C C, Kupcu A, Papamanthou C, et al. Dynamic provable data possession [C]// Proceedings of the 16th ACM conference on Computer and communications security. New York: ACM, 2009: 213-222

[10] Wang Q, Wang C, Li J, et al. Enabling public verifiability and data dynamics for storage security in cloud computing[J]. Lecture Notes in Computer Science, 2009, 5789/2009: 355-370

[11] Bowers K D, Juels A, Oprea A. Proofs of retrievability: Theory and implementation[C]// Proceedings of the 2009 ACM Workshop on Cloud Computing Security, CCSW 2009. New York: ACM, 2009: 43-54

[12] Bragantini R, Conti M, Di Pietro, et al. Security in Outsourced Storage: Efficiently Checking Integrity and Service Level Agreement Compliance[C]// Proceedings of CIT. 2010. New York: IEEE, 2010: 1096-1101

[13] Kamara S, Lauter K. Cryptographic Cloud Storage[C]// Financial Cryptography and Data Security. Berlin: Springer, 2010: 136-149

[14] 罗堃,吴朝宏. 哈希树[EB/OL]. http://wenku.Baidu.com/view/16b2c7abd1f34693daef3e58.html, 2011-11-13