一种支持更新操作的数据空间访问控制方法

潘 颖* 元昌安 李文敬 程茂华
(广西师范学院计算机与信息工程学院 南宁 530023)

摘 要：数据空间是一种新型的数据管理方式，能够以“pay-as-you-go”模式管理海量、动态、异构的数据。然而，由于数据空间环境下数据的动态演化、数据描述的细粒度和极松散性等原因，难于构建有效的访问控制机制。该文提出一个针对数据空间环境下极松散结构模型，重点支持更新操作的细粒度和动态的访问控制框架。首先定义更新操作集用于数据空间的数据更新，提出支持更新操作的映射方法，可将动态数据映射到关系数据库中；给出支持更新操作权限的数据空间访问控制规则的定义，并分析与关系数据库的访问控制规则二者转换的一致性；然后提出具有可靠性和完备性的访问请求动态重写算法，该算法根据用户的读/写访问请求检索相关访问控制规则，使用相关权限信息重写访问请求，从而实现支持动态更新的细粒度数据空间访问控制。理论和实验证明该框架是可行和有效的。

关键词：访问控制；数据空间；Pay-as-you-go；极松散结构
中图分类号： TP309 **文献标识码**：A **文章编号**：1009-5896(2016)08-1935-07
DOI: 10.11999/JEIT151212

Access Control Method for Supporting Update Operations in Dataspace

PAN Ying YUAN Chang'an LI Wenjing CHENG Maohua
(*College of Computer and Information Engineering*, *Guangxi Teachers Education University*, *Nanning* 530023, *China*)

Abstract: Dataspace is a new type of data management, which can manage the mass, heterogeneous, and dynamic data in a pay-as-you-go fashion. However, it is difficult to construct an effective access control mechanism in dataspace environment, because of the data dynamic evolution, the fine-grained and extremely loose data description. A fine-grained and dynamic access control mechanism supporting secure updates is presented in this paper for very loosely structured data model which is commonly used in dataspace. Firstly, a set of update operations are defined for modifying data in the dataspace, and the mapping functions are provided for mapping the updates data into relational databases. Secondly, the fine-grained access control rule supporting secure updates is given, and the consistency of the conversion between this rule and relational database access control rule is analyzed. Thirdly, an access request rewriting algorithm, which is sound and complete, is also presented for dynamically controlling read/write access to the data. The algorithm retrieves the related access control rules based on user's access request, and then rewrites the request by utilizing the relevant authority. Finally, the validity of the work in this paper is proved by the theory and the experiment.

Key words: Access control; Dataspace; Pay-as-you-go; Very loosely structured

1 引言

现代数据具有海量、多样和动态等特点，使得数据集成和管理需要遵循“pay-as-you-go”模式[1,2]，即数据集成和管理应该是渐进的、逐步完善的、功能由简单到复杂的过程，有别于传统的数据库系统和数据集成系统“高成本高功能”的集成和管理技术。为此，数据空间[3,4]，一种以“pay-as-you-go”模式管理海量、动态、异构数据为目标的新型数据管理方式应运而生。然而，由于难于构建有效的数据空间访问控制机制，极大阻碍了数据空间的发展和应用。

和传统的访问控制研究相比，数据空间的访问控制问题更具有挑战性：(1)数据空间需要描述和访问不同层次、不同粒度的数据，因此访问控制必须是细粒度的；(2)数据空间的数据是动态演变的，访问者的权限随着环境条件、数据属性等因素动态变化，因此访问控制必须是动态的。同时，由于数据空间经常发生动态更新，使得访问控制不仅需要考虑数据读操作，更重要的是考虑数据写(更新)操作

收稿日期：2015-11-03；改回日期：2016-03-25；网络出版：2016-05-05
*通信作者：潘颖 panying@gxtc.edu.cn
基金项目：国家自然科学基金(61363074)，广西自然科学基金(2013GXNSFAA019346)，广西教育厅科研项目(2013YB148)
Foundation Items: The National Natural Science Foundation of China (61363074), The Natural Science Foundation of Guangxi Province of China (2013GXNSFAA019346), The Scientific Research Fund of Guangxi Education Department of China (2013YB148)

的安全要求；(3)数据空间普遍采用的极松散结构数据模型便于描述动态异构数据，但其结构的极松散性、数据描述的不完全性等特点使得数据空间支持更新操作的访问控制问题变得复杂。

文献[5]已经建立了基于关系数据库的数据空间访问控制框架，对访问控制规则、查询重写算法等进行了初步的研究，但未涉及数据空间发生动态更新的情形。本文重点研究支持动态更新的细粒度数据空间访问控制，主要内容如下：(1)定义更新操作集用于数据更新；(2)提出支持更新操作的映射方法，将数据空间更新的数据映射到关系数据库中；(3)给出支持更新操作的数据空间访问控制规则的定义，分析其与关系数据库的访问控制规则二者转换的一致性；(4)提出访问请求的动态重写算法，在用户执行读/写操作时，检索相关访问控制规则，将其权限限制条件添加到访问请求中，执行改写后的访问请求，从而动态控制用户对数据的访问。

访问控制问题一直是数据空间领域研究的难点。文献[3]阐述了实现数据空间系统需解决的若干技术难点，其中包括了访问控制技术，并指出存在该难点的原因主要有：现实数据具有分布、动态、异构等特性，多个数据空间存在权限重叠等，但文献没有涉及问题的解决。文献[6]将索引和查询技术嵌入到防篡改硬件中，进而安全管理大型个人数据空间。文献[7]提出一个科学数据空间原型系统，支持全球不同研究机构访问和发布分布式呼吸气体数据和分析资源，该系统的数据访问和数据发布遵循基于智能卡(smart card)的严格的访问机制。文献[8]将 4 元组的数据空间模型 iDM[9]扩展成为 8 元组的数据模型，用于描述基于规则的权限信息，从而支持用户对 iDM 的安全访问。

支持更新操作的访问控制的研究比较多，并取得了不错的成果。文献[10]给出 XML 更新操作和动作类型的定义，提出了一个支持这些更新操作的 XML 访问控制模型。大多数访问控制描述语言只支持读访问权限，不支持添加、删除、修改等更新操作的写访问权限，为此文献[11]提出了一种考虑了写权限的细粒度访问控制描述语言 XACU，通过权限信息标注方式在 XML DTD 中添加写访问权限。文献[12]提出一种基于 XML 更新验证的重写方法，通过定义访问控制描述语言 XACU 更新操作的重写规则，有助于研究访问控制策略的一致性。文献[13]也对 XML 的写访问控制策略的一致性和维护问题进行了研究。文献[14]探讨了支持 XML 查询和更新的访问控制策略的安全性和有效性问题。文献[15,16]研究了支持 RDF 图更新操作的访问控制问题。针对云计算中数据拥有者写-用户读/写的应用场景，文献[17]引入可信平台模块，提高了云计算外包数据访问机制的安全性。文献[18]研究了云计算环境下数据的敏感信息隐藏问题，通过更改签名文件，使其隐私部分不再呈现。

综上所述，数据空间的访问控制研究目前多数停留在理论探讨上，还没有普遍认可的有效的访问控制机制；支持更新操作的访问控制研究主要集中在 XML, RDF 等数据模型上，对极松散结构数据模型鲜有研究。极松散结构模型和 XML, RDF 有较大的区别，例如，XML, RDF 的结构较严格，XML 为树形结构，RDF 严格使用主体、谓词和客体的三元组来描述资源，而极松散结构模型是极松散的图模型，对节点和边的形式没有严格要求；此外，XML, RDF 难于描述复杂关系，如 XML 主要描述父子关系，RDF 不便描述空关系等，而极松散结构模型可以方便描述简单关系(如空关系)和复杂关系(如节点间的多重关系)。因此，基于 XML, RDF 的支持更新操作的访问控制技术不适用于极松散结构模型，我们需要进一步研究面向极松散结构模型的支持更新操作的细粒度动态访问控制方法。

2 支持动态更新的数据空间访问控制框架

支持动态更新的数据空间访问控制框架如图 1 所示。(1)通过映射函数 C 和更新操作集将数据空间的动态数据映射到关系数据库；(2)通过支持更新操作的映射函数 M 将数据空间的访问规则映射到关系表的访问规则，从而将数据空间的细粒度访问控制转换为对相应关系表的访问控制；(3)当用户提出访问请求时，访问请求重写算法首先检索和该用户的读/写操作相关的访问控制规则，然后根据这些规则重写访问请求，使其包含相关权限信息，执行重写后的访问请求，对映射的关系表进行细粒度的访问。

2.1 数据模型

在现有的数据空间模型[5,9]的基础上，为了方便描述动态数据和不确定数据，本文提出一种更为通用的极松散结构模型，它的形式定义如下：

定义 1 数据空间的数据模型是用来描述数据空间中的数据及其关系的图模型，记为 $G:=(N,E)$，其中 N 为节点集 $\{N_1,N_2,\cdots,N_k\}$，节点 N_i 由属性-值对 (attribute-value) 组成，记为 $N_i=\{(A_1^i,V_1^i),(A_2^i,V_2^i),\cdots,(A_n^i,V_n^i)\}$，$A_1^i,A_2^i,\cdots,A_n^i$ 代表节点 N_i 具有的属性系列，$V_1^i,V_2^i,\cdots,V_n^i$ 代表该属性系列对应的值。当 $N_i=\Phi$ 时，称 N_i 为空节点。E 是边的集合，边记为 (N_i,N_j,L)，其中 $N_i,N_j\in N, i\neq j$，L 表示边的标签，且 L 可为 null 值。

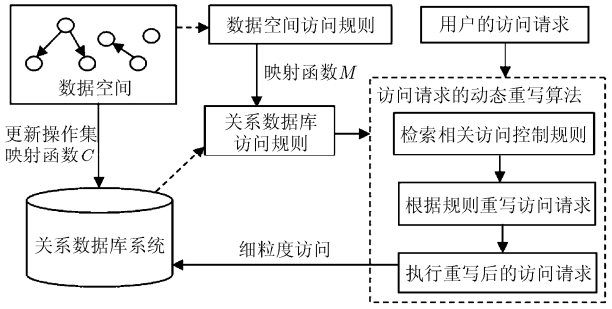

图 1 支持动态更新的数据空间访问控制框架

由定义 1 可知，该数据模型可以描述数据空间里各种粒度的逻辑实体及其关系(如节点可以描述整个文档、文档章节、一段文字等不同粒度的逻辑实体；边可以描述逻辑实体间的各种关系)。此外，空节点为没有包含属性-值对的节点，可用来描述节点的不完全信息；边为 null 值可方便描述不确定的关联信息。

2.2 更新操作集

更新操作集包括 create, delete 和 update 操作，含义如下：

(1)create 操作：

create (nodeid, attribute, value)：添加节点及其属性-值对；

create (nodeid, label, nodeid)：添加节点间的边(关系)。

(2)delete 操作：

delete (nodeid)：删除整个节点，其属性-值对也被删除；

delete(nodeid, attribute, value)：删除节点中的属性-值对；

delete(nodeid, label, nodeid)：删除节点间的边(关系)。

(3)update 操作：

update (nodeid, attribute, value)：更新节点中的属性-值对的内容；

update (nodeid, label, nodeid)：更新节点间的边(关系)的内容。

2.3 支持动态更新的数据模型到关系表的映射

数据空间中的数据是动态演变的，数据模型对数据的描述是从简单到具体、逐渐完善的过程。因此，该数据模型到关系表的映射必须支持数据的动态更新。考虑到数据空间的数据更新操作可以对应 SQL 的 insert, update 和 delete 语句，因此本文通过这些 SQL 语句将更新的数据分别映射到不同的关系表中。

定义 2　G 中数据记为 D_G，关系表中数据记为 D_R，则 G 到关系表的映射函数记为 $C(D_G)=D_R$，具体映射规则如下：

(1) G 的边映射到关系表 Edge(source, label, target)。其中，字段 source 和 target 分别为边的引出节点和引入节点，字段 label 为边的标签。

(2) G 的节点的属性-值对映射到关系表 Attribute(nodeid, value)。其中，字段 nodeid 为节点的 ID，字段 value 为节点的属性值。

(3)数据空间发生 create 操作的情形：当 create (nodeid, attribute, value)时，则使用 insert 语句将所添加的节点及其属性-值对的内容插入到相应的 Attribute 表，如果该属性没有 Attribute 表，则先新建该 Attribute 表再做插入；当 create (nodeid, label, nodeid)时，则使用 insert 语句将所添加节点间的边(关系)插入到 Edge 表。

(4)数据空间发生 delete 操作的情形：当 delete (nodeid)时，则使用 delete 语句删除 Attribute 表和 Edge 表中该节点的所有内容。具体地说，如果该节点有属性-值对，则使用 delete 语句删除 Attribute 表中这些属性-值对；如果该节点有边和其它节点相连，则使用 delete 语句删除 Edge 表中的这些边。当 delete(nodeid, attribute, value)时，则使用 delete 语句删除 Attribute 表中该节点的属性-值对。当 delete(nodeid, label, nodeid) 时，则使用 delete 语句删除 Edge 表中该边(关系)。

(5)数据空间发生update操作的情形：当update (nodeid, attribute, value)时，则使用 update 语句更新 Attribute 表中该节点的属性-值对的内容；当 update(nodeid, label, nodeid)时，则使用 update 语句更新 Edge 表中该边(关系)的内容。

2.4 数据空间的访问控制规则

定义 3　关系表的访问控制规则描述访问者是否对关系表中被访问的资源拥有某些操作权限，包含主体(subject)、客体(object)、操作(action)、标识(sign) 4 个基本元素，记为：$R_R=\{\mathrm{su}_R,\mathrm{ob}_R,\mathrm{ac}_R,\mathrm{si}_R\}$，其中，

(1) su_R 表示对关系表进行访问的用户或角色；

(2) ob_R 表示关系表的访问控制对象，即被访问的资源。这里，我们使用 SQL 语句“select F from T where P”来描述 ob_R，其中 T 代表权限涉及的表(table)的集合；F 代表受权限约束的字段(fields)的集合，且 F 包含在 T 的字段集合中，记为：$F\subseteq T.\mathrm{fields}$；$P$ 是与 T.fields 相关的谓语(predicate)，代表限制条件；

(3) ac_R 表示主体对资源进行的操作，如 read, create, delete 和 update 操作；

(4) si_R 包括标识“+”和“−”，其中，“+”表示肯定授权，“−”表示否定授权。

定义 4 $M:\mathrm{ob}_G \to \mathrm{ob}_R$ 表示 ob_G 到 ob_R 的映射函数，记为：$M(\mathrm{ob}_G)=\mathrm{ob}_R=$ select $F$from $T$where P，其中 ob_G 表示 G 中被访问的资源，$\mathrm{ob}_R \in D_R$，$\mathrm{ob}_G \in D_G$，$T \subseteq$(edge, attribute)，$F \subseteq$(edge.fields ∪attribute.fields)，$P=P_{\mathrm{attribute}} \wedge P_{\mathrm{edge}} \wedge P_{\mathrm{join}}$，$P_{\mathrm{attribute}}$ 中的字段均来自表 Attribute，P_{edge} 中的字段均来自表 Edge，连接谓词 P_{join} 将来自多个表的字段连接起来(为了方便描述，本文在公式中使用 Edge, Attribute 分别表示表 Edge 和表 Attribute)。

这里，ob_G 可以表示任意粒度的 G 中数据，例如可以表示 G，也可以表示 G 中某个属性-值对 对或边中的某部分(如 N_i,N_j 或者 L)等细粒度的数据。

定义 5 数据空间的访问控制规则描述访问者是否对 G 中被访问的资源拥有某些操作权限，记为：$R_G=\{\mathrm{su}_G, M(\mathrm{ob}_G), \mathrm{ac}_G, \mathrm{si}_G\}$。在此，数据空间的访问控制规则通过关系表的访问控制规则来描述。

例 1 定义如下数据空间的访问控制规则，使得用户 user1 只能添加、修改或删除自己的 Email 信息：

{user1, select * from Aemail where user1id = nodeid, create (nodeid, email, value), +}

{user1, select * from Aemail where user1id = nodeid, delete (nodeid, email, value), +}

{user1, select * from Aemail where user1id = nodeid, update (nodeid, email, value), +}

其中，条件 where user1id = nodeid 表示：当用户的登录 ID 等于节点 ID 时，该用户可以访问节点的信息，从而保证用户只能添加、修改或删除自己的 Email 信息。

定义 6 (访问请求)访问请求是指用户/角色 u 对访问资源 o 执行读/写操作 a 的请求，记为 $A:=(u,o,a)$，访问请求的 SQL 描述是指 u 对 o 执行操作 a 的 SQL 表示，a 包括 4 种操作，分别对应 SQL 的 select, create, delete 和 update 语句，访问资源 o 的 SQL 描述方式类似 ob_R。特别地，数据空间的访问请求记为 $A_G:=(u_G,o_G,a_G)$，其中 $u_G \in \mathrm{su}_R$，$o_G \in \mathrm{ob}_G$，$a_G \in \mathrm{ac}_R$。

定义 7 (访问请求的结果)对于一个访问请求 $A:=(u,o,a)$，访问请求的结果是指访问操作 a 在访问资源 o 上的肯定授权执行结果、否定授权执行结果和不确定结果。这里的不确定结果指既不是肯定授权，也不是否定授权的情况。

现实应用中，数据量大时难以定义所有数据的访问权限，因此不确定结果是客观存在的。

定理 1 (访问规则映射的一致性) 对于数据空间的任意一个访问请求 $A_G:=(u_G,o_G,a_G)$，则其访问请求的结果为一个。

证明 根据定义 3，定义 5 和定义 6 可知，数据空间的访问控制规则除了 $M(\mathrm{ob}_G)$，其他内容均和关系数据库的访问控制规则一样，有 $u_G \in \mathrm{su}_R$，$o_G \in \mathrm{ob}_G$，$a_G \in \mathrm{ac}_R$。根据定义 6 和定义 7 可知，对于数据空间的任意一个访问请求，其访问结果为 ac_R 操作在 $M(\mathrm{ob}_G)$ 上的 3 种执行结果之一。因此要证明该定理，只需证明 $M(\mathrm{ob}_G)$ 的值是唯一的，即数据空间上的访问资源到关系数据库的映射结果是唯一的。根据定义 2 和定义 4 可知，对于任意粒度的 ob_G，总存在唯一的 ob_R，使得 $M(\mathrm{ob}_G)=\mathrm{ob}_R$；反过来，考虑 ob_R 不同粒度的情况：表粒度时，ob_R 为属性表或边表，分别对应 G 中包含特定属性的所有节点的该属性-值对或整个边集；行粒度时，ob_R 为属性表或边表中的一行记录，分别对应 G 中一个属性-值对或一条边；列粒度时，ob_R 为属性表或边表中的一列，分别对应 G 中包含特定属性的所有节点的该属性的值或所有边的标签；元素粒度时，ob_R 为属性表或边表中的一个元素，分别对应 G 中一个特定节点的特定属性的值或一条特定边的标签。由此得到：对于任意粒度的 ob_R，也总存在唯一的 ob_G，使得 $M(\mathrm{ob}_G)=\mathrm{ob}_R$。 证毕

定理 1 说明了数据空间的访问规则和关系数据库的访问控制规则二者转换的一致性。

2.5 访问请求的动态重写算法

访问请求的动态重写过程如表 1 的算法 1 所示，该算法的主要思想：根据访问请求 A_G 所涉及的用户/角色、访问资源和操作，检索相关访问控制规则 $\{R_{G1}, R_{G2}, \cdots, R_{Gn}\}$，逐一计算 $M(\mathrm{ob}_{Gi})=\mathrm{ob}_{Ri}$，如果 si_{Gi} 为“+”，则析取肯定授权资源赋值给 ob_R^+，如果 si_{Gi} 为“−”，则合取否定授权资源赋值给 ob_R^-，计算 $S_R=\mathrm{ob}_R^+-\mathrm{ob}_R^-$，然后将 S_R 添加到访问请求的 SQL 描述的 where 条件里，得到包含了权限限制的访问请求 $A_G^{'}$。表 1 通过动态重写访问请求，使其符合相关访问控制规则，进而控制用户/角色对数据的访问。

例 2 假定 user1 的访问请求为：更新 email 信息，即 A_G =(user1, $M(\mathrm{ob}_G)$ =select * from Aemail, update)，相关数据空间的访问控制规则为：{user1, $M(\mathrm{ob}_G)$ =select*from Aemail where user1id= nodeid, update (nodeid, email, value), +}。由算法 1 得：$A_G^{'}$ =(user1, $M(\mathrm{ob}_G)$ =select*from Aemail where user1id=nodeid, update)，其中访问控制规则中的 ob_R^+ 信息被添加到 where 条件中，使得重写

表 1 访问请求的动态重写算法

算法 1　访问请求的动态重写
输入：用户的数据空间访问请求 A_G。
输出：添加了权限限制的访问请求 $A_G^{'}$。
步骤 1　根据 A_G 涉及的用户/角色、访问资源和操作，检索数据空间相关访问控制规则 $\{R_{G1}, R_{G2}, \cdots, R_{Gn}\}$
步骤 2　for i=1 to n do
步骤 3　　计算 $M(\mathrm{ob}_{Gi})$=ob_{Ri}=select F_i from T_i where P_i
步骤 4　　If si_{Gi} 为“+” then $\mathrm{ob}_R^+ = \bigcap_{i=1}^{n} \mathrm{ob}_{Ri}$ //析取肯定授权资源，SQL 的交运算
步骤 5　　If si_{Gi} 为“−” then $\mathrm{ob}_R^- = \bigcup_{i=1}^{n} \mathrm{ob}_{Ri}$ //合取否定授权资源，SQL 的并运算
步骤 6　end for
步骤 7　$S_R \leftarrow \mathrm{ob}_R^+ - \mathrm{ob}_R^-$ //SQL 的差运算
步骤 8　将 S_R 添加到 A_G 的 SQL 描述的 where 条件里，得到并返回 $A_G^{'}$

过的 $A_G^{'}$ 包含权限信息。$A_G^{'}$ 的 SQL 描述为：update Aemail set email=value where user1id = nodeid，访问请求 $A_G^{'}$ 的结果为：update 操作在 select * from Aemail where user1id = nodeid 上的肯定授权执行结果。

下面证明该算法具有可靠性与完备性。设 $S_G = \mathrm{ob}_G^+ - \mathrm{ob}_G^-, x_G \in D_G$。该算法是可靠的，意味着：当 $x_G \notin S_G$ 时，不能通过算法 1 返回的访问请求 $A_G^{'}$ 对 x_G 执行读/写访问操作。该算法是完备的，意味着：当 $x_G \in S_G$ 时，可以通过算法 1 返回的访问请求 $A_G^{'}$ 对 x_G 执行读/写访问操作。

定理 2 (可靠性) 已知 $x_G \in D_G$ 且 $x_G \notin S_G$，则不存在 $A_G^{'}$ 可以对 x_G 执行读/写访问操作。

证明　反证法。假定当 $x_G \notin S_G$ 时，存在 $A_G^{'}$，可以对 x_G 执行读/写访问操作。根据算法 1 可知，$S_R = \mathrm{ob}_R^+ - \mathrm{ob}_R^-$，$S_R$ 为允许执行读/写访问操作的 ob_R^+ 除去禁止执行读/写访问操作的 ob_R^-，即 S_R 中的数据均允许执行读/写访问操作。根据定义 5 和定理 1 可知，$\exists M(x_G) = x_R$，又已知 $x_G \notin S_G$，则有 $x_R \notin S_R$ (可由反证法得证，此略)。由假定条件可知，存在 $A_G^{'}$，可以对 x_G 执行读/写访问操作，根据定理 1 访问规则转换的一致性可知，$A_G^{'}$ 可以对 x_R 执行读/写访问操作，即 $x_R \in S_R$，这与前面的结论 $x_R \notin S_R$ 矛盾。　　证毕

定理 3 (完备性) 已知 $x_G \in D_G$ 且 $x_G \in S_G$，则存在 $A_G^{'}$，可以对 x_G 执行读/写访问操作。

证明　由定理 1 可知，$\exists M(x_G) = x_R$，又由 $x_G \in S_G$ 可知，$x_R \in S_R$ (可由反证法得证，此略)。由 $S_G = \mathrm{ob}_G^+ - \mathrm{ob}_G^-$ 可知，S_G 为允许执行读/写访问操作的 ob_G^+ 除去禁止执行读/写访问操作的 ob_G^-，由 $x_G \in S_G$ 可知，$x_G \in (\mathrm{ob}_G^+ - \mathrm{ob}_G^-)$，即允许对 x_G 执行读/写访问操作。下面证明存在算法 1 返回的访问请求 $A_G^{'}$ 可以对 x_G 执行读/写访问操作。假设访问请求 A_G 为读操作，由算法 1 可得，$A_G^{'}$ 的 SQL 描述为 select 语句，其中，where 子句中定义了可对 S_R 执行读操作的条件，由 $x_R \in S_R$ 及定理 1 可知，$A_G^{'}$ 可对 x_G 执行读操作；同理，假设访问请求 A_G 为修改操作，由算法 1 可得，$A_G^{'}$ 的 SQL 描述为 update 语句，可对 x_G 执行 update 操作；假设访问请求 A_G 为添加操作，由算法 1 可得，$A_G^{'}$ 的 SQL 描述为 insert 语句，可对 x_G 执行 insert 操作；假设访问请求 A_G 为删除操作，由算法 1 可得，$A_G^{'}$ 的 SQL 描述为 delete 语句，可对 x_G 执行 delete 操作。　　证毕

3　实验结果及分析

下面通过实验测试本文方法的有效性和可行性，并和文献[8]的 GDM 方法进行比较分析。实验数据主要来源于在线学术信息服务平台：学者网 schol@t(http://www. scholat.com)，涉及学者的个人信息、论文信息和学术会议信息等。在数据空间下该数据集的节点数为 10000 个，属性-值对 85062 对，其中单个节点包含的属性-值对不超过 10 个，边数为 7904，通过 3.3 节的映射规则转换到 Oracle 12c 中，生成 1 个 Edge 表和 26 个 Attribute 表，近 10 万条记录数。实验的硬件环境为：Intel(R) core(TM) i5-3210M CPU 2.5 GHz, 内存 4 GB，硬盘 430 GB，操作系统 Windows 7 Ultimate with Service Pack 1。

理想状态下，涉及特定用户、特定访问资源的同一类操作的访问规则不会超过 1 个(肯定授权或否定授权)。然而，现实中定义的访问规则可能存在相互矛盾、访问资源重叠授权的情况，相互矛盾的访问规则可以作为不授权处理，为了考察访问请求动态重写算法的有效性，本文重点考虑同用户、同操作下访问资源重叠授权的情况。因此我们定义如下 5 类共 500 个数据空间访问控制规则，存在访问资源重叠授权，涉及表、行、列、元素 4 种粒度：

(1)用户可以读、写所有数据；

(2)用户可以查看部分数据内容，不能进行写操作；

(3)用户可查看所有数据内容，但只能添加部分边/属性-值对；

(4)用户可查看所有数据内容，但只能更新部分边/属性-值对；

(5)用户可查看所有数据内容，但只能删除部分边/属性-值对。

部分数据空间访问控制规则如表 2 所示。

表 2 部分数据空间访问控制规则

{user1, select * from Aemail where user1id= nodeid, create (nodeid, email, value), +}
{user1, select * from Aemail where user1id= nodeid, update (nodeid, email, value), +}
{user1, select * from Aemail where user1id= nodeid, delete (nodeid, email, value), +}
{user1, select * from Aemail, create (nodeid, email, value), +}
{user1, select * from Aemail, update (nodeid, email, value), +}
{user1, select * from edge, create (nodeid, label, nodeid), −}
{user1, select * from edge, update (nodeid, label, nodeid), −}
{user1, select * from edge, delete (nodeid, label, nodeid), −}
{user2, select * from Aemail, update (nodeid, email, value), +}
{user2, select * from Aemail, create (nodeid, email, value), +}
{user2, select * from Aemail, delete (nodeid, email, value), −}
{user2, select * from name where user1id= nodeid, create (nodeid, name, value), +}
{user2, select * from name where user1id= nodeid, delete (nodeid, name, value), +}
{user2, select * from name where user1id= nodeid, update (nodeid, name, value), +}

定义 4 组访问请求 A_{S1}, A_{S2}, A_{S3} 和 A_{S4}，分别对应 Read 操作、create 操作、update 操作和 delete 操作，每组访问请求包含 4 个请求，涉及表、行、列、元素 4 种粒度。

为测试本文方法的可扩展性，分别在不同的节点数上执行 4 组访问请求，每组访问请求执行 10 次，求其平均时间，实验结果如图 2 所示。随着节点数量的增长，执行时间也逐渐增长，时间开销与节点个数成线性关系，本文方法具有较好的可扩展性。

根据文献[8]的思想，我们在数据空间中做如下仿真实验：预先添加访问规则到 iDM 中，采用 iQL 语言处理访问请求。将本文方法、GDM 方法、Oracle 环境下无访问控制进行比较，每组访问请求分别执行 10 次，求其平均时间，得到的实验结果如图 3 所示。采用本文方法时，4 组访问请求的执行时间比无权限控制时的执行时间略高，但在可接受的范围。GDM 方法的仿真实验预先将控制规则写入数据模型中，在查询阶段省略了访问权限的检索和计算，所以执行时间较短，但在现实使用中，将规则写入数据模型、权限的检索和计算会花费更多时间。本文方法的读操作访问请求 A_{S1} 比 3 组写操作访问请求 $A_{S2}-A_{S4}$ 的执行时间低，最高的执行时间是 create 操作 A_{S2}，这是因为与 A_{S2} 相关的访问控制规则存在较多的重叠授权，造成了 S_R 的计算量较大。考虑到可以预先对访问控制规则进行优化，计算同一操作的 ob_R^+ 和 ob_R^-，从而降低访问请求执行时 S_R 的计算量。总体上看，本文的访问控制框架所增加的负担在可接受和可控的范围，因此是可行的。

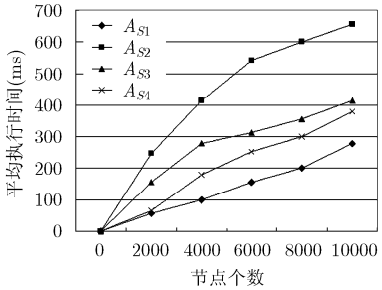

图 2 不同节点数 4 组访问请求的执行时间比较

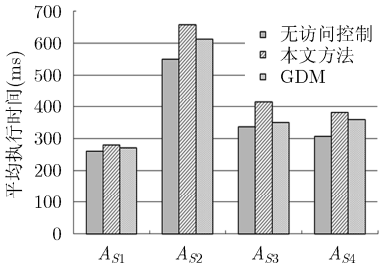

图 3 3 种方法访问请求执行时间比较

4 结束语

本文提出了一个基于关系数据库的支持动态更新的数据空间访问控制框架，在用户对数据空间的数据进行更新时，能够即时根据用户的访问控制权限，阻止或者允许用户的行为。关系数据库的访问控制技术较为成熟，基于关系数据库的 XML, RDF 等模型的访问控制一直是研究热点[19,20]。因此，本文在现有关系数据库访问控制研究的基础上探讨数据空间的访问控制框架，也是可行和必要的。此外，本文的研究对解决动态异构数据的更新操作安全问题有一定的借鉴价值。将来拟重点探讨用户授权问题，考虑数据动态演变、数据质量低等特殊情况下，如何高效、准确地为不同用户指定其可以访问的数据。

参 考 文 献

[1] MARX V. Biology: The big challenges of big data[J]. *Nature*, 2013, 498(7453): 255–260.

[2] NGUYEN Q V H, NGUYEN T T, MIKLÓS Z, *et al*. Pay-as-you-go reconciliation in schema matching networks[C]. International Conference on Data Engineering (ICDE). Chicago, IL, USA, 2014: 220–231.

[3] HALEVY A, FRANKLIN M, and MAIER D. Principles of dataspace systems[C]. Proceedings of the 25th ACM

SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems(PODS). Chicago, IL, USA, 2006: 1–9.

[4] 李玉坤, 孟小峰, 张相於. 数据空间技术研究[J]. 软件学报, 2008, 19(8): 2018–2031.
LI Yukun, MENG Xiaofeng, and ZHANG Xiangyu. Research on dataspace[J]. *Journal of Software*, 2008, 19(8): 2018–2031.

[5] 潘颖, 汤庸, 刘海. 基于关系数据库的极松散结构数据模型的访问控制研究[J]. 电子学报, 2012, 40(3): 600–606.
PAN Ying, TANG Yong, and LIU Hai. Access control in very loosely structured data model using relational databases[J]. *Acta Electronica Sinica*, 2012, 40(3): 600–606.

[6] LALLALI S, ANCIAUX N, SANDU POPA I, *et al*. A secure search engine for the personal cloud[C]. Proceedings of the ACM SIGMOD International Conference on Management of Data. Melbourne, VIC, Australia, 2015: 1445–1450.

[7] ELSAYED I, LUDESCHER T, SCHWARZ K, *et al*. Towards realization of scientific dataspaces for the breath gas analysis research community[C]. CEUR Workshop Proceedings, Temuco, Chile, 2009: 1–8.

[8] JIN Lei, ZHANG Yawei, and YE Xiaojun. An extensible data model with security support for dataspace management[C]. Proceedings of the 10th International Conference on High Performance Computing and Communications (HPCC). Dalian, China, 2008: 556–563.

[9] DITTRICH J P and SALLES M A V. iDM: a unified and versatile data model for personal dataspace management[C]. Proceedings of the 32nd International Conference on Very Large Data Bases. Seoul, Korea, 2006: 367–378.

[10] LIM C H, PARK S, and SON S H. Access control of XML documents considering update operations[C]. Proceedings of the ACM Workshop on XML Security. ACM, Fairfax, VA, USA, 2003: 49–59.

[11] FUNDULAKI I and MANETH S. Formalizing XML access control for update operations[C]. Proceedings of the 12th ACM Symposium on Access Control Models and Technologies. Sophia Antipolis, France, 2007: 169–174.

[12] JACQUEMARD F and RUSINOWITCH M. Rewrite-based verification of XML updates[C]. Proceedings of the 12thInternational ACM SIGPLAN Symposium on Principles and Practice of Declarative Programming. Hagenberg, Austria, 2010: 119–130.

[13] BRAVO L, CHENEY J, FUNDULAKI I, *et al*. Consistency and repair for XML write-access control policies[J]. *The VLDB Journal*, 2012, 21(6): 843–867.

[14] MIRABI M, IBRAHIM H, FATHI L, *et al*. A dynamic compressed accessibility map for secure XML querying and updating[J]. *Journal of Information Science and Engineering*, 2015, 31(1): 59–93.

[15] SAYAH T, COQUERY E, THION R, *et al*. Inference Leakage Detection for Authorization Policies over RDF Data[M]. Data and Applications Security and Privacy. Berlin, Germany, Springer International Publishing, 2015: 346–361.

[16] RACHAPALLI J, KHADILKAR V, KANTARCIOGLU M, *et al*. Towards fine grained RDF access control[C]. Proceedings of the 19th ACM Symposium on Access Control Models and Technologies. London, ON, Canada, 2014: 165–176.

[17] 付东来, 彭新光, 杨玉丽. 基于可信平台模块的外包数据安全访问方案[J]. 电子与信息学报, 2013, 35(7): 1766–1773. doi: 10.3724/SP.J.1146.2012.01321.
FU Donglai, PENG Xinguang, and YANG Yuli. Trusted platform module-based scheme for secure access to outsourced data[J]. *Journal of Electronics & Information Technology*, 2013, 35(7): 1766–1773. doi: 10.3724/SP.J.1146.2012.01321.

[18] 刘西蒙, 马建峰, 熊金波, 等. 云计算环境下基于属性的可净化签名方案[J]. 电子与信息学报, 2014, 36(7): 1749–1754. doi: 10.3724/SP.J.1146.2013.01154.
LIU Ximeng, MA Jianfeng, XIONG Jinbo, *et al*. Attribute based sanitizable signature scheme in cloud computing[J]. *Journal of Electronics & Information Technology*, 2014, 36(7): 1749–1754. doi: 10.3724/SP.J.1146.2013.01154.

[19] EL-AZIZ A, AHMED A E A, and KANNAN A. XML access control: mapping XACML Policies to relational database tables[J]. *International Arab Journal of Information Technology*, 2014, 11(6): 532–539.

[20] PAPAKON STANTINOU V, MICHOU M, FUNDULAKI I, *et al*. Access control for RDF graphs using abstract models[C]. Proceedings of the 17th ACM Symposium on Access Control Models and Technologies. Newark, NJ, USA, 2012: 103–112.

潘　颖：女，1972 年生，教授，博士，硕士生导师，研究方向为大数据管理、信息安全.

元昌安：男，1964年生，教授，博士，硕士生导师，研究方向为数据库与知识工程、智能计算.

李文敬：男，1964 年生，教授，硕士生导师，研究方向为数据库与知识工程、信息安全.

程茂华：男，1991 年生，硕士生，研究方向为大数据管理、服务计算.