文章编号:1001 - 9081(2009)09 - 2336 - 03

快速有效的 XML 访问控制新方案

李时文,卢建朱
(暨南大学 信息科学与技术学院,广州 510632)
(80lsw163@163.com)

摘 要:随着可扩展标记语言(XML)文档的广泛使用和用户安全意识的加强,XML 数据的安全问题显得日益重要。结合索引/标记方案,设计了一种安全的、能有效查询和快速更新的 XML 访问控制新方案。该方案利用多种授权实现了权限不同的多个用户灵活、安全地查询 XML 文档数据;利用空对象和备注子节点实现了 XML 数据的删除和插入。

关键词:可扩展标记语言;访问控制;有效查询;快速更新;索引/标记方案
中图分类号: TP309.2 **文献标志码**:A

New scheme of XML-based access control for effective querying and quick updating

LI Shi-wen, LU Jian-zhu
(*College of Information Science and Technology, Jinan University, Guangzhou Guangdong* 510632, *China*)

Abstract: With the wide use of Extensible Markup Language (XML) documents and the strengthening security awareness, the security issues of XML data become increasingly important. The authors proposed a XML-based access control scheme, which implemented effective querying by combining indexing/labeling scheme. To achieve quick updating, the scheme used NULL objects to delete XML data and memo sub-nodes to insert XML data respectively.

Key words: Extensible Markup Language (XML); access control; effective querying; quick updating; indexing/labeling scheme

XML(Extensible Markup Language)作为一种新兴的标识语言,在 Web 上得到广泛的应用,并逐步成为 Internet 存储交换数据的一种机制。随着 XML 的广泛应用,用户对其数据的安全使用和共享提出了更高的要求,尤其是对于存放有机密信息的 XML 数据库。早期的 XML 安全研究侧重于传输过程中的数据加密和数据签名,但这两种技术不能满足以查询为目的的 XML 数据库的安全要求。基于授权的访问控制[1]380,[2]77加强了主体对访问对象的限制,使用户能访问其获得许可的 XML 文档片断,这是 XML 数据安全的发展方向。但已有的访问控制模型大多采用节点过滤技术[1]379,[3]1529。节点过滤技术的缺点是在处理过程中要对 XML 文档进行分解、标记(允许为"+",禁止为"-")和剪枝处理;且每个用户的每一个操作都需重复的分解、标记、剪枝处理。

本文在已有节点过滤技术[3]1531 - 1532,[4]185 - 187的基础上,结合文献[5]提出的索引/标记方案,设计了一种能有效查询和快速更新的 XML 访问控制新方案。其主要思想如下:1)采用基于身份与基于位置的主体授权和多种数据授权,确保了同类数据在不同部门之间的过滤。2)结合已有的索引/标记方案[5],提出了一种基于访问控制策略的索引/标记方案。3)首次讨论了在不改变 XML 和 DTD 文档结构的条件下,利用空对象和备注子节点实现 XML 数据的删除和插入。

1 XML 访问控制模型

XML 是一种以半结构化的方式来表示数据的文本语言,其数据结构是嵌套的,它用一系列规则来保证 XML 文档的良构性,并提供了验证 XML 文档有效性的两个机制:DTD 和 XML Scheme。目前,采用 DTD 验证 XML 的机制应用仍较广泛,本文也采用这种机制。假设某公司由行政,人事,财务,生产,业务等部门组成,其员工信息的 XML 文档如下所示:

```
<?xml version = "1.0"? >
<?DOCTYPE employeelist SYSTEM "employeelist.dtd",
<employee >
    <dept name = "businesss" > 业务部 </dept >
    <staff >
        <name > xiao wu </name. >
        <contact > 133 ***** </contact >
        <edu > college </edu >
        <ID_num > 430301 ***** </ID_num >
        <payroll >
            <b_pay > $ 300.00 </b_pay >
            <bonus > $ 542.06 </bonus >
            <memo > 工资已结算 </memo >
        </payroll >
        <perf >
            <p_volume > 1200 件 </p_volume >
            <a_output > 800 件 </a_output >
            <memo > 比上月有提高 </memo >
        </perf > </staff >
    <memo >
        <public > 紧急联系方式: 159 *** </public >
        <private > 个人简介 </private >
        <protected > 建议 </protected >
    </memo > </employee >
```

收稿日期:2009 - 03 - 10;**修回日期**:2009 - 05 - 04。
基金项目:国家自然科学基金资助项目(60773083);省部产学研项目(2008B090500201)。
作者简介:李时文(1980 -),男,湖南湘潭人,硕士研究生,主要研究方向:信息系统、网络信息安全; 卢建朱(1965 -),男,湖南郴州人,副教授,主要研究方向:多媒体数据处理、网络安全通信、计算机网络与安全。

1.1 模型定义

一个访问控制模型由访问者,受控对象和操作三个部分组成,本文研究的多级安全访问控制模型由一个五元组(Sub,Obj,Sec,Pro,valid())组成。令角色集 $R = \{r_1, r_2, \cdots, r_m\}$,数据集 $D = \{d_1, d_2, \cdots, d_n\}$,用户操作集 $O = \{read, modify, insert, delete\}$,安全标识集 $S = \{s_1, s_2, \cdots, s_i\}$,其定义如下。

1) $Sub \in R$ 为具有某一角色的授权用户或用户组。

2) $Obj \in D$ 表示客体,由一组 XML、DTD 文档的 URIs(Uniform Resources Identifiers)或由 XPath 指定的 XML 或 DTD 文档的元素、属性、实体等组成。

3) $Sec \subseteq O \times S$ 为操作权限,令"≥"表示权限的偏序关系,$Sec(s)$、$Sec(o)$ 分别表示 Sub、Obj 的授权,根据 BLP 模型的基本规则[6]:① 当且仅当 $Sec(s) \geq Sec(o)$ 时,读操作被允许;② 当且仅当 $Sec(s) \leq Sec(o)$ 时,写操作被允许。出于客体完整性的考虑,实际中的写操作通常被限制为 $Sec(s) = Sec(o)$。

4) $Pro \in \{LDH, RDH, L, R, LD, RD, LS, RS\}$[4]185,用来定义授权的传播与继承策略。

5) $valid()$ 为更新操作有效性检查函数,其值为 true/false。

该模型定义用户、角色和客体,通过建立三者的授权关系实现访问控制。角色被分配一定的权限,而用户被分配相应的角色来获得权限,使用户和权限之间不再直接联系,简化了权限的分配和管理。通过定义客体的权限来选择和过滤用户访问的数据,确保用户只能访问其工作范围内的最小数据。结合主体和客体授权实现同类数据在不同部门之间的过滤,确保不同部门的用户只能操作本部门的数据,适用于数据保密性要求高的 XML 数据库的访问控制。

1.2 索引/标识方案

XML 索引/标记为每个节点分配唯一编码值,利用索引/标记值来查找和过滤数据,仅需比较节点编码值而不必访问原 XML 文档,就可以快速有效地确定所需数据。令 $L[n] \in \{a, b, c, \cdots, z, zb, \cdots, zz, zzb, \cdots\}$ 表示节点 n 的尾标,同一节点的子节点从左到右 $L[n]$ 的值从"b"开始,按尾标集中顺序依次取值。则利用文献[5]定义 XML 节点标记规则如下。

1) XML 文档根节点标记为"0"。

2) 第 $k+1$ 层的节点 n,若它的父节点的标记为 $k-1 panernt(n)$,则 $label(n) = k\ parent(n).L[n]$。

3) 在第 $k+1$ 层的节点 m 前后插入一个新节点 p 时,分三种情形考虑,如图 1 所示,其中虚线表示插入节点。

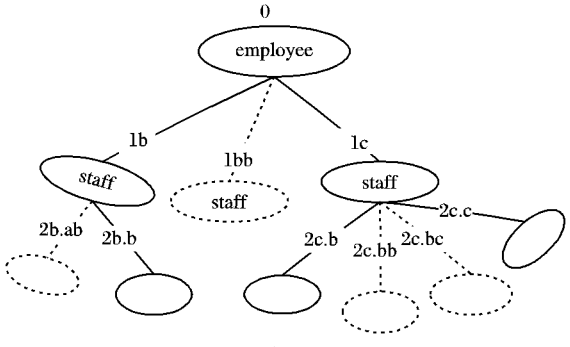

图 1　动态索引/标记方案

① 若节点 m 没有左兄弟,且新插入节点 p 为其左兄弟时,则节点 p 的标记为:

$$label(p) = \begin{cases} k\ panent(m).L[m-1]b, & L[m] \text{ 最后字母为 } b \\ k\ panetnt(m)L[m-1], & \text{其他} \end{cases}$$

② 若插入节点 p 位于节点 m,n 之间时,节点 p 的标记为:

$$label(p) = \begin{cases} k\ panent(m).L[m+1], & label(p) \neq label(n) \\ k\ panetnt(m)L[m]b, & label(p) = label(n) \end{cases}$$

③ 若节点 m 无右兄弟,且新插入节点 p 为其右兄弟时,则节点 p 的标记为:

$$label(p) = k\ parent(m).L[m+1]$$

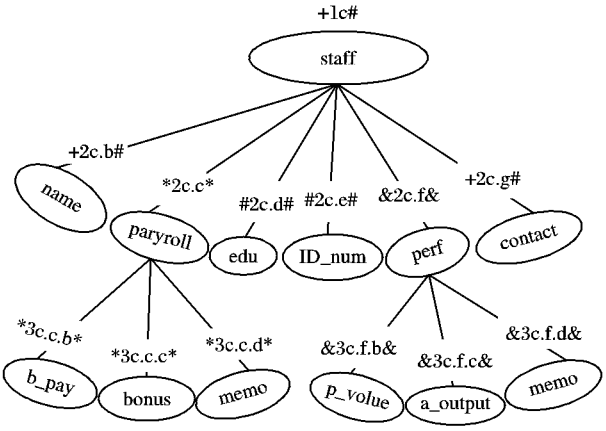

图 2　扩展的动态索引/标记方案

为了使索引/标记方案能应用于 XML 访问控制系统中,我们扩展了以上标记规则,用形式(1)

$$< read(n) >< label(n) >< write(n) > \quad (1)$$

将节点 n 的读、写权限与索引/标记值相结合,实现在访问控制条件下利用索引/标记来加快数据的查询和查找,如图2所示。这里 $read(n)$、$write(n)$ 分别为节点 n 的读、写权限。

2 系统结构及案例分析

本文设计的系统体系结构如图 3 所示。用户用自己的用户名和密码登录系统,由用户管理模块验证用户的合法性。通过验证的用户由角色管理模块分配合适的角色,并实现用户与角色的权限映像。XML 预处理模块验证某一角色的用户权限,判断用户请求的合法性,并调用 valid() 函数检查用户操作是否改变 XML 文档的有效性。将索引/标识方案与访问控制策略结合,通过动态索引/标记方案来查找数据,能加速数据的查询与查找处理。

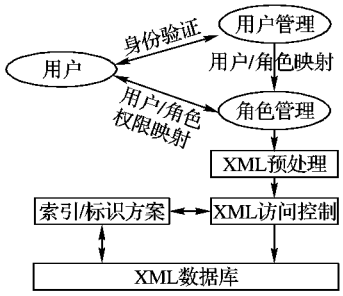

图 3　系统体系结构

在上文所提的某公司中,规定权限为禁止、公开、秘密、机密、财务、个人五级,对应符号为"-"、"+"、"#"、"&"、"*"、"$",用户权限的优先级为:禁止 < 公开 < 秘密 < 机密,客体权限的优先级为:公开 < 秘密 < 机密 < 禁止,财务类信息只能由财务人员访问。员工信息的授权文件 XAA(XML Access Authorization)和用户授权文件 XGA(XML Group Authorization)分别如下:

```
<XAA doc = "employeelist. xml" >
<rule object = "/employee/staff/name"read = " + "
write = "#" type = "R"/ >
<rule object = "//contact" read = " + " write = " + " / >
<rule object = "//dept" read = " + " write = "#" / >
<rule object = "//edu" read = "#" write = "#" / >
<rule object = "//ID_num" read = "#" write = "#" / >
<rule object = "//payroll" read = " * " write = " * " type = "R"/ >
<rule object = "//perf" read = "&" write = "&" type = "R" / >
<rule object = "//public" read = " + " write = " + " / >
<rule object = "//private" read = "#" write = "#" / >
<rule object = "//protected" read = "&" write = "&" / >
…
</XAA >
<XAG doc = "employeelist. xml" >
<group ID = "public" read_type = " + " write_type = " - " / >
<group ID = "staff" read_type = " +, $ " write_type = " + " / >
<group ID = "admin" read_type = " +,#, $ " write_type = " - " / >
<group ID = "head" read_type = " $,&" write_type = "&" / >
<group ID = "202.116.52. * " read_type = " $, +, * " write_type
    = " * " / >
<group ID = "201.116.53. * " read_type = " $,#" write_type = "#"/ >
<group ID = "CEO" read_type = "&, * " write_type = " - " / >
…
</XGA >
```

根据规定列出部分规则如下：

规则1 小王是该公司一名普通员工，他能查看自己的个人信息和公司其他员工的公开信息，其访问授权规则为：< group ID = "staff" read_type = " +, $ " write_type = " + "/ >。

规则2 财务部和人事部员工的授权可以通过基于IP地址来实现。如：< group ID = "202.116.52. * " read_type = " $, +, * " write_type = " * " / >。

规则3 部门领导只能访问本部门员工除财务之外的所有信息，以及其他部门员工的公共信息，其访问授权定义如下：

```
<group ID = "head" read_type = " + " write_type = " - "/ >
<rule object = "//dept[@name = '本部门']" read_type =
    "&" write_type = "&" type = "R"/ > </group >
```

3 基于XML访问控制的快速更新

XML的写操作主要包含三类基本操作：修改、删除和插入。XML文档与普通文本文档或数据文档不同，在更新XML数据时，需进行有效性检查。也就是说，在基于XML访问控制策略上更新数据，不仅要考虑访问控制策略，还需考虑DTD所定义的结构，即一个更新操作应该在被确认是“安全有效”的情况下才能被执行。

修改操作只更新XML元素或属性值，不会引起XML文档结构的改变。插入和删除操作存在需改变XML文档结构和不需改变XML文档结构两种情况。当需改变XML文档结构时，授权用户需修改DTD使XML文档重新满足有效性。当前关于XML文档的删除和插入操作都存在着需要改变XML文档结构[2]79,[7]1104。为了解决这个问题，提出了一种快速更新方案。即在删除数据时，采用将所删对象值置空的方法来实现删除操作；在插入数据时，采用将插入信息写入预留备注元素的方法来实现插入操作。

3.1 删除XML数据

删除XML数据时，首先由XML预处理模块调用valid()函数判断该操作是否需改变XML文档结构，查询处理器再根据valid()的值来决定下一步操作。其操作的主要过程如下：

1）用户向预处理进程发送删除数据的请求；

2）系统根据需删除数据的访问类型验证该用户的权限，如果该操作合法，则继续，否则禁止；

3）系统预处理程序调用valid()函数判断该操作是否改变XML文档结构，若改变，则返回true，否则返回false；

4）查询处理器利用索引/标记值定位需删除的数据，然后根据valid()函数的返回值来决定下一步操作；

5）如果返回值为true，则将删除对象值置换成空对象，否则，直接删除数据。

3.2 备注元素定义与插入XML数据

在优化备注元素的定义位置时，为了避免因定义过多的备注元素而造成空间的浪费，或因定义备注元素的位置不合理而造成插入数据的不方便，规定如下。

1）在可预见的将来，如果某父节点包含子节点较多且增加数据节点的几率较大时，通常在该节点的最后子节点后预留备注节点。从兄弟节点具有较大的相似性考虑，将该类备注元素的访问类型定义为其兄弟节点中出现次数最多的访问类型。如：payroll元素的memo子节点。

2）通常在根节点的最后子节点后预留备注元素，且将该备注元素定义为具有多个子节点的形式。我们将备注元素子节点的访问类型分别定义为终端节点所具有的各种类型。如：employee元素的memo分枝树。

3）其他情况如无特殊要求或说明，从节省空间的角度考虑，通常不再定义备注元素节点。

插入XML数据时，按如下步骤进行。

1）用户请求在指定位置插入节点n，并给出节点n的访问类型。

2）系统根据输入的访问类型验证该用户的权限，如果该操作合法，则继续，否则禁止。

3）系统预处理程序调用valid()函数判断该操作是否改变XML文档结构，若需改变，则返回true，否则返回false。

4）查询处理器利用索引/标记值定位插入位置，然后根据valid()函数的返回值来决定下一步操作；若返回值为false，则插入数据。

5）否则，通过索引/标记值搜索其兄弟节点，查找是否有因删除而置空的空元素，如果找到且访问类型相符，则将数据插入到该空元素中。若未找到，则查找其兄弟节点中是否有符合要求的备注元素，如果找到则将数据插入到该备注元素中。

6）如果在其兄弟节点中未找到符合要求的插入位置，则返回其父节点中按以上方法继续查找，直到找到为止。

因为备注元素通常会在根节点的最后子节点位置出现，且其子节点的访问类型分别定义为终端节点所具有的各种类型，这样就能保证最终都能找到合适的插入位置。当需多次插入时，可将DTD中备注元素的量词设为*，来实现0次或多次插入。

4 安全性和效率分析

本文方案通过定义用户角色来授权用户只具有完成其正常工作的最小权限，通过定义数据的权限来选择和过滤用户访问的对象，确保用户只能访问其工作范围内的最小数据。与基于角色的访问控制模型相比，本文方案结合用户授权和

（下转第2341页）

值选取如下所示:

q = "8780710799663312522437781984754049815806883199414208211028653399266475630880222957078625179422662221423155858769582317459277713367317481324925129998224791"

r = "7307508186654516213611192455715049014059765596617"

$r = 2^{159} + 2^{107} + 1$(Solinas 素数)

h = "1201601226489114607938882136674053420480295440125131182291961513104720728935970453110284480218390653778677 6"

p = "[3163580956772149639925912500766810073109666440270410289968026398982380938528821100065051990342893360704324303698244672599590418044198150276814580345072723,1865875802148101784198652765771498409027880624129171910648521084280016474914846379432515547108209224628155167940046585683527588501231517546468799494459083]"

实验测试模拟的重签名结构如图 1 所示,3 个成员进行部分签名,最后由签名合成者完成重签名。实验测试结果为三个成员和重签名合成者完成重签名共耗时 546 ms,验证重签名时间开销为 141 ms。其签名和验证签名时间开销是文献[5]方案的 3/4。

4　结语

企事业机构中,一个决议的通过要经过多个部门领导的签名,要满足一定签名次序和结构,结构化多重签名就是对这类问题的一种解决方案。本文基于双线性对提出了一种基于身份的结构化多重签名方案,该方案以用户的身份信息,如电子邮箱地址,IP 地址、电话号码等作为用户公钥,从而降低了建立和管理公钥基础设施的代价,避免了用户对公钥及其证书的存储和传递等问题。最后给出该种重签名方案的仿真实现。

参考文献:

[1] BONEH D, LYNN B, SHACHAM H. Short signatures from the Weil pairing [C]// Proceedings of Advances in Cryptology - Asia - crypt'2001. Berlin: Springer-Verlag, 2001: 514 - 532.

[2] BONEH D, BOYEN X, SHACHAM H. Short group signatures [C]// Proceedings of Cryptology - CRYPTO 2004. Berlin: Springer-Verlag, 2004: 41 - 59.

[3] BOLDYREVA A. Efficient threshold signature, multi-signature and blind signature schemes based on the Gap-Diffie-Hellman-group signature scheme [C]// Proceedings of Practice and Theory in Public Key Cryptography. Berlin: Springer-Verlag, 2003: 31 - 46.

[4] CHEN X, ZHANG F, KIM K. A new ID-based group signature scheme from bilinear pairings [C]// WISA 2003: Proceedings of the 2003 International Workshop on Information Security Applications. Berlin: Springer-Verlag, 2003: 585 - 592.

[5] 吴克力. 一个带签名者意向的结构化多重签名方案[J]. 电子与信息学报, 2006, 28(5): 825 - 826.

[6] SHAMIR A. Identity-based cryptosystems and signature schemes [C]// Proceedings of Advances in Cryptology - Crypto'84. Berlin: Springer-Verlag, 1984: 47 - 53.

(上接第 2338 页)

数据权限实现了不同部门的用户只能操作本部门的数据,而传统的基于角色的访问控制模型只提供同类数据的过滤功能,不能实现同类数据在不同部门之间的过滤[8]。相比而言,我们设计的新方案具有较高的数据安全保密性。

基于用户处理查询时,节点过滤技术需分解 XML 文档成 DOM 树,然后标记 DOM 树的各节点,再对标记后的 DOM 树剪枝,以便生成用户视图。且对每个用户的每次请求都需进行重复的分解、标记和剪枝处理。而本文方案只需利用索引/标记值将需要访问的数据过滤出来即可,能减少和优化处理步骤,加快查询和查找速度,节省系统资源。

本文的快速更新方案在实现插入、删除等更新操作时不需改变 DTD 和 XML 文档结构,以浪费少量的空间来换取更新操作的时间,具有较快的处理速度。且不需要修改授权访问规则,减少了对授权访问规则的频繁访问,这是其他更新方案[2]82 - 83,[7]1100 - 1103所不具备的。

5　结语

针对 XML 数据安全日趋重要的现实,本文提出了一种细粒度的同时支持读写权限的访问控制新方案。该方案以灵活有效的方法控制用户对 XML 数据库系统中不同安全等级数据的访问;利用空对象和备注子节点实现 XML 数据的删除和插入,确保了数据的有效性和安全性;结合动态索引/标记方案来实现查找,减少了处理步骤,加快了查找速度。

参考文献:

[1] DAMIANI E, FANSI M, GABILLON A, *et al.* A general approach to securely querying XML [J]. Computer Standards and Interfaces, 2008, 30(6): 379 - 389.

[2] DUONG M, ZHANG Y. An integrated access control for securely querying and updating XML data [C]// Proceedings of the 19th Conference on Australasian Database. Darlinghurst, Australia: Australian Computer Society, 2008: 75 - 84.

[3] 李斓, 何永忠, 冯登国. 面向 XML 文档的细粒度强制访问控制模型[J]. 软件学报, 2004, 15(10): 1528 - 1537.

[4] DAMIANI E, De CAPITANI di VIMERCATI S, PARABOSCHI S, *et al.* A fine-grained access control system for XML documents [J]. ACM Transactions on Information and System Security, 2002, 5(2): 169 - 202.

[5] DUONG M, ZHANG Y. LSDX: A new labeling scheme for dynamically updating XML data [C]// Proceedings of the 16th Australasian Database Conference. Darlinghurst, Australia: Australian Computer Society, 2005: 185 - 193.

[6] ZHAO G S, CHADWICK D W. On the modeling of Bell-LaPadula security policies Using RBAC [C]// Proceedings of the 17th IEEE International Workshops on Enabling Technologies. Los Alamitos, CA: IEEE Computer Science, 2008: 257 - 262.

[7] DAMIANI E, FANSI M, GABILLON A, *et al.* Securely updating XML [J]. KES 2007: Proceedings of the 11th International Conference on Neural Networks, LNCS 4694. Berlin: Springer-Verlag, 2007: 1098 - 1106.

[8] MASSACCI F, MYLOPOULOS J, ZANNONE N. Hierarchical hippocratic databases with minimal disclosure for virtual organizations [C]// Proceedings of the 12th ACM Conference on Computer and Communications Security in Computer Security. Heidelberg: Springer, 2005: 438 - 454.