基于XML的数据客体与安全标记绑定方法

曹利峰[1,3] 李 中[2] 陈性元[2] 冯 瑜[1]
(解放军信息工程大学四院 郑州 450004)[1] (解放军信息工程大学三院 郑州 450004)[1]
(数学工程与先进计算国家重点实验室 郑州 450004)[3]

摘 要 安全标记与数据客体的绑定,是制约多级安全真正走向网络实用化的关键问题。针对这一问题,在深入分析XML的基础上,描述了XML客体安全标记及其约束规则,提出了安全标记与数据客体的绑定方法,讨论了安全标记查询、客体内容裂解等相关操作,给出了基于XML安全标记的安全通信实施机制。该绑定方法不仅能够满足多级信息系统间安全通信的需要,而且能够实施粒度更细的访问控制,提高信息客体的利用率。

关键词 多级安全,等级保护,XML,安全标记,数据客体
中图法分类号 TP303.08 **文献标识码** A

Method of Binding Secure Label to Data Object Based on XML

CAO Li-feng[1,3] LI Zhong[2] CHEN Xing-yuan[2] FENG Yu[1]
(The Fourth Institute, PLA Information Engineering University, Zhengzhou 450004, China)[1]
(The Third Institute, PLA Information Engineering University, Zhengzhou 450004, China)[2]
(State Key Laboratory of Mathematical Engineering and Advanced Computing, Zhengzhou)[3]

Abstract How to bind secure label to data object is a key problem in multi-level network that restricts MLS from practicality on network. This paper analyzed deeply xml, and expounded secure label of object based on xml and its restrictions, then put forward a method of binding secure label to data object based on XML. At the same time, some operations were discussed in detail, such as query of secure label, decomposition of object. Finally, secure communication based on secure label was described in multi-level network. The method can not only meet the need of secure communication in multi-level network, but also accomplish fine-grained mandatory access control, which may improve availability of information and reduce complexity of binding.

Keywords MLS, Classified security protection, XML, Secure label, Data object

1 引言

等级保护[1]是依据信息系统的重要程度,针对信息系统面临的风险,采用不同的信息安全防护。其理论支撑最早可追溯到美国橘皮书提出的多级安全,多级安全通过为主体、客体分配安全标记,依据安全标记的比较,达到主体对客体访问的目的。可见,安全标记是多级安全的基础,也是等级保护三级及三级以上级信息系统构建的关键。但是,安全标记并非为数据客体固有的属性,缺乏有效的绑定措施,因此存在着被替换、假冒、不一致与网络灵活性差等问题,这制约着多级安全真正网络化、实用化,从而也影响着信息系统等级保护的整改工作。

针对数据客体与安全标记的绑定,传统的方法是在数据或客体里添加安全标记,比如文档的头部或尾部、电子邮件信息正体的第一行,并通过对数据或客体进行数字签名,来增强安全标记的安全性。传统的方法尽管可以实现安全标记与数据客体的绑定,但是仍存着一定的问题,主要表现在:

①传统方法未考虑客体的数据格式,每增加一种类型的数据客体,就要研究一种新的绑定方法,难度较大,也较为复杂。特别是在多级信息系统间异构数据交换时,访问控制较难[8,9]。因此,如何实现安全标记对数据客体的统一绑定,是降低安全标记绑定复杂性、解决多级信息系统异构数据安全交换问题的关键。

②传统方法对数据客体进行的是整体打标记,难以实施细粒度的访问控制。当数据客体由若干不同安全级别的子单元组成时,数据客体仅仅能满足安全级别不小于数据客体安全级别(所有子单元的最大安全级别)的主体访问,而不能被其他主体所访问,这也将大大降低数据客体的利用率。

因此,本文在深入分析数据客体特征的基础上,定义了XML客体安全标记,依托XML树形结构,灵活方便地实现了安全标记与数据客体的绑定。该方法不仅能够实现安全标记与数据客体绑定方法的统一,而且可以提高安全标记绑定

到稿日期:2012-10-18 返修日期:2013-03-03 本文受国家863高技术研究发展计划项目(2009AA01Z438),国家973计划前期研究专项(2011CB311801),河南省杰出科技创新人才计划(114200510001h)资助。

曹利峰(1981-),男,博士生,讲师,主要研究方向为网络安全,E-mail:caolf302@sina.com;李 中(1969-),男,副教授,主要研究方向为网络安全;陈性元(1963-),男,博士,教授,博士生导师,主要研究方向为信息安全;冯 瑜(1989-),女,硕士生,主要研究方向为网络安全。

的灵活性、可用性以及数据客体的利用率。

2 XML 客体安全标记模型

2.1 XML 客体安全标记

通常情况下，安全标记包括安全级别与范畴。为便于数据客体与安全标记的绑定，本文对客体安全标记进行了新的定义。

定义 1 XML 客体安全标记是一个八元组：

$XOLabel=(N_{rs},D_{unit},D_{elem},D_{attr},CL,CK,Subf,Lfun)$

其中，N_{rs} 为客体资源索引标识，由客体名、资源标识符等组成。资源标识符在同一领域内是唯一的。

D_{unit} 为数据单元集，一个客体资源由若干数据单元组成，即 $O=\sum d_i, d_i \in D_{unit}$。

D_{elem} 为数据元素集，一个数据单元由数据元素组成，即 $d_i=\sum el_k, el_k \in D_{elem}$。

D_{attr} 为数据属性集，属性是元素自身固有的性质，是元素在某一方面质的表现。一个元素通常可表现出多种属性，即 $el_k=(att_1,\cdots,att_i), att_i \in D_{attr}$。

CL 为安全级集合，用于表示主体的安全等级以及客体的敏感程度；CK 为范畴集，是指主体能够活动的领域，或者指的是客体作用的领域。

$Subf$ 是一个二元关系，若 $d_i, d_j \in D_{unit}$，则 $Subf(d_j, d_i)$ 表示数据单元 d_j 是数据单元 d_i 的子单元；若 $el_i, el_j \in D_{elem}$，则 $Subf(el_i, el_j)$ 表示元素 el_i 是元素 el_j 的子元素。

$Lfun$ 是一个二元关系，令 $Tag=(CL,CK)$，Tag 是安全标签，为安全级别和范畴的二元对，$Lfun \subseteq Tag \times (D_{unit} \cup D_{elem} \cup D_{attr})$，若 $d_i \in D_{unit}$，或 $el_i \in D_{elem}$，或 $att_i \in D_{attr}$，那么 (tag_i, d_i)、(tag_j, el_j)、(tag_k, att_k) 分别表示 tag_i 为数据单元 d_i 的安全标签、tag_j 为数据元素 el_k 的安全标签、tag_k 为数据属性 att_k 的安全标签。

2.2 XML 安全标记约束规则

客体安全标记实施时，应遵循以下原则。

规则 1 XML 客体安全标记中，若 $d_i, d_j \in D_{unit}$、$el_i, el_j \in D_{elem}$，且存在 $Subf(d_j, d_i)$、$Subf(el_i, el_j)$，则有 $CL(d_i) \leqslant CL(d_j)$、$CL(el_i) \leqslant CL(el_j)$。

规则 1 说明：若数据单元或数据元素间存在父子/祖先二元关系，则其子单元或子元素安全级别大于等于父单元或父元素。

规则 2 XML 客体安全标记中，若 $d_i \in D_{unit}$、$el_i \in D_{elem}$，且存在 $Subf(el_i, d_i)$，则有 $CL(d_i) \leqslant CL(el_i)$。

规则 2 说明：若 el_i 是数据单元 d_i 的元素，则数据元素 el_i 的安全级别大于或等于数据单元 d_i 的安全级别。

规则 3 XML 安全标记中，若 $el_i \in D_{elem}$、$att_i \in D_{attr}$，且存在 $Subf(attr_i, el_i)$，则有 $CL(el_i) \leqslant CL(attr_i)$。

规则 3 说明：若 att_i 是数据元素 el_i 的属性，则属性 att_i 的安全级别大于或等于数据元素 el_i 的安全级别。

规则 4 XML 安全标记中，若 $att_i, att_j \in D_{attr}$，且有 $att_i \rightarrow att_j$，则必有 $CL(att_j)=CL(att_i)$。

规则 4 说明：若已知属性 att_i，可以推出属性 att_j，则 att_i 的安全级别必须等于 att_j 的安全级别。否则，若 $CL(att_j) > CL(att_i)$，则必然引起信息的泄漏。

规则 5 XML 安全标记中，若 $att_i, att_j, att_k \in D_{attr}$，$att_i, att_j \rightarrow att_k$，$att_i \not\rightarrow att_k$，$att_j \not\rightarrow att_k$，则有 $CL(att_k) \geqslant \max(CL(att_i), CL(att_j))$。

规则 5 说明：若已知属性 att_i、att_j，可以推出属性 att_k，则 att_k 安全级别必定要大于 att_i、att_j 安全级别。

规则 6 若 $att_i, att_j \rightarrow att_k$，主体 $CL(S) \geqslant CL(att_i)$、$CL(S) \geqslant CL(att_j)$，且 $CL(S) < CL(att_k)$，则 S 对 att_i、att_j 的访问是互斥。

规则 6 说明：当已知属性 att_i、att_j 可以推出属性 att_k 时，若主体 S 的安全级别小于属性 att_k 的安全级别，且 S 的安全级别不小于 att_i、att_j 的安全级别，则 S 对属性 att_i、att_j 的访问是互斥的。

3 安全标记与客体绑定方法

3.1 安全标记绑定基本思想

数据客体，按照数据结构类型的不同，可分为结构化、半结构化以及非结构化数据等[5]。尽管客体类型较为复杂，但是依据其内部数据关系，客体均能采用数据树进行表示[5,6]。因此，本文依据典型的树形结构 XML，通过客体数据树的遍历，有效地实现了数据客体与 XML 安全标记的绑定。

3.2 数据客体逻辑分割

为将数据客体表示为树形结构，首先必须要对数据客体进行逻辑的分割，每一部分为一个独立单元，作为树的节点。结构化数据可采用二维表进行逻辑表达，通常存储在关系数据库中。数据库中的表，从本质上讲，都可以表达成深度为 2 的树形结构，表名为树根，表项为叶子节点；而半结构化数据 XML 文档本身就是一个典型的逻辑树形结构，XML 根元素为树根，其他元素、属性为树节点，因此，在数据客体的逻辑分割中，本文仅仅讨论非结构化数据。非结构化数据，是相对于结构化数据而言，不便于用二维表存储的数据。比较典型的代表有文本文件、视频、图片等。

(1)文本文件。文本文件是具有层次化结构的，因此，其分割与逻辑表示原则为：

• 若文本文件具有目录结构，则按照其结构构建逻辑树，文本标题为树根，其他为节点；若无目录结构，则按照文本内容的重要程度，划分为若干具有完备语义的片段，文本标题为树根，片段为树的节点。

• 若节点内容可继续分割为相互独立的片段，则可将片段作为该节点的子节点。节点内容无法分割时，则将节点属性作为该节点的孩子节点。

• 为表示方便，可将文本中的图像、表格等非结构化数据视为整体进行处理，作为某一节点的子节点。

• 具有偏序蕴含关系的节点遵循 XML 安全标记约束规则。

(2)视频。视频是一帧、一帧组成的，帧是按照秒进行划分的，某段连续的帧构成一个视频的画面。视频分割与逻辑表示原则为：

• 视频中的帧映射到数据树的节点，帧名为节点名，节点类型为元素。

• 连续帧构成数据树的子树，连续帧之间是一种序列关系。

• 若存在若干画面可推导出下一个画面，则以画面作为节点，形成具有推导关系的子树。节点类型为元素。

(3)图像。图像包括像素图像和矢量图像,像素图像是由众多像素所组成的,而矢量图像则是带有方位位置的像素图像。因此,本文将组成一个可识别图像单元的像素集合,称之为图像节点,一个图像由若干节点组成。图像分割与逻辑表示原则为:

• 图像节点映射为数据树节点,图像节点名为数据树节点名,节点类型为元素。

• 图像进行分隔后,依据图像特点,选取合理的图像节点作为树的根节点,其他图像节点则按照节点间的相对位置,映射为树的其他节点。

• 图像节点间关系,映射为树节点之间关系。图像节点之间的关系通常为并列对等关系,因此,尽管树呈现的是层次化的节点关系,但是树节点关系应与图像节点间关系保持一致。

3.3 安全标记与客体绑定算法

在数据客体分割的基础上,本文给出了数据客体与安全标记的绑定算法(Algorithm of Binding Secure Label to Data Object,ABSLDO),基本思想为:通过树的前序遍历,采用人工制定安全级别和范畴,为客体中的数据单元、元素、属性分配安全标签,并在树遍历与安全标签分配中,转换节点名到XML标签,从而生成客体安全标记映像。ABSLDO算法如表1所列。

表1 数据客体与安全标记绑定算法ABSLDO

```
Input:tr(trees which belongs to the same object);
      Stag(set of tags that belongs to node of tr)
Output:xsldoc(xml doc about secure label)
(1)dt_preorder(dt){
(2)  if(dt<>null)then{
(3)    xml=xml_createl(dt. name);
(4)    dt_preorder(dt->lchild);//left child tree
(5)    dt_preorder(dt->rchild);//right child tree
(6)  }
(7)  xml=xml_create2();
(8)  return xml;
(9)}
(10)xml_createl(name){
(11)write <name> to xml file;
(12)write <secattr> to xml file;
(13)write <level> level. value</level> to xml file;
(14)write<domain>domain. value</domain> to xmlfile;
(15)write </secattr> to xml file;
(16)push </name> stack;// enter a stack;
(17)}
(18)xml_create2(){
(19)  while(stack<>null){
(20)  pull from stack;// emerge from a stack;
(21)  write</name> to xml file;
(22)  }
(23)}
(24)n=|tr|;// the number of trees
(25)for i=1 to n do
(26)begin
(27)  xsldoc[]=dt_preorder(tr);
(28)endfor
(29)return xsldoc[];
```

算法主要分为2大部分,第1部分(1-9行)为树的前序遍历,遍历树中每一个节点,为安全标签的分配奠定基础;第2部分(10-23行)为采用入栈/出栈方法,为客体树节点分配安全标签,创建XML安全标记文档。算法的时间复杂度与树的节点数有关,为O(n)。

下面给出一个通过ABSLDO绑定算法所形成的XML安全标记文档的例子,其树形结构如图1所示。

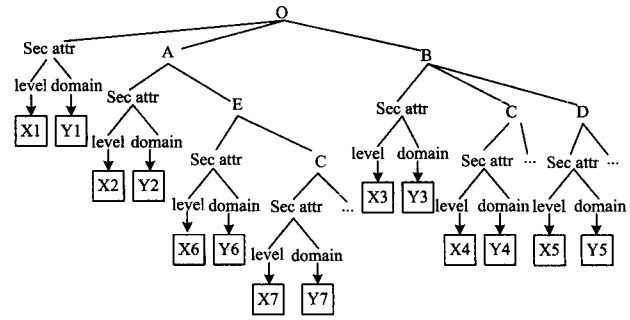

图1 XML客体安全标记树形结构

4 XOLabel相关操作

XML安全标记操作包括安全标记查询、删除、更新以及客体片段提取等,其中标记查询与客体片段提取是多级网络客体操作的关键。安全标记查询用于判断主体对客体的访问权限,而客体片段提取则依据安全标记返回主体能够访问客体的具体内容。本文重点讨论这两个操作。

4.1 XOLabel查询

针对XML安全标记的查询,本文基于XQuery查询语言给出了XOLabel的查询模式。

定义2(路径表达式) 令 $p=\{v,\{a_1,\cdots,a_l\}\}$代表一个节点及其孩子节点,$a_i$ 不仅指节点 v 的属性节点,也包括其下级节点。路径表示式定义为:

(1)p 是一个路径表示式;

(2)若 $p_i(i=1,\cdots,j)$是一个路径表达式,则$\{p_1\rightarrow p_2\rightarrow\cdots\rightarrow p_j\}$也是一个路径表示式;

(3)$*$代表任意的路径表达式,则$\{*\rightarrow p_i\rightarrow\cdots\rightarrow p_k\}$、$\{p_1\rightarrow p_2\rightarrow *\rightarrow p_k\}$均称为一个路径表达式。

定义3(XML查询) 一个XML查询 Q 遵循如下形式:

(1)FOR $\$var$ in p_1:路径 p_1 中任意的查询变量。

(2)LET $\$var=p_k$:为查询变量赋值。

(3)WHERE condition clause:具有前提假设的条件查询语句。

(4)ORDER BY clause:按照条件进行排序。

(5)RETURN p_j:返回查询值 $p_j=\{v,\{\overline{\$var_1},\cdots,\overline{\$var_m}\}\}$。

其中,$\$var$ 为查询变量;p_i 为路径表达式;$\overline{\$var_i}$也为路径表达式,即$\{\$var_i\rightarrow p_l\}$,$p_l$ 为路径表达式。

一般来说,XQuery的查询变量也称为查询项,查询项主要分为①仅含路径信息的查询,如图2中的查询项$\$s$;②仅含关键字查询,如图2中的查询项$\$t$;③既含路径又含关键字的查询,如图2中的查询项$\$m$。下列的查询表达式均基于图1所描述的安全标记。

```
FOR $s in doc /O//D/secattr /level   FOR $s in doc /O//D/secattr /level   FOR $m in doc/O/A/E/F
LET $x=$s-1                          WHERE contain ($t,"X3")               WHERE contain ($m,"X8")
RETURN <result> {$x}</result>        RETURN <result> {$t}</result>         RETURN <result> {$x}</result>
```

图2 XQuery的查询方式

但是,通常情况下,在编写XQuery查询表达式时,用户仅仅能够了解部分的路径,如图3所示,而且XML安全标记

文档中，数据单元、元素、属性等名称存在着一致性，使得XML文档的查询存在着不确定性，很难明确用户的查询意图，容易造成无意义的查询结果。

```
FOR $a in doc//secattr/level                FOR $a in doc//C/secattr/level
    $b in doc//secatrr/domain                   $b in doc//C/secatrr/domain
WHERE $a/text()="X5"                        WHERE $a/text()="X5"
RETURN <result>{ $a ,$b}</result>           RETURN <result>{ $a ,$b}</result>
          (a)                                         (b)
```

图3　不完全路径下XML文档查询

图3为不完全路径下对XML安全标记的查询。依据XQuery查询机制，通过图3(a)的查询表达式获得的查询结果为〈X5,Y1〉、〈X5,Y2〉、〈X5,Y3〉、〈X5,Y4〉、〈X5,Y5〉、〈X5,Y6〉、〈X5,Y7〉、〈X5,Y8〉；通过图3(b)的查询表达式获得的查询结果为〈X5,Y5〉、〈X5,Y7〉。从这些查询结果来看，〈X5,Y1〉、〈X5,Y2〉、〈X5,Y3〉、〈X5,Y4〉、〈X5,Y6〉、〈X5,Y7〉、〈X5,Y8〉、〈X5,Y7〉均是无效的。显然，在用户不完全知道XML文档结构的情况下，用户的查询意图很难得到明确，这也将会影响查询的有效性、准确性。因此，本文为了保证XOLabel查询的有效性，给出了符合XOLabel查询的有意义判断原则，使得安全标记查询更加合理，进而防止安全标记查询错误导致的泄密以及无法访问问题。

定义4(XOLabel有意义性查询判别原则)　给定一个XOLabel的查询Q，设r_Q为查询结果，$r_Q=\{\langle v_i,v_j\rangle\}$，若$r_Q$为有意义的，当且仅当对于任意的$v_i$和$v_j$，有：

(1)v_i和v_j必为叶子节点，且$v_i\neq v_j$。

(2)若v_k是v_i的父节点，v_k'是v_j的父节点，则必有$v_k=v_k'$，即v_i和v_j有共同的父节点。

(3)v_i和v_j的父节点必为XOLabel所对应的客体数据树中节点的孩子节点。

表2　XOLabel的查询算法

```
input:Q={t1,t2,…,tn};//ti is Query item
      XOLabel;//XOLabelis XML document queried
Output:R(result of query)
(1)for i=1 to n do
(2)begin
(3)  if ti only contains path information then {
(4)    find all paths from XML document by ti;
(5)  for each p do{// p is a path
(6)    append text of leaf at the end of p into QRi[];}}
(7)  if ti only contains keyword information then {
(8)    find all nodes from XML document by keyword;
(9)    for each n do{
(10)     append text of n into QRi[];}}
(11)  if ti contains not only keyword but also path then {
(12)    find all nodes from XML document by keyword;
(13)    find all paths from XML document by ti;
(14)    for each p do { //p is path;n is node
(15)      filter the node set of p by n;
(16)  append text of leaf node at the end of p into QRi[];}}
(17)endfor
(18)  TR[]=QR1[]×QR2[]×…×QRn[];
(19)for k=1 to |TR| do
(20)  begin
(21)  v1,v2,…,vn∈TRi;
(22)  if vi==vj or parent(vi)≠parent(vj)then
(23)    del TRi;
(24)endfor
(25)append TR to R;
(26)return R;
```

依据XQuery查询方法以及XOLabel有意义性查询判断原则，给出了XOLabel的查询算法(Algorithm about Query of XML Secure Label，AQXSL)，如表2所列。该算法的主要思想为依据查询项的查询方式分别进行查询，然后，对查询的结果进行有意义性的判断，最终返回要查询的结果。

4.2　数据客体片段的提取

主体安全标记不同，能够获取的数据客体信息也不同，因此，本文依据上述安全标记规则，设计了数据客体片段的提取算法。以图4为例来进行阐述。

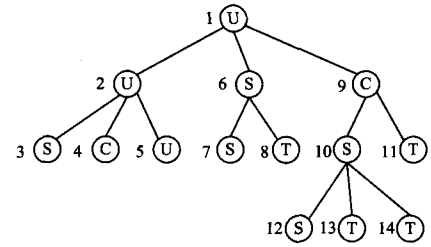

图4　多级数据客体XML组织方式

图4是一个多级XML数据客体文档，依据主体安全标记获取的数据片段不一样。其提取过程如表3所列。

表3　客体数据片段提取算法

```
Input:sub_label,Mls_xdoc;//sub_label is label of subject
Output:xdoc_frag;
(1)dt=bi_dt(Mls_xdoc);//转换为二叉树
(2)nd=dt.root;
(3)xdoc_frag=dt;
(4)sl=AQXSL(xdoc_frag/../nd);
(5)if(sl>sub_label){//删除所有高密级节点
(6)  if nd==dt.root then{
(7)      return null;
(8)    }else{
(9)      nd=null;
(10)   }
(11)  }else{
(12)  if nd.lchild()null then{
(13)          nd=nd.lchild;
(14)          goto 4;
(15)  }
(16)  if nd.rchild()null then{
(17)      nd=nd.rchild;
(18)      goto 4;
(19)   }
(20)  }
(21)return xdoc_frag;
```

通过对多级XML文档实施提取操作后，可获得与主体级别对应的数据客体片段，如图5所示，不同片段中所含有信息内容的敏感程度不相同，能够提供给不同级别的主体。

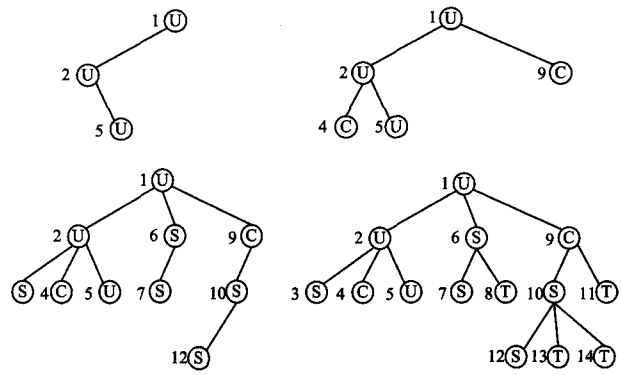

图5　各个敏感级别的文档片段

5 XML 安全标记绑定实例

为说明上述数据客体与安全标记绑定的有效性、可用性，本文以文本文档为例阐述了其绑定过程。

假定客体 o 为文本文档，按照其目录结构进行逻辑分割，形成的树形关系如图 6 所示，每一部分都由独立可读的具有不同安全级别的内容所组成。

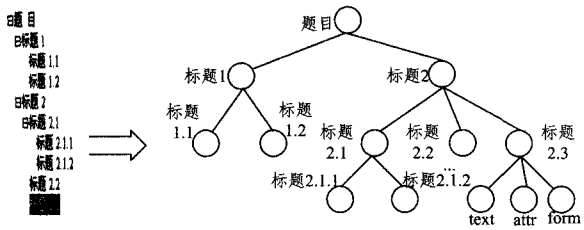

图 6 客体 o 逻辑分割后的树形结构示意

依据安全标记绑定算法 ABSLDO 遍历客体 o 的树形节点，其节点顺序为“题目、标题 1、标题 1.1、标题 1.2、标题 2、标题 2.1 标题 2.1.1、标题 2.1.2、标题 2.2、标题 2.3、text、attr、form”，若为其分配的安全标记分别为〈(U, $D1$)、(U, $D1$)、(U, $D1$)、(U, $D1$)、(C, $D2$)、(C, $D2$)、(C, $D2$)、(C, $D2$)、(C, $D3$)(S, $D2$)、(S, $D2$)、(S, $D2$)、(S, $D2$)〉，则由 ABSLDO 算法所生成的安全标记文档如图 7 所示。

```
〈题目〉
 〈secattr〉〈level〉U〈/level〉〈domain〉D1〈/domain〉〈/secattr〉
 〈标题 1〉
  〈secattr〉〈level〉U〈/level〉〈domain〉D1〈/domain〉〈/secattr〉
  〈标题 1.1〉
 〈secattr〉〈level〉U〈/level〉〈domain〉D1〈/domain〉〈/secattr〉
  〈/标题 1.1〉
  〈标题 1.2〉…〈/标题 1.2〉
 〈/标题 1〉
 〈标题 2〉
   〈secattr〉〈level〉C〈/level〉〈domain〉D2〈/domain〉〈/secattr〉
   〈标题 2.1〉〈secattr〉〈level〉C〈/level〉〈domain〉D2〈/domain〉〈/secattr〉
    …
   〈/标题 2.1〉
   〈标题 2.2〉…〈/标题 2.2〉
   〈标题 2.3〉
    〈secattr〉〈level〉S〈/level〉〈domain〉D2〈/domain〉〈/secattr〉
  〈text〉
     〈secattr〉〈level〉S〈/level〉〈domain〉D2〈/domain〉〈/secattr〉
     〈/text〉
     〈attr〉…〈/attr〉
     〈form〉…〈/form〉
   〈/标题 2.3〉
    〈/标题 2〉
〈/题目〉
```

图 7 客体 o 的安全标记文档

客体 o 的安全标记文档树形结构如图 8 所示。

假定信息系统中，某一主体 s 要访问客体 o，其安全级别为秘密级(C)，则由 4.2 客体片段抽取算法可获得 s 访问的内容。首先对客体 o 树形节点进行二叉树转换，并保持其安全属性不发生变化，然后采用安全标记查询算法 AQXSL 进行节点安全属性查询，其查询项为：

```
FOR $a in xdoc_frag/../$nd //secattr/level
    $b in xdoc_frag/../$nd//secatrr/domain
WHERE o. node/text()="$nd"
RETURN 〈result〉{$a ,$b}〈/result〉
```

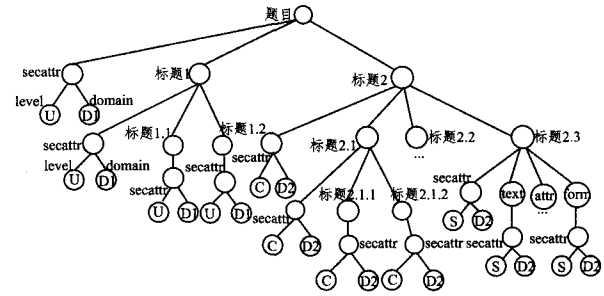

图 8 客体 o 安全标记文档的树形结构

通过返回的安全属性值，判断主体是否可以进行访问，若 s 安全级别(以安全标记中的级别为例)小于此节点安全级别，则置其为空，直至所有节点遍历完毕。最后返回 s 可获得的客体片段，如图 9 所示。

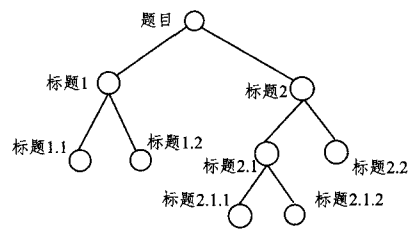

图 9 主体 s 可获取的客体片段

6 基于 XML 安全标记多级安全通信机制

在 XML 安全标记绑定实例的基础之上，本文给出了基于 XML 安全标记的多级安全通信机制，如图 10 所示。

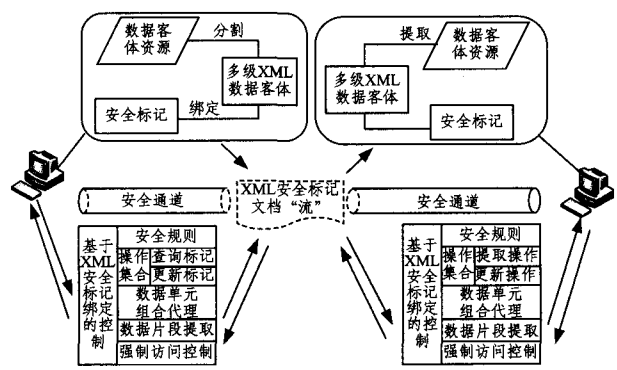

图 10 基于 XML 安全标记绑定的多级安全通信

从图 10 可以看出，基于 XML 安全标记的多级安全通信主要由安全标记绑定、安全标记查询、客体片段提取、客体数据单元组合、强制访问控制以及安全通道处理等组成。其中，安全标记绑定完成客体的逻辑分割、分配安全标记等功能；安全标记查询依据客体节点内容查询其安全标记，用于强制访问控制；客体片段提取则依据主体安全标记提取其能够获知的客体片段；客体数据单元组合则是依据客体类型将客体数据单元进行组合，形成可识别的内容；强制访问控制依据安全标记限定对客体的访问；安全通道处理则对数据进行封装、加密、认证等安全处理，确保通信双方传输数据的安全性。其执

(下转第 145 页)

[3] Wong R, Li J, Fu A, et al. (α,k)-anonymous data publishing[J]. Journal of Intelligent Information Systems, 2009, 33, (2): 209-234

[4] Ninghui L, Tiancheng L, Venkatasubramanian S. t-Closeness: Privacy beyond k-anonymity and l-diversity[C]// Proceedings of the 23rd International Conference on Data Engineering. Inst. of Elec. and Elec. Eng. Computer Society, Istanbul, Turkey, 2007: 106-115

[5] Lefevre K, Dewitt D J, Ramakrishnan R. Incognito: Efficient full-domain k-anonymity [C] // ACM SIGMOD International Conference on Management of Data. United states. Association for Computing Machinery, Baltimore, Maryland, 2005: 49-60

[6] Kabir M E, Wang H, Bertino E. Efficient systematic clustering method for k-anonymization [J]. Acta Informatica, 2011, 48, (1): 51-66

[7] Aggarwal G, Panigrahy R, Tom, et al. Achieving anonymity via clustering [J]. ACM Trans. Algorithms, 2010, 6(3): 1-19

[8] 王智慧,许俭,汪卫,等. 一种基于聚类的数据匿名方法[J]. 软件学报,2010,21(04):680-693

[9] Kenig B, Tassa T. A practical approximation algorithm for optimal k-anonymity[J]. Data Mining and Knowledge Discovery, 2012, 25(1): 134-168

[10] Ni W, Chong Z. Clustering-oriented privacy-preserving data publishing[J]. Knowledge-Based Systems, 2012, 35: 264-270

[11] Sweeney L. k-anonymity: A model for protecting privacy[J]. International Journal of Uncertainty Fuzziness and Knowledge-Based Systems, 2002, 10(5): 557-570

[12] Xu J, Wang W, Pei J, et al. Utility-based anonymization using local recoding [C] // Philadelphia, PA, USA. Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. USA: ACM, 2006: 785-790

[13] Li C, Biswas G. Unsupervised learning with mixed numeric and nominal data[J]. IEEE Transactions on Knowledge and Data Engineering, 2002, 14(4): 673-690

[14] Meyerson A, Williams R. On the complexity of optimal k-anonymity [C]// Proceedings of the twenty-third ACM SIGMOD-SIGACT- SIGART symposium on Principles of database systems ACM. 2004: 223-228

[15] Xiao X, Yi K, Tao Y. The hardness and approximation algorithms for L-diversity[C] // 13th International Conference on Extending Database Technology: Advances in Database Technology. Association for Computing Machinery, Lausanne, Switzerland, 2010: 135-146

[16] Ghinita G, Karras P, Kalnis P, et al. A framework for efficient data anonymization under privacy and accuracy constraints[J]. ACM Transactions on Database Systems, 2009, 34(2)

(上接第 128 页)

行步骤如下:

(1)客体资源 o 进行逻辑分割,形成 XML 数据树,树中每个节点为客体中独立的数据单元。

(2)依据 XML 安全标记绑定算法,将安全标记与数据客体进行绑定,形成多级 XML 数据客体。

(3)当某 s 访问数据客体 o 时,将 o 的多级 XML 数据客体通过安全通道发送给 s。安全通道是依据通信双方安全标记协商的具有安全级别的逻辑通道,通过该通道可保证传输数据的机密性、完整性。

(4)当数据客体到达 s 后,依据 s 安全标记对其访问的数据客体进行片段抽取,获得 s 能够访问的数据单元集。

(5)依据客体 o 的类型以及 o 中数据单元之间的关系,对数据单元进行组合,使得 o 相对于主体 s 来说是可读的。同时,s 保存 o 到本地数据库中。至此,完成了主体对客体的安全访问。

结束语 安全标记与数据客体的绑定,是等级保护网络中数据安全共享的关键。本文通过采用 XML 方式,有效地实现了网络数据客体的统一表示,定义了 XML 安全标记,合理巧妙地完成了数据客体与安全标记的绑定。该方法不仅提高了安全标记绑定的灵活性,实现了数据客体与安全标记绑定的统一,而且能够实施更为细粒度的访问控制。同时,数据客体的统一表示与 XML 客体安全标记,也能够解决多级信息系统间异构数据交换访问控制难的问题。当然,安全标记绑定还有许多方面待于研究,比如进程与安全标记、数据流与安全标记的绑定等,今后我们将针对这些方面做进一步的研究。

参 考 文 献

[1] GB/T 22239-2008. 信息安全技术信息系统安全等级保护基本要求[S]. 中国国家标准化管理委员会,2008

[2] Bell P D E, Padula L J L. Secure computer system: unified exposition and multics interpretation [R]. ESD-TR-75-306. MTR 2997 Rev. 1, The MITRE Corporation, 1976

[3] 季庆光,卿斯汉,等. 一个改进的可动态调节的机密性策略模型[J]. 软件学报,2004,15(10):1547-1557

[4] 何建波,卿斯汉,等. 对两个改进的 BLP 模型分析[J]. 软件学报,2007,18(6):1501-1509

[5] Peng P C, Rohatgi P, Keser C. Fuzzy multi-level security: an experiment on quantified risk-adaptive access control[C] // IEEE Symposium on Security and Privacy. Oakland, CA, May 2007: 222-230

[6] Magnani M, Montesi D. A Unified Approach to Structured, Semistructured and Unstructured Data[R]. UBLCS- 2004-9. University of Bologna, 2004

[7] Lee T Y. Formalisms on Semi-structured and Unstructured Data Schema Computations [D]. University of Hong Kong, Hong Kong Special Administrative Region, 2010

[8] 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

[9] Oudkerk S. A Proposal for an XML Confidentiality Label and Related Binding of Metadata to Data Objects [R]. RTO-MP-IST-091 -22. NATO C3 Agency. 2010

[10] Blazic A J, Saljic S. Confidentiality Labeling Using Structured Data Types[C]// 2010 Fourth International Conferences on Digital Society. ST, Maarten, Feb. 2010: 182-187

[11] Pernul G, Winiwarter W, Tjoa A M. The entity-relationship model for multilevel security[C]// Proceedings of the 12th international conference on the entity-relationship approach: entity-relationship approach. Arlington, Texas, USA, December 1994: 166-177