

文章编号:1001-9081(2010)04-1099-04

# XML 加密数据查询方法的研究与设计

柯于义, 夏士雄, 汪楚娇

(中国矿业大学 计算机科学与技术学院, 江苏 徐州 221116)

(jxhlskyy@163.com)

**摘要:**充分利用 XML 数据库文档的结构特性,结合 Dewey 编码的编码原理,设计了一种数据服务(DAS)模式下的 XML 加密数据的查询算法(ILISA)。将树型结构上的数据检索变换为顺序链表的数据检索,应用插值搜索算法替代深度与广度优先遍历,带来了良好的时间复杂性。设计了一种 XML 索引表数据结构,使得检索空间大幅缩减。最后给出 ILISA 的复杂性描述,证明了该算法具有良好的效果。

**关键词:**数据服务;XML;加密;数据查询;电子商务

**中图分类号:** TP311 **文献标志码:** A

## Research and design of querying approach for XML encrypted data

KE Yu-yi, XIA Shi-xiong, WANG Chu-jiao

(School of Computer Science and Technology, China University of Mining and Technology, Xuzhou Jiangsu 221116, China)

**Abstract:** The structural characteristics of XML database document have been fully made use of and combined with the encoding theory of Dewey code to design an encrypted XML algorithm called Index Layered Itersection Scan Algorithm (ILISA) under the model of Database as A Service (DAS). The data retrieval of tree structure was turned into the data retrieval of sequential list form. The authors applied the interpolation search algorithm instead of depth-first or breadth-first search, which had brought a good time complexity. And a data structure of XML index table was designed to reduce the search space greatly. Finally, the complexity of ILISA was described and the efficiency of ILISA was proved.

**Key words:** Database as A Service (DAS); XML; encryption; data querying; E-commerce

## 0 引言

近年来,随着网络与计算机技术的快速发展,XML 应用得也越广泛。随着面向服务的体系架构的数据服务(Database as A Service, DAS)模式的产生,它为数据拥有者提供完善的创建、存储和访问数据的服务<sup>[1]</sup>,已被应用于关系数据库系统和 XML 数据库系统等领域<sup>[2]</sup>。因为在 DAS 模式下,存储在数据服务商的客户端数据中有些是敏感的,这在客户端的安全范围之外,所以需要采取措施以保护数据不被滥用<sup>[3]</sup>。将数据存储在数据服务商,而通常第三方是不被完全信任的,故常用的解决方案是数据加密。这意味着,DAS 模式存储在数据服务商的数据是加密后的数据。另一方面,近年的网络与电子商务的快速发展,也带来了一定的安全性威胁,如:网络传输的信息易被窃听、截取、篡改。所以在网络上传输的数据也必须经过加密,以抵御此类安全性威胁。

既然数据加密已经成为现代电子商务的必要手段,那么对加密后数据的操作也成为必然。DAS 模式下,最为普遍的一项操作是对存储在数据服务商加密数据的查询。因此,研究加密数据查询对网络、电子商务等技术的发展有极重要的作用。

## 1 相关技术

### 1.1 XML 加密

网络信息数据交换主要涉及的安全性问题包括:保密性、

完整性、认证性、不可抵赖性以及访问控制。其中保密性是指隐藏部分或全部的通信信息,防止未授权的用户窃取数据。XML 加密可以选择性地采取不同的密钥、不同的加密算法进行加密。加密语法的核心是 EncryptedData 元素,该元素从 EncryptedType 抽象类型派生,加密的数据可以是任意数据。根据需要,XML 文档的加密和解密可以有三种不同的粒度:整个 XML 文档的加解密、基于 XML 文档元素的加解密以及 XML 文档元素内容的加解密。该模式下通常使用基于文档元素的加密。

### 1.2 Dewey 编码

Dewey 码是基于 XML 文档的树型结构以及广度优先遍历所得的一种编码。Dewey 编码的定义:根节点  $r$  的 Dewey 编码为“0”,在宽度优先遍历过程中,如果节点  $v$  是节点  $u$  的第  $i$  个子节点,那么,节点  $v$  的 Dewey 码为  $D(u).i-1$ ,其中的  $D(u)$  表示节点  $u$  的 Dewey 码<sup>[4]</sup>。该编码符合人们的思维习惯,也比较适合于描述树型结构的节点,使用起来比较容易被接受。

### 1.3 XML 索引表

直接对源文档的遍历完成对 XML 数据的查询会耗费大量的时间,故该搜索方式不可行。若要实现高效的查询,则必须建立一套索引机制来支持查询的实现。近些年来,研究者们提出了多种索引方法来支持 XML 查询,如 XISS、XASR、Index Fabric、MPMGJN、Re-tree、PRIX、TIndex 等<sup>[5-10]</sup>。XML 的索引方法有树匹配、路径索引、串索引、本体索引等<sup>[11]</sup>。

**收稿日期:**2009-10-10;**修回日期:**2009-12-09。 **基金项目:**国家自然科学基金资助项目(50674086);高等学校博士学科点专项科研基金资助项目(20060290508);江苏省社会发展科技计划项目(BS2006002);中国矿业大学青年科研基金资助项目(2006A047)。

**作者简介:**柯于义(1984-),男,江西瑞昌人,硕士研究生,主要研究方向:XML 数据处理及其信息安全、计算机应用技术;夏士雄(1961-),男,黑龙江鹤岗人,教授,博士生导师,博士,主要研究方向:模式识别、人工智能;汪楚娇(1976-),女,江苏徐州人,讲师,博士,主要研究方向:人工智能、模式识别。

## 2 设计原理

在 DAS 模式下,XML 加密数据存储在第三方,即数据服务商。加密数据查询的主要工作在数据服务商处完成。本文设计了一种数据结构:基于节点索引的 XML 索引表,将用户标识(用户编号)与其对应的所有用户节点对应起来。每个用户编号对应一个或多个对应的用户子节点的 Dewey 编码,该数据结构及 XML 索引表结构如表 1 所示,其中第一行为“Final”索引项,它用来记录当前 XML 数据库文档中用户子树根节点 Dewey 编码的最大值。

**定义 1** 用户子树根节点与用户子树。对 XML 文档的树型结构而言,从某节点出发的所有后代节点均是某用户编号对应的属性、子孙元素或子孙元素的属性,则称该节点是该用户的用户子树根节点,以该节点为根的子树则称为该用户的用户子树。

性质 1 每个用户订单每次可以包含多个用户子树根节点。

性质 2 一个用户子树根节点只与一个用户编号对应,一个用户通常有多个用户子树根节点。

性质 3 根据 XML 文档极好的对称性,所有用户的用户子树根节点处于同一层次上,它们 Dewey 编码的长度也相同。

表 1 XML 索引表

用户编号	用户子树根节点的 Dewey 编码
Final	0.6
001	0.0,0.2,0.3
002	0.1
003	0.5
004	0.4,0.6
...	...

DAS 模式的电子商务网站系统中,客户端生成 XML 订单文档,并经过加密/签名后经由服务器交付给作为外部数据源的第三方——数据服务商来存储和管理。单个用户的一张订单往往在树型结构上是很小的,树的分支也较少。而在数据服务商处存储的 XML 数据库文档会集多个用户以及多次交易订单于一体,形成一棵非常“宽”而高度较低的树型结构的文档,因而直接搜索整棵树来寻找目标节点将会造成大量的浪费。另外,基于 XML 文档结构具有极好的对称性,对称性指的是属于同一级别的节点(如不同用户的子树根节点属同一级)处于整个文档树型结构的同一层次上,即它们互为兄弟节点的关系。

基于层次的交叉扫描算法(Layered Intersection Scan Algorithm, LISA)方法是基于最小子树根节点(Smallest Lowest Common Ancestor, SLCA)节点按“层”分布的规律,采取了逐层求解 SLCA 节点的思路,LISA 除了只需对应于关键字的节点集合信息以外,不再需要其他复杂的辅助数据结构<sup>[4]</sup>。

基于 XML 数据库文档这些特性,本文提出了一种结合 XML 索引表和 LISA 的搜索算法应用于 XML 加密数据查询,称为 ILISA(Index Layered Intersection Scan Algorithm)算法。XML 索引表数据结构的设计将原本在树型结构上进行的搜索转化为线性表的搜索,而且是有序线性表。

## 3 算法描述

ILISA 算法主要分为预置索引表、XML 索引表搜索、数据解密查询三个阶段,分别完成索引表的初始化,用户节点查找与结果文档构造,以及最终结果的生成。

### 3.1 预置索引表

预置索引表是进行 XML 加密数据查询的前提和基础。

客户端生成订单数据后,经加密/签名后由服务器交付给数据服务商。数据服务商则需对新增的订单文档进行整合,将其添加至 XML 数据库文件。根据 XML 的文档对称性,如果已知数据库文件中用户节点 Dewey 编码的最大者,则可以直接对新增的小文档进行 Dewey 编码,再将该小文档作为子树以最后一个孩子的形式添加到数据库文档中,同时更新 XML 索引表。XML 索引表采用有序线性表方式组织。

对索引表的更新方式如下:当数据服务商接收到来自服务器的 XML 订单数据时,数据服务商对 XML 索引表进行搜索,其目的是添加新的用户节点。该搜索过程可能会出现以下两种结果。

1) 当前 XML 索引表中存在当前用户的索引项时,算法描述如下(用户编号为 User\_No;新增文档为 New\_Doc;XML 数据库文档为 Old\_Doc):

```

(1) Find (Final): D(Final)
(2) While (have new nodes)
(3) {
(4)   Encoding(current sub-tree): D(Final + 1)
(5)   Add (D(Final + 1), Old_Doc)
(6)   RenewIndexTab (User_No, D(Final + 1))
(7)   Renew (Final, D(Final + 1))
(8)   Final = Final + 1
(9) }
```

首先:第(1)步是在 XML 索引表中获得当前 XML 数据库文档中最大的用户子树根节点的 Dewey 值;其次,对新增的 XML 小文档中所含的各用户子树根节点进行以下操作:第(4)步是对新增用户子树进行 Dewey 编码,将用户子树根节点编码为原文档最大 Dewey 值加 1,因为要将其添加到文档的最后;第(5)步是将用户子树根节点添加到 XML 数据库文档;第(6)步更新 XML 索引表中该用户编号对应的索引项;第(7)步更新 Final 索引项的数据域(即更新当前文档中用户子树根节点的最大 Dewey 码)的值;第(8)步用来循环递增,进行下一个新增节点操作。

2) XML 索引表中不存在当前用户的索引项时,其算法与上述第一种情况类似,只是将上述第(6)步改为:InsertIndexTab(NewNode (User\_No, D(Final + 1))),其目的是在 XML 索引表中添加该用户编号对应的索引项。

### 3.2 XML 索引表搜索

XML 加密数据查询分为全部查询和部分查询。

**定义 2** 全部查询和部分查询。全部查询指的是用户需要对用户子树根节点的所有属性、后代元素以及后代元素的属性进行查询,该查询返回的结果文档由完整的用户子树组成;部分查询是只需要对用户子树根节点的所有属性、后代元素以及后代元素属性中的部分内容进行查询,查询返回的结果文档是以用户子树根节点为根,和满足要求的子元素以及加密元素组成。

不难看出,全部查询比部分查询较为简单。全部查询可以得到比较完整的查询结果,可获得较多的信息,查询消耗稍小,但查询结果不够精确;部分查询则更具针对性、更附有语义性,能较为准确地查询到所需要信息,消耗稍大些。

该步骤主要是从 XML 索引表中查找出所有的用户子树根节点,并按查询条件进行筛选,将符合要求的节点组织在新建的 XML 文档中,作为查询结果返回给客户端。基于上述 XML 加密数据查询的分类,从部分查询来探讨。

通过 3.1 节的处理后,查询过程可以有这样两种方案:方案 1 是先在数据服务商处查询所有用户节点,但要对这些节点的子节点中的未加密部分进行条件匹配,这将删除一部分后代节点而保留符合条件的节点以及加密节点,将这些组织

到初步结果文档中,再在客户端完成初步结果文档的解密与条件匹配查询,称为 ILISA-1;方案2是先在数据服务商处查询所有用户节点,而不进行其他的条件匹配处理,将这些用户子树全部组织到初步结果文档中,再在客户端完成初步结果文档的解密与条件匹配查询,称为 ILISA-2。按不同方案进行处理,那么在3.3节中则有两种不同的处理过程。

ILISA-1 算法的主要步骤如下:

```
(1) Find( User_No ): DeweyList
(2) New result_Doc
(3) While ( Dewey is not null)
(4) {
(5)   New this_node = the nodewhose Dewey is Current Dewey
(6)   While( this_node has child_nodes)
(7)   {
(8)     If ( child_nodes that match to the query conditions except
         the encrypted ones)
(9)     { organized these nodes as a sub_tree}
(10)    Add( sub_tree, result_Doc)
(11)    The next node
(12)  }
(13)  Dewey = the next Dewey
(14) }
```

其中:第(1)行使用插值法对 XML 索引表进行搜索,获得用户的所有用户子树根节点所对应的 Dewey 码的值,存储于 DeweyList 中;第(2)行新建 XML 结果文档,用来存储查询结果;第(3)行对 DeweyList 进行遍历;第(5)行取 DeweyList 中当前节点的 Dewey 值;第(6)行是对该节点的所有后代节点进行遍历;第(8)行是在其所有后代子节点中(加密子节点除外)寻找符合条件的那些,此处应用 LISA 的分层思想;第(9)行是将上述符合条件的子节点与当前子树的根节点组织成为结果子树;第(10)行将上述子树添加到结果文档 result\_Doc 中。

ILISA-2 与 ILISA-1 第(6)到(11)行不同。ILISA-2 不需要对用户子树根节点下的后代节点进行判断,而是直接将整个子树添加到结果文档 result\_Doc 中,也就是说 ILISA-2 的第(6)到(11)行直接 Add( sub\_tree, result\_Doc)就够了,只是该 sub\_tree 不需要重新组织而就是用户子树本身。

数据服务商在执行上述查询后,无论是 ILISA-2 还是 ILISA-1,都返回查询的初步结果文档,而该文档中的加密元素还未解密,因为密钥对数据服务商这样的第三方来说是保密的,数据服务商无权解密。此两方案不同的是,ILISA-2 初步结果文档包含全部的用户子树节点,而 ILISA-1 只保留了满足查询条件的那些;相同的是,初步结果文档中每棵用户子树仍然包含它的加密子元素。

### 3.3 数据解密查询

数据服务商生成初步结果文档将返回至客户端,同时给服务器一个响应信号,此时服务器将与客户端进行会话以传送该用户的密钥。由于上述提到两种方案都会返回查询的初步结果文档,数据解密查询是客户端针对3.2节中的查询结果文档进行的处理,即为二次查询,将有不同的处理过程。

ILISA-1 的数据解密查询过程如下:

```
(1) Decrypt( key, result_Doc, element)
(2) New Final_Doc
(3) While( the sub_nodes that been decrypted)
(4) {
(5)   While( the sub_node has child_nodes)
(6)   {
(7)     If( child_nodes match to the query conditions)
(8)     { organized these nodes as a sub_tree }
(9)     Add( sub_tree, Final_Doc )
```

```
(10)   The next node
(11)   }
(12)   This_node = The next node of the sub_nodes
(13) }
```

第(1)行是用密钥 key 对 result\_Doc 中的 element 元素进行解密,生成解密文档;第(2)行新建 XML 结果文档,用来存储查询的最终结果;第(3)行对文档中所解密元素节点本身进行遍历;第(5)行是对解密元素的所有后代节点进行遍历;第(7)行是在解密元素的后代中找出符合条件的那些节点,该步骤可用到 LISA 分层比较的思想;第(8)行是将上述符合条件的子节点与当前子树的根节点组织成为结果子树;第(9)行将上述子树添加到最终结果文档 Final\_Doc 中。算法执行完毕后形成用户所需要的精确结果。

ILISA-2 与 ILISA-1 算法不同主要在于第(3)行与第(5)行。ILISA-2 第(3)行应为 While( the user\_nodes ),是对所有用户子树根节点进行遍历;第(5)行为 While( this user\_node has child\_nodes ),是对当前用户子树根节点的所有后代节点进行遍历。那么在 ILISA-2 的第(7)行到第(9)行的任务是对当前用户根节点的所有后代元素进行条件判断,并将符合条件的添加到结果文档中。

## 4 算法分析

本文通过设计如表1所示的 XML 索引表数据结构,将 XML 树型文档中的用户子树根节点存储到链表中,而且每次添加元素时考虑索引项“用户编号”之间的排列顺序,目的是将链表中元素进行排序。这样以来有如下好处:1)原本在树型结构上进行的用户子树根节点搜索用顺序链表搜索即可完成;2)可用顺序链表高效的搜索算法进行搜索,如插值法;3)XML数据库文档往往会非常大,而通过本文 XML 索引表的处理,将其搜索空间缩小了很多,因为 XML 数据库文档中每个用户编号通常对应多个用户子树,而在本文设计的 XML 索引表中每个用户编号对应一个索引项。对于表1对应的 XML 文档来说,其 XML 数据库所含用户子树根节点数目为  $n(n=7)$ ,而该索引表中索引项(除 Final 索引项外)仅包含  $m(m=4)$  项,通常情况下  $m$  比  $n$  要小得多。如果 XML 数据库文档中,从文档的根节点到用户子树根节点所在的层次之间边的数目为  $e(e$  为  $n$  数量级),那么树的深度或广度优先搜索时间复杂度为  $O(n+e)$ ,而用插值法搜索索引表的时间期望复杂度为  $O(\log(\log m))$ ,最坏时间复杂度为  $O(m)^{[12]}$ 。

相对于不使用 XML 索引表的方法来说,ILISA 增加了步骤:1)对 XML 索引表进行搜索,以便找到该用户编号对应的索引项应该插入的位置,属于预置索引表的步骤,所耗时间期望复杂度也为  $O(\log(\log m))$ ,最坏时间复杂度为  $O(m)$ ;2)搜索到用户子树节点所对应的 Dewey 值后,到 XML 数据库文档中寻找对应的节点 node,根据 Dewey 编码的特性包含树的层次及该层的第几个子节点,则可在树型文档中直接定位到该节点,因此所耗时间为  $O(1)$ 。

因此,ILISA 的期望时间复杂度为:

$$O(\log(\log m)) + O(\log(\log m)) + O(1) = O(\log(\log m))$$

最坏时间复杂度为:

$$O(m) + O(m) + O(1) = O(m)$$

上述是预置索引表的时间复杂性,它比起直接在树结构中搜索所耗时间  $O(n+e)$  要小得多。紧接着是在含有  $k$  个( $k$  为该用户所对应应用子树根节点的数目)用户子树根节点的树型结构空间里进行搜索,该搜索过程采取分层思想,时间复杂性同 LISA,故整个查询过程的时间需加上规模为  $k$  的 LISA

时间。

设计了 XML 索引表结构,故需要多增  $m+1$  个单元的存储空间。另外,在数据服务商处设置一个阈值,用于规定是否存储查询的初始结果,原理是当某用户查询的频率超过该值时,就存储其初始查询结果,那么它是利用一部分的空间代价来取代时间代价。基于现代的硬件发展,达到更好的时间复杂性比达到更好的空间复杂性要求更强烈。

## 5 结语

ILISA 充分考虑了 XML 数据库文档的结构:它是一棵非常“宽”而高度较低的树型结构的文档,而它具有极好的对称性。基于此类性质,本文设计的数据结构可以将树型结构数据的检索变换为顺序链表检索,这不但带来了更优的检索时间复杂性,而且也大大减小了数据的检索空间,即数据规模从  $n$  个用户子树的检索减少到  $m$  个索引项的检索,通常  $n$  比  $m$  大得多。另外,从 Dewey 编码的构造原理,添加用户子树到 XML 数据库文档时,只需要获取当前 XML 索引表中的 Final 数据域的值,对新增用户子树进行 Final 数据域值加 1 的 Dewey 编码,再将其添加到文档的最后,每次的 Dewey 编码空间将会很小。ILISA-1 与 ILISA-2 有相同的时间复杂性,而它们比起  $O(n+e)$  的时间复杂度要优异得多。

### 参考文献:

- [1] HACIGUMUS H, IYER B, MEHROTRA S. Providing database as a service [C]// Proceedings of the 18th International Conference on Data Engineering. Washington, DC: IEEE Computer Society, 2002: 29.
- [2] 孔令波,唐世渭,杨冬青,等. XML 数据索引技术[J]. 软件学报, 2005, 16(12): 2063-2079.

- [3] JAMMALAMADAKA R C, MEHROTRA S. Querying encrypted XML documents [C]// Proceedings of the 10th International Database Engineering and Applications Symposium. Washington, DC: IEEE Computer Society, 2006: 129-136.
- [4] 孔令波,唐世渭,杨冬青,等. XML 信息检索中最小子树根节点问题的分层算法[J]. 软件学报, 2007, 18(4): 919-932.
- [5] 王静,孟小峰,王宇,等. 以目标节点为导向的 XML 路径查询处理[J]. 软件学报, 2005, 16(5): 827-837.
- [6] 姚全珠,丁晓剑,任雪利. 一种新的基于 XML 的索引机制[J]. 计算机工程, 2006, 32(15): 15-17.
- [7] GRUST T. Accelerating XPath location steps [C]// Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data. New York: ACM, 2006: 4-6.
- [8] GRUST T, van KEULEN M, TEUBNER J. Accelerating XPath evaluation in any RDBMS [C]// ACM Transactions on Database Systems. New York: ACM, 2006: 91-131.
- [9] DeHAAN D, TOMAN D, CONSENS M P, et al. A comprehensive XQuery to SQL translation using dynamic interval encoding [C]// SIGMOD: Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data. San Diego: ACM Press, 2003: 623-634.
- [10] AMAGASA T, YOSHIKAWA M, UEMURA S. QRS: A robust numbering scheme for XML documents [C]// ICDE'03: Proceedings of 19th International Conference on Data Engineering. Washington, DC: IEEE Computer Society, 2003: 705-707.
- [11] RUNAPONGSA K. Methods for efficient storage and indexing in XML databases [D]. Ann Arbor: University of Michigan, 2004: 18-23.
- [12] 刘大有,唐海鹰,孙舒杨,等. 数据结构 [M]. 北京: 高等教育出版社, 2001.

(上接第 1098 页)

通过实验,本实例使用直接搜索与使用递进目标搜索传输到客户端的总的数据量比较如表 1 所示。

表 1 不同方式下的多媒体查询数据传输大小比较

搜索方式	说明	传输的 XML 文件大小/B		
		总尺寸	附加的二进制尺寸	MQF 尺寸
直接搜索	直接得到结果	114 743	97 979	16 764
递进目标搜索	结果集描述	6 805	—	6 805
	结果	19 186	12 163	7 023
	合计	25 991	12 163	13 828

从表 1 可以看出,在不使用递进目标搜索的情况下,向客户端传输文件的总尺寸为查询结果的二进制文件与 MQF XML 文件大小之和;而在使用递进目标搜索的情况下,传输的总尺寸为两块:一块为先返回的结果集描述数据;另一块则为最终的查询结果数据。两者比较可以看出,使用递进目标搜索的传输总量只有直接搜索的传输总量的 22.7%,传输的数据量下降了 78%,显然这将大大降低数据传输所需的带宽与速度,所以特别适合于各种移动终端使用。

## 4 结语

本文探讨的是一种基于内容的多媒体信息搜索的方法,通过将 MQF 与 FRU/FUU 相结合,通过从服务器返回结果集描述的元数据来实现递进目标搜索,而不需要返回结果集本身。这种方法特别适合于使用移动终端来进行多媒体数据库的搜索,从而大大降低了客户端与服务器的数据通信量。另外,只要采用一些标准的压缩算法(如 MPEG-7 中的 TeM 标

准)还可以进一步对传输的数据进行压缩,使得传输的数据量明显减少。

### 参考文献:

- [1] MARTÍNEZ J M, KOENEN R, PEREIRA F. MPEG-7: The generic multimedia content description standard, Part 1 [EB/OL]. [2009-06-10]. [http://www.chiariglione.org/mpeg/tutorials/papers/IEEEEMM\\_mp7overview\\_withcopyright.pdf](http://www.chiariglione.org/mpeg/tutorials/papers/IEEEEMM_mp7overview_withcopyright.pdf).
- [2] ZHAO L. DC 元数据介绍 [EB/OL]. [2009-05-10]. <http://www.dlresearch.cn/download/metadata/xiamen2.pdf>.
- [3] 曹忠升,吴宗大,王元珍. 多媒体查询语言及其评价准则 [J]. 计算机科学, 2009, 36(3): 9-13.
- [4] IBM, Watson Research Center. MARVEL: Multimedia analysis and retrieval system [EB/OL]. [2009-05-20]. <http://www.research.ibm.com/marvel/>.
- [5] Google. Google images [EB/OL]. [2009-04-10]. <http://www.google.com/imghp?hl=en&tab=wi&q>.
- [6] 李国辉,王辰,柳伟. 基于内容的多媒体数据库系统引擎 CDB [J]. 小型微型计算机系统, 2004, 25(7): 1113-1118.
- [7] 张正兰,李冬梅,赵春燕. 多媒体数据库查询技术研究 [J]. 微机发展, 2004, 14(3): 70-73.
- [8] ADISTAMBHA K, RITZ C, BURNETT I S. MQF: An XML based multimedia query format [C]// ICME 2007: Proceedings of the 2007 IEEE International Conference on Multimedia and Expo. Washington, DC: IEEE Computer Society, 2007: 264-267.
- [9] MCHROY M. Reverse polish notation [EB/OL]. [2009-05-18]. <http://mathworld.wolfram.com/ReversePolishNotation.html>.