安全云存储中高效的多关键词查找方案

李 倩 岳风顺 王国军
(中南大学信息科学与工程学院 长沙 410083)

摘 要 用户在租赁了云服务提供商(Cloud Service Provider,CSP)的存储空间之后,为了保护隐私,通常将文件以密文的形式存储在CSP上。这给在密文数据上进行关键词查找带来了问题。在本应用场景中,CSP被视为潜在攻击者。提出了支持多关键词查找的安全高效的云计算方案。该方案基于二叉排序树结构,在保证用户数据的安全和隐私的前提下,支持授权用户对文件的多关键词查询,让用户可以随时随地享受多关键词查询服务。

关键词 多关键词查找,二叉排序树,云服务提供商,云存储
中图法分类号 TP393.4 **文献标识码** A

Efficient Multi-keyword Search over Secure Cloud Storage

LI Qian YUE Feng-shun WANG Guo-jun
(School of Information Science and Engineering, Central South University, Changsha 410083, China)

Abstract To protect users' privacy, data stored in cloud service provider (CSP) usually needs to be encrypted before being sent to CSP. It brings about a problem that how users search files using keywords over encrypted cloud data. In lots of scenarios, CSP is considered as a potential attacker. According to characteristics of cloud computing, we proposed an efficient privacy-preserving approach to support multi-keyword search over encrypted data (short for PPMKS), which is based on binary sort tree for search (short for BSTS). In our PPMKS, authorized users can easily search ciphertext files using multiple keywords, which can make users enjoy the service of multi-keyword search over encrypted data anywhere and anytime.

Keywords Multi-keyword search, BSTS, CSP, Cloud storage

云计算是一种通过网络统一灵活调用云内各种ICT资源而实现高性能信息处理的服务方式,是网格计算、虚拟化、服务外包、托管等已有概念的整合与发展[1]。同时,云计算和其他技术及理论的有机结合,也是解决理论研究和实际应用的重要途径[2]。随着云计算技术发展越来越成熟,越来越多的用户将数据和信息存储在云服务器上。但是,在给用户带来便捷服务的同时,云计算的安全性和可靠性成为用户最关注的问题之一。用户使用云计算服务,等于是把自己的数据交给一家以盈利为目的的企业代为保管,存在数据泄露危险。由于云计算环境具有虚拟化、可扩展性、灵活性等特性[3],明文搜索不能直接应用到云计算环境中。为了保证云用户数据的安全,隐私数据在上传到云服务器之前需要加密。

用户可能仅对某部分数据而不是云端所有的数据感兴趣。用户可以通过关键词查找得到自己想要的相关文件。随着实际应用的发展,对云中的加密文件进行单关键词查找已经不能满足用户的需求,因此,引入多关键词查找变得非常迫切。多关键词查找可以让用户输入多个关键词对云中的加密文件进行搜索,也可以自定义关键词之间的与或逻辑关系。例如,用户可以搜索既包含关键词"information"又包含关键词"retrieval"的文件序列,也可以搜索包含关键词"cloud"或者包含关键词"keywords"的文件序列。引入多关键词查找,可以提高搜索结果的准确性,也更加符合用户的搜索习惯。

1 相关工作

本文主要研究如何对云中的加密数据进行安全高效的多关键词查找。目前,已存在的和本文相关的研究成果包括明文模糊关键词搜索、搜索加密、模糊关键词搜索等。

明文模糊关键词搜索。模糊搜索的重要性在信息明文检索中得到了关注。文献[4-6]中提出了明文模糊关键词搜索的解决方案,这些方案都是基于类似的字符串匹配的技术。在文献[4]中,作者提出了一个高效的合并算法,算法中应用了各种各样的索引技术。在文献[5]中,作者提出了一种新方案,该方案能够大大减少存储索引树的代价,索引树是为了相关字符串匹配而建立的。在文献[6]中,作者改善了支持交互性模糊明文字符串匹配的技术。但是,这些方案不能直接在加密搜索中应用,因为它们很容易受到字典和统计攻击,不能达到隐私保护查询的目的。

搜索加密。文献[7-12]中使用传统密码学技术实现了搜索加密。在这些研究中,大部分都集中在效率的提高和安全定义的形式化上。Song等人[8]最先提出了搜索加密模型。

到稿日期:2012-02-23 返修日期:2012-06-11 本文受国家自然科学基金(61073037),教育部博士点基金(20110162110043)资助。
李 倩(1987-),女,硕士生,CCF会员,主要研究方向为关键词搜索、云计算安全,E-mail:qianerfei@gmail.com;岳风顺(1985-),男,硕士生;王国军(1970-),男,教授,博士生导师。

在他们的方案中，文件的每个词都用一个特殊的二层加密结构进行单独加密[13]。Boneh 等人[10]提出了基于公钥加密的搜索加密方案。在该方案中，A 用户给云服务器发送一个公钥，允许云服务器通过已存在的关键词搜索 A 的数据条目，其中关键词用 A 的公钥进行了加密。Goh 等[9]提出了一个有效的安全索引模型 Z-IDX，该模型使用了伪随机函数和 Bloom 过滤器。一个 Bloom 过滤器包含了为每个文件所有关键词建立的陷门，它们都存储在云服务器上。检索词语时，用户将这个词语的陷门发送给服务器，服务器查找 Bloom 过滤器中是否包含所要查找词语的陷门，如果成功，则返回相关文件的标识符。Curmola[12]和 Chang[14]等人都提出了类似索引的方法，该方法为所有的文件建立了一个加密哈希表索引。在这个索引表中，每一项由一个关键词和包含这个关键词的相关文件的文件标识符的加密集合构成。文献[8]的作者给出了安全保障更严格的定义，并设法解决某些恶意的用户根据之前查询的结果调整现在的查询来窃取数据的问题。文献[9,15]的作者提出了对加密数据进行联合关键词查找、子集查询，以及归并查找的方法。但是，这些关键词查找技术的查询效率比较低下，不太适合运用到实际中来。

模糊关键词搜索。在文献[13]中，作者提出了一个模式，其允许对加密数据进行高效的模糊关键词查找。但是该方案需要存储所有的关键词索引，这种模式的存储量会随着加密文件集合的扩大而迅速增长。除此之外，该方案没有考虑到多关键词的搜索语法，比如，要搜索“information retrieval”相关的所有文件，需要转化成两个查询，一个用来查找“information”的相关文件，一个用来查找“retrieval”的相关文件。然后用户需要解密返回来的两个查询列表，决定哪一个列表是所需要的(假定返回来的列表中包含加密文件内容的某些元数据描述)。最坏的情况是，当元数据描述无效时，用户需要请求两个查询中的所有加密文件并对其解密，才能选择所需要的文件。这个方法既耗费时间，效率又低(从通信带宽和客户端的计算量的角度考虑)。在文献[16]中，作者提出了支持多关键词有序搜索的加密数据方案。这个方法需要首先得知所有文件的所有关键词，以便于建立特定的二维矩阵。这就意味着当有新的文件加入时，需要重新构造这个矩阵，因此该方案效率非常低下。在文献[17]中，虽然实现了模糊多关键词查找，但是它的查找效率不是特别高，云存储量也比较大。

2 问题规划

A. 系统模型

和文献[13]一样，我们考虑云数据服务中的 3 个实体，即数据拥有者、云服务器和授权用户。如图 1 所示，数据拥有者拥有数据文件集合 F。F 中的所有文件经过加密，形成了一个加密文件集合 C。为了在 C 上搜索，数据拥有者为 F 建立了一个可搜索的加密索引 I，然后将索引 I 和加密文件集合 C 存储到云服务器上。使用关键词搜索文件时，经授权的用户通过某些搜索控制机制为这些关键词建立相关的陷门 T。云服务端从数据用户接收陷门 T 之后，搜索索引 I，然后将相关的加密文件集合返回给用户。

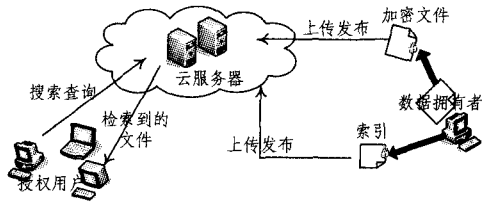

图 1 查询加密文件的系统结构

B. 安全模型

在我们的工作中，将使用与文献[13]描述类似的安全模型。我们假定云服务器是“诚实但是好奇的”。确切地说，云服务器不会删除存储的加密数据文件和索引，它会遵循授权用户的查询请求进行查询操作，并返回未经篡改的查询结果。但是，云服务器又是好奇的，它可能会出于某种盈利目的分析存储在它上面的数据来获得额外的信息。基于云服务提供商有可能获得用户的私密信息，设计方案确立了以下两个安全模型[15]：

已知密文模型。在该模型中，云服务提供商被期望只知晓加密的数据集 C 和用于关键词搜索的索引集 I。

已知背景模型。在该模型中，云服务提供商被假定对数据集有额外的行为，获得除了在已知密文模型中能访问的信息之外的主题和相关的静态数据。因此，云服务器能够利用访问文件的频率和关键词的频率[18]来推断查询队列中的关键词。

C. 研究目的

为了支持对存储在云服务器上的加密文件进行多关键词查询，基于上文中提出的系统模型和安全模型，我们提出的方案应该达到以下几个目标。

隐私保护：解决方案应该满足隐私需求，防止云服务器从任何数据集及其相关的索引得到额外的信息。

多关键词查询：搜索模型不仅支持单个关键词查询，而且应该支持多关键词查询，云服务器将包含单个或多个关键词的相关加密数据文件列表返回给用户。除此之外，搜索模型还应该支持多个关键词进行逻辑与和逻辑或关系查找，以满足客户不同要求的查询。

高效：方案应该在较低通信、存储和计算开销的前提下，完成对云中加密文件的隐私保护查询功能。

D. 预备知识

为了更好地理解本文提出的方案，首先给出一些重要的相关定义。

(1)二叉排序树：每个结点至多只有两棵子树，即每个结点的度小于等于 2 的树称为二叉树。二叉树的子树有左右之分，其次序不能任意颠倒。二叉排序树是具有下列性质的二叉树：(1)若它的左子树不为空，则左子树上所有结点的值均小于它的根结点的值；(2)若它的右子树不为空，则右子树上所有结点的值均大于它的根结点的值；(3)它的左、右子树也分别为二叉排序树。

二叉排序树的查找过程为：首先将给定值和根结点的关键词进行比较，若它们相等，则查找成功；否则，将依据给定值和根结点的关键词的大小关系，决定是继续在左子树还是右子树上查找。通常，二叉排序树采用二叉链表作为存储结构。

(2)关键词的陷门：关键词陷门的实现是应用了一个单向

函数：给定一个关键词 w，一个私钥 sk，w 的陷门表示为 $T_w=H(w,sk)$。关键词的陷门帮助我们保护查询和索引的隐私。

3 PPMKS 方案的构建

本文提出了一种适用于云计算环境中的高效的支持多关键词查询的方案（简称 PPMKS 方案）。该方案能够在不泄露任何用户云数据信息的情况下，达到高效查询加密云数据的目的。

为了达到安全高效的查询目的，本方案建立了一个查询索引顺序树。用户除了将自己的文件加密后发送到云服务器端，还要将其各个文件所对应的关键词集合发送到云服务器上，并将其组织成一个查询索引顺序树。该索引顺序树是以各个不同的关键词为树结点而形成的二叉排序树。树的结点结构里面包括两个实体：第一个实体是经过加密后的关键词；第二个实体是该关键词所对应的各个文件的 ID 所形成的链表（索引结点的顺序大小依据结点的第一个实体，即经过加密后的关键词）。有了这个查询索引顺序树，授权用户就可以通过客户端键入关键词来高效而准确地查询到与该关键词相关联的文件。

利用索引顺序树构造的支持多关键词查询的方案（PPMKS）的具体步骤实现如下。

（1）系统建立：在客户端，密钥生成器（PKG）输入一个安全参数 k，返回系统参数 $params$。系统参数包括用户私钥 sk、加密函数 E。

（2）加密文件和关键词：在客户端，授权用户输入系统参数 $params$、文件 F。该算法首先使用用户私钥 sk 对文件进行前期处理。即 $F'=E(F,sk)$。同时，该算法使用加密函数 E 分别对与该文件所对应的关键词集合中的各个关键词（$W=\{w_1,w_2,\cdots w_n\}$）进行处理。即 $w_i'=E(w_i,sk)$，最终得到 $W'=\{w_1',w_2',\cdots,w_n'\}$。最后，文件拥有者将 F' 和 W' 发送到云服务器端（CSP）。

（3）索引顺序树更新：

a. 文件拥有者添加某文件到 CSP

文件拥有者将加密后的文件 F' 和该文件对应的关键词集合 W' 一并发送到云服务器端。云服务器端负责将该文件所对应的关键词集合插入到已建立的用于查询的索引顺序树（BSTS，Binary Sort Tree for search）中。对于 w_i'，首先执行二叉排序树查询算法，在 BSTS 中查找 w_i'。如果找到，假设其结点为 Node w_i'，则在该结点结构的第二部分（包含该关键词的文件 ID 号链表）插入该文件 F' 的 ID；如果没有找到，则将 w_i' 和 F' 的 ID 号形成一个新的树结点，并将其插入到已建立好的索引顺序树中。插入过程：若二叉排序树为空，则将待插入结点作为根结点插入到空树中；当其非空时，将待插结点关键词和树根关键词进行比较，若待插结点关键词等于树根关键词，则无需插入；若待插结点关键词小于树根关键词，则插入到根的左子树中；若待插结点关键词大于树根关键词，则插入到根的右子树中。而子树中的插入过程与在树中的插入过程相同，如此进行下去，直到把待插结点作为一个新的树叶插入到二叉排序树中，或者直到发现树已有相同关键词的结点为止。

b. 文件拥有者向 CSP 申请删除某个文件

CSP 将文件删除，并更新索引树。CSP 输入索引顺序树、文件 F'。CSP 从文件头部得到与之相关联的关键词集合 W'。对于 w_i'，该算法在索引顺序树中查找到包含该关键词 w_i' 的结点位置，并将该文件 F' 的 ID 从该关键词所对应的各个文件的 ID 所形成的链表中删除（如果删除之后使链表为空，则将该结点从索引顺序树中删除）。

（4）多关键词查找：授权用户输入由多个关键词组成的与或逻辑表达式，希望查询文件库中包含的关键词符合该逻辑表达式的所有文件。授权用户输入由多个关键词组成的与或逻辑表达式（例如 $Query=w_1\cup w_2\cdots\cup w_i\cap w_{i+1}\cdots\cap w_j$），在客户端，授权用户首先使用加密函数 E 和用户私钥 sk 对 $Query$ 进行处理。即 $Query'=w_1'\cup w_2'\cdots\cup w_i'\cap w'_{i+1}\cdots\cap w_j'$，然后将 $Query'$ 发送给 CSP 服务器。基于索引顺序树，CSP 对 $w_1',\cdots,w_i',w_{i+1}'\cdots,w_j'$ 依次调用查询算法，分别得到 $\{ID_1\},\{ID_2\},\cdots,\{ID_j\}$。根据逻辑表达式 $Query'=w_1'\cup w_2'\cdots\cup w_i'\cap w'_{i+1}\cdots\cap w_j'$，计算符合条件的文件 $\{ID\}$（$\{ID\}=\{ID_1\}\cup\{ID_2\}\cdots\cup\{ID_i\}\cap\{ID_{i+1}\}\cdots\cap\{ID_j\}$）。

客户端明文恢复：CSP 将查询结果 $\{ID\}$ 集合发送给授权用户。最后，授权用户利用私钥 sk 与由 CSP 返回来的结果进行解密操作 $f=E^{-1}(f',sk)$，最终得到符合查询条件的明文文件。

PPMKS 方案的主要工作过程如图 2 所示。

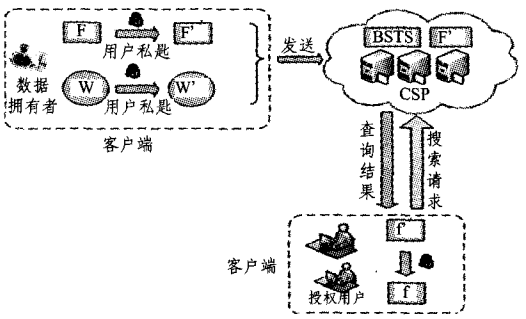

图 2 PPMKS 方案的主要工作过程

可以将 PPMKS 方案形式化为以下几个算法。

（1）系统建立（Setup）：密钥生成器（PKG）输入一个安全参数 k，返回系统参数 $params$。系统参数包括用户私钥 sk、加密函数 E。

（2）加密文件算法：用户输入系统参数 $params$、文件 F。该算法使用用户私钥 sk 对文件 F 进行加密。

（3）加密关键词算法：用户输入系统参数 $params$ 和与文件相关联的关键词集合 $W=\{w_1,w_2,\cdots,w_n\}$。该算法利用加密函数 E 对关键词集合中的关键词进行处理。即 $w_i'=E(w_i,sk)$，最终得到 $W'=\{w_1',w_2',\cdots,w_n'\}$。

系统初始化，加密文件和关键词算法的伪代码表示为：

```
params=Setup(K);
F'=E(F,sk);
for each keyword w_i∈W
{
  w'=E(w_i,sk);
}
SendtoCSP(F',W');
```

（4）树结点插入算法：该算法的功能是将新的关键词插入到索引结构中。CSP 输入索引顺序树、加密后的文件 F' 和该文件对应的关键词 w'。该算法将 w' 和 F' 的 ID 形成一个新的树结点，并将其插入到索引顺序树中。

(5)树结点链表域添加算法:该算法的功能是向云服务端插入新的文件。CSP 输入索引顺序树、文件 F' 和该文件所对应的关键词 w'。该算法在索引顺序树中查找到包含该关键词 w' 的结点位置,并将该文件 F' 的 ID 添加到该关键词所对应的各个文件的 ID 所形成的链表中。

添加文件算法的伪代码可以表示为:

```
GetFileAndKeyword(F',W');
for each keyword w_i'∈W'
{
node_i=BinarySearchTree(w_i',BSTS);
  if(node_i! =NULL)
    Insert(F',Node_iID);
  else
  {
    newNode=new Node(w_i',F');
    Insert(newNode,BSTS);
  }
}
```

(6)树结点链表域删除算法:该算法的功能是删除云服务器端的数据文件。CSP 输入索引顺序树、文件 F' 和该文件所对应的关键词 w'。该算法在索引顺序树中查找到包含该关键词 w' 的结点位置,并将该文件 F' 的 ID 从结点中该关键词所对应的各个文件的 ID 所形成的链表中删除。

删除文件算法的伪代码可以表示为:

```
    GetFileAndKeyword(F',W');
for each keyword w_i'∈W'
{
  node_i=BinarySearchTree(w_i',BSTS);
  Delete(node_i,F');
  if(node_iID==NULL)
    Delete(node_i,BSTS);
}
```

(7)多关键词查找算法:CSP 输入索引顺序树和由多个关键词组成的与或逻辑表达式,该算法输出符合该与或逻辑表达式的所有文件的 ID。

多关键词查找算法的伪代码表示为:

```
Query=GetQueryFromUser(W',user_ID);
for each w_i'∈Query
{
  {ID_i}=BinarySearchTree(w_i',BSTS);
}
{ID}={ID_1}∪{ID_2}…∪{ID_i}∩{ID_i+1}…∩{ID_n}
SendToUser({ID},user_ID);
```

4 性能分析

该小节将分析 PPMKS 方案的安全性、查找时间开销和索引树更新开销。

(1)安全性

在 PPMKS 方案中,在将数据发送给 CSP 之前,使用用户私钥 sk 对数据文件和关键词进行了加密,用户私钥 sk 只有授权用户知道,CSP 和非授权用户都不知道。文件和关键词经过加密之后,非授权用户和 CSP 都无法得知具体的关键词信息和文件信息,从而能够在不泄露任何其他信息的前提下,达到隐私保护关键词查询数据的目的。

(2)查找时间开销

PPMKS 方案只需要存储文件以及基于二叉排序树结构的索引顺序树,树的结点的顺序是参照加密之后关键词的顺序。进行查找时,首先将需要查找的关键词(发送到 CSP 端也是加密的)同索引树的根结点比较,根据是相等、小于、大于决定是返回文件标识符列表还是继续往树的左子树、右子树查找。现在计算二叉排序树的平均查找性能。

假设在含有 $n(n\geqslant 1)$ 个关键词的序列中,i 个关键词小于第一个关键词,$n-i-1$ 个关键词大于第一个关键词,则由此构造而得的二叉排序树在 n 个记录的查找概率相等情况下的平均查找长度为

$$P(n,i)=\frac{1}{n}[1+i*(P(i)+1)+(n-i-1)(P(n-i-1)+1)]$$

式中,$P(i)$ 为含有 i 个结点的二叉排序树的平均查找长度,则 $P(i)+1$ 为查找左子树中每个关键词时所用比较次数的平均值,$P(n-i-1)+1$ 为查找右子树中每个关键词时所用比较次数的平均值。又假设表中 n 个关键词的排列是"随机"的,则可对上面的公式从 i 等于 0 至 $n-1$ 取平均值,即

$$P(n)=\frac{1}{n}\sum_{i=0}^{n-1}P(n,i)$$

$$P(n)=1+\frac{1}{n^2}\sum_{i=0}^{n-1}[iP(i)+(n-i-1)P(n-i-1)]$$

方括号里第一项与第二项对称。另,$i=0$ 时,$iP(i)=0$,则上式可以化简为

$$P(n)=1+\frac{2}{n^2}\sum_{i=0}^{n-1}iP(i)\quad n\geqslant 2$$

显然,$P(0)=0$,$P(1)=1$,则

$$\sum_{j=0}^{n-1}jP(i)=\frac{n^2}{2}[P(n)-1]$$

又 $\sum_{j=0}^{n-1}jP(i)=(n-1)P(n-1)+\sum_{j=0}^{n-2}jP(i)$

即可得

$$P(n)=(1-\frac{1}{n^2})P(n-1)+\frac{2}{n}-\frac{1}{n^2}$$

由递归公式和初始条件 $P(1)=1$,可推得

$$P(n)=2(1+\frac{1}{n})(\frac{1}{2}+\frac{1}{3}+\cdots\frac{1}{n})+\frac{2}{n}-1$$

则,当 $n\geqslant 2$ 时,有

$$P(n)\leqslant 2(1+\frac{1}{n})\ln n$$

由此可见,在随机的情况下,二叉排序树的平均查找时间与 $\log_2 n$ 是等数量级的。由此可得知,PPMKS 方案的平均查找时间为 $O(\log_2 n)$(其中 n 为索引树结点的个数,即关键词的个数),根据时间复杂度定义可知,对数级的时间复杂度相当低,能够实现高效的查找云中的加密数据。而且该方案使用户只需取回具有某些特定关键词的文件进行解密,用户进行解密的时间开销也非常小。以下对二叉排序树与一般二叉树的查找时间进行了比较(见表 1),说明排序树在查找时间方面非常高效。

表 1 二叉排序树与一般二叉树的比较

	最好情况	最坏情况	平均查找时间
二叉排序树	1	$\log_2 n$	$O(\log_2 n)$
一般二叉树	1	n	O(n)

(下转第 176 页)

filtering recommendation algorithms[C]// Proceedings of the 10th international conference on World Wide Web. New York, N. Y., USA: ACM, 2001: 285-295

[2] Pazzani M J. A framework for collaborative, content-based, and demographic filtering [J]. Artificial Intelligence Review, 1999, 13(5/6): 393-400

[3] Huang Zan, Chung Wing-yan. A graph model for e-commerce recommender systems [J]. J. ASIST, 2003, 55(3): 259-274

[4] Zhang Guang-wei, Kang Jian-chu. Context based collaborative filtering recommendation algorithm [J]. Journal of system Simulation, 2006, 18(S1): 595-602

[5] 姚忠，吴跃，常娜. 集成项目类别与语境信息的协同过滤推荐算法[J]. 计算机集成制造系统，2008，14(7)：1449-1455

[6] Bell R M, Koren Y. Improved neighborhood-based collaborative filtering[C]// KDDCup'07. San Jose, California, USA, August 2007

[7] Bell R, Koren Y, Volinsky C. The BellKor Solution to the Netflix Prize[R]. 2009

[8] Baluja S, Seth R. Video Suggestion and Discovery for YouTube: Taking Random Walks through the View Graph[C]// Proc. 17th Int'l World Wide Web Conf. 2008

[9] Fouss F, Pirotte A, Renders J-M, et al. Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation [J]. IEEE Trans. Knowl. Data Eng., 2007, 19: 355-369

[10] 周军军，王明文，何世柱. 基于随机游走和聚类平滑的协同过滤推荐算法[J]. 广西师范大学学报：自然科学版，2011，29(1)：173-178

[11] Sarwar B M, Karypis G, Konstan J A. Analysis of Recommendation Algorithms for E-Commerce [J]. Proceedings of the ACM EC'00 Conference, 2000, 40(3): 158- 167

[12] Ha V, Haddawy P. Toward Case-Based Preference Elicitation: Similarity Measures on Preference Structures[C]// Proceedings of 14th Conference on Uncertainty in Artificial Intelligence. 1998: 193-201

[13] Lemire D, Maclachlan A. Slope One Predictors for Online Rating-Based Collaborative Filtering[C]// Proceedings of SIAM Data Mining (SDM'05). 2005

(上接第 161 页)

(3)索引树更新开销

在两种情况下需要对索引顺序树进行更新操作。第一，当文件拥有者添加文件到 CSP 时，CSP 只需将该文件所对应的关键词集合插入到索引顺序树中。找到插入位置的平均时间为 $O(\log_2 n)$(其中 n 为索引树结点的个数，即关键词的个数)。第二，当文件拥有者向 CSP 申请删除某文件时，CSP 除了将文件库中的对应文件删除之外，只需将该文件 *ID* 从索引顺序树中与该文件相关联的所有关键词所对应的结点的文件 *ID* 链表中删除。查找删除结点位置的平均时间也为 $O(\log_2 n)$。由此可见，更新索引树也非常高效。

结束语 随着用户要求的提高，单关键词查找已经不能满足用户的需求。本文针对云计算环境的特点以及安全性，提出了一个适用于云计算平台的高效、安全的多关键词查找方案，即 PPMKS 方案。该方案在保证用户数据隐私的前提下，能够实现授权用户多关键词查找云服务器端的加密文件；此外，还引入了与或逻辑表达式，满足用户自定义查找的各种需求。在该结构下，用户删除、新增文件都相当便捷。PPMKS 方案充分利用了云服务器的特性，既保证了云数据的安全和隐私，又实现了高效地查找云端密文数据。

参 考 文 献

[1] 高巍. 2010 年中国通信产业十大关键词点评文章(三)—云计算[J]. 数据通信，2011(1)：5-6

[2] 李乔，郑啸. 云计算现状综述[J]. 计算机科学，2011(4)：32-37

[3] 史美林，姜进磊，孙瑞志. 云计算[M]. 向勇，译. 北京：机械工业出版社，2009

[4] Li C, Lu J, Lu Y. Efficient merging and filtering algorithms for approximate string searches[C]// Proceedings of ICDE. 2008

[5] Behm A, Ji S, Li C, et al. Space-constrained gram-based indexing for efficient approximate string search[C]// Proceedings of ICDE. 2009

[6] Ji S, Li G, Li C, et al. Efficient interactive fuzzy keyword search [C]// Proceedings of ACM WWW. 2009

[7] Bellare M, Boldyreva A, Neil A O. Deterministic and efficiently searchable encryption[C]// Proceedings of Crypto 2007. LCNS, Vol 4622, 2007

[8] Song D, Wagner D, Perrig A. Practical techniques for searches on encrypted data[C]// Proceedings of IEEE Security and Privacy. 2000

[9] Goh E-J. Secure indexes[R]. 2003

[10] Boneh D, Crescenzo G D, Ostrovsky R, et al. Public key encryption with keyword search[C]// Proceedings of EUROCRYP. 2004

[11] Waters B, Balfanz D, Durfee G, et al. Building an encrypted and searchable audit log[C]// Proceeding of 11th Annual network and Distributed System. 2004

[12] Cutmola R, Garay J A, Kamara S, et al. Searchable symmetric encryption: improved definition and efficient constructions[C]// Proceedings of ACM CCS. 2006

[13] Li J, et al. Fuzzy Search over Encrypted Data in Cloud Computing[C]// Proceedings of IEEE INFOCOM. 2010

[14] Chang Y C, Mitzenmacher M. Privacy preserving keyword searches on remote encrypted data[C]// Proceedings of ACNS. 2005

[15] Shi E, Bethencourt J, Chan T H H, et al. Multidimensional range query over encrypted data[C]// IEEE Symposium on Security and Privacy. 2007

[16] Cao N, et al. Privacy-Preserving Multi-keyword Ranked Search over Encrypted Cloud Data[C]// Proceedings IEEE INFOCOM. 2011

[17] Chuah M, Hu W. Privacy-aware Bed Tree Based Solution for Fuzzy Multi-keyword Search over Encrypted Data[C]// Proceedings of ICDCS Workshops. 2011: 273-281

[18] Zerr S, et al. Zerber: r-confidential indexing for distributed documents[C]// Proceedings of EDBT. 2008: 287-298