doi:10.3969/j.issn.1000-436x.2013.07.016

云计算环境下密文搜索算法的研究

项菲[1,2]，刘川意[2,3]，方滨兴[1,2]，王春露[1,2]，钟睿明[1,2]
（1. 北京邮电大学 计算机学院，北京 100876；
2. 北京邮电大学 可信分布式计算与服务教育部重点实验室，北京 100876；
3. 北京邮电大学 软件学院，北京 100876）

摘 要：为确保数据私密性，用户选择将数据加密后再上传到云端，但云无法为密文数据提供管理和搜索等服务。密文搜索技术可以把保护用户数据私密性和有效利用云服务结合起来。在分析云环境下密文搜索算法的基础上，提出基于云环境的密文搜索体系结构，研究其中的关键技术，指出云环境应用密文搜索技术存在的问题和改进的方向。
关键词：云存储；密文搜索；倒排索引；Bloom filter
中图分类号：TP393 **文献标识码：**A **文章编号：**1000-436X(2013)07-0143-11

Research on ciphertext search for the cloud environment

XIANG Fei[1,2], LIU Chuan-yi[2,3], FANG Bin-xing[1,2], WANG Chun-lu[1,2], ZHONG Rui-ming[1,2]
(1. School of Computer Science, Beijing University of Posts and Telecommunications, Beijing 100876, China; 2. Key Laboratory of Trustworthy Distributed Computing and Service of the Ministry of Education of China, Beijing University of Posts and Telecommunications, Beiji g 100876, China;
3. School of Software Engineering, Beijing University of Posts and Telecommunications, Be jing 100876, China)

Abstract: In order to ensure the data privacy, users have to encrypt the data before outsourcing them to the cloud, However, the encrypted data cannot take full advantage of the rich service function and powerful computation ability of the cloud platform. Ciphertext search technology can combine the protection of user data's privacy with the efficient usage of cloud platform services. On the basis of analyzing the ciphertext search technologies in cloud computing environment, a ciphertext search system architecture based on cloud c g environment was proposed and the key technologies of ciphertext search were studied. Finally the ciphertext search technologies' main problems at present and the important research direction in future were pointed out.
Key words: cloud storage; ciphertext search; reverse index; Bloom filter

1 引言

以云计算[1]为基础的云存储系统将网络化和虚拟化技术相结合实现了强大的计算和存储能力，并且降低了海量数据的管理成本，还能够向用户提供可扩展的、满足 QoS 要求的、高通用性的、可按需分配的存储资源。但安全问题是云计算目前发展面临的最关键问题，云用户对云存储服务提供商的不信任已成为制约云存储推广的重要因素。在云存储模式下，用户的数据存储在云提供商（CP, cloud provider）中，这意味着用户的数据（包括敏感数据）完全由云提供商进行管理和存储，数据处于用户不可控域中，用户担心云提供商可能会窃取和篡改自己的敏感数据（例如政府和公司的财政数据、个人的医疗信息等）[2,3]。因此，用户希望能够将数据交由云提供商存储和管理的同时又不向云提供商泄露任何数据相关信息[3]。

最简单直接的一种保护用户数据安全的方法就是由用户在客户端将数据加密成一个或若干个密文数据分组后上传，这样存储在云端的是加密后的

收稿日期：2013-03-22；修回日期：2013-06-25
基金项目：国家自然科学基金资助项目（61202081）；国家科技重大专项基金资助项目（2012ZX03005010）
Foundation Items: The National Natural Science Foundation of China (61202081); The National Science and Technology Major Project of the Ministry of Science and Technology of China (2012ZX03005010)

数据信息，因此能够在传送和存储过程中保护敏感数据的隐私性和机密性[4]。这样不但可以保证云提供商无法获取存储在云端的用户信息，也无法未经授权就盗用用户数据以获利；而且即使黑客在云端窃取了用户数据，也得不到明文信息因为其无法解密。

数据加密后上传虽然解决了隐私安全问题，但是当用户需要使用某个文件时，用户必须将上传至云端的密文数据分组全部下载下来，在本地解密后搜索出自己需要的内容[5]。这无疑浪费了带宽资源，且搜索效率极低。这样就产生了新的问题，如何能够使用户不需要对整个文档进行下载解密就可以有效地直接从密文中搜索出所需要的内容？

对云中的密文数据进行操作和处理已引起越来越多的重视，各国的研究人员也都提出了相应的解决方案。本文分析了当前云存储中所面临的安全问题，按照直接对密文进行线性搜索和基于安全索引 2 个大类对各经典密文搜索算法的核心思想进行了分类和总结，提出了未来云存储中密文搜索技术的重要科研和应用方向，以期为未来的云存储安全问题的科研、应用发展做出有益的探索。

2 基于云存储的密文搜索体系结构

云存储是以云计算为基础发展起来的新的存储方式。与传统的存储方式相比，云存储使得个人或企业可以按当前所需付费获取云计算中无限的资源，而不必按其最大需求购买昂贵的存储设备。云存储在降低存储成本的同时也带来了安全问题的新挑战，主要表现在数据的所有者与数据分离，数据处于所有者不可控的区域，数据存储在云中的时候，数据所有者不会知道云对数据进行了哪些操作。云存储这样的服务性质，导致了在云计算环境下特有的安全隐患。加上近年来用户存储在云中的数据的泄露事件时有发生，用户对云是否信任直接决定了云存储服务的未来发展前景，因此如何解决和保证存储在云中数据的安全性问题应该引起足够的重视。

解决上述安全问题最直接有效的方法就是数据所有者将数据加密后再上传至云端，加密的方法保证了数据在未经授权的情况下不会被窃取和篡改，数据在云端可控。但是加密处理导致密文数据失去了明文数据原有的相关特性，使得数据所有者无法在云端对自己的数据进行诸如查询、更新等操作，从而导致云提供的服务大打折扣，因此研究人员将密文搜索技术应用于云存储系统用以兼顾云存储数据的安全性和可操作性。

基于云计算环境的密文搜索的基本思想可以描述为由数据所有者在客户端用一种特殊的加密方式对其数据进行加密获得加密后的密文并生成对应的查询单射函数；用户发起数据搜索请求时，云可以利用用户发送的查询单射函数，通过密文匹配操作，查找出用户所需数据，且在操作过程中云不会获知数据的明文内容。因此基于云存储的密文搜索技术不但保证了数据存储的安全性，还保证了云的功能性（不会沦为简单的数据存储池）。

总结起来，目前密文搜索技术主要有 2 种典型方法：第 1 种直接对密文进行线性搜索，即对密文中每个单词进行逐个比对，确认关键字是否存在与文档中以及统计其出现的次数；第 2 种基于安全索引，先对文档建立关键字索引，然后将文档和索引都加密上传至云端，搜索时从索引中查询关键字是否存在于某个文档中。现有工作能否以统一的体系结构组织起来，并为应用和下一步的研究工作提供基础？

经过对现有的密文搜索工作的调研和整理，本文提出基于云存储的密文搜索体系结构，如图 1 所示。密文搜索的组成个体主要有 3 部分：数据所有者（data owner）、数据用户（data user）和云提供商。数据所有者先将数据传给 proxy，由 proxy 根据密文搜索方案决定对数据是否需要建立关键字索引，如果不需要建立索引（步骤 2A），直接对明文数据进行加密上传到云端即可；如果需要建立索引，那么先建立关键字索引（步骤 2B-1），然后依照一定的可搜索加密机制分别对索引（步骤 2B-2）和数据进行加密，再将加密后的数据和索引上传到云端，在云端进行存储。进行密文搜索时，数据所有者分别给予不同的数据用户以不同的访问权限的查询单射函数和解密密钥，使得用户可以通过云提供商提供的查询接口进行密文搜索，并且可以对搜索到的密文进行解密。

图 1 中由 proxy 建立的是倒排索引表，KW_i 代表关键字；D_{ij} 是分配给文档的标识号，是唯一的；TF 是关键字的词频(term frequency)信息，即该关键字在对应文档中出现的次数。Pos 表示关键字的位置信息。通过搜索倒排索引表可以迅速地找出某个

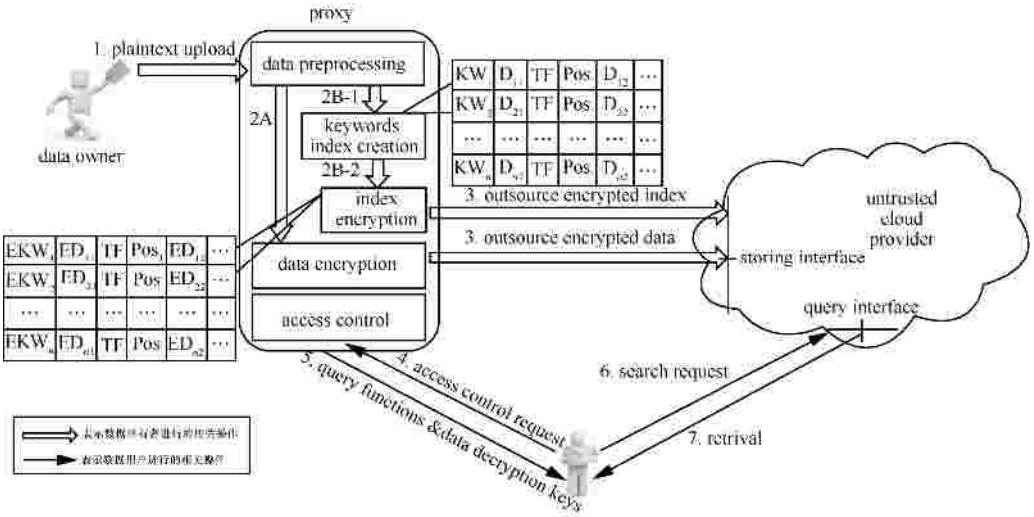


图 1　基于云存储的密文搜索流程

关键字具体出现的位置和存在于哪篇文档中。EKW_i 是加密后的关键字，ED_{ij} 表示加密后的文档标识号，加密方案可以是对称加密，也可以是非对称加密。

对于密文搜索在云中的工作流程可以概括地描述为：云的查询接口接收到用户的查询请求，根据用户提交的查询单射函数和关键字（可以是密文也可以是明文）进行相应的操作（在不同的密文搜索方案中的操作是不同的，比如 Song 等人[6]的方案中查询单射函数是异或操作，Goh[5]的方案中是 Bloom filter），根据操作后得到的结果判断该查询关键字是否存在于文档之中。

3　直接对密文线性搜索的方法

无索引情况下，对密文全文直接进行线性搜索的方法可以进一步分为 2 个子类：第 1 子类是基于对称密钥的加密搜索，代表文章是 Song 等人[6]提出的可搜索对称密钥加密 (SSKE, searchable symmetric key encryption)方案，该方案的基本思想是采用流密码(stream cipher)方法对字符型数据进行加密处理，存储时，它使用流密码算法将原数据与随机发生器产生的随机数进行按位异或后得到的密文存储在加密的文件中；查询时，可以无需解密直接在加密文本中搜索关键字，即用户用给定的加密后的关键字在密文中进行逐个异或运算，根据异或的结果是否等于该关键字的查询单射函数，确定该关键字是否存在于密文中。

SSKE 方案为每个词都进行了特殊的两层加密，具体流程如图 2 所示。

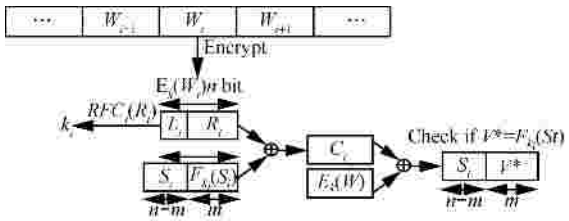


图 2　可搜索对称密钥加密（SSKE）工作流程

Server 通过给定的查询单射函数可以剥去外层的加密，然后能够判断内层的密文是否存在于文档中，其基本步骤如下。

1) 输入分组加密函数 E、密钥 k 及明文中的单词 W_i（长度经过处理，皆为 n bit），生成一次加密后的单词 X_i（该过程与 W_i 的所处位置无关）：$X_i=E_k(W_i)$。

2) 用 stream cipher 生成一串伪随机数 $S_1, S_2,\cdots,S_k$，$S_i(i\in[1,k]$，k 表示文档中单词的个数)的长度皆为 $n-m$ bit。

3) 将经过一次加密后的单词 X_i 分成 L_i 和 R_i 两部分，L_i 的位数是 $n-m$，R_i 的位数是 m。

4) 输入散列函数 f 和 L_i，生成密钥 k_i：$k_i=f_k(L_i)$。

5) 输入伪随机函数 F 和 k_i 对 S_i 进行操作，生成其余的 m bit，即 $F_{k_i}(S_i)$。将 $F_{k_i}(S_i)$ 和 S_i 合并得到 $T_i = S_i \| F_{k_i}(S_i)$，$T_i$ 的长度为 n bit。

6) 将 T_i 与 X_i 按位进行异或运算，得到二次加密后的密文：$C_i= T_i\oplus X_i$。

7) 搜索时，输入加密后的待搜索单词 $E_k(W)$依次与密文中的每个单词进行异或运算，即 $T= C_i\oplus E_k(W)$。检测是否存在某个 S_i 的 T_i 与 T 相等，如果相等，则表示文档中存在单词 W；否则，表示

文档中不存在单词 W。

8) 解密时，先将 S_i 与 C_i 的前 n- m bit 进行按位异或运算，得到 L_i 的值，再由 $k_i = f_k(L_i)$可以求出 k_i 的值，得到 k_i 后就可以解密密文文档。

其中步骤 7）在云端完成，云从客户端接受到加密后的待搜索单词 $E_k(W)$后与存在云中的密文中的每个单词进行异或运算，寻找密文中是否存在用户查询的单词。其余步骤皆在客户端完成。

SSKE 这种方法几乎没有额外存储空间的开销，加解密速度快，在搜索过程中只是用到简单的异或运算和函数求值，执行效率高，简单易行。但为了保证密文不受到明文攻击和统计攻击，流密码算法中密钥序列则不能重复，这样会导致密钥管理难度增大，并且 SSKE 方案只能证明是一个安全加密方案，而不能证明是一个安全的密文搜索方案。SSKE 不是安全搜索方案的原因在于以下几点。

1) 其底层明文的分布结构在抗统计性分析攻击面前是很脆弱的。

2) SSKE 方案是通过逐词匹配密文信息来搜索关键字，所以使得这种搜索方法在海量数据的情况下难以应用且会泄露搜索关键字在文档中的所处位置。

3) 只能实现对自己加密数据的搜索，与当前所有的文件加密体系都不兼容。

由于 SSKE 方案的这些局限性，研究人员又进一步研究出基于非对称密钥的加密搜索方案，即第 2 子类。代表文章是 Boneh 等人[7]提出的基于关键字的公钥加密搜索（PEKS, public-key encryption with keyword search）方案，该方案可以让接收者从发送者发出的文件中搜索出是否包含需要的关键字。该方案的基本思想是数据发送者用公钥分别对指定的关键字集合 W 里的若干关键字 w_1~w_n 执行加密操作运算，计算得到的结果 C_{w_1}~ C_{w_n} 附在发送的消息后面，且由服务器保存；数据接收者用私钥生成查询关键字 w'的查询函数，并将结果 T_w 提交给服务器。服务器通过执行比对函数，根据结果输出 1 或者 0 判断两者是否是同一个单词。PEKS 方案的执行流程如图 3 所示。

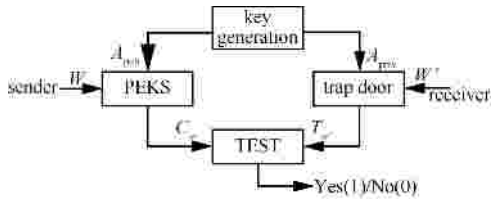


图 3 公钥加密搜索工作流程

PEKS 主要由 4 部分组成，该方案执行过程如下。

1) 由安全参数 s 生成公钥私钥对$(A_{\text{pub}}, A_{\text{priv}})$：

$$(A_{\text{pub}}, A_{\text{priv}}) = \text{KeyGen}(s)$$

2) 输入公钥 A_{pub} 和指定关键字集合 W，分别对集合中的关键字 w_1~w_n 进行加密，输出得到 W 的密文集合 C_W：

$$C_W = \text{PEKS}(A_{\text{pub}}, W)$$

3) 输入接收者的 A_{priv} 和某一查询关键字 w'，生成 w'的查询单射函数 T_w：

$$T_{w'} = \text{Trapdoor}(A_{\text{priv}}, w')$$

4) 搜索时，输入公钥 A_{pub}、查询单射函数 $T_{w'}$ 和密文集合 C_W，计算 $\text{Test}(\text{Trapdoor}(A_{\text{priv}}, w'), \text{PEKS}(A_{\text{pub}}, w_m))$，$m \in [1,n]$。如果结果为 1，即 $w_m = w'$，表示搜索到关键字；否则，表示没有搜索到关键字。

其中步骤 1)~3)在客户端完成，步骤 4)由云端完成，云收到用户发送的查询单射函数后，根据用户所给的公钥，在密文集合里进行 Test 操作，查找关键字是否存在于密文集合中。

在 PEKS 中，任何用户只要握有公钥就可以将数据加密上传，但是只有数据所有者可以用私钥生成查询单射函数进行搜索，该方法可用于邮件服务器。PEKS 方法的缺点也很明显：非对称加密方案的运算相对复杂，加密解密效率较低，搜索效率也不高，另外还会泄露用户的搜索类型。

为了提高搜索效率，降低通信开销，Song 等人利用 Paillier 加密系统[8]设计了一种新的密文线性搜索方案[9, 10]。该方案的核心思想是用一些加密的缓存器来存储相关的内容。由 Paillier 加密系统的性质生成公钥 K_{pub} 和私钥 K_{priv}，首先在客户端将所有用户可能感兴趣的关键字选出来生成加密的查询序列 Q；然后将 K_{pub} 和 Q 传给服务器，因为 Paillier 加密系统的同态性质，使得服务器可以对密文执行相关查询操作后返回结果给用户，且服务器不知道用户的搜索内容；最后用户用 K_{priv} 对结果进行解密得到查询结果。

直接对密文线性搜索算法应用在云存储系统中时，其对文本的加密是在客户端完成，然后将加密后的数据上传至云端，如果想要搜索存储在云中的数据，用户根据所用的加密算法生成对应的查询单射函数发送给云，然后由云在不会获知明文内容的情况下完成对密文的等值匹配搜索过程。

直接对密文线性搜索算法能够实现将数据安全地存储在云中，保证数据的私密性和可控性，支持在云计算环境中的密文搜索。但是云中的数据高度集中，因此安全措施必须满足能够处理海量信息的需求。而直接对密文线性搜索算法在密钥管理方面代价太大且效率较低，所以不太适合用于大规模云存储情况的密文搜索情况。

4　基于索引的密文搜索方法

直接对密文线性搜索的方法缺点在于搜索效率不高，且无法应对海量数据的搜索场景，为了解决这个问题，研究人员考虑到为密文建立索引以提高搜索速度和搜索范围。基于索引的密文搜索也可以分为 2 个子类：第 1 子类是针对结构化的数据，以数据库为代表，不需要进行分词操作；第 2 子类是针对非结构化的数据，以文件系统和 Web 网页内容为代表，需要进行分词操作，且分词会直接影响到搜索效果。

4.1　针对结构化数据的密文搜索

对于结构化数据，数值明文在加密完成后，密文不再具有原先的任何大小特征。这使得对结构化数据最为常用的操作：等值搜索和范围搜索在密文数据中无法进行。为了在密文数据中也能进行相关操作，研究人员考虑用不同的加密方法使得仍然可以对密文进行等值搜索和范围搜索。

H.Hacigumus 等在数据库即服务（DAS, database as a service）[11]的背景下，提出了一种数据库分桶建立索引算法，将元组作为最小粒度对数据库进行加密，用桶划分的方法为每一列密文建立索引，从而缩小解密范围，可以提高对加密数据库的查询速度[12]。该算法的核心思想是：将数据根据数值范围划分为多个分区，然后分别为每个分区分配一个唯一的 ID 作为标识，称为桶号，即索引号，记为 $ident_{R.A_i}(p_j)$，其中，R 表示关系，A_i 表示关系中的一个属性，p_j 表示数据的某一段数值范围。为了避免信息泄漏，通常都采用散列算法进行数据库分区,即对数值进行散列后，根据散列值再进行分区。下面给出一个用分桶算法建立索引的实例。表 1 是员工信息明文，图 4 给出了 eid 这列数值经过散列后的分区与桶号的对应关系，然后 eid 经分桶后建立的索引如表 2 所示，$\mathrm{Map}_{\mathrm{emp_eid}}(v)$即为属性值 $v(v \in p_j)$所建立的索引号。其余各列也用同样的方法建立索引，最后得到加密后的员工信息密文表及索引表，如表 3 所示，其中 etuple 是对明文元组加密后的数据串。用户在进行数据库密文搜索时，首先通过关键字的数值大小判断关键字落在哪一个分区内，进而根据数值范围确定数据库中哪些记录可能符合搜索条件。该方法在执行搜索时一般是将符合条件的分区里的所有数据都返回给用户，用户还需要在返回数据中进一步搜索出自己实际需要的数据，因此当每个数据库分区中的数据记录较多时，这种搜索方式的搜索效率就比较低。由此可以看出这种方式的分区数量越多，搜索性能会越好，但这样又会带来数据冗余，所以 H.Hacigumus 等后来又进一步讨论了该模式下密文数据查询策略的优化问题[13,14]，核心思想是将原始查询语句进行了逻辑再重组，重组后再按照不同属性进行分桶，最后进行过滤算法可以快速有效地搜索出所需结果，且不会造成太大的数据冗余。

图 4　分区与桶号的对应关系

表 1　员工信息明文

eid	ename	salary	addr	did
23	Tom	70 000	Maple	40
860	Mary	65 000	Main	80
320	John	50 000	River	50
875	Jerry	55 000	Hopewell	110

表 2　eid 列经过分桶后建立的索引号

eid value v	$\mathrm{Map}_{\mathrm{emp\_eid}}(v)$
23	2
860	4
320	7
875	4

表 3　员工信息加密后的密文及索引

etuple	eid^S	ename^S	salary^S	addr^S	did^S
1100110011110010…	2	19	81	18	2
1000000000011101…	4	31	59	41	4
1111101000010001…	7	7	7	22	2
1010101010111110…	4	71	49	22	4

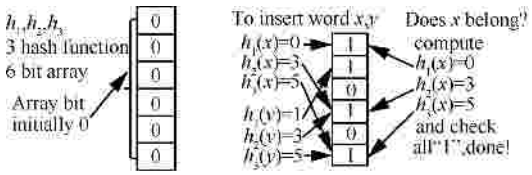
Hore 等人[15]在 Haeigumus 等人的数据库分桶算法的基础上，提出了一种改进的数据库分桶策

略。该策略利用最优桶划分算法，在数据库搜索过程中能够将解密和传输的开销降到最低，提高了密文数据库的搜索命中率和效率。最优桶划分算法的基本思想是首先根据安全需求确定分桶数目，然后不断地把原始桶划分的最优策略问题等价为求解其 2 个子划分的最优策略问题，直到满足分桶数目为止，记录其子划分的划分路径及最小代价，最后合并子问题的最优结果得到最优桶划分策略[16]。Hore 在文中另外还提出了一种可控制的扩展算法，能够自适应地按照不同的需求调整变化数据安全等级，虽然降低了一定的密文搜索效率，但是使得安全策略更为机动灵活。Hore 在文中还证明了利用分桶算法建立密文索引时，对多个敏感属性的划分次数和划分桶的数目与查询命中率是成正比关系的，同时与信息的泄露程度也是成正比的。因此如果在建立索引的属性值明文空间相对较小的情况下，桶划分的数目越多时，数据的安全性会大大地降低。但是 Hore 等人在文中没有具体指出如何能够提高密文索引的安全性。

IBM 研究中心的 R.Agrawal 等[17]提出一种保序加密(OPES, order preserving encryption)算法，这种算法在加密过程中保证了数值的有序性。该算法的核心思想是：假设数据库中有任意 2 个明文数值 $p_i < p_j$，则分别对其加密后的密文数值满足 $c_i < c_j$。在这种有序性的基础上，不但降低了数据冗余，还可以实现对密文数据库最大值、最小值的搜索以及范围搜索，同时也可以实现 COUNT、GROUP BY 和 ORDER BY 等操作。但是由于密文数值与对应明文数值的大小顺序是完全一致的，当在明文空间较小的情况下，如果攻击者事先知道某些明文和密文就可以根据密文的大小关系对明文数值进行猜测，数据的安全性不高。因此 R.Agrawal 等又将 OPES 与数据库分区策略结合起来[11]：即先将数据库按数据范围分区，然后在每个数据库分区中，将加密数据服从某一目标函数进行分布，这样在每个分区中明文与密文的数值顺序可以保持相同，这种混合方式在一定程度上加强了密文数据库的安全，但是引入数据库分区策略确实增加了数据冗余。

4.2 针对非结构化数据的密文搜索

与结构化数据类型不同，非结构化数据中最常见的搜索是等值搜索或模糊搜索。Eu-Jin Goh[14]设计了一种安全加密索引及其密文搜索方法。安全索引的要解决的基本问题有 2 个：一是如何对海量密文数据进行搜索；二是如何提高搜索效率。为了解决这 2 个问题，Goh 利用 Bloom filter 为每个文档生成索引，用户可以通过这个索引来确定关键字是否存在于这个文档中。搜索时，只需要对搜索词 W 进行 h_1 到 h_r 的散列函数处理，如果计算出的值对应于 m bit 数组中的位置上的值全为 1，则表示文档中包含搜索词 W；否则搜索词 W 不在文档中，如图 5(b)所示。采用 Bloom filter 技术的优势在于使得攻击者很难通过解密的方式从索引获知关键字的明文信息。Goh 的方案对于“非自适应选择关键字攻击”是语义安全的（IND-CKA），即若一个索引是 IND-CKA 安全的，表示 2 个大小相等的加密文档的索引应该看起来有着相同数目的关键字。Goh 方案的安全性已经足够抵御 CKA 攻击，加上 Bloom filter 使得搜索效率更高，再配合 Pseudo-random functions(PRF)生成了它的最终方案 Z-IDX[14]，Z-IDX 的整体工作流程如图 5 所示。



(a) 初始化 Z-IDX 的 Bloom filter (b) Z-IDX 的索引生成和查询过程

图 5 Z-IDX 工作流程

Z-IDX 方案主要包括以下 4 个算法。

Keygen(s)：给定一个安全参数 s，选择一个伪随机函数 f: $\{0,1\}^n \times \{0,1\}^s \to \{0,1\}^s$，生成主密钥 $K_{priv}=(k_1,\cdots,k_r) \in \{0,1\}^{sr}$。

Trapdoor(K_{priv},w)：输入主密钥 K_{priv} 和单词 w，输出单词 w 的查询单射函数 $T_w=(f(w,k_1),\cdots,f(w,k_r)) \in \{0,1\}^{sr}$。

BuildIndex(D,K_{priv})：输入由唯一标识符 $D_{id} \in \{0,1\}^n$ 及单词$(w_0,\cdots,w_t) \in \{0,1\}^{nt}$ 组成的文档 D 和 $K_{priv}=(k_1,\cdots,k_r) \in \{0,1\}^{sr}$。

对每一个唯一单词 $w_i, i \in [0,t]$，计算：

Trapdoor: $(x_1=f(w_i,k_1),\cdots, x_r=f(w_i,k_r)) \in \{0,1\}^{sr}$

w_i 在 D_{id} 中的 codeword 为：$(y_1=f(D_{id},x_1),\cdots, y_r=f(D_{id},x_r)) \in \{0,1\}^{sr}$。

将 $y_1,\cdots,y_r$ 插入到文档 D_{id} 的 Bloom filter 中。

计算文档 D 中的单词数上限值 u。例如，u 的极值可以假定为文档 D 中字节的个数（加密后）。

令 v 表示在$(w_0,\cdots,w_t)$ 单词集合中所有出现的单词数目(重复出现的只记一次)，然后将$(u-v)r$ 个

1 均匀随机地插入 Bloom filter 内。这相当于在索引中加入 $u-v$ 个随机单词，且不需要进行任何伪随机函数计算。

输出 D_{id} 的索引 $I_{D_{\mathrm{id}}}=(D_{\mathrm{id}},\mathrm{BF})$

SearchIndex(T_w,I_D)：输入单词 w 的查询单射函数 $T_w=(x_1,\cdots,x_r)\in\{0,1\}^{sr}$ 和文档 D_{id} 的索引 $I_{D_{\mathrm{id}}}=(D_{\mathrm{id}},\mathrm{BF})$。

计算 D_{id} 内 w_i 的 codeword：$(y_1=f(D_{\mathrm{id}},x_1),\cdots,y_r=f(D_{\mathrm{id}},x_r))\in\{0,1\}^{sr}$。

检测 $y_1,\cdots,y_r$ 所表示的 r 个位置在 BF 内是否全为 1。

如果全为 1，输出 1；否则，输出 0。

Z-IDX 应用在云存储系统中时，其中 SearchIndex 算法在云端完成，根据查询单射函数和文档生成的 Bloom filter 判断查询关键字是否存在于文档中；其余 3 个算法皆在客户端完成。

Bloom filter[18]因为其数据结构是随机化的，所以它的空间存储效率非常高。构成 Bloom filter 的 2 个基本部分为：k 个相互独立的散列函数和一个 m 位的数组。初始时该数组的每一位都为 0，如图 5(a)所示。如果要表示某个包含 n 个元素的集合，只需要计算出其中的每个元素 k 个在$\{1,\cdots,m\}$之中的散列值，然后在数组中找到对应的 k bit：若比特为 0，置 1；若已置 1，则保持不变。但是 Bloom filter 虽然高效，却存在有一定的正向误检(false positive)概率：即查询时，计算查询元素的 k 个散列函数值，只要在数组中对应的比特有 1 个为 0 时，表示该集合一定不含此元素；但即便计算出来的 k 个散列值所对应的数组位全为 1，该集合中却未必包含该元素。

因为 Bloom filter 存在这种正向误检概率，所以 Goh 的方案不适用于“零错误”情况。于是 Chang 等人[19]提出了 IND2-CKA 方案：没有引入公钥加密体系，只用到了启发式伪随机函数。该方案不但可以避免 Goh 方案的正向误检情况，而且抗选择关键字攻击能力也比 Goh 方案强，可以抗“自适应选择关键字攻击”，即便攻击者知道以前的搜索信息也无法获知查询函数。

但是 Goh 和 Chang 的方案依然泄露了搜索类型信息（即 2 次搜索的是否是同一关键字），因此 Boneh 等人[20]又设计了能够隐藏访问类型的基于索引的公钥加密关键字搜索(PEKS, public-key encryption with keyword search with index)方案，该方案的主要思想是每个文件都伴随一个加密的关键字列表（该列表由所有可能被搜索的关键字组成），由改进的 Bloom filter 生成。该 Bloom filter 中不仅存储元素，还存储元素的位置信息，即原本用“1”填充的改用元素加密后的位置信息进行填充，搜索时对搜索词 W 进行 h_1 到 h_r 的散列函数处理，得到 r 个散列值，然后将 Bloom filter 数组中对应的 r 个位置上存储的值进行“n”操作，如果为 0，表示搜索词不在文件中；如果不为 0，那么计算出的结果经过解密后即为搜索词在文件中所处的位置，表示搜索词存在于文件中，如图 6 所示。但是该方案的搜索时间开销与数据库规模大小的平方根成正比关系，这个方案的搜索效率也比较低。

基于索引的密文搜索算法应用在云存储系统中时，客户端首先对文件集合建立索引，然后分别对索引和文件集合进行加密后上传至云端；当用户想要搜索存储在云中的数据时，用户根据所用的加密算法生成对应的查询单射函数发送给云，然后由云在加密索引中完成密文的搜索过程并返回查找结果，云在搜索过程中不会获知搜索的明文内容。

基于索引的密文搜索方法是目前的研究主流，原因是其搜索效率更好，安全性能更高，适合用于大规模的云存储密文搜索系统。

5　密文搜索的重要问题和研究方向

基于对上述具有代表性的相关密文搜索算法的分析和研究，表 4 给出了它们一些主要特性的比较。

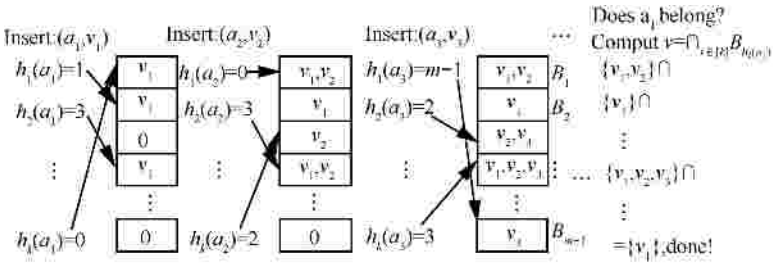


图 6　PEKS_index 工作流程

表 4 具有代表性的密文搜索算法的比较

算法	加密方式	可以实现的操作	是否泄露访问类型	安全等级	适用数据规模/类型
文献[4]	对称	精确搜索	是	CPA	小/非结构化
文献[5]	非对称	精确搜索	是	CKA	小/非结构化
文献[6]	Hash	精确搜索	否	CKA	大/非结构化
文献[11]	对称	MAX,MIN,COUNT,等值查询,范围搜索等	是	不具有语义安全性	大/结构化
文献[14]	Hash	精确搜索	是	CKA	大/非结构化

从表 4 可以看出，各算法的加密方式、功能、性能等各不相同，但都能够实现用户对存储在云中的数据可控操作。并且，安全等级和可以实现的操作是互为矛盾的 2 个方面，安全等级低的密文数据能进行的搜索操作相对要多一些，比如范围搜索，最大最小值搜索等；而安全等级高的，密文数据能进行的操作相对少一些，比如只能进行精确搜索。另外各算法适用的数据规模和类型也不尽相同。因此在实际应用中可以根据实际需求进行合适的算法选择。比如如果用户的数据集规模不大且希望加解密的效率高，那么可以考虑用文献[6]；如果用户的数据是结构化的且规模较大，并且希望能够进行多种搜索操作并对安全性要求不高，那么可以考虑用文献[17]；如果用户希望加解密效率高，安全性要求也高，且数据规模较大的情况下，可以考虑用文献[5]和文献[20]。

同样，也可以看出目前的密文搜索机制仍不够完善，如何将密文搜索应用于实际云存储中仍面临不少亟待解决的问题。密文搜索未来的研究可以包括以下几方面。

在不受信任的服务器环境下，密文搜索的安全性还需要进一步研究。这是因为为了实现密文搜索，对加密算法是有一定特殊要求的，目前的应用于密文搜索的可搜索加密技术都是确定性加密，如保持关键字的相关特征[7,19]，或要保持关键字在密文中的存储位置[6,20]，或要保证加密后的大小关系[17]，这些特殊的加密算法都存在信息泄露问题。比如 OPES[17]就会泄露数据的大小关系，而对于等值匹配搜索[6,7,20]一般都是直接将明文索引词加密后在密文中进行匹配搜索，这样攻击者可以很容易通过搜索频率等进行分析后破解。另外目前索引表里的词频和存储位置等大部分是以明文形式出现的，这也给攻击者进行频率分析提供了便利条件。因此如果能够设计出非确定性的可搜索加密技术，可以大大提高密文搜索技术的安全性。

目前基于索引的密文搜索算法的索引构建大部分是静态的[6,19,21~23]，即不能随着密文数据的增加删减进行动态的更新，只有少部分算法的索引是可以动态更新的[5,19,24,25]。Goh[5]是利用为每个文件建立 Bloom filter 数组实现索引动态更新的，在增加文件时只要新建一个与之对应的 Bloom filter 数组作为该文件的索引，删除文件时只要将该文件对应的 Bloom filter 数组删除即可。实际上存储在云上的数据每天会新增大量的数据，同时也会有许多无效数据需要删除，所以密文搜索的动态索引技术是云存储的必然要求。

目前密文搜索着重关注的都是单一关键字[4~6]或 Boolean 关键字[26~32]的搜索，很少涉及对密文搜索结果进行有效排序，因此返回的无差别的密文搜索结果质量不高，用户仍需要在大量的搜索结果中再次进行搜索找出自己想要的内容。而基于多条件多关键字搜索可以更精确且更有效、快速找到用户需要搜索的内容，因此现实中用户更倾向于使用多个关键字而不是单一关键字进行搜索。Wang 等人[33,34]已经注意到了多条件多关键字的密文搜索问题，他们利用词频信息能够将最符合搜索要求的而不是无差别的结果返回给用户，但是仍然只给出了在密文中基于单一关键字的安全排序搜索的解决方法。Cao 等人[35]进一步提出了云计算环境中对于密文的基于隐私保护的多关键字排序搜索的一种解决方法。首先用“坐标匹配”方法在搜索关键字与文档之间建立联系，接下来用“内积相似性”来计算每个文档与搜索关键字之间的相似权重，最后用“K 近邻”算法对相关度进行排序，将排名靠前的 k 个搜索结果返回给用户。

在很多场合数据是需要进行共享的，因此，对密文搜索访问控制策略的效率性及安全性的研究是很有意义的。如何合理地设置访问控制策略以及有效的密钥管理机制也应该是研究的重点。对于密文搜索系统，数据所有者和数据用户之间的密钥管

理机制和访问控制策略是必不可少的。在现有的密文访问控制策略中，数据所有者的 proxy 需要对每一个数据用户维护和发放数据密钥。当用户数量众多时，proxy 会成为应用的瓶颈。另外在支持多用户的信息搜索系统中，拥有不同权限的多个用户都可以对共享文档或网络资源进行搜索，但由于用户权限不同，即使输入相同的查询条件，返回的查询结果也应该不同。Damiani 等人[36]实现了对密钥的动态管理。

当前关于如何把密文搜索技术应用于云存储的实际场景中，研究人员进行了大量有意义的探索和实践。

对于加密的结构化数据进行操作的最新应用是 MIT 的 Popa 等人设计的 CryptDB[37]。CryptDB 的核心思想是：以列为单位进行加密，根据不同的列可能会进行的操作分别是对其进行加密。具体表示为对不需要进行任何操作的列进行随机加密（也称为不确定加密），即相同的数值加密后的密文是不同的；对需要进行等值搜索的列进行确定加密，即相同的数值加密后的密文也是相同的；对需要进行范围搜索的列进行保序加密；对需要进行数学计算的列用全同态加密[38]。

Kamara 等人[39]提出了一个可搜索密文的云存储系统 CS2。CS2 密文云存储系统能够保证数据的机密性、完整性和可验证性且没有牺牲效率实用性，并且还能实现 2 种搜索模式即标准关键字搜索和辅助关键字搜索。CS2 主要由 3 个算法组成：可搜索对称加密（SSE）算法，搜索认证（SA）算法和完整性证明（PoS）算法。其中 SSE 用于构造索引实现对云中密文的关键字进行搜索，且不会泄露任何信息；SA 可以帮助用户判断返回的查询结果是否符合用户需求；PoS 能够帮助用户随时验证存储在云中的数据有无被篡改。

Kamara 等人[25]在 2012 年又最新提出了动态可搜索对称加密方案。该方案解决了目前所有 SSE 方案都不能同时满足的 3 个重要性质：搜索时间要尽可能地短；能够抗自适应选择关键字攻击；能够对索引进行动态的更新，即压缩和高效的增加和删除文件。该方案基于倒排索引的数据结构，是 SSE-1 的扩展[21,40]。该方案根据倒排索引分别生成搜索表 T_s、删除表 T_d、搜索数组 A_s 和删除数组 A_d。搜索和删除数组里的每一位存储的是一对值，分别表示关键字和关键对应的文档编号，在搜索数组里根据倒排索引生成对应的指针，将出现相同关键字的文档连接起来，在删除数组里根据正排索引将所有相同编号的文档用指针连接起来。数组里的指针用伪随机函数进行了加密。搜索表存储的是各关键字在 A_s 中的起始位置，删除表存储的是各文档在 A_d 中的起始位置，同样用伪随机函数（与加密指针的不同）进行了加密。查找时，先从 T_s 中解密得到关键字 W 在 A_s 里的起始位置，然后到 A_s 中根据指针指向把对应的所有包含关键字的文档搜索 W 出来。增加新文档时，先在 A_s 中找到空闲位，将新的对值放入空闲位，并生成新的指针指向包含相同关键字的文档。删除文档 F 时，先从 T_d 中找到 F 在 A_d 中的起始位置，然后从起始位置开始将指针连接的相同编号 F 的位置“0”，然后在 A_s 中将包含 F 的对应位置置为“free”，并且重新调整 A_s 中的指针指向。

Ferreira 和 Domingos[41]提出了一种将同态加密技术和动态索引机制结合起来的中间件(middleware)结构，能够实现安全地将数据存储在云中，保证数据的隐私性和可控性，支持在云计算环境中的密文搜索。该方案可以应用于异构的云中，返回搜索结果是进行相关性排序后的，便于用户选择，同时该方案还具有更高的安全性且不会带来过多的时间和经济开销。

6　结束语

云存储是当前发展前景十分广阔的新兴产业，但其也面临相当大的的安全问题。在云存储模式下，数据的所有者和对数据的操作是分开的，明文数据存储于不可信的云端，存在着相当大的安全隐患，所以用户数据必须以密文的形式存储在云端，但是这样云平台就沦为简单数据池，而无法发挥其各种云服务功能以及强大计算能力。因此需要有相关的技术能够对密文进行操作，要求在保证用户数据安全的同时能够实现对密文数据的高效搜索。因此，密文搜索技术应该得到广泛的重视。本文对经典的密文搜索技术进行了分类总结和说明，并以此为基础给出了云计算环境中密文搜索的体系结构，同时还给出了国内外在密文搜索领域中的最新研究成果，目前存在的问题和未来可能的研究方向。云存储和与之对应的密文搜索技术未来可以广泛地应用于存储和保护个人认证信息、政府或企业的财政数据、个人的电子医疗记录等。可以说，云存

储中的密文搜索技术的研究具有重要意义且有着广泛的应用前景。

参考文献：

[1] ARMBRUST M, FOX A, GRIFFITH R. Above the clouds: a berkeley view of cloud computing[EB/OL]. http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.html, 2009.

[2] BUGIEL S, NURNBERGER S, SADEGHI A. Twin clouds: an architecture for secure cloud computing[A]. Workshop on Cryptography and Security in Clouds[C]. Zurich, Switzerland, 2011. 1-11.

[3] KAMARA S, LAUTER K. Cryptographic cloud storage[J]. Fi ial Cryptogra phy and Data Security, 2010,6054:136-149.

[4] LI J, KROHN M, MAZI D. Secure untrusted data repositor (SUNDR)[A]. Proceedings of the 6th Symposium on Operat ng Systems Design and Implementation[C]. San Francisco, CA, USA, 2004. 91-106.

[5] GOH E J. Secure indexes[A]. Proceedings of the 2004 Workshop on Information Security Applications[C]. Jeju Island , Korea, 2004.73-86.

[6] SONG D X, WAGNER D, PERRIG A. Practical techniques or searches on encrypted data[A]. Proceedings of the IEEE Symposium on Security and Privacy[C].CA,USA,2000.36-49.

[7] BONEH D, CRESCENZO G D, OSTROVSKY R. Public key encryption with keyword search[A]. Proc of EUROCRYP'04[C]. Interlaken, Switzerland, 2004.506-522.7

[8] PAILLIER P. Public-key cryptosystems based on composite degree residuosity classes[A]. Proceedings of the International Conference on the Theory and Applications of Cryptograhic Techniques (EUROCRYPT'99)[C]. Prague, Czech Republic, 1999.223-238.

[9] BETHENCOURT J, SONG D, WATERS B. New constructions and practical applications for private stream searching (extended abstract)[A]. Proceedings of the IEEE Symposium on Security and Privacy (SP'06)[C]. Oakland, California, USA, 2006. 134-139.

[10] BETHENCOURT J, SONG D, WATERS B. New techniques for private stream searching[J]. ACM Transactions on Infor ion and System Security, 2009, 12(3):1-32.

[11] HACIGUMUS H, LYER B, MEHROTRA S. Providing database as a service[A]. Proceedings of the International Conference on Data Engineering(ICDE 2002)[C]. SAN JOSE, CA, USA, 2002. 29-38.

[12] HACIGUMUS H, LYER B, LI C. Executing SQL over encrypted data in the database-server-provider model[A]. Proceedings of ACM SIGMOD[C]. Madison, Wisconsin, USA, 2002. 216-227.

[13] HACIGUMUS H, LYER B, LI C. Efficient executing of aggre ion queries over encrypted relational database[A]. Proceed ngs of Database Systems for Advanced Applications(DASFAA 2004)[C]. Jeju Island, Korea,2004. 125-136.

[14] HACIGUMUS H, LYER B, LI C. Query optimization in encrypted database system[A]. Proceedings of Database Systems for Advanced Applications(DASFAA 2005)[C]. Beijing, China,2005. 216-227.

[15] HORE B, MEHROTRA S, TSUDIK G. A privacy-index for range queries[A]. Proceedings of the 30th VLDB Conference[C]. Toronto, Canada, 2004. 223-235.

[16] **王迪,刘国华,于醒兵.基于最佳桶划分策略的密文索引技术**[J].**小型微型计算机系统**,2008,29(4):649-652.

WANG D, LIU G H, YU X B. Cryptograph index technology based on strategy of optimal bucket partitioning[J]. Journal of Chinese Computer Systems,2008,29(4):649-652.

[17] AGRAWAL R, KIERNAN J, SRIKANT R. Order preserving encryption for numericdata[A]. SIGMOD2004[C]. Paris, France, 2004. 13-18.

[18] BLOOM B. Space/time trade-offs in hash coding with allowable errors[J]. Communications of the ACM,1970,13(7):422-426.

[19] CHANG Y C, MITZENMACHER M. Privacy preserving keyword searches on remote encrypted data[J]. Applied Cryptography and Network Security Lecture Notes in Computer Science, 2005, 3531: 442-455.

[20] BONEH D, KUSHILEVITZ E, OSTROVSKY R. Public-key encryption that allows PIR queries[A]. 27th Annual International Cryptology Conference[C]. Santa Barbara, CA, USA, 2007. 50-67.

[21] CURTMOLA R, GARAY J, KAMARA S. Searchable symmetric encryption: improved definitions and efficient constructions[A]. Proceedings of ACM Conference on Computer and Communications Security (CCS)[C]. Alexandria, VA, USA,2006. 79-88.

[22] CHASE M, KAMARA S. Structured encryption and controlled disclosure[A]. Proceedings of Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT 2010)[C]. Singapore, 2010.577-594.

[23] KUROSAWA K, OHTAKI Y. UC-secure searchable symmetric encryption[A]. Proceedings of Financial Cryptography and Data Security(FC)[C]. Bonaire, Dutch Caribbean, 2012. 285-298.

[24] LISEDONK V P, SEDGHI S, DOUMEN J. Computationally effic nt searchable symmetric encryption[A]. Proceedings of Workshop on Secure Data Management(SDM)[C]. Singapore, 2010. 87-100.

[25] KAMARA S, PAPAMANTHOU C, ROEDER T. Dynamic searchable symmetric encryption[A]. Proceedings of the 2012 ACM conference on Computer and Communications Security (CCS '12)[C]. New York, NY, USA, 2012.965-976.

[26] BALLARD L, KAMARA S, MONROSE F. Achieving efficient conjunctive keyword searches over encrypted data[A]. Proc of ICICS[C]. Beijing, China, 2005.414-426.

[27] BONEH D, WATERS B. Conjunctive, subset, and range quer es on encrypted data[A]. Proc of TCC[C]. Amsterdam, The Netherlands, 2007.535-554.

[28] BRINKMAN R, DOUMEN J, JONKER W. Using secret sharing for searching in encrypted data[J]. Secure Data Management, 2004, 3718:18-27.

[29] BRINKMAN R. Searching in Encrypted Data[D]. University of Twente, the Netherlands,2007.

[30] GOLLE P, STADDON J, WATERS B. Secure conjunctive keyword search over encrypted data[A]. Proc of ACNS[C]. Yellow Mountain, China, 2004. 31-45.

[31] HWANG Y H, LEE P J. Public key encryption with conjunctive keyword search and its extension to a multi-user system[A]. First International Conference[C]. Tokyo, Japan, 2007. 2-22.

[32] KATZ J, SAHAI A, WATER B. Predicate encryption supporting disjunctions, polynomial equations, and inner products[A]. Proc of EUROCRYPT[C]. Istanbul, Turkey, 2008.146-162.

[33] WANG C, CAO N, REN K. Enabling secure and efficient ranked keyword search over outsourced cloud data[J]. IEEE Transactions on