计算机研究与发展
Journal of Computer Research and Development
DOI:10.7544/issn1000-1239.2018.20180433
55(10):2149-2163,2018

双服务器模型下支持相关度排序的多关键字密文搜索方案

李宇溪　周福才　徐　剑　徐紫枫
(东北大学软件学院　沈阳　110819)
(eliyuxi@gmail.com)

Multiple-Keyword Encrypted Search with Relevance Ranking on Dual-Server Model

Li Yuxi, Zhou Fucai, Xu Jian, and Xu Zifeng
(*College of Software*, *Northeastern University*, *Shenyang* 110819)

Abstract　Focusing on the problem of confidentiality and availability of user data in cloud storage environment, we study the encrypted search method with multi-keyword. Aiming at the practical demand, we propose a multi-keyword encrypted search scheme with relevance ranking (MES-RR) in dual-server model, which can not only achieve secure multi-keyword encrypted search, but also ensure efficient search result sorting. We construct a relevance-based keyword index with the tools of TF-IDF weighting scheme and Paillier homomorphic cryptosystems, which not only obtains optimize computational complexity but also reduces storage complexity. We design a dual-server model architecture to perform the collaborated mechanism. Based on that, we design a secure sorting protocol between the two collaborated servers to sort the encrypted search results, which outputs private ranking result to user. In terms of security, we design the security model of MES-RR under honest but curious threat scenario, and give formal security analysis. The result shows that MES-RR can resist adaptive chosen keyword attacks under the random oracle model (IND-CKA2). The performance analysis shows that compared with the previous multi-keyword encrypted search scheme that supports result sorting, MES-RR reduces the storage cost and interactions, and is applicable to the cloud storage environment in the real world.

Key words　encrypted search; relevance ranking; dual-server; homomorphic cryptosystems; adaptive chosen keyword attack

摘　要　围绕当前云存储环境中用户数据机密性以及可用性的问题,对多关键字密文搜索方案展开研究,提出双服务器模型下支持相关度排序的多关键字密文搜索方案(multi-keyword encrypted search with relevance ranking, MES-RR),在能够保证高效地实现多关键字密文搜索的同时,实现对于搜索结果的排序.方案基于 TF-IDF 加权技术并融合 Paillier 同态加密体制,构建关键字相关度安全索引,优化计算代价并降低了存储复杂度;设计双服务器模型架构,引入安全可信的协同处理机制来构造安全排序协议,实现对于搜索结果的高效排序.在安全性方面,在诚实与好奇的威胁场景下构建方案的安全模型,

收稿日期:2018-06-10;**修回日期**:2018-08-13
基金项目:国家自然科学基金项目(61772127,61472184);国家科技重大专项基金项目(2013ZX03002006);辽宁省科技攻关项目(2013217004);中央高校基本科研业务费专项资金项目(N151704002)
This work was supported by the National Natural Science Foundation of China (61772127, 61472184), the Major National Science and Technology Program (2013ZX03002006), the Liaoning Province Science and Technology Projects (2013217004), and the Fundamental Research Funds for the Central Universities (N151704002).
通信作者:周福才(fczhou@mail.neu.edu.cn)

并对安全性进行严格分析,结果表明方案能够在随机预言模型下抵抗自适应性选择关键字攻击,具有IND-CKA2安全性.性能分析表明:该方案用户生成q个关键字搜索令牌仅需要常数级时间$O(q)$,而且仅需和服务器进行1次交互即可得到搜索结果,与以往的支持排序的多关键字密文搜索方案相比,该方案大大降低了计算代价和访问交互次数,适用于实际的云存储环境.

关键词 密文搜索;相关度排序;双服务器;同态加密;自适应性选择关键字攻击

中图法分类号 TP391

密文搜索是指当数据以加密形式存储在设备中时,在确保数据安全的前提下检索数据的技术.因为数据加密后会严重影响用户数据的可用性,因此如何在保证用户隐私和数据安全性的前提下,确保密文数据的可用性,就成为了密文搜索领域的重要研究内容.总体来说,密文搜索主要包括两大实体:用户和服务器.以及2个步骤:初始化阶段和搜索阶段.在初始化阶段,用户在客户端加密文件集合与索引数据结构,并上传给服务器加密后的文件集合和一个加密索引;在搜索阶段,用户想要服务器返回包括某关键字的所有文件.为了执行这一请求,用户发送一个令牌,这一令牌能够封装这一搜索请求,并且不会泄露任何关于关键字的信息.服务器收到用户发送的令牌后,用这个令牌以及加密索引,通过某种方式返回符合搜索条件的加密文件.

当前,密文搜索方案大都基于单服务器模型,用户将密文数据存储在单个云服务提供商中.受限于预计算量较大、交互轮数较多等实际缺陷,云服务器只能进行简单的搜索请求,大大影响了密文搜索方案的可用性.这种情况下云服务器不能在完全在保证用户数据隐私的前提下,对密文搜索功能进行丰富完善,使其真正投入企业级大规模应用收到了限制.实际情况下,云服务提供商并不仅限于单个实体,可能协同服务提供商同时提供服务.多服务器模型在近几年很多相关工作已被广泛使用[1-3].由于云服务提供商之间存在一定的竞争关系,因而不会为了短期利益而勾结在一起攻击用户隐私数据.在现实的应用场景中,协同云服务通常由大公司来提供,例如亚马逊、微软和谷歌等这些公司不会因为商业利益来与云存储服务器工商进行合谋,这不仅会破坏他们的声誉,从长远角度来看也会影响其商业利润和公司发展.

然而,很多情况下,粗粒度的返回全部搜索结果并不能满足用户的需求.在数据量较大的情况下,用户在进行多关键字搜索时,不仅需要获得匹配到的文件,与此同时希望服务器根据其与搜索关键字的相关度,对搜索的文件进行排序,从而更有效地得到想要的结果.然而,多数密文搜索方案都是将匹配到关键字搜索条件的全部搜索结果返回给用户,并没有搜索结果进行筛选或排序,尤其是在多关键字搜索场景下.虽然现在已有相关工作试图解决支持排序的密文搜索问题,但其基于较高的系统模型假设(例如,引入可信第三方)或较低的安全目标(例如,允许服务器获得关键词信息)或由于通过服务器重加密或其他复杂运算而导致效率较低.因此,完善现有的支持结果排序的密文搜索机制,是未来需要研究的一个重要方向.

针对上述问题,并结合实际应用场景中对多关键字密文搜索的可用性需求及安全性需求,面向双服务器模型,提出了支持相关度排序的多关键字密文搜索方案,在能够保证安全高效地实现密文搜索的同时,满足对于搜索结果的排序.

1) 方案基于TF-IDF加权技术并融合同态加密体制,构建包含相关度权值的关键字安全索引,优化搜索过程中服务器计算代价并降低了存储复杂度.方案利用TF-IDF加权规则来计算关键字与文件之间的相关度权值,并利用Paillier同态加密算法将其加密,使得服务器在搜索过程中,无需对令牌及密文进行解密的前提下能够对权值密文进行相加,从而获得多关键字权值密文之和,满足多关键字搜索请求下对结果的排序.

2) 构建安全可信的协同处理机制实现对于搜索结果安全排序.方案构建2个不可共谋的服务器S和coS来共同扮演服务提供者的角色.其中S负责存储密文和索引信息,coS负责协助S来进行搜索结果排序工作.方案引入两方安全密文比较协议,在不向协同服务器泄露搜索模式与访问模式的前提下,实现对于搜索结果的高效排序.与单服务器模型相比,降低了用户与服务器之间通信代价的同时,减少了向服务器泄露的消息.

3) 在安全性方面,在诚实并好奇的半可信场景下给出MES-RR方案的安全模型,并依据所提出的

安全模型对该方案的安全性进行严格证明,证明方案在随机语言模型下能够抵抗敌手自适应选择关键字攻击,满足 IND-CKA2 安全性. 性能分析表明,与以往的支持搜索结果排序的多关键字密文搜索方案相比,本方案大大降低了用户存储代价和访问交互次数,适用于实际的云存储环境.

1 相关工作

1.1 通用密文搜索方案

密文搜索由 Song 等人在 2000 年首次提出[4],作者使用了一个特殊的 2 层的加密构造,将明文文件划分为"单词"并对其分别加密,通过对整个密文文件扫描和密文单词进行比对,就可确认关键词是否存在,甚至统计其出现的次数,然而这导致了搜索复杂度较高,与文件集合的大小呈线性关系. 之后,密文搜索方案按照实体不同分为对称和非对称 2 种应用场景. 在对称场景中,数据拥有者加密自己的数据. 加密的数据和对应的数据结构被存储在服务器中,只有数据拥有者以及被数据拥有者认证的用户才能够对其进行搜索. Curtmola 等人在 2006 年对对称场景的密文搜索模型进行了形式化定义,并基于倒排索引提出了一个高效的构建方案[5],方案只需常数级时间即可完成搜索操作,然而,执行文件的添加或删除操作需要重新构建索引,时间开销较大. 除此之外,他们提出了基于广播加密技术的多用户密文搜索构造思想,即使只有单一数据拥有者,仍然可以允许任意被授权的用户,通过向数据拥有者请求搜索令牌,来向服务器发送搜索请求. 另一个对称场景的经典方案是 Kamara 等人在 2012 年提出的支持搜索结果完整性验证的密文搜索方案[6],方案利用基于双线性映射累加器和扩展欧几里德算法的集合证明理论,实现对于搜索结果的离线验证. Boneh 等人在 2004 年首次提出非对称场景的密文搜索方案[7],并且他们的创新性思想为后来很对研究成果提供了启发. 在非对称场景中,数据拥有者将数据加密并上传至服务器,任何拥有解密密钥的用户可以执行搜索请求(例如邮件服务器场景).

在很多情况下,用户想要搜索包含多个关键字信息的文件,基于此需求,Golle 等人首先提出了多关键字密文搜索机制并给出了具体构造方案[8],但是方案中搜索令牌的大小与加密文件数呈线性关系,用户计算代价过高. Cash 等人在 2013 年提出了第一个面向任意结构化数据的支持交集的密文搜索方案[9],并证明方案满足 IND-CKA2 安全性. 2014 年,Cash 等人提出了一个支持连接词查询的高效密文搜索方案[10],方案针对数据集中关键字的出现频率设计 3 层安全索引结构,使搜索过程更具有灵活性,更适合大数据场景. 2017 年,Wang 等人提出了面相多关键字的模糊密文搜索方案[11],方案不需要提前设置索引存储空间,从而大大降低了方案的搜索复杂度.

1.2 支持排序的密文搜索方案

在通用密文搜索方案中,用户必须解密所有返回的搜索结果,才能得到所需文件. 然而,在数据量较大的情况下,查询结果中可能仅仅有极少部分为用户所需文件. 支持排序的密文搜索方案可以按照关键词与文件的相关度对搜索结果进行排序,并返回最相关的文件来优化搜索结果,用户只需根据排序结果解密少量加密文件即可获得所需文件,这能在一定程度上降低网络负载并且增强系统可用性.

早期支持排序的密文搜索方案[12-14]大多利用保序加密方法(order-preserving encryption, OPE)实现了针对单关键字密文搜索结果的排序. 用户在上传文件之前,使用 OPE 对关键字相关性分数加密,搜索过程中服务器根据每个文件对搜索关键字相关性分数的密文结果进行排序,并将经过排序后的前 k 个文件返回给用户. 这种方法的优点是,在每次搜索的过程中,用户可以得到在特定相关度权衡模型下对该关键字最为相关的文件,节省了带宽和因为解密带来的计算开销,但是服务器也掌握了每个文件对于某个关键字相关性程度的信息.

Cao 等人在 2014 年首次提出支持搜索结果排序的多关键字密文搜索方案[15],方案中基于内积相似性(inner product similarity, IPS)来计算每个文件中匹配到的关键字的数量,以此来计算文件权值并对文件进行排序. 内积相似性中文件的权值与每个文件中匹配的关键字的个数相关. 但是仅仅依赖 IPS 技术,使得方案没有考虑关键字对于整个文件集合的重要性等信息. TF-IDF 加权技术为信息检索领域常用来评估某个关键词对于一个文件集或一个语料库中的其中一份文件的重要程度的方法. 许多支持排序的密文搜索方案基于 TF-IDF 加权技术来实现:Strizhov 等人在 2014 年融合 TF-IDF 加权技术和 IPS 设计了新型密文搜索排序方案[16],为了防止服务器获取相关度信息,方案中假设服务器仅仅执行搜索功能,并不能对结果进行排序,服务器将搜索到的全部文件返回给用户,用户在本地基于

相关度权值对搜索结果进行排序. 方案的不足之处在于,为了实现排序,相关度由全同态加密方案进行加密,使得构建索引阶段时间复杂度较高,而且方案不支持文件动态更新. 同年,Sun 等人利用余弦度量技术设计一个支持排序的可验证密文搜索方案[17]. 他们的方案能够在稍稍牺牲搜索准确度的前提下,提高搜索效率. Zhang 等人提出了一个多数据拥有者模型下的支持排序的密文搜索方案[18]. 不同的数据拥有者利用不同的私钥来加密文件和关键字,经过认证的用户能够在不知道拥有者私钥的前提下发起查询. 2015 年,Fu 等人提出了支持并行搜索的可排序密文搜索方案[19],方案利用向量空间模型构造索引,模型中将每一个 TF-IDF 权值转化成 2 个随机向量搜索过程中将搜索向量转化成矩阵乘积形式,并且计算搜索向量与数据库中所有密文文件向量的点相似度来判定相关性.

Orencik 等人在 2016 年提出了支持排序的多关键字密文搜索方案[20],他们利用位置敏感 Hash 函数(location sensetive Hash, LSH)来聚类相似文件. 然而,LSH 适用于相似性搜索但是不能提供精确排序. 同文献[15]类似,他们的排序机制仅仅依赖文件中关键字的出现频率,导致搜索准确性较差,具有一定误差. 同年,Xia 等人提出了基于树结构的动态多关键字密文搜索方案[21],论文利用向量空间模型构建树型索引结构,并利用深度优先搜索方案进行关键字搜索. 2017 年,Song 等人提出支持全文检索的密文搜索方案[22],他们使用布隆过滤器(bloom filter)作为加密索引,并且提出了索引关键字熵的概念来计算搜索令牌中关键字和文件的相关度. 同年,Shen 等人提出了一个基于 RSA 和 ElGamal 加密算法的相关度排序方案[23]. 方案利用高阶聚类索引(hierarchical clustering index)构造索引结构来提高搜索效率. 为了提高安全性,方案中权值的计算由服务器执行,结果的排序由用户执行,大大提高了搜索效率. 在上述方案中,当文件集合的大小呈指数增长时,搜索时间也成线性增长.

2 预备知识

2.1 同态密码体制

本方案的构建基于同态密码体制,主要应用 Paillier[24] 以及 Goldwasser-Micali(GM)加密方法[25]. 上述 2 个方法均支持加法同态性,即给定 2 个密文 $Enc(m_1)$ 和 $Enc(m_2)$,应用加同态操作可以得到 2 个明文相加之后的密文 $Enc(m_1+m_2)$.

详细来讲,Paillier 同态加密算法由 3 个函数组成:

1) $P.Gen(1^k)$. 输入安全参数 1^k,随机选择 2 个 k 位素数 P 和 Q,并置 $N=PQ$. 随机选择一个数 g,满足 $\gcd(L(g^{\lambda(N)} \bmod N^2), N)=1$,其中函数 $L(u)=(u-1)/N$,$\lambda(N)=\mathrm{lcm}(P-1,Q-1)$,选择一个随机数 $a\in_R \mathbb{Z}_{N^2}^*$,令 $h=a^N \bmod N^2$,可知 h 在模 N^2 下的阶为 $\lambda(N)$. 加密系统公钥 PK_P 为 (N,g,h),私钥 SK_P 为 (P,Q).

2) $P.Enc_{PK_P}(m)$. 输入消息 $m\in\mathbb{Z}_N$,选择随机数 $r\in\mathbb{Z}_N$,计算密文 $[m]=g^m h^r \bmod N^2$.

3) $P.Dec_{SK_P}([m])$. 输入密文 $c\in\mathbb{Z}_{N^2}^*$,计算明文 $m=(L([m]^{\lambda(N)} \bmod N^2)/L(g^{\lambda(N)} \bmod N^2)) \bmod N$.

除此之外,Lipmaa 等人在 2005 年提出 Paillier 双重加密的方法[26],设 $\|m\|$ 为双重加密之后的密文,其中明文 m_1 满足 $m\in\mathbb{Z}_{N^2}$,则 Paillier 双重加密系统满足如下性质:$\|[m_1]\|^{[m_2]}=\|[m_1][m_2]\|=\|[m_1+m_2]\|$.

同样,GM 加密算法包含 3 个函数:

1) $GM.Gen(1^k)$. 输入安全参数 1^k,选择 2 个随机 k-bit 素数 P 和 Q,并置 $N=PQ$. 选取模 n 的一个二次非剩余 $\delta\in\mathbb{Z}_n^*$ 满足雅可比符号 $\left(\frac{\delta}{n}\right)=1$. 公钥 $PK_{GM}=(\delta,n)$,私钥是 $SK_{GM}=(P,Q)$.

2) $GM.Enc_{PK_{GM}}(m)$. 输入消息 $m\in(0,1)$,选择随机数 $r\in\mathbb{Z}_N$,计算密文 $\|m\|=r^2\delta^m \bmod n$.

3) $GM.Dec_{SK_{GM}}(\|m\|)$. 输入密文 $\|m\|\in\mathbb{Z}_{N^2}^*$,计算明文

$$m=\begin{cases}0, & \|m\| \text{ 是 } n \text{ 的二次剩余};\\ 1, & \|m\| \text{ 不是 } n \text{ 的二次剩余}.\end{cases}$$

GM 算法的明文空间为 1 位,并且 $[m]$ 指的是一位比特值 m 利用 GM 方法加密之后的密文. GM 的密钥对表示为 $K_{GM}=(PK_{GM},SK_{GM})$. GM 方法的同态性表示为 $\|m_1\|\|m_2\|=\|m_1\oplus m_2\|$.

2.2 TF-IDF 加权技术

为了能够实现搜索结果的相关度排序,本文使用 TF-IDF 加权技术[27]来计算某次搜索请求中匹配文件的相关度权值. TF-IDF 加权技术是一种用于信息检索与数据挖掘的常用加权技术,用以评估某个关键词对于一个文件集或一个语料库中的其中一份文件的重要程度. TF-IDF 加权技术主要基于

2 个独立的参数:词频(term frequency, TF)以及逆向文件频率(inverse document frequency, IDF).

定义 1. 词频. 词频指的是某一个给定的关键字 t_i 在某个文件 d 中出现的频率,可表示为

$$tf_{t_i,d}=n_{t_i,d}\Big/\sum_{j=1}^{k}n_{t_j,d},\tag{1}$$

其中,$n_{t_i,d}$表示关键字 t_i 在文件 d 中出现的次数,$\sum_{j=1}^{k}n_{t_j,d}$表示文件 d 中所有关键字出现的次数之和.

定义 2. 逆向文件频率. 逆向文件频率是一个词语在整个文件集中的普遍重要性的度量,可表示为

$$idf_{t_i}=\mathrm{lb}(|D|/\{j:t_i\in d_j\}),\tag{2}$$

其中$|D|$表示总文件数目,j 表示含有关键字 t_i 的文件数目. 本文采用式(3)来权衡关键字 t_i 在文件 d 中的相关度:

$$tf\text{-}idf_{t_i,d}=\mathrm{lb}(tf_{t_i,d}+1)\times\mathrm{lb}(1+n/idf_{t_i}).\tag{3}$$

2.3 两方安全密文比较协议

本文利用文献[28]中的两方安全密文比较协议来作为方案排序协议构建模块. 协议的交互双方中一方持有 2 个利用 Paillier 加密方案加密的密文,其需要在不解密的情况下比较 2 个密文对应的明文值的大小. 协议要求在比较过程中不泄露任何关于明文的信息. 现如今的方法中不存在有效的方案使其能够在上述调前的情况下比较 2 个密文的大小. 因此,需要另一方与其配合进行比较协议.

详细来说,假设 A 和 B 为交互双方,其中 A 拥有密文$[a]\leftarrow P.Enc_{PK_P}(a)$和$[b]\leftarrow P.Enc_{PK_P}(b)$,$B$ 拥有私钥 SK_P. 比较协议的目标是在不向任意一方揭示 a,b 明文的情况下,判定 2 个长度为 l 位的数值 a,b 的大小. 协议主要思想是计算 $2^l+b-a(l+1)$ 并验证第 $l+1$ 位比特值来判定 a,b 的大小. 如果等于 1,则 $a\leqslant b$,否则 $a>b$. 协议结束之后,A 得到比较结果的密文,而 B 不会得到任何有用信息. 方案在诚实并好奇敌手模型下被证明满足语义安全.

2.4 Batcher 排序网络

Batcher 排序网络[29]能够将一个包含 N 个元素的数组进行高效排序,可看作是一个具有 $O((\log N)^2)$个连贯层级的排序算法,每个层级要比较 $O(N)$对元素. 具体来说,如图 1 所示,设 A_i 是第 i 层的序列,其中 $A_i\{j\}$为序列 A_i 的第 j 个元素. 每一个层级 i 输入 A_i,输出 A_{i+1},其中 i 层被指定需要排序的元素对在 A_{i+1}中的是有序的. 例如 $A_2\{1\}$和 $A_2\{2\}$包含有序的 $A_1\{1\}$和 $A_1\{2\}$,$A_2\{3\}$和 $A_2\{4\}$包含有序的 $A_1\{3\}$和 $A_1\{4\}$. $pair_i$ 表示层级 i 中需要排序的元素对集合,$pair_i.next$ 表示下一个需要排序的元素对集合.

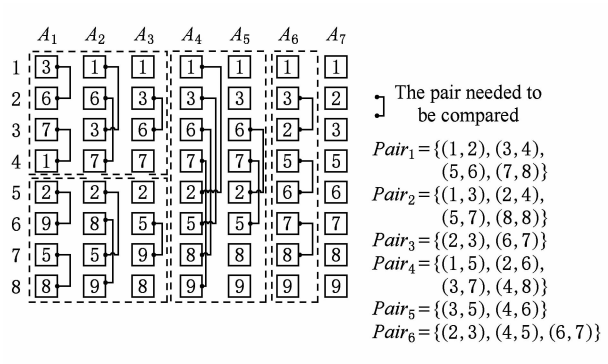


Fig. 1 An illustration of Batcher sort procession

图 1 Batcher 排序实例

图 1 为 Batcher 排序网络实例,输入序列 $A_1=\{3,6,7,1,2,9,5,8\}$,首先将序列分为前半部分和后半部分,2 个子序列同时进行奇偶排序,最终得到 $A_4=\{1,3,6,7,2,5,9,8\}$. 随后,对整个序列进行奇偶归并,得到 $A_6=\{1,3,2,5,6,7,8,9\}$,最后将相邻元素进行邻近排序,得到最终的排序之后的序列 $A_7=\{1,2,3,5,6,7,8,9\}$. 本文利用 Batcher 排序网络来实现搜索结果排序.

3 MES-RR 方案

MES-RR 由 3 类实体构成:用户 U、云存储服务器 S 以及协同服务器 coS. 其中,用户 U 为数据拥有者,云存储服务器 S 为向用户提供数据存储服务,根据用户特搜索定请求完成对应的搜索操作,协同服务器 coS 为辅助服务器,协助云存储服务器 S 完成多关键字搜索结果的排序.

在本场景中,数据以文件的形式进行组织,每个文件都带有若干个对应的关键字. 使用“令牌”来描述用户的请求. 例如,在进行搜索时,用户 U 首先生成一个搜索令牌,并发送给服务器 S. 服务器 S 使用该令牌作为输入来执行搜索算法. 随后服务器 S 在协同服务器 coS 的协助下完成搜索结果的排序. 由于方案只关注文件的标识和关键字索引,因此文件的类型与大小不会对性能产生影响.

3.1 方案模型

由图 2 可以看出,方案主要包含 4 个阶段:文件预处理阶段、搜索令牌生成阶段、搜索排序阶段以及搜索结果返回阶段. 用户 U 将文件和索引加密后,外包至服务器 S 端进行存储,用户 U 选择若干关键字进行集合搜索时,根据关键字集合生成对应的搜索

令牌发送给服务器 S,服务器 S 在加密索引上执行搜索操作,得到每个关键字对应的密文集合,对密文集合进行对应的集合操作,并与协同服务器 coS 共同进行搜索结果排序.随后服务器 S 将排序后的搜索结果返回给用户,用户可以根据该搜索结果来选择相应的文件.

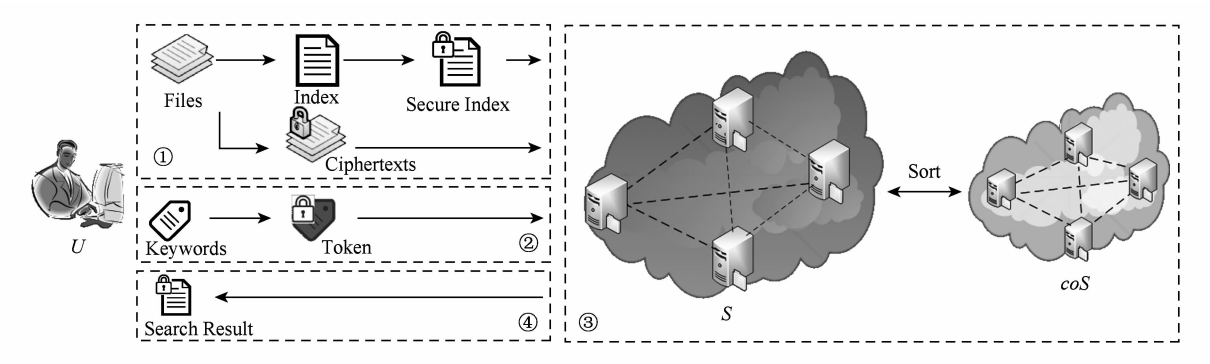


Fig. 2 Scheme architecture

图 2 方案架构

3.1.1 形式化定义

MES-RR 由 6 个多项式时间算法或协议组成,即 $MES\text{-}RR=(KeyGen, Setup, SrchToken, Search, Sort, Dec)$,具体描述如下:

1) 密钥生成算法 $(K,SK,PK)\leftarrow KeyGen(1^k)$.由用户 U 执行,生成用户对称密钥 K 以及公私钥 PK,SK.

2) 初始化算法 $(\gamma,c)\leftarrow Setup(K,PK,F)$.由用户 U 执行,用于初始化安全索引结构并对要上传的数据进行加密.算法输入密钥 K,PK 和数据文件集合 F,输出加密索引 γ 和密文文件集合 C.

3) 搜索令牌生成算法 $\tau\leftarrow SrchToken(K,W)$.由用户 U 执行,用于生成搜索令牌 τ.算法输入密钥 K 和关键字集合 W,输出搜索令牌 τ.

4) 搜索算法 $I_W\leftarrow Search(\gamma,T)$.服务器 S 执行,用于实现搜索.算法输入加密索引 γ 和搜索令牌 τ,输出文件标识集合 I_W.

5) 排序协议 $(S:\tilde{I}_W)\leftarrow Sort(S:I_W,PK;coS:SK)$.由服务器 S 和协同服务器 coS 之间交互运行,用于对搜索结果进行排序.在协议中,服务器 S 输入公钥 PK、搜索结果 I_W;协同服务器输入私钥 SK.协议结束后,服务器得到将 I_W 按相关度排序的重加密序列 $\tilde{I}_W$.

6) 解密算法 $(id_1,id_2,\cdots,id_m)\leftarrow Dec(K,\tilde{I}_W)$.由用户 U 执行,用于对接收到的搜索结果进行解密.算法输入密钥 K 和搜索结果 $\tilde{I}_W$,输出按相关度排序的文件标示符序列 $id_1,id_2,\cdots,id_m$.

方案的正确性意味着,对于所有的安全参数 k、所有由 $KeyGen(1^k)$ 生成的 K,SK,PK、所有的 F 和 δ、所有由 $Setup(K,F)$ 输出的 (γ,C),对于所有由 $\tau\leftarrow SrchToken(K,W)$ 生成的搜索令牌 τ,搜索算法 $I_W\leftarrow Search(\gamma,C,\tau)$ 都会得出正确的搜索结果 I_W,并且经过排序协议 $Sort(S:I_W,PK;coS:SK)$ 总会得到有序的搜索结果 $\tilde{I}_W$,意味着用户可以通过解密算法 $Dec(K,\tilde{I}_W)$ 对序列解密成功,得到按搜索关键字相关度排序的文件标识符序列 $id_1,id_2,\cdots,id_m$.

3.1.2 安全模型

与经典的密文搜索方案的安全模型不同,MES-RR 的安全模型中包括 2 个不合谋的服务器:服务器 S 以及协同服务器 coS,均为满足“诚实但好奇”模型的半可信实体,即具有可以分析用户的搜索过程的能力,但是不会试图共谋来获取用户数据.服务器 S 会诚实地为用户提供文件存储服务,并且不会主动更改或损坏用户的数据,但可能试图查看、分析或使用用户的数据、索引和令牌信息;协同服务器 coS 配备了用于保存密钥的密码处理器,仅仅进行辅助工作,不会获得任何有用数据,并且 2 个服务器之间不会共谋.具体来说,在 MES-RR 安全模型中,S 得到加密的文件集合以及安全索引的情况下,响应用户发送的搜索请求,并通过的 coS 的协助,对搜索结果按照相关度进行排序,并返回给用户.在上述过程中,S 能够学习到文件的个数以及在倒排索引中关键词的个数,以及在多项式次搜索过程中用户的每一次搜索令牌信息.coS 只能学习到文件的个数以及用户进行搜索请求的时间.然而,coS 不会得到任何有关搜索内容的信息.

文献[5]曾提到,目前已知的大多数高效的密文搜索方案均或多或少泄露了用户的私有信息,诸如用户的搜索模式访问模式等.因此在本方案中,将方案在执行的过程中用户向服务器所泄露的信息抽象

成一组泄露函数 $\mathcal{L}_1$,$\mathcal{L}_2$ 和 $\mathcal{L}_3$ 来描述在方案执行期间向服务器 S 和协同服务器 coS 泄露的信息,分别代表初始化过程,搜索过程以及排序协议中泄漏的信息.并在这些泄露函数存在的条件下给出安全性的定义.其主要目标是证明,所有具有多项式时间计算能力(probabilistic polynomial-time, PPT)的敌手都无法从数据和查询中获取到任何信息,除了在泄露函数中所刻画的之外.以下为方案的安全性定义.

定义 3. IND-CKA2 安全性.设 A_S 和 A_{coS} 为 2 个自适应敌手,分别代表方案中的服务器 S 与协同服务器 coS,自适应表示敌手可以根据之前的质询结果来发起新的质询.$view$ 为敌手在实验过程中所能够收集到的所有信息.Sim_1 和 Sim_2 为 2 个具有多项式时间计算能力的模拟器,$\mathcal{L}_1$,$\mathcal{L}_2$ 和 $\mathcal{L}_3$ 为方案向 A_S 和 A_{coS} 泄露的信息构成的泄漏函数.

设 $Real_A(k)$ 为敌手 A_S 和 A_{coS} 和用户之间的交互式实验,使用真实的方案算法.在该实验中,用户 U 随机选择 $b\in(0,1)$,A_S 向用户发送 2 个拥有相同的文件数目和相同的关键字数的文件集合 F_0,F_1,用户 U 随机选择 $b\in(0,1)$,生成 F_b 的安全索引 γ_b 并发送给 A_S.随后,A_S 向用户发起至多 n_q 次的质询,其中 n_q 为多项式级:每次质询 i 中,A_S 选择 2 个长度相等的多关键字搜索请求 $W_0^{(i)}$,$W_1^{(i)}$ 发送给用户,满足 $|W_0^{(i)}|=|W_1^{(i)}|$,用户生成 $W_b^{(i)}$ 的搜索令牌 $\tau_b^{(i)}$ 并返回给敌手 A_S,A_S 执行搜索算法 $Search(\gamma_b,\tau_b^{(i)})$ 得到搜索结果 $I_{W_b}^{(i)}$.随后,A_S 与 A_{coS} 执行排序协议 $Sort(A_S:I_{W_b}^{(i)},PK;A_{coS}:SK)$ 将结果进行排序,生成 $I_{W_b}^{(i)}$.由于 A_S 是自适应敌手,所以其每次质询的结果都可以作为输入来产生下一次质询.在所有 n_q 次质询结束后,将 A_S 得到的信息记做 $view_{A_S}$.A_{coS} 仅参与与 A_S 的交互环节,并从中得到的 $view_{A_{coS}}$.实验结束后,A_S 输出 $Output_{A_S}$,A_{coS} 输出 $Output_{A_{coS}}$.

设 $Ideal_{Sim}(k)$ 为模拟器 Sim_1,Sim_2 和用户 U 之间的交互式实验,该实验与 $Real_A(k)$ 实验的区别是,在 $Ideal_{Sim}(k)$ 中,模拟器不运行真实方案中的算法,而是利用泄漏函数 $\mathcal{L}_1$,$\mathcal{L}_2$ 和 $\mathcal{L}_3$ 来作为唯一输入,来生成随机化的数据来回应用户搜索请求.每次质询 i 中,Sim_1 选择 2 个长度相等的多关键字搜索请求 $W_0^{(i)}$,$W_1^{(i)}$ 发送给用户,满足 $|W_0^{(i)}|=|W_1^{(i)}|$,用户生成 $W_b^{(i)}$ 的搜索令牌 $\tau_b^{(i)}$ 并返回给 Sim_1,Sim_1 执行搜索算法 $Search(\mathcal{L}_2,\tau_b^{(i)})$ 得到搜索结果 $I_{W_b}^{(i)}$.随后,Sim_1,Sim_2 利用泄漏函数执行排序协议将结果进行排序,生成 $I_{W_b}^{(i)}$.在所有 n_q 次质询结束后,将 Sim_1 得到的信息记做 $view_{Sim_1}$.Sim_2 仅参与与 Sim_1 的交互环节,将 Sim_2 得到的信息记做 $view_{Sim_2}$.实验结束后,Sim_1 输出 $Output_{Sim_1}$,Sim_2 输出 $Output_{Sim_2}$.

我们称 MES-RR 方案满足 IND-CKA2 安全性,当且仅当用户不能区分与其交互的为真实的敌手还是模拟器,即当且仅当以下 2 个条件成立:

1) 对于敌手 A_S,存在一个多项式时间的模拟器 Sim_1 满足 2 个分布式计算不可区分性:$Output_{A_S}^{\text{Real}}\approx Output_{Sim_1}^{\text{Ideal}}$.

2) 对于敌手 A_{coS},存在一个多项式时间的模拟器 Sim_2 满足 2 个分布式计算不可区分性:$Output_{A_{coS}}^{\text{Real}}\approx Output_{Sim_1}^{\text{Ideal}}$.

3.2 MES-RR 详细设计

本节给出了 MES-RR 的具体构建方案.在 MES-RR 中,文件集合 F 和倒排索引 δ 为初始输入.与文件索引不同的是,倒排索引是一种以关键字为起始的索引结构.每个关键字后面都存在一组包含该关键字的文件集合.方案中用到了多种数据结构,如链表 L、数组 A 和查找表 T.值得说明的是,方案中所用到的查找表 T 是一种以 $key/value$ 对为基本元素的数据结构,其元素的形式为 (σ,v),其中 σ 表示键,v 表示对应于键 σ 的值.记号 $T[\sigma]$ 表示在查找表 T 中键 σ 所对应的元素值,$T[\sigma]=v$ 表示将键 σ 的元素值置为 v.记号 $\sharp T$ 表示查找表 T 中的元素个数.

初始化阶段用户生成用于加密文件以及生成安全索引结构使用的密钥 K,以及 Paillier 同态加密算法的公私钥 (PK_P,SK_P) 和 GM 加密算法的公私钥 $(PK_{\text{GM}},SK_{\text{GM}})$.随后将用户的批量文件加密并生成加密索引,随后将密文文件 C,加密索引 γ 以及公钥 (PK_P,PK_{GM}) 一同上传至服务器 S,并将密钥对 (PK_P,SK_P) 和 $(PK_{\text{GM}},SK_{\text{GM}})$ 利用密钥分发算法分享给协同服务器 coS.

随后用户构建文件索引结构,首先从他的文件集 F 中提取到所有的关键字集合 W,计算每个关键字 w 在文件集 F 中针对所有文件的 $tf\text{-}idf$ 相关度权值,并生成倒排索引 δ,δ 中每个关键字对应一个长度为 N 的数组 F_w,数组 F_w 中的位置 d 中的元素代表了关键字 w 在文件 d 中相关度权值,如果权值为 0 则表示关键字 w 不存在文件 d 中.

δ 中列的个数为 M(w 中关键字的个数)行数为 N(F 中文件的个数).为了满足后续密码学运算,用户将每一个相关度权值映射到一个空间为 Z_n 的

整数值 $score$ 之中，并将其用 Paillier 进行加密(明文空间为 Z_n)。因为 Paillier 密文具有不可区分性，所以倒排索引 δ 不泄露任何关于关键字的文件个数的信息。随后，用户根据倒排索引 δ 构造安全索引 γ，安全索引 γ 包含 2 个数据结构：查找表 T 以及搜索数组 A。查找表 T 包含了系统中的所有关键字，同时表中的每个关键字都指向搜索数组中的一个链表。而在搜索数组 A 中，元素是以“文件/关键字”对的形式出现的，每个元素都是一个唯一的“文件/关键字”对，并通过以关键字为引导的链表进行组织具体来说，令 F,G 和 P 为 3 个伪随机函数，H 为 Hash 函数。$K=(K_1,K_2,K_3)$ 为用户生成的对称密钥。针对用户的文件集合 F 和倒排索引 δ，对其中的每个关键字 w 构建一个链表 L_w。每个链表都由 $\#F_w$ 个节点 $(N_1,N_2,\cdots,N_{\#f_w})$ 构成，并且存储在搜索数组 A 中的随机位置。定义节点 N_i 的结构为 $N_i=(\langle id_i,[score_i],addr(N_{i+1})\rangle\oplus H(P_{K_3}(w),r_i),r_i)$，其中 id_i 为文件 f_i 的唯一标识，文件中包含对应的关键字，$[score_i]$ 是关键字 w 在文件 f_i 中的相关度权值的 Paillier 密文：$[score_i]\leftarrow P.Enc_{PK_p}(score_i)$。$addr(N)$ 表示节点 N 在数组 A 中的位置。链表 L_w 中的所有节点都使用 $H(P_{K_3}(w),r_i)$ 进行异或加密，其中 r_i 为储存在节点中的随机值，K_3 为伪随机函数 P 的密钥。数组 A 中未使用的节点都随机利用 0 或 1 进行填充。

对于每个关键字 w，查找表 T 中的对应元素以 $F_{K_1}(w)$ 为键，以指向链表 L_w 头节点的指针为值，其中 K_1 为伪随机函数的密钥。指针使用 $G_{K_2}(w)$ 进行异或加密，其中 K_2 是伪随机函数 G 的密钥。即：$T[F_{K_1}(w)]=addr(N_1)\oplus G_{K_2}(w)$。值得说明的是，数组 A 和查找表 T 共同作为安全索引存储至服务器 S 端。文件利用标准加密算法进行加密，生成文件密文集合 $C=\{c_1,c_2,\cdots,c_{\#F}\}$，同样上传至服务器 S。

为了搜索某个关键字集合 $W=\{w_1,w_2,\cdots,w_n\}$，用户需要首先生成一个搜索令牌 τ 并发送给服务器 S。搜索令牌 τ 中包含了针对每个关键字 w_i 的 $F_{K_1}(w_i)$,$G_{K_2}(w_i)$ 和 $P_{K_3}(w_i)$，即 $t_i=(F_{K_1}(w_i),G_{K_2}(w_i),P_{K_3}(w_i))$，输出搜索令牌 $\tau=(t_1,t_2,\cdots,t_n)$。

对于每个关键字 $w_i\in W$，服务器 S 使用 $F_{K_1}(w_i)$ 作为键来在查找表 T 中找到加密的指针，然后使用 $G_{K_2}(w)$ 来解密得到 L_w 的明文指针并定位到数组 A 中的节点，即 $\alpha=T[t_1]\oplus t_2=T[F_{K_1}(w)]\oplus G_{K_2}(w)$，其中 α 代表 N_1 在 A 中的坐标，最后使用 $P_{K_3}(w)$ 和储存在各自节点中的随机值 r 来解密整个链表节点，即 $[id_i,[score_i],addr(N_{i+1})]=l_i\oplus H(P_{K_3}(w),r_i)$，进而得到针对关键字 w 的文件标识符集合 $S=\{\langle id_1,[score_1]\rangle,\langle id_2,[score_2]\rangle,\cdots,\langle id_m,[score_m]\rangle\}$。为了不在后续排序协议中泄露信息，服务器处理每个关键字 w_i 对应的搜索结果集合 S_i，最终得到所有关键字的集合 $\{S_1,S_2,\cdots,S_n\}$。对于集合中每个文件标示符 id_j，对其对应的全部关键字的权值密文 $[score_j]$ 进行累积相加，得到最终的权值 e_j，即 $e_j=\sum_{i=1}^{n}[score_j]_{w_i}$。因此交集为 $I_W=\{\langle id_1,e_1\rangle,\langle id_2,e_2\rangle,\cdots,\langle id_m,e_m\rangle\}=S_1\cap S_2\cap\cdots\cap S_n$，并集为 $I_W=\{\langle id_1,e_{id_1}\rangle,\langle id_2,e_{id_2}\rangle,\cdots,\langle id_m,e_{id_m}\rangle\}=S_1\cup S_2\cup\cdots\cup S_n$。

图 3 为搜索过程的示例。用户想要搜索文件集中包含 w_1 以及 w_3 的文件，生成 $w_1\cup w_3$ 的搜索令牌 $\tau=(t_1,t_2)$，其中 $t_1=(F_{K_1}(w_1),G_{K_2}(w_1),P_{K_3}(w_1))$，$t_2=(F_{K_1}(w_3),G_{K_2}(w_3),P_{K_3}(w_3))$。用户将令牌发送给服务器 S，服务器 S 以 $F_{K_1}(w_1)$ 为键，对应到查找表 T 中加密的指针 $0\oplus G_{K_2}(w_1)$，再利用 $G_{K_2}(w_1)$ 解密，得到搜索数组中链表的头节点位置 0 的值 l_1，使用 $P_{K_3}(w_1)$ 和储存在节点 0 中的随机值 r_1 来解密，得到 $(id_2\parallel[score_2]\parallel 4)=l_1\oplus H(P_{K_3}(w_1),r_1)$，即得到指向位置 4 的指针，继续解密位置 4，得到的指针指向了位置 0，然后结束遍历。对于 w_3 服务器进行相同的操作，最终所有包含关键字 w_1 和 w_3 的文件都可以被找到，提取出文件标识符合和对应权值密文，组成 $S_{w_1}=\{\langle id_2,[score_2]_{w_1}\rangle,\langle id_4,[score_4]_{w_1}\rangle\}$，$S_{w_3}=\{\langle id_1,[score_1]_{w_3}\rangle,\langle id_2,[score_2]_{w_3}\rangle\}$ 然后服务器对于相同标识符 id_2 的权值密文进行同态相加得到 $e_2=[score_2]_{w_1}\oplus[score_2]_{w_3}$，从而得到最终搜索结果 $I_{w_1\cap w_3}=\{\langle id_1,e_1\rangle,\langle id_2,e_2\rangle,\langle id_4,e_4\rangle\}$。

搜索阶段结束之后，服务器 S 得到搜索结果 $I_W=\{\langle id_1,e_1\rangle,\langle id_2,e_2\rangle,\cdots,\langle id_m,e_m\rangle\}$，即文件标示符与对应的相关度权值密文组成的二元组的集合。然而，由于权值由 Paillier 同态加密进行加密，其随机性导致密文并不会表达明文的顺序，所以服务器 S 并不能通过权值的密文来得到排序之后的结果。为了实现结果排序，服务器 S 通过协同服务器 coS 的协助得到最终有序的搜索结果。然而，本文的安全模型定义不允许协同服务器 coS 获得和文件有关的任何信息，因此为了不向其泄漏消息，在进行协议之前服务器对 I_W 进行变换，将文件标示符利用 Paillier 加密方法进行加密，得到标示符密文 $[id_i]\leftarrow$

$P.Enc_{PK_p}(id_i)$,并将$[id_i]$替代集合中的id_i,生成$I_{W,1}=\{\langle[id_1],e_1\rangle,\langle[id_2],e_2\rangle,\cdots,\langle[id_m],e_m\rangle\}$.随后服务器$S$与协同服务器$coS$对$I_{W,1}$中的$\langle[id_i],e_i\rangle$进行两两比较并排序,排序以$e_i$为主键.具体来说,对于Batcher排序网络中每个层级i中$pairs_i$对应的每对元素$(\langle[id_x],e_x\rangle,\langle[id_y],e_y\rangle)$,服务器$S$计算$[z]:=[2^l]\times[e_x]\times[e_y]^{-1}\bmod n^2(l>k)$并对其进行盲化处理:选择$r\in(0,1)^{l+k}$,计算$[d]:=[z]\times[r]\bmod n^2$,并发送$[d]$给$coS$.同样,$S$与$coS$分别计算$r':=r\bmod 2^l$与$d'\leftarrow P.Dec_{SK_p}[d]\bmod 2^l$,随后双方执行DGK两方安全比较协议[29]比较$d'$和$r'$的大小,DGK协议结束后,云存储服务器得到比较结果的GM密文$\|\lambda\|$,满足$\lambda=1\Leftrightarrow d'\leqslant r'$.随后,$coS$提取出$d'$的第$l$位比特值,记做$d_l$,发送给$S$,同时,$S$提取出$A_{coS}$的第$l$位比特值,记做$\tilde{\lambda}$,计算$\|v\|=\|d_l\|\times\|r_l\|\times\|\lambda\|$,并与$coS$运行比特重加密协议以及密文选择协议[30],得到重加密之后的排序好的$\|\tilde{\lambda}\|$.图4为协同排序过程示例.

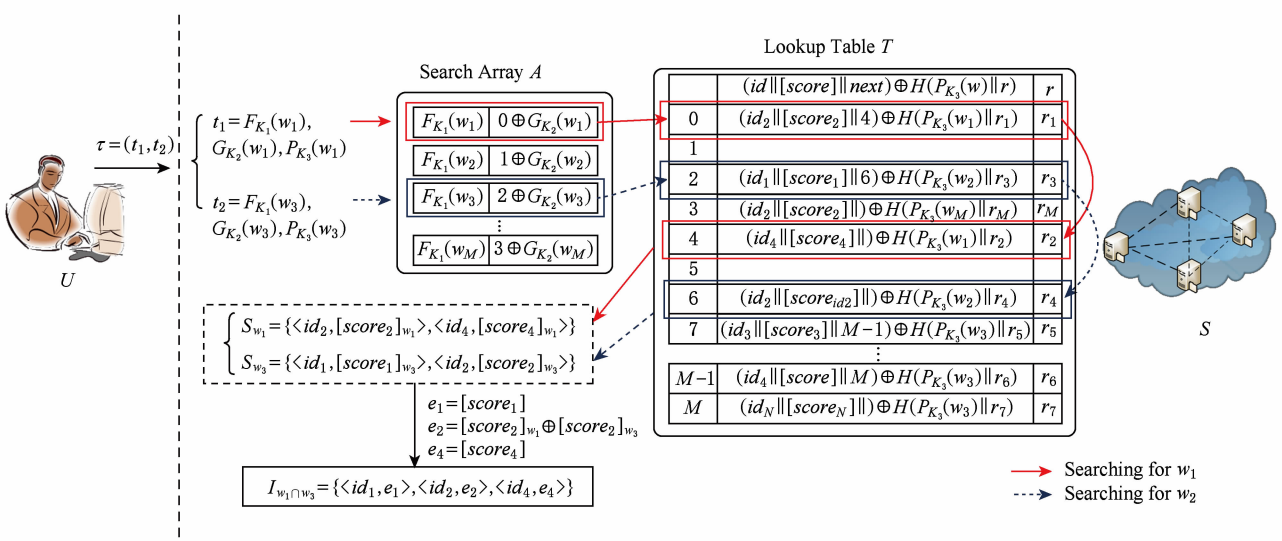


Fig. 3 Searching procedure

图3 搜索过程

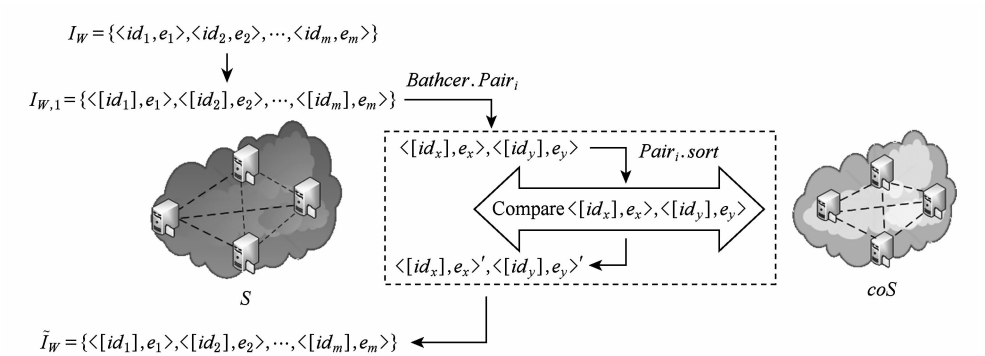


Fig. 4 Collaborated comparison procedure

图4 协同排序过程

进行$O((\mathrm{lb}\,m)^2)$轮Bathcer排序之后,最终服务器S得到根据权值排序并重加密的搜索结果$\tilde{I}_W=\{\langle[id_1],e_1\rangle,\langle[id_2],e_2\rangle,\cdots,\langle[id_m],e_m\rangle\}$,满足:$P.Dec_{SK_p}(e_1)\leqslant P.Dec_{SK_p}(e_2)\leqslant\cdots\leqslant P.Dec_{SK_p}(e_m)$,并将$\tilde{I}_W$返回给用户,用户利用Paillier解密密钥$SK$将$\tilde{I}_W$中的权值密文$(e_1,e_2,\cdots,e_m)$解密,得到文件标示符$(id_1,id_2,\cdots,id_m)$,并请求服务器返回相应的文件密文$(c_1,c_2,\cdots,c_m)$,通过解密算法得到文件明文$f_1,f_2,\cdots,f_m$.

3.3 详细算法设计

本节详细介绍MES-RR的算法与协议.在运行之前应该首先选择确定所使用的公共参数,包括内容:

选择选择伪随机函数:$F\{0,1\}^k\times\{0,1\}^*\rightarrow\{0,1\}^k$、$G:\{0,1\}^k\times\{0,1\}^*\rightarrow\{0,1\}^*$和$P:\{0,1\}^k\times\{0,1\}^*\rightarrow\{0,1\}^k$;选择Hash函数:$H\{0,1\}^*\rightarrow\{0,$

$1\}^*$;选择 Paillier 同态加密方法 $P=(P.Gen,P.Enc,P.Dec)$以及 GM 加密方法 $GM=(GM.Gen,GM.Enc,GM.Dec)$.

3.3.1 密钥生成算法

$KeyGen(1^k)\rightarrow(SK,PK)$由用户 U 执行,生成用户密钥以及加密算法公私钥.主要步骤如下:

1) 调用 $P.Gen(1^k)$生成 Paillier 加密算法的公私钥(PK_P,SK_p);

2) 调用 $GM.Gen(1^k)$生成 GM 加密算法的公私钥(PK_{GM},SK_{GM});

3) 随机选择 3 个长度为 k 位的数值 K_1,K_2,K_3;

4) 用户 U 将生成的密钥在本地进行存储,将(PK_P,PK_{GM})发布给云存储服务器 S,并将$(PK_P,SK_P,PK_{GM},SK_{GM})$发送给协同服务器 coS.

3.3.2 初始化算法

$Setup(K,PK,F)\rightarrow(\gamma,C)$由用户 U 执行,用于生成安全索引结构并对要上传的数据进行加密.创建有足够空间的查找表 T 以及一个有足够空间的搜索数组 A.

1) 构建倒排索引 δ

从文件集 F 中提取关键字集合 W,对于每个关键字 $w_i\in W$:

① 对于每个关键字 $w_i\in W$ 计算其在每个文件中的的 TF-IDF 相关度权值;

② 创建一个长度为 N 的数组 F_{w_i},将 w_i 在文件 r 中权值插入数组 F_{w_i} 中的位置 d 中.如果权值为 0 则表示这个关键字不存在文件 d 中.每个关键字后面都存在一组包含该关键字的文件集合;

③ 所有关键字的数组组成倒排索引 δ,δ 中列的个数为 M(W 中关键字的个数)行数为 N(F 中文件的个数).

2) 构建安全索引 γ

创建有足够空间的查找表 T 以及一个有足够空间的搜索数组 A.对于每个关键字 $w\in W$:

① 将 δ 中每一个相关度权值映射到一个空间为 Z_n 的整数值 $score_i$ 之中,其中 i 为对应的文件 f_i 标识,并将其用 Paillier 加密算法进行加密(明文空间为 Z_n),生成密文$[score_i]\leftarrow P.Enc_{PK_p}(score_i)$;

② 创建一个有 $\sharp w$ 个节点的链表 $L_w=(N_1,N_2,\cdots,N_{\sharp w})$,节点的结构定义为 $N_i=(\langle id_i,[score_i],addr(N_{i+1})\rangle\oplus H(P_{K_3}(w),r_i),r_i)$,其中$[score_i]$是 w 在 f_i 中的权值密文,r_i 为一个随机的 k 位比特串,$addr_s(N_{\sharp w+1})=addr_s(N_0)=\mathbf{0}^{\log\sharp A}$.将链表中的节点存储在搜索数组 A 的随机位置;

③ 通过以下方式将链表 L_w 的头节点的地址储存在查找表 T 中:$T[F_{K_1}(w)]=addr(N_1)\oplus G_{K_2}(w)$.

算法输出 $\gamma=(A,T)$为加密索引,$C=(c_1,c_2,\cdots,c_{\sharp F})$为密文集合.

3.3.3 搜索令牌生成算法

$SrchToken(K,W)\rightarrow\tau$ 由用户执行,用于用户在搜索时生成搜索令牌.主要步骤如下:

1) 输入对称密钥 K 和待搜索关键字集合 W,对于每一个待搜索关键字 $w_i\in W$,计算 $t_i=(F_{K_1}(w_i),G_{K_2}(w_i),P_{K_3}(w_i))$;

2) 输出搜索令牌 $\tau=(t_1,t_2,\cdots,t_n)$.

3.3.4 搜索算法

$Search(\gamma,C,\tau)\rightarrow I_W$ 由服务器 S 执行,用于实现搜索.对与搜索令牌中每一个 $t_i\in\tau$,依次按以下步骤:

1) 计算 $\alpha=T[t_1]\oplus t_2=T[F_{K_1}(w)]\oplus G_{K_2}(w)$,找到关键字链表 L_w 的头结点 N_1 的坐标,其中 α 代表 N_1 在 A 中的坐标值;

2) 查找节点 $N_1=A[\alpha_1]$,使用 t_3 解密该节点,例如 $N_1\Rightarrow(l_1,r_1)$,计算$\langle id_1,[score_1],addr(N_2)\rangle=l_1\oplus H(\tau_3,r_1)$;

对于 $j\geqslant 2$,重复 2)中过程,直至 α_{j+1}为 $\mathbf{0}$.

3) 将得到的结果记为 $S_i=\{\langle id_1,[score_1]\rangle,\langle id_2,[score_2]\rangle,\cdots,\langle id_m,[score_m]\rangle\}$;

4) 针对集合关键字布尔搜索模式,处理每个关键字令牌对应的搜索结果集合 S_i:

对于集合中每个文件标示符 id_i,对其权值密文进行相加,得到最终的权值 $e_i=\sum_{j=1}^{n}[score_i]_{w_j}$,因此交集为 $I_W=\{\langle id_1,e_1\rangle,\langle id_2,e_2\rangle,\cdots,\langle id_m,e_m\rangle\}=S_1\cap S_2\cap\cdots\cap S_n$,并集为 $I_W=\{\langle id_1,e_1\rangle,\langle id_2,e_2\rangle,\cdots,\langle id_m,e_m\rangle\}=S_1\cup S_2\cup\cdots\cup S_n$.

3.3.5 排序协议

$Sort(S:I_w,PK;coS:SK)\rightarrow(S:\tilde{I}_W)$是一个服务器 S 和协同服务器 coS 之间运行的交互式协议,用于对搜索结果进行排序.主要步骤如下:

1) 服务器将 I_W 中的每个文件标示符 id_i 进行加密,得到$[id_i]\leftarrow P.Enc_{PK_p}(id_i)$;

2) 生成 $I_{W,1}=\{\langle[id_1],e_1\rangle,\langle[id_2],e_2\rangle,\cdots,\langle[id_m],e_m\rangle\}$;

3) 服务器 S 与协同服务器 coS 利用 Batcher 排序网络对 $I_{W,1}$中的 e_i 进行两两比较并排序,具体来说,对于每个层级 i(总层级为$(\mathrm{lb}\,m)^2$,且 $1\leqslant i\leqslant$

$(\mathrm{lb}\,m)^2-1$)顺序执行以下步骤:

① 标记 $I_{W,1}$ 的 $pairs_i$ 为($\langle[id_x],e_x\rangle$,$\langle[id_y],e_y\rangle$):

Ⅰ. S 计算$[z]:=[2^l]\cdot[e_x]\cdot[e_y]^{-1} \bmod n^2$ ($l>k$);

Ⅱ. 选择 $r\in(0,1)^{l+k}$,并计算 $r':=r \bmod 2^l$,计算并发送$[d]:=[z]\cdot[r] \bmod n^2$,给代理服务器 coS;

Ⅲ. coS 运行 $P.Dec$ 算法解密$[d]\to d$,并计算 $d':=d \bmod 2^l$;

Ⅳ. 利用 DGK 协议,S 与 coS 联合计算 GM 密文 $\|\lambda\|$,满足 $\lambda=1\Leftrightarrow d'\leqslant r'$;

Ⅴ. coS 用 GM 加密方案加密 d_l 并发送 $\|d_l\|$ 给 S;

Ⅵ. S 加密 r_l 并计算 $\|v\|=\|d_l\|\times\|r_l\|\times\|\lambda\|$;

Ⅶ. S 与 coS 运行比特重加密协议与密文选择协议[23],得到有序的 $pairs_i$:($\langle[id_x],e_x\rangle'$,$\langle[id_y],e_y\rangle'$);

Ⅷ. S 将得到的重加密之后的有序($\langle[id_x],e_x\rangle'$,$\langle[id_y],e_y\rangle'$)存入 Bathcer 排序网络中层级 $i+1$ 的序列 $I_{W,i+1}$ 对应位置.

② 将 $I_{W,1}$ 在层级 i 的下一组元素继续执行步骤(1),直到 $I_{W,1}$ 完成层级 i 的比较.

4) 在 $\log(m)^2$ 次层级排序结束后,S 得到根据权值排序之后的结果的重加密序列 $\tilde{I}_W=\{\langle[id_1],e_1\rangle,\langle[id_2],e_2\rangle,\cdots,\langle[id_m],e_m\rangle\}$,满足:$P.Dec_{SK_P}(e_1)\leqslant P.Dec_{SK_P}(e_2)\leqslant\cdots\leqslant P.Dec_{SK_P}(e_m)$;

5) 服务器将 $\tilde{I}_W$ 返回给用户.

3.3.6 解密算法

$Dec(K,\tilde{I}_W)\to(id_1,id_2,\cdots,id_m)$ 由用户 U 执行,用于对接收到的搜索结果进行解密.

用户 U 利用密钥 PK_P 和搜索结果 $\tilde{I}_W=\{\langle[id_1],e_{id_1}\rangle,\langle[id_2],e_{id_2}\rangle,\cdots,\langle[id_m],e_{id_m}\rangle\}$,对序列中的文件标识符 id_i,运行 $P.DEC_{PK_P}([id_i])$ 算法,解密输出按相关度排序的文件标示符序列 $id_1,id_2,\cdots,id_m$.

3.4 安全性分析

本节首先对本文提出的 MSE-RR 方案的安全性进行概述,然后对其所具有的安全性进行详细的数学证明.

在接下来的内容中,首先分析在方案的执行过程中有哪些信息泄露给了用户,然后给出泄露函数的形式化定义.

1) 初始化. 定义安全索引 $\gamma=(T,A)$ 和密文集合 C,在初始化阶段,服务器 S 可以获知数组 A 的大小、T 中的集合$[id(w)]_{w\in W}$和每个文件$[|f|]_{f\in F}$的长度. 将这些信息记为 $\mathcal{L}_1$,即:$\mathcal{L}_1=(\sharp A,[id(w)]_{w\in W},[|f|]_{f\in F})$.

2) 搜索. 在多项式次搜索操作中,方案向服务器 S 泄露了搜索令牌中关键字 $w\in W$ 的标识 $id(w)$,每个关键字标识 $id(w)$ 和包含关键字 w 的文件标识之间的关联,以及用户的搜索历史. 具体来说,设 Q 为 n_q 次多关键字搜索的搜索令牌集合 $Q=\{q_1,q_2,\cdots,q_{n_q}\}$,$l_{\max}$ 表示在集合 Q 中长度最长的搜索令牌,n_q 次搜索之后的搜索历史可表示为一个二进制四阶的矩阵 $\boldsymbol{Q}_{n_q\times n_q\times l_{\max}\times l_{\max}}$. 将这些信息记为 $\mathcal{L}_2$,即:

$$\mathcal{L}_2=([id(f)]_{f\in F_w},id(w),\boldsymbol{Q}_{n_q\times n_q\times l_{\max}\times l_{\max}})_{w\in W}.$$

3) 排序. 在对每次搜索结果进行排序的过程中,需要 S 与 coS 进行交互,coS 能够学习到的知识包括和 S 运行两方安全密文比较协议中的泄露的消息 $\mathcal{L}_{Sort}=\{(e_x,e_y,l)_{A_S},(\|z\|,[\lambda])_{A_{coS}}\}$ 以及需要排序的文件的个数 m. 其中将这些信息记为 $\mathcal{L}_3$,即:

$$\mathcal{L}_3=(\{(e_x,e_y,l)_{A_S},(\|z\|,[\lambda])_{A_{coS}}\},m).$$

现在利用定理来证明方案在泄漏函数($\mathcal{L}_1$,$\mathcal{L}_2$,$\mathcal{L}_3$)存在的情况下,在随机语言模式下具有 IND-CKA2 安全性.

定理 1. 若 Paillier,GM 加密方案是 CPA 安全的,F,G 和 P 为伪随机函数,并且 DGK 协议在随机模型下可证明安全,则本文所提出的方案在随机预言模型下针对自适应敌手 A_S 与 A_{coS} 攻击是安全的,即满足 IND-CKA2 安全性.

证明. 在随机语言模型下构造 2 个模拟器 Sim_1 和 Sim_2,能够在仅获得泄露函数中包括的内容中的情况下,在理想游戏中通过模拟敌手 A_S 与 A_{coS} 的行为生成相应的模拟值来与用户 U 进行交互. 安全目标是这些模拟值应该和真实游戏中的相对应的值具有不可区分性.

给定泄漏函数 $\mathcal{L}_1$ 的信息,模拟器 Sim_1 可以学习到数组 A 的大小、T 中的集合$[id(w)]_{w\in W}$和每个文件$[|f|]_{f\in F}$的长度. 随后,它可以使用随机值来构建这些结构,并输出作为模拟值($\tilde{A}$,$\tilde{T}$,$\tilde{F}$). 伪随机函数的伪随机性及离散对数假设保证了模拟值($\tilde{A}$,$\tilde{T}$,$\tilde{F}$)与真实值(A,T,F)是无法区分的. 即

$$Output_{A_S}^{\mathrm{Real}}(A,T,F)\approx Output_{Sim_1}^{\mathrm{Ideal}}(\tilde{A},\tilde{T},\tilde{F})$$

给定泄漏函数 $\mathcal{L}_2$ 的信息,Sim_1 可以学习到搜索令牌中关键字 $w\in W$ 的标识 $id(w)$,每个关键字

标识 $id(w)$ 和包含关键字 w 的文件标识之间的关联,以及用户的搜索历史.用户 U 能够挑选不同个数关键字组成的集合并且用任意搜索请求 q_i 向系统发起质询.设 Q 为 n_q 次多关键字搜索的搜索令牌集合 $Q=\{q_1,q_2,\cdots,q_{n_q}\}$,$l_{\max}$ 表示在集合 Q 中长度最长的搜索令牌,n_q 次搜索之后的搜索历史可表示为一个二进制四阶的矩阵 $\boldsymbol{Q}_{n_q\times n_q\times l_{\max}\times l_{\max}}$.为了响应质询,$Sim_1$ 需要使用 $\mathcal{L}_2$ 来构建模拟搜索令牌,使其与真实令牌是无法区分的.Sim_1 需要跟踪这些质询之间的依赖关系,来保证各个令牌之间的一致性.通过定义额外的辅助结构 iA 和 iT 来记录 $\mathcal{L}_2$ 所揭示的依赖关系[3],与此同时 $\tilde{r}=(\widetilde{A},\widetilde{T})$ 数据依然保持随机.这使得 Sim_1 能够像真实 A_S 一样应答用户的质询,不同的是其使用的是模拟值,即

$$Output_{A_S}^{\text{Real}}(A,T)\approx Output_{Sim_1}^{\text{Ideal}}(\widetilde{A},\widetilde{T}).$$

给定泄漏函数 $\mathcal{L}_3$ 的信息,Sim_2 能够学习到两方安全密文比较协议中的泄露的消息 $\mathcal{L}_{Sort}=\{(e_x,e_y,l)_{A_S},(\|z\|,[\lambda])_{A_{coS}}\}$ 以及需要排序的文件的个数 m.由于 A_S 与 A_{coS} 的交互为多次密文比较协议组成的 $(\log m)^2$ 层 Bathcer 排序网络,所以通过证明 Sim_1 和 Sim_2 能够通过泄漏函数来模拟密文比较协议中的 A_S 与 A_{coS},进而模拟 $(\log m)^2$ 层 Bathcer 排序网络.

在进行每次密文比较协议时,A_S 的 $view$ 可表示为 $view_{A_s}=(e_x,e_y,l,PK_{QR},PK_P,r;\|\lambda\|,[z_l])$.给定 (e_x,e_y,l,PK_{QR},PK_P),我们能够构建出一个能够模拟 A_S 的 Sim_1:

1) 选择 $\tilde{r}\leftarrow(0,2^{\lambda+l})\bigcap Z$;

2) 选择随机数 $\tilde{\lambda},\tilde{z}_l$,生成 2 个 GM 方案的密文 $\|\tilde{\lambda}\|,\|\tilde{z}\|$;

3) 输出 $view_{Sim_1}=(e_x,e_y,l,PK_{\text{GM}},PK_P,\tilde{r};[\tilde{z_l}])$.

对于 $view_{A_s}$ 和 $view_{Sim_1}$,r 从一个均匀分布 $(0,2^{\lambda+l})\bigcap Z$ 中提取,而 $\|\tilde{z}\|$ 为 Paillier 的二重密文,具有随机性.所以 $(e_x,e_y,l,PK_{\text{GM}},PK_P)=(e_x,e_y,l,PK_{\text{GM}},PK_P,r,[\tilde{z}_l])$.除此之外,$r$ 和 $\tilde{r}$ 的都是从同一个均匀分布中提取,基于 GM 方案的语义安全性,我们能得出结论:A_S 的 $view$ 和 Sim_1 的 $view$ 具有计算不可区分性.

同理,A_{coS} 的 $view$ 可表示为 $view_{A_{coS}}=(SK_{\text{GM}},SK_P,\|z\|,\|\lambda\|)$,其中 SK_{GM} 是 GM 方案的私钥,SK_P 是 Paillier 加密方案的私钥.

给定 $(SK_{\text{GM}},SK_P,\|z\|,[\lambda])$,我们能够构建出一个能够模拟 A_{coS} 的 Sim_2:

1) 随机选择一个随机数 $\tilde{\lambda}$,计算 $\tilde{\lambda}$ 的 GM 方案的密文 $\|\tilde{\lambda}\|$,用于表示 $P.Dec_{SK_P}(e_x)\leqslant P.Dec_{SK_P}(e_y)$;

2) 随机选择 $\tilde{z}\leftarrow(0,2^{\lambda+l})\bigcap Z$;

3) 利用 Paillier 方案将 $\tilde{z}$ 加密,生成密文 $[\tilde{z}]$;

4) 输出 $view_{Sim_2}=(SK_{QR},SK_P,l,[\tilde{z}],\|\tilde{\lambda}\|)$.

可以得出 $(SK_{\text{GM}},SK_P,[\tilde{z}],\|\tilde{\lambda}\|)=(SK_{\text{GM}},SK_P,l,[\tilde{z}],\|\tilde{\lambda}\|)$.

在真实试验中,$z=x+r$,其中 x 是一个 l 位长的整数并且 r 是一个 $l+\lambda$ 比特长的整数,所以 $\tilde{z}$ 的分布和 z 具有不可区分性.我们能够直接得出 $(SK_P,[\tilde{z}])=(SK_P,[z])$ 并且,因为 $\tilde{z}$ 和 z 的分布独立于 t,所以 $(SK_{\text{GM}},SK_P,l,[\tilde{z}],\|\tilde{\lambda}\|)=(SK_{\text{GM}},SK_P,l,[z],\|\lambda\|)$.

除此之外,由于 $(SK_{\text{GM}},\|\tilde{\lambda}\|)=(SK_{\text{GM}},\|e_x\leqslant e_y\|)$,所以 $(SK_{\text{GM}},SK_P,l,[z],\|\tilde{\lambda}\|)=(SK_{\text{GM}},SK_P,l,[z],\|e_x\leqslant e_y\|)$.由于 DGK 协议为随机语言模型下具有语义安全性,所以:

$$Output_{A_{coS}}^{\text{Real}}(SK_{\text{GM}},SK_P,\|\lambda\|,[z])\approx Output_{Sim_2}^{\text{Ideal}}(SK_{\text{GM}},SK_P,\|P.Dec_{SK_P}(e_x)\leqslant P.Dec_{SK_P}(e_y)\|,[\tilde{z}],l).$$

由于 Sim_1 和 Sim_2 能够从泄漏函数 $\mathcal{L}_3$ 中获取需要排序的文件的个数 m,所以 Sim_1 和 Sim_2 能够模拟每次密文比较协议中的 A_S 与 A_{coS},进而模拟 $(\log m)^2$ 层 Bathcer 排序网络.所以,给定泄漏函数 $\mathcal{L}_3$ 的信息,Sim_1 和 Sim_2 能够像真实 A_S 与 A_{coS} 一样应答用户的质询,不同的是其使用的是模拟值.

综上所述,对于敌手 A_S,存在一个多项式时间的模拟器 Sim_1 满足 2 个分布式计算不可区分性:

$$Output_{A_S}^{\text{Real}}\approx Output_{Sim_1}^{\text{Ideal}}.$$

对于敌手 A_{coS},存在一个多项式时间的模拟器 Sim_2 满足 2 个分布式计算不可区分性:

$$Output_{A_{coS}}^{\text{Real}}\approx Output_{Sim_1}^{\text{Ideal}}.$$

因此,本文 MES-RR 方案在随机预言模型下针对自适应选择关键字攻击是安全的,即满足 IND-CKA2 安全性. 证毕.

3.5 性能分析

本节将 MES-RR 方案与现有具有代表性的支持排序的多关键字密文搜索方案[15-16,20]进行了比较.在本文中评估的指标有以下 6 种:同态技术、搜索结果的准确性(Accuracy)、排序机制(Rank Method)、用户生成搜索令牌的时间复杂度 Com_U,服务器生成最终搜索结果的时间复杂度 Com_S 以及安全性等.表 1 中 IPS(inner scalar product)表示内积相似性,

N 指的是文件个数,M 指的是文件集合中关键字的个数,指的是搜索关键字个数,CKA-1 和 CKA-2 分别指指选择关键字安全性和自适应性选择密文攻击安全.时间复杂度均为渐进的.

Table 1 Schemes Comparison

表 1 性能对比

Scheme	Accuracy	Rank Method	Updatable	Server Model	Security	Homomorphic Tools	Com_U	Com_S
Ref [15]	×	IPS	×	Single-Server	CKA1		$O(M^2)$	$O(NM^2)$
Ref [16]	√	TF-IDF/IPS	×	Single-Server	CKA2	LWE-Brakerski	$O(qN)$	$O(qN+M^2)$
Ref [20]	×	TF-IDF	×	Dual-Server	CKA1	Paillier	$O(N\,\mathrm{lb}\,N)$	$O(N\,\mathrm{lb}\,N)$
Ours	√	TF-IDF	√	Dual-Server	CKA2	Paillier/GM	$O(q)$	$O(N(\mathrm{lb}\,N)^2)$

搜索结果准确性方面,由于文献[15]中的方案仅仅基于内积相似性来计算每个文件中匹配到的关键字的数量,以此来计算文件权值并对文件进行排序.然而,因为这种方法失去了关键词对于文件的重要性这一信息,所以结果具有一定的误报率.文献[20]中的方案基于位置敏感 Hash 函数来聚类相似文件.然而,LSH 适用于相似性搜索但是同样不能提供精确排序.本方案利用 TF-IDF 加权规则来计算关键字文件相关度权值,实现对于关键字的精确排序.

安全性方面,文献[15]利用了启发式方法来隐藏搜索模式和访问模式,所以安全性不在比较范围之内.文献[16]基于全同态加密构建搜索索引,安全性较高,方案假设一个单独的服务器仅仅执行搜索功能,并不能对结果进行排序,服务器找到需要的密文文件以及权值,将其全部返回给用户,用户在本地基于 TF-IDF 进行对搜索结果排序.文献[20]中引入 2 个不共谋服务器模型,但是服务器之间的交互过程中,其中一个服务器能够访问每一个搜索结果的明文,这对于用户的数据安全以及搜索模式和访问模式造成了极大的威胁,存在信息泄露的隐患,安全性较弱.本方案基于同态加密算法对相关度权值进行加密,使得服务器无需解密即可对权值密文进行计算,从而在不获取额外信息的前提下对搜索结果进行排序.除此之外,本文方案中协同服务器仅仅在排序过程中起到了协助作用,在交互过程中得到的全部是密文,并不会得到任何有关搜索模式和用户真实数据相关的信息,安全性方面较高.

效率方面,文献[15]的搜索阶段中时间复杂度为 $O(M^2)$,搜索令牌的大小 $O(M)$,与文件集合关键字个数呈线性关系.为了响应查询请求,服务器需要进行 $O(NM^2)$次计算,其中 N 是文件集中文件个数.文献[16]中为了实现排序,频率信息由全同态加密方法进行加密,导致服务期计算代价为 $O(qN+M^2)$,效率较其他方案相比较差,且方案不支持动态更新.文献[20]中的协同服务器中存储复杂度是与文件集呈线性关系,存储代价稍高.本方案中搜索过程中需要 2 个服务器进行两方安全交互计算,导致服务器的搜索复杂度为 $O(N(\mathrm{lb}\,N)^2)$,略高于其他方案,但是本方案中用户生成长度为 q 的搜索令牌仅仅需要常数级 $O(q)$的时间复杂度,与集合中文件数量和关键字数量无关,与其他方案相比客户端计算代价较低.在如今分布式云存储环境下,如何降低客户端存储与计算代价为首要考虑问题,所以本方案在效率更适用于实际云存储部署需求.

4 总 结

本文对多关键字密文搜索和搜索结果排序方法展开研究,提出一种双服务器模型下支持相关度排序的多关键字密文搜索方案,在能够保证高效地实现多关键字密文搜索的同时,实现对于搜索结果的排序.方案基于 TF/IDF 加权技术以及 Paillier 同态加密算法,构建关键字相关度安全索引,优化计算代价并降低了存储复杂度;方案设计双服务器模型,引入安全可信的协同服务器处理机制来构造安全排序协议,在不向协同服务器泄露搜索模式与访问模式的前提下,实现对于搜索结果的高效排序.在安全性方面,本文在诚实与好奇的威胁场景下,构建方案的安全模型,并对该方案的安全性进行严格分析,结果表明在结果表明在有限泄漏函数存在的情况下,方案能够在随机语言模型下具有语义安全性.性能分析表明,与以往的支持搜索结果排序的多关键字密文搜索方案相比,本方案大大降低了存储代价和访问交互次数,适用于实际的云存储环境.未来工作包括对提出的方案部署进真实云环境进行部署试验并在服务器交互次数和计算代价方面进行进一步提高.

参 考 文 献

[1] Yousef E, Bharath K S, Wei J. Secure k-nearest neighbor query over encrypted data in outsourced environments [C] // Proc of the 30th IEEE Int Conf on Data Engineering. Piscataway, NJ: IEEE, 2014: 664-675

[2] Raphael B, Raluca A P, Stephen T, et al. Machine learning classification over encrypted data [C/OL] //Proc of 2015 Network and Distributed System Security (NDSS) Symp. Reston, VA: The Internet Society, 2015 [2018-06-10]. http://www. internetsociety. org/doc/machine-learning-classification-over-encrypted-data

[3] Liu An, Zheng Kai, Li Lu, et al. Efficient secure similarity computation on encrypted trajectory data [C] //Proc of the 31st IEEE Int Conf on Data Engineering. Piscataway, NJ: IEEE, 2015: 66-77

[4] Song D, Wagner D, Perrig A. Practical techniques for searches on encrypted data [C] //Proc of 2000 IEEE Symp on Security and Privacy (S&P 2000). Piscataway, NJ: IEEE, 2000: 44-55

[5] Curtmola R, Garay J, Kamara S, et al. Searchable symmetric encryption: Improved definitions and efficient constructions [C] //Proc of the 13th ACM Conf on Computer and Communications Security. New York: ACM, 2006: 79-88

[6] Kamara S, Papamanthou C, Roeder T. Dynamic searchable symmetric encryption [C] //Proc of 2012 ACM Conf on Computer and Communications Security. New York: ACM, 2011: 965-976

[7] Boneh D, Di C G, Ostrovsky R, et al. Public key encryption with keyword search [G] //LNCS 3027: Proc of Int Conf on the Theory and Applications of Cryptographic Techniques. Berlin: Springer, 2004: 506-522

[8] Golle P, Staddon J, Waters B. Secure conjunctive keyword search over encrypted data [G] //LNCS 3089: Proc of Int Conf on Applied Cryptography and Network Security. Berlin: Springer, 2004: 31-45

[9] Cash D, Jarecki S, Jutla C, et al. Highly-scalable searchable symmetric encryption with support for boolean queries [G] // LNCS 8042: Proc of Advances in Cryptology-CRYPTO 2013. Berlin: Springer, 2013: 353-373

[10] Cash D, Jaeger J, Jarecki S, et al. Dynamic searchable encryption in very-large databases: Data structures and implementation [C] //Proc of 2014 Network and Distributed System Security (NDSS) Symp. Reston, VA: The Internet Society, 2014: 23-26

[11] Wang Kaixuan, Li Yuxi, Zhou Fucai, et al. Multi-keyword fuzzy search over encrypted data [J]. Journal of Computer Research and Development, 2017, 54(2): 348-360 (in Chinese)
(王恺璇, 李宇溪, 周福才, 等. 面向多关键字的模糊密文搜索方法[J]. 计算机研究与发展, 2017, 54(2): 348-360)

[12] Swaminathan A, Mao Y, Su G M, et al. Confidentiality-preserving rank-ordered search [C] //Proc of the 2007 ACM Workshop on Storage Security and Survivability. New York: ACM, 2007: 7-12

[13] Zerr S, Olmedilla D, Nejdl W, et al. Zerber+r: Top-k retrieval from a confidential index [C] //Proc of the 12th Int Conf on Extending Database Technology: Advances in Database Technology. New York: ACM, 2009: 439-449

[14] Wang Cong, Cao Ning, Ren Kui, et al. Enabling secure and efficient ranked keyword search over outsourced cloud data [J]. IEEE Trans on Parallel and Distributed Systems, 2012, 23(8): 1467-1479

[15] Cao Ning, Wang Cong, Li Ming, et al. Privacy-preserving multi-keyword ranked search over encrypted cloud data [J]. IEEE Trans on Parallel and Distributed Systems, 2014, 25(1), 222-233

[16] Strizhov M, Ray I. Multi-keyword similarity search over encrypted cloud data [G] //IFIPAICT 428: Proc of IFIP Int Information Security Conf. Berlin: Springer, 2014: 52-65

[17] Sun Wenhai, Wang Bing, Cao Ning, et al. Verifiable privacy-preserving multi-keyword text search in the cloud supporting similarity-based ranking [J]. IEEE Trans on Parallel and Distributed Systems, 2014, 25(11): 3025-3035

[18] Zhang Wei, Xiao Sheng, Lin Yaping, et al. Secure ranked multi-keyword search for multiple data owners in cloud computing [C] //Proc of the 44th Annual IEEE/IFIP Int Conf on Dependable Systems and Networks. Piscataway, NJ: IEEE, 2014: 276-286

[19] Fu Zhangjie, Sun Xingming, Liu Qi, et al. Achieving efficient cloud search services: Multi-keyword ranked search over encrypted cloud data supporting parallel computing [J]. IEICE Trans on Communications, 2015, 98(1): 190-200

[20] Orencik C, Kantarcioglu M, Savas E. A practical and secure multi-keyword search method over encrypted cloud data [C] //Proc of 2013 6th Int Conf on Cloud Computing. Piscataway, NJ: IEEE, 2013: 390-397

[21] Xia Zhihua, Wang Xinhui, Sun Xingming, et al. A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data [J]. IEEE Trans on Parallel and Distributed Systems, 2016, 27(2): 340-352

[22] Song Wei, Wang Bing, Wang Qian, et al. A privacy-preserved full-text retrieval algorithm over encrypted data for cloud storage applications [J]. Journal of Parallel and Distributed Computing, 2017, 99: 14-27

[23] Shen Peisong, Chen Chi, Zhu Xiaojie. Privacy-preserving relevance ranking scheme and its application in multi-keyword searchable encryption [C] //Proc of the 13th Int Conf on Security and Privacy in Communication Systems. Berlin: Springer, 2018: 128-146

[24] Paillier P. Public-key cryptosystems based on composite degree residuosity classes [G] //LNCS 1592: Proc of Int Conf on the Theory and Applications of Cryptographic Techniques. Berlin: Springer, 1999: 223-238

[25] Goldwasser S, Micali S. Probabilistic encryption [J]. Journal of Computer and System Sciences, 1984, 28(2): 270-299

[26] Lipmaa H. An oblivious transfer protocol with log-squared communication [G] //LNCS 3650: Proc of Int Conf on Information Security. Berlin: Springer, 2005: 314-328

[27] Wikipedie: tf-idf [EB/OL]. [2018-07-02]. https://en.wikipedia.org/wiki/Tf%E2%80%93idf

[28] Veugen T. Comparing encrypted data [EB/OL]. [2018-06-10]. https://www.researchgate.net/publication/266527434_COMPARING_ENCRYPTED_DATA, 2011[2015-04-29]

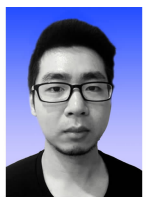
[29] Damgard I, Geisler M, Kroigard M. Homomorphic encryption and secure comparison [J]. International Journal of Applied Cryptography, 2008, 1(1): 22-31

[30] Baldimtsi F, Ohrimenko O. Sorting and searching behind the curtain [G] //LNCS 8975: Proc of Int Conf on Financial Cryptography and Data Security. Berlin: Springer, 2015: 127-146

Li Yuxi, born in 1990. PhD candidate. Her main research interests include secure cloud storage.

Zhou Fucai, born in 1964. PhD, professor and PhD supervisor. Senior member of CCF. His main research interests include cryptography and network security, trusted computing, and critical technology in electronic commerence.

Xu Jian, born in 1978. PhD, associate professor. His main research interests include secure computing.



Xu Zifeng, born in 1990. PhD candidate. His main research interests include secure cloud storage.