云环境下一种隐私保护的高效密文排序查询方法

程芳权 彭智勇 宋 伟 王书林 崔一辉
(武汉大学软件工程国家重点实验室 武汉 430072)
(武汉大学计算机学院 武汉 430072)

摘 要 数据前端加密是保护云环境下外包数据隐私的一种有效手段,但却给数据查询等操作带来挑战.针对云环境下多数据拥有者数据外包及选择性访问授权特征,为支持大规模加密云数据上高效且隐私保护的用户个性化密文查询,文中提出了一种隐私保护的高效密文排序查询方法RQED.通过设计无证书认证的PKES(支持关键词检索的公钥加密),并构建RQED框架来实现强隐私保护的密文查询.基于该框架,设计了更合理的多属性多关键词密文查询排序函数,并提出了基于层次动态布隆过滤器的RQED索引机制,提高密文查询时空效率.理论分析和实验性能对比证明:RQED在确保查询强隐私保护和高准确性的同时,具有较明显的时空效率优势.

关键词 云计算;布隆过滤器;排序查询;层次索引;无证书认证
中图法分类号 TP309 **DOI号**: 10.3724/SP.J.1016.2012.02215

An Efficient Privacy-Preserving Rank Query over Encrypted Data in Cloud Computing

CHENG Fang-Quan PENG Zhi-Yong SONG Wei WANG Shu-Lin CUI Yi-Hui
(*State Key Laboratory of Software Engineering*, *Wuhan University*, *Wuhan* 430072)
(*Computer School*, *Wuhan University*, *Wuhan* 430072)

Abstract In cloud computing, for protecting the privacy of the sensitive cloud data, an effective methodology is to encrypt the data before outsourcing. However, data encryption makes data utilization, e.g. querying, a very challenging task. Though many solutions have been proposed, they are insufficient or even ineffective to achieve efficient multi-keyword rank query and flexible selective query authorization with multiple data owners while keeping strong privacy preserving. In this paper, we propose an efficient privacy-preserving rank query over encrypted data (RQED). Through the improved searchable public key encryption (PKES) with certificateless authentication, we establish the RQED framework. Based on the RQED framework, we design a more sound and privacy-preserving RQED rank function, and propose a hierarchical index based on dynamic Bloom filters. The theoretical analysis and experimental evaluation show that the proposed solutions indeed achieve powerful privacy guarantee, efficient query performance, low communication overhead and effective query authorization control.

Keywords cloud computing; bloom filter; rank query; hierarchical index; certificateless authentication

收稿日期:2012-06-05;最终修改稿收到日期:2012-08-07.本课题得到国家自然科学基金(61070011,61100019,61202034)、华为技术有限公司创新研究计划基金"可信云存储服务关键技术研究"(YJCB201001078)、武汉市学科带头人计划基金(201150530139)、教育部博士点基金(20110141120033)、软件工程国家重点实验室开放基金(SKLSE2010-08-20)、湖北省自然科学基金国际合作重点项目资助.**程芳权**,男,1985年生,博士研究生,主要研究方向为数据库、可信数据管理等.E-mail:cheng@whu.edu.cn.**彭智勇**,男,1963年生,博士,教授,博士生导师,主要研究领域为复杂数据管理、Web数据管理、可信数据管理等.**宋 伟**(通信作者),男,1978年生,博士,讲师,主要研究方向为分布式系统、可信数据管理等.E-mail:songwei@whu.edu.cn.**王书林**,男,1987年生,硕士研究生,主要研究方向为可信数据管理.**崔一辉**,男,1981年生,博士研究生,主要研究方向为可信数据管理.

1 引 言

云计算是一种新型网络计算模式并被喻为信息领域的工业化革命[1]. 云环境下,以"X-as-a-Service"和"Pay-as-you-go"为核心理念,驱动企业或用户将本地数据及其复杂管理外包给云服务提供商[2-3](Cloud Service Provider,CSP),从而利用"云"充足的计算、存储和网络等资源,使用户只需较小的代价就能快速获得高质量的数据服务及应用. 但数据安全和隐私保护在云计算问题中首当其冲. 外包于云端的隐私数据完全脱离其拥有者的直接物理控制,将面临外部网络攻击者和 CSP 中不可信(诚实而好奇[4])管理员的双重威胁.

采用前端加密是解决上述安全问题的一种有效方法[2-4],即数据拥有者在本地可信前端对数据加密后再外包给 CSP,查询时则从 CSP 获取满足请求的密文数据并在前端解密. 如此一来,密文数据查询则不再具有明文查询特性而成为重要挑战,且云环境下密文查询有以下两方面需求:

一方面,面向云端海量数据特征和用户个性化查询需求,有必要提供针对多关键词、多属性请求的高效密文排序查询[3],充分表达用户查询兴趣,高效返回更能精确满足用户需求且具有重要度区分的所有排序结果或 Top-k 排序结果,使用户能快速定位与查询最相关密文数据,同时也减小网络传输开销.

另一方面,不同于传统企业级单数据拥有者数据外包,云环境下数据外包应用更具有多数据拥有者数据发布及选择性访问授权、多源数据查询等特征. 如 Email 系统等,任何用户都可以作为数据拥有者外包数据并选择性授权其所有或部分数据给任何其他用户查询访问,用户查询则执行其被授权的来源于不同数据拥有者的所有数据. 该特征下,将使用户面临更具威胁的攻击,如不可信 CSP 与部分恶意数据拥有者之间合谋对其他任何用户隐私的攻击. 同时考虑到云环境下的"Pay-as-you-go"理念、轻量级用户查询客户端及高要求用户体验等特征,极有必要提供针对上述特征的时空高效密文查询及查询授权控制,并确保其针对各种攻击的强隐私保护能力. 其时空高效表现为:数据发布时数据、索引等的低传输/存储代价,数据查询时请求转换的客户端低计算代价以及云端密文查询的低执行代价. 而隐私保护则要求除了保护数据隐私,还需确保密文索引和查询过程无隐私泄露.

目前针对数值型数据[4]和字符型数据[5-12]的密文查询都有大量的方法提出,Cao 等人[3]也在 2011 年第一次提出并解决了单数据拥有者云环境下的隐私保护多关键词密文排序查询. 然而,现有方法仍无法满足上述云环境下的密文查询需求:

首先,未全面考虑多关键词查询相似度计算的贡献因素,如多属性数据查询、关键词隐私保护权重、用户被授权可访问的数据范围等,无法有效实现更客观、合理的密文排序查询.

其次,无法在支持多数据拥有者数据外包及其选择性数据查询授权的同时,兼顾密文查询的强隐私保护能力、时空高效性、查询及结果排序准确性,如无法在保证查询隐私前提下,支持大规模数据高效、准确的多关键词密文排序查询并保持较低索引空间(传输/存储)开销等.

针对以上问题,本文提出了一种支持隐私保护的高效密文排序查询方法(RQED),旨在满足上述云环境下密文查询需求,主要贡献:

(1)提出了针对云环境下多数据拥有者数据外包及选择性数据查询授权特征的多属性多关键词密文排序查询,并定义其系统模型和攻击模型. 基于无证书认证思想[13-14]设计不依赖于可信第三方和安全传输信道的可认证 PKES(支持关键词检索的公钥加密[8-9]),并构建支持系统模型的 RQED 框架以增强查询隐私保护.

(2)基于 RQED 框架,设计支持多属性数据隐私保护词权重、查询词权重及用户被授权可访问数据范围等更客观、合理的密文查询排序函数.

(3)考虑到单数据量不确定性、数据和授权更新不确定性以及大规模密文数据查询,为提高密文查询执行与密文索引的时空效率,提出了基于层次动态布隆过滤器的 RQED 索引机制.

(4)理论分析了 RQED 机制对密文查询各种隐私的保障,并通过真实数据集上的实验分析展示出 RQED 机制在兼顾查询高准确性和强隐私保护的同时,还具备了较高的时空效率优势.

2 相关工作

目前国内外学者和研究人员针对加密数据查询方面做了大量工作,旨在实现密文数据查询的功能丰富、高执行效率、低空间开销和强隐私保护,本文相关工作大致分为以下方面:

(1)关键词密文查询. 现有基于关键词的密文

查询方法主要分为单关键词查询和多关键词查询，其目的都是通过索引关键词构建密文数据索引以实现相关查询，并确保其针对无查询密钥的非授权用户不泄露数据隐私. Song 等人[5]第一次提出基于对称密钥的单关键词可检索加密方法，Goh[7]基于布隆过滤器对其效率进行改进，同时 Curtmola 等人[10]在 Song 的基础上给出更严格的安全性定义和更高效的对称密钥可检索加密方法构造. Wang 等人[11]考虑关键词词频信息，提出基于对称密钥保序加密技术 OPSE 的单关键词分级密文排序查询方法，该方法对相似度计算并未全面考虑，因为需要扫描所有文档而不易进行索引更新. 然而，单关键词查询不足以表达和满足用户的个性化查询需求，且涉及过多查询结果而导致巨大的网络通信负载. Li 等人[12]针对关键词精确匹配的不足提出基于编辑距离的加密字符串模糊检索方案. 然而这些方法却仅仅针对“all-or-nothing”的查询方式，并返回给用户完全无区分性的查询结果. 为了进一步满足用户个性化查询需求，Cao 等[3]第一次提出多关键词密文排序查询问题，并基于安全 KNN 查询技术[6]中索引向量与查询向量间“内积相似度”来实现排序查询并进行隐私保护增强. 然而，该方法并未考虑数据中索引关键词权重及查询关键词权重等因素，从而不能返回满足用户查询请求的准确排序结果；并且该方法需要构建系统枚举索引关键词词典并在用户端存储，这无疑不符合实际应用且造成较大客户端负担. 例如，当词典规模为 N 时，其用户需要安全传输和存储的密钥矩阵空间开销为 $O(N^2)$ 则随着 N 增大将带给客户端难以承受的负担.

(2) 密文查询隐私保护. 事实上，现有方法都试图在具有一定攻击能力的特定攻击模型下执行查询隐私保护. 基于对称密钥加密的查询隐私保护[3,5-6]具有较高的执行效率，然而一方面，若采用同一个密钥进行所有用户的查询授权控制，则完全无法抵御内部用户与 CSP 的合谋攻击；若实施选择性查询授权控制，即数据拥有者对不同用户授权不同的查询密钥，显然将带来复杂的密钥安全分发和管理问题(随着系统数据和用户规模扩大，其密钥安全管理极具挑战[15]且近似于 NP 完全困难问题). 文献[3]试图通过扰乱排序实现查询隐私保护，但其内部用户完全可以根据自己所掌握的密钥与 CSP 合谋推理出其他用户的查询请求隐私. 针对上述缺陷，Boneh 等人[8]首次基于 DH 密钥交换协议提出关键词可检索的公钥加密方法(PKES)，通过公钥加密和私钥查询很好的避免了上述选择性查询授权中的复杂密钥管理问题. Boneh 等人[9]在此基础上，进一步提出了支持 PIR 查询的公钥加密方法，但需要繁重的客户端与服务器端间交互. 相对于基于对称密钥的方法，PKES 显然更具有优势. 然而，现有 PKES 方法存在局限性：算法构造中忽视公钥的认证；当存在多数据拥有者查询授权时，用户查询过程难以抵御恶意数据拥有者与 CSP 的合谋攻击；索引缺乏动态伸缩性致使空间开销较大；查询时直接匹配每个数据索引导致查询效率低下；无法高效支持云环境下大规模数据的多关键词排序查询.

3　问题定义

3.1　系统模型

如图 1 所示，我们认为一个典型的云环境下多数拥有者数据外包服务系统主要包括以下 3 个实体：云服务提供商 CSP、数据拥有者用户和被授权

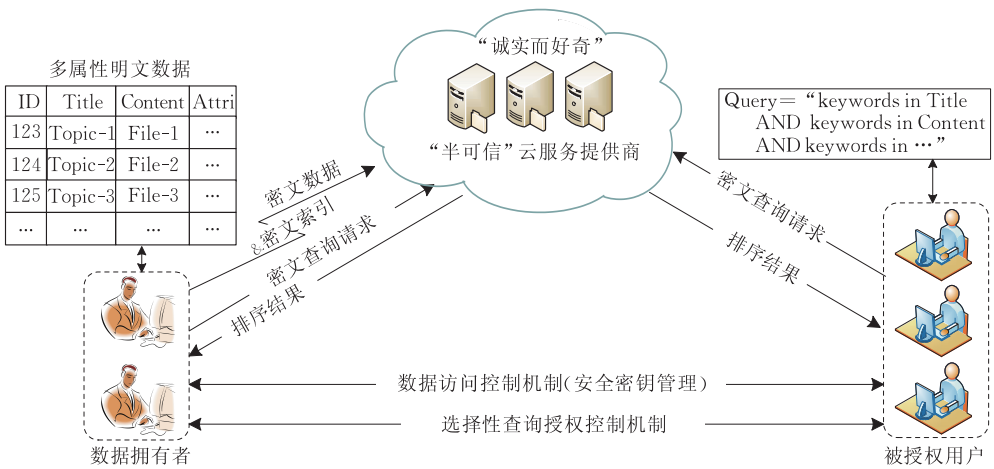


图 1　云环境下隐私保护的密文排序查询结构图

用户.系统任何用户都可以作为数据拥有者发布数据,也可以同时作为被授权用户获得其他用户的数据访问权限.假设数据拥有者用户$\bar{u}$有数据集$\mathbb{D}_{\bar{u}}$需要外包给CSP,为保障数据安全,首先将$\mathbb{D}_{\bar{u}}$加密为密文数据$\mathbb{C}_{\bar{u}}$.而$\bar{u}$可以根据自己的需求选择性地授权$\mathbb{D}_{\bar{u}}$中所有或部分数据给被授权用户u进行访问.为了实现在密文数据$\mathbb{C}_{\bar{u}}$上的查询能力,$\bar{u}$在外包数据前还需要根据$\mathbb{D}_{\bar{u}}$构建其密文索引$\mathbb{I}_{\bar{u}}$.然后再将密文数据及对应密文索引$\{\mathbb{C}_{\bar{u}},\mathbb{I}_{\bar{u}}\}$外包给CSP.

当被授权用户u提交查询$\mathcal{Q}$时,通过如图1中所示查询授权控制机制,u能够正确转换$\mathcal{Q}$为查询陷门$\mathcal{T}_{\mathcal{Q}}$(又称为Trapdoor),并提交CSP.CSP利用$\mathcal{T}_{\mathcal{Q}}$在系统所有密文索引集$\mathbb{I}$和密文数据集$\mathbb{C}$上执行查询,并返回满足查询条件且按查询相关度排序的所有结果或Top-k结果.最终,通过如图1所示的数据访问控制机制[15],u从相应数据拥有者获取结果数据对应密钥安全解密数据内容.

本文旨在设计上述系统模型下强隐私保护且时空高效的多属性多关键词密文排序查询方法.

3.2 攻击模型

攻击者.按照其攻击能力由弱至强分为:(1)外部攻击者,即网络攻击者,在系统中最普遍且攻击能力最弱;(2)CSP,即不可信(诚实而好奇)系统管理员,他承诺正确执行与用户的数据托管协议,但无法避免其恶意窥窃或分析用户数据、索引及查询请求,从而获取用户隐私信息;(3)内部攻击者,包括恶意数据拥有者和被授权用户,通过其已掌握的部分信息(如部分数据、索引、Trapdoor、密钥等)来试图攻击其他用户隐私信息;(4)合谋攻击者,包括被授权用户与CSP合谋和数据拥有者与CSP合谋,后者较前者有更强的攻击能力.

攻击方式.正如文献[3]所述,存在两种攻击方式:(1)已知密文攻击,攻击者仅仅掌握密文数据$\mathbb{C}$、密文索引$\mathbb{I}$及密文查询请求Trapdoor;(2)已知背景攻击,攻击者在已知密文基础上,还掌握部分关键词或关键词频等静态信息、部分给定关键词与数据的关联信息以及部分给定Trapdoor间的关联信息等.后者有更强攻击能力,而外部攻击者只能执行前者.

本文目标是在上述攻击模型下,保障密文排序查询的隐私需求[3],详见5.1节安全性分析.

3.3 主要符号说明

u;U↔任何一个用户;系统所有用户集合.

$\bar{u}$;OU↔任何数据拥有者;所有数据拥有者集合.$OU\subseteq U$.

$\mathbb{D}$;$\mathbb{C}$;$\mathbb{I}$↔系统明文数据集;密文数据集;密文索引集.

$\mathcal{A}$;$\mathbb{A}$↔任何一个数据属性标识;所有属性集合.

$\mathbb{D}_{\bar{u}}$;$\mathbb{C}_{\bar{u}}$;$\mathbb{I}_{\bar{u}}$↔$\bar{u}$外包的明文数据集;密文数据集;密文索引集.$\mathbb{D}=\bigcup_{\bar{u}\in OU}\mathbb{D}_{\bar{u}}$;$\mathbb{I}=\bigcup_{\bar{u}\in OU}\mathbb{I}_{\bar{u}}$;$\mathbb{C}=\bigcup_{\bar{u}\in OU}\mathbb{C}_{\bar{u}}$.

$\mathcal{D}_{\bar{u}}$;$\mathcal{C}_{\bar{u}}$;$\mathcal{I}_{\bar{u}}$↔$\bar{u}$外包的其中一条明文数据;密文数据;密文索引.即$\forall\mathcal{D}_{\bar{u}}\in\mathbb{D}_{\bar{u}}$;$\forall\mathcal{C}_{\bar{u}}\in\mathbb{C}_{\bar{u}}$;$\forall\mathcal{I}_{\bar{u}}\in\mathbb{I}_{\bar{u}}$.

$\mathcal{D}_{\bar{u},\mathcal{A}}$↔$\mathcal{D}_{\bar{u}}$中属性$\mathcal{A}$上的数据值,即$\forall\mathcal{D}_{\bar{u},\mathcal{A}}\in\mathbb{D}_{\bar{u}}$.

$W_{\mathcal{A}}$↔$\mathcal{D}_{\bar{u},\mathcal{A}}$中所包含的索引关键词集合.

$W_{\mathcal{A},l}$↔$W_{\mathcal{A}}$中第l个索引关键词.

$EW_{\mathcal{A},l}$↔$W_{\mathcal{A},l}$加密后的密文索引词.

$EW_{\mathcal{A}}$↔$W_{\mathcal{A}}$加密后的密文索引词集合.

EW↔$\mathcal{D}_{\bar{u}}$经处理后的所有密文索引词集合.

$\mathcal{Q}$↔u的查询请求.

$\mathcal{T}_{\mathcal{Q}}$↔$\mathcal{Q}$的查询Trapdoor.

$\widetilde{\mathcal{A}}$;$\widetilde{\mathbb{A}}$↔$\mathcal{Q}$中任何一个属性及其所有属性集合.

$W_{\widetilde{\mathcal{A}}}$↔$\mathcal{Q}$中属性$\widetilde{\mathcal{A}}$上的$q$个查询关键词集合.

$W_{\widetilde{\mathcal{A}},l}$↔$W_{\widetilde{\mathcal{A}}}$中第$l$个查询关键词.

$\mathcal{T}_{\bar{u},W_{\widetilde{\mathcal{A}},l}}$↔$W_{\widetilde{\mathcal{A}},l}$针对数据拥有者$\bar{u}$的查询子Trapdoor.

$H^{BF}_{i=[1,K]}$↔布隆过滤器的K个Hash函数簇.

$H^{BF}_{i=[1,K]}(x)$↔x映射到布隆过滤器的K个位游标.

$BF[H^{BF}_{i=[1,K]}(x)]$↔位于布隆过滤器BF中K个位游标处的K个值.

4 RQED机制

4.1 RQED框架

概念1. 双线性配对映射[13-14].假设G_1和G_2是两个阶都为大素数p的乘法群,映射$\hat{e}$:$G_1\times G_1\rightarrow G_2$称为双线性配对映射,则映射$\hat{e}$满足如下性质:

(1)双线性.对于$\forall P,Q\in G_1$和$\forall a,b\in Z_p$,都有$\hat{e}(P^a,Q^b)=\hat{e}(P,Q)^{ab}$.

(2)非退化性.$\exists P,Q\in G_1$,使得$\hat{e}(P,Q)\neq 1$.

(3)可计算性.对于$\forall P,Q\in G_1$,都会有一个有效的多项式时间算法来计算$\hat{e}(P,Q)$.

针对上述系统模型和安全模型,我们首先构造RQED框架.该框架中,通过设计不依赖于可信第三方和安全传输信道的无证书认证PKES来实施复杂的多数据拥有者选择性查询授权控制并增强查

询隐私保护.基于该框架,我们设计多属性、多关键词的隐私保护 RQED 排序函数,并进一步提出基于层次动态布隆过滤器索引的 RQED 机制.

无证书公钥密码[13-14]的基本思想即通过引入半可信第三方 KGC 来辅助用户公私钥对生成及公钥认证,但 KGC 却并不能掌握威胁用户私钥的相关信息.本文在 Vipul Goyal[14]所提出的无证书思想基础上,构造高效的可认证 PKES.在本文系统模型和框架中,我们认为 CSP 是一个典型的半可信第三方实体(诚实而好奇),可以充当 KGC 角色.

RQED 框架由以下部分组成:

Scenario.为了阐述更为清晰,我们这里仅以数据拥有者$\bar{u}$的单条数据$\mathcal{D}_{\bar{u}}$为例,而$\bar{u}$的其它数据处理过程完全相同.$\mathcal{D}_{\bar{u}}$对应的密文数据及索引分别为$\mathcal{C}_{\bar{u}}$、$\mathcal{I}_{\bar{u}}$,且$\mathcal{D}_{\bar{u}}=\{\mathcal{D}_{\bar{u},\mathcal{A}}\mid\mathcal{A}\in\mathbb{I}\}$.同时假设$\bar{u}$将数据$\mathcal{D}_{\bar{u}}$的访问权限授权给用户 u.然后,用户 u 提交查询请求$\mathcal{Q}$,且$\mathcal{Q}$包含有属性集$\widetilde{\mathbb{A}}$,其中每一个属性$\widetilde{\mathcal{A}}$上含有 q 个查询关键词 $W_{\widetilde{\mathcal{A}}}=\{W_{\widetilde{\mathcal{A}},l}\mid l=[1,q]\}$.

Setup(1^{ℓ}).KGC 生成系统主密钥 α 并秘密保存,同时选取阶都为 p 的乘法群 G_1 和 G_2 以及满足概念 1 所述性质的双线性映射$\hat{e}:G_1\times G_1\to G_2$.令 g、h 是 G_1 的两个生成元,并设置 $g_1=g^{\alpha}$.同时,选取一个 Hash 函数 $H_1:\{0,1\}^*\to Z_p^*$ 和一个 Hash 函数族 $H_{i=[1,K]}^{BF}=\{H_i^{BF}\mid i=[1,K]\}$,且$\forall H_i^{BF}:\{0,1\}^*\to m$(针对后面 4.3 节,$K$ 为 Hash 函数个数,m 为任何标准布隆过滤器[16-17]的位长度).最终系统公开参数为 $p_{\text{pub}}=\{G_1,G_2,\hat{e},g,g_1,\alpha,H_1,m,H_{i=[1,K]}^{BF}\}$.

KeyGen(α,P_{pub}).任何用户 u 加入系统时,该算法通过 u 与 KGC 的交互生成 u 的公私钥对$(K_{\text{pub}}^{u},K_{\text{pri}}^{u})$:

$u\to$KGC:u 随机选取一个大整数作为秘密值 a_u并秘密保存,同时计算其公开值 $R_u=h^{a_u}$.然后传输对值(R_u,ID_u)给 KGC,其中 ID_u是用户 u 的身份标识.

KGC$\to u$:KGC 接收到(R_u,ID_u),首先对其进行零知识证明[14],若证明失败则放弃交互,否则计算 $Q_{ID_u}=H_1(ID_u)$以及用户 u 的部分公钥 $H_u=(R_u)^{1/(\alpha-Q_{ID_u})}=(h)^{a_u/(\alpha-Q_{ID_u})}$.然后返回 H_u给用户 u.

u:u 获得 H_u后首先对其进行真实性认证,即判断$\hat{e}(H_u^{1/a_u},g_1g^{-Q_{ID_u}})\stackrel{?}{=}\hat{e}(h,g)$.若等式成立,则确定其公私钥对$(K_{\text{pub}}^{u},K_{\text{pri}}^{u})=(H_u,a_u)$.否则,公私钥对生成失败.

PreProcData($\mathcal{D}_{\bar{u}}$).对于$\forall\mathcal{D}_{\bar{u},\mathcal{A}}\in\mathcal{D}_{\bar{u}}$,将其预处理为索引关键词的集合 $W_{\mathcal{A}}$.并对$\forall W_{\mathcal{A},l}\in W_{\mathcal{A}}$,计算其权重相关信息.

PKES($\mathcal{D}_{\bar{u}},ID_u,H_u,a_{\bar{u}}$).对于$\forall\mathcal{A}\in\mathbb{A}$,$\forall W_{\mathcal{A},l}\in W_{\mathcal{A}}$,数据拥有者$\bar{u}$对 $W_{\mathcal{A},l}$进行 PKES 加密成为密文索引词 $EW_{\mathcal{A},l}$.该加密处理涉及$\bar{u}$的私钥 $K_{\text{pri}}^{\bar{u}}=a_{\bar{u}}$和被授权用户 u 的可认证公钥 $K_{\text{pub}}^{u}=H_u$,其具体过程如下:

(1)计算 $Q_{ID_{\bar{u}}}=H_1(ID_{\bar{u}})$和 $Q_{ID_u}=H_1(ID_u)$.

(2)判断$\hat{e}(H_u,g_1g^{-Q_{ID_u}})\stackrel{?}{=}\hat{e}(R_u,g)$.若等式成立,执行对 u 授权,否则放弃授权.

(3)计算
$$EW_{\mathcal{A},l}=\hat{e}((H_u)^{a_{\bar{u}}\cdot H_1(W_{\mathcal{A},l}\parallel\mathcal{A})},g_1g^{-Q_{ID_u}})=\hat{e}(h^{a_{\bar{u}}\cdot a_u\cdot H_1(W_{\mathcal{A},l}\parallel\mathcal{A})},g).$$

最终得$\mathcal{D}_{\bar{u},\mathcal{A}}$对应的所有密文索引词为 $EW_{\mathcal{A}}$,$\mathcal{D}_{\bar{u}}$对应的所有密文索引词为$EW=\{EW_{\mathcal{A}}\mid\mathcal{A}\in\mathbb{A}\}$.

RQED 机制中,我们将属性标识$\mathcal{A}$引入 PKES 加密及下面的 Trapdoor 计算,以此来区分数据及查询请求中的不同属性关键词.相对于传统单属性查询,如此处理的优势是多属性数据仅需采用同一个索引,提高索引空间效率.

BuildIndex(EW).输入密文索引词 EW,该算法输出$\mathcal{C}_{\bar{u}}$的密文索引$\mathcal{I}_{\bar{u}}$.

PreProcQuery($\mathcal{Q}$).令$\mathcal{Q}$中任何属性$\widetilde{\mathcal{A}}$中包含 q 个查询关键词集合 $W_{\widetilde{\mathcal{A}}}$,则对于$\forall\widetilde{\mathcal{A}}\in\widetilde{\mathbb{A}}$,$\forall W_{\widetilde{\mathcal{A}},l}\in W_{\widetilde{\mathcal{A}}}$,为 $W_{\widetilde{\mathcal{A}},l}$赋予相应的查询词权重.

Trapdoor($\mathcal{Q},a_u$).令已授权给 u 的数据拥有者集合为 OU_u,则对于$\forall\widetilde{\mathcal{A}}\in\widetilde{\mathbb{A}}$,$\forall W_{\widetilde{\mathcal{A}},l}\in W_{\widetilde{\mathcal{A}}}$,$\forall\bar{u}\in OU_u$,利用 u 的私钥 $K_{\text{pri}}^{u}=a_u$和$\bar{u}$的公开值$R_{\bar{u}}$计算查询子 Trapdoor $\mathcal{T}_{\bar{u},W_{\widetilde{\mathcal{A}},l}}$,即
$$\mathcal{T}_{\bar{u},W_{\widetilde{\mathcal{A}},l}}=\hat{e}(g^{a_u\cdot H_1(W_{\widetilde{\mathcal{A}},l}\mid\mid\widetilde{\mathcal{A}})},R_{\bar{u}}=h^{a_{\bar{u}}}).$$
最终得查询$\mathcal{Q}$的查询陷门 Trapdoor:
$$\mathcal{T}_{\mathcal{Q}}=\{\mathcal{T}_{\bar{u},W_{\widetilde{\mathcal{A}},l}}\mid\bar{u}\in OU_u,\widetilde{\mathcal{A}}\in\widetilde{\mathbb{A}},l\in[1,q]\}.$$

Query($\mathcal{T}_{\mathcal{Q}},\mathbb{I},\mathbb{C}$).输入$\mathcal{T}_{\mathcal{Q}}$、$\mathbb{I}$和$\mathbb{C}$,CSP 执行该算法并返回满足条件的所有排序结果或 Top-k 结果.

4.2　RQED 排序函数

基于上述 RQED 框架,我们设计了更合理且隐私保护的多属性多关键词密文查询排序函数.信息检索(IR)领域,数据(文档)与给定查询请求间的相似度是用于查询排序的最常用标准,相似度越高则越满足用户的查询兴趣.单关键词排序查询中相似度计算广泛采用的是 $TF\times IDF$ 规则,其中 TF 指数据中的简单关键词频,反映关键词在数据中的重要度;IDF 指反数据(文档)词频,反映关键词在整

个数据集中的区分度.

然而,在本文系统模型下,仅仅考虑单关键词与数据间的相似度是不可行的[3,11],特别是针对隐私保护的多关键词排序查询(例如,关键词词频泄露会导致遭受已知背景统计攻击[3]). RQED 机制中,在保护隐私前提下全面分析相似度计算的贡献因素,涉及两个重要概念:一方面最重要的是针对数据不同属性的隐私保护关键词权重(Privacy-preserving Data Keyword Weight, PDKW),另一方面则是用户自定义的查询请求不同属性关键词权重(Query Keyword Weight, QKW).

归纳起来,RQED 机制的关键点体现在:(1)数据发布时如何计算 PDKW;(2)构建索引时如何存储 PDKW;(3)查询时如何获取 PDKW;(4)如何利用 PDKW 和 QKW 计算查询结果相似度并排序. 考虑到多关键词查询中"OR"查询逻辑更为普遍,且"AND"查询逻辑可以通过"OR"实现,因此本文仅以"OR"查询为例进行阐述.

RQED 框架的 PreProcData 中,所做的主要处理任务是计算数据$\mathcal{D}_{\tilde{u}}$中每一个关键词的 PDKW,对于$\forall W_{\mathcal{A},l}\in W_{\mathcal{A}}$,定义其 PDKW 为$\mathcal{DW}_{W_{\mathcal{A},l}}$. 我们基于对传统经典方法的分析设计了更合理的 PDKW 计算函数,如下

$$\mathcal{DW}_{W_{\mathcal{A},l}}=\delta_{\mathcal{A}}\cdot\varepsilon_{W_{\mathcal{A},l}}\cdot (TF_{W_{\mathcal{A},l}}/(\lg(\max_{l\in[1,z_p^*]}\{TF_{W_{\mathcal{A},l}}\}\cdot|\mathcal{D}_{\tilde{u},\mathcal{A}}|))\cdot IDF_{W_{\mathcal{A},l}} \tag{1}$$

式(1)中,$\delta_{\mathcal{A}}$ 和 $\varepsilon_{W_{\mathcal{A},l}}$ 是针对实际应用特点,为增强 PDKW 计算客观性而设计的调整因子. 在数据挖掘领域的文摘抽取、QA 等技术中也有类似思路,而现有安全云数据排序查询方法却完全未考虑. $\delta_{\mathcal{A}}$ 用于关键词在不同属性中的重要度区分,即数据属性特征赋予了其值中所有关键词一个重要度. 以邮件系统为例,邮件主题中的关键词显然应比邮件正文中的关键词具有更高的重要度. $\varepsilon_{W_{\mathcal{A},l}}$ 用于不同长度关键词的重要度区分,一般情况下关键词越长说明其具有越高的语义区分性,其重要度就越高;$TF_{W_{\mathcal{A},l}}$ 指关键词 $W_{\mathcal{A},l}$ 在数据属性值$\mathcal{D}_{\tilde{u},\mathcal{A}}$中的简单词频(出现次数),$\max_{l\in[1,z_p^*]}\{TF_{W_{\mathcal{A},l}}\}$ 指$\mathcal{D}_{\tilde{u},\mathcal{A}}$中最大的关键词词频;$|\mathcal{D}_{\tilde{u},\mathcal{A}}|$指$\mathcal{D}_{\tilde{u},\mathcal{A}}$的数据长度,注意这里是指$\mathcal{D}_{\tilde{u},\mathcal{A}}$中所有关键词的长度总和,而非简单的$\mathcal{D}_{\tilde{u},\mathcal{A}}$中字个数,目的是降低无效停用词的影响;最终定义$(TF_{W_{\mathcal{A},l}}/(\lg(\max_{l\in[1,z_p^*]}\{TF_{W_{\mathcal{A},l}}\}\cdot|\mathcal{D}_{\tilde{u},\mathcal{A}}|))$作为相对关键词词频. 如此做:一方面考虑了数据长度对关键词权重的反作用,另一方面,通过相对词频实现关键词真实词频信息的隐私保护.

对于反数据(文档)词频 $IDF_{W_{\mathcal{A},l}}$,传统计算方式:$IDF_{W_{\mathcal{A},l}}=\mathbb{K}/f_{W_{\mathcal{A},l}}$($\mathbb{K}$指系统所有数据量$|\mathbb{D}|$,$f_{W_{\mathcal{A},l}}$指系统数据集$\mathbb{D}$中包含关键词$W_{\mathcal{A},l}$的数据量). 然而,在本文系统模型中,任何数据拥有者基本上不可能掌握系统所有数据,则传统 $IDF_{W_{\mathcal{A},l}}$ 计算显然不可行. 更重要的是,我们考虑了用户可访问数据范围,对于只能访问系统部分数据的用户,利用系统所有数据$\mathbb{D}$来进行其查询结果排序并不精确甚至无意义. 因此,我们在 RQED 框架的 BuildIndex 中,仅仅计算索引关键词 $W_{\mathcal{A},l}$ 的部分 PDKW,即$\mathcal{PW}_{W_{\mathcal{A},l}}$,见式(2). 并将$\mathcal{PW}_{W_{\mathcal{A},l}}$存储于$\mathcal{D}_{\tilde{u}}$的密文索引$\mathcal{I}_{\tilde{u}}$中. 而在 Query 中,首先从密文索引中获取查询关键词 $W_{\tilde{\mathcal{A}},l}$ 的部分 PDKW,即$\mathcal{PW}_{W_{\tilde{\mathcal{A}},l}}$. 然后再计算 $W_{\tilde{\mathcal{A}},l}$的真实 PDKW,即$\mathcal{DW}_{W_{\tilde{\mathcal{A}},l}}$,见式(3). 如下所示:

$$\mathcal{PW}_{W_{\mathcal{A},l}}=\delta_{\mathcal{A}}\cdot\varepsilon_{W_{\mathcal{A},l}}\cdot (TF_{W_{\mathcal{A},l}}/(\lg(\max_{l\in[1,z_p^*]}\{TF_{W_{\mathcal{A},l}}\}\cdot|\mathcal{D}_{\tilde{u},\mathcal{A}}|))) \tag{2}$$

$$\mathcal{DW}_{W_{\tilde{\mathcal{A}},l}}=\mathcal{PW}_{W_{\tilde{\mathcal{A}},l}}\cdot IDF_{W_{\tilde{\mathcal{A}},l}} \tag{3}$$

$$IDF_{W_{\tilde{\mathcal{A}},l}}=\mathbb{K}_u/f_{W_{\tilde{\mathcal{A}},l},u} \tag{4}$$

式(4)计算 $IDF_{W_{\tilde{\mathcal{A}},l}}$ 的最大特点是去除用户无法访问的无关数据对其查询排序的影响,令查询用户 u 所能访问的所有数据为$\mathcal{D}_u$,则其中$\mathbb{K}_u=|\mathcal{D}_u|$,$f_{W_{\tilde{\mathcal{A}},l},u}$则指$\mathcal{D}_u$中包含查询关键词 $W_{\tilde{\mathcal{A}},l}$的数据量.

最后在 Query 中,为使排序更合理,我们在查询相似度$\mathcal{STM}(\mathcal{C}_{\tilde{u}},\mathcal{Q})$的计算中引入熵,具体如下

$$\mathcal{STM}(\mathcal{C}_{\tilde{u}},\mathcal{Q})=\sum_{\tilde{\mathcal{A}}\in\tilde{\mathbb{A}}}\left\{ENT(W_{\tilde{\mathcal{A}}})+\frac{\sum_{l=1}^{q}\mathcal{DW}_{W_{\tilde{\mathcal{A}},l}}\cdot\mathcal{QW}_{W_{\tilde{\mathcal{A}},l}}}{\sum_{l=1}^{q}\mathcal{DW}^2_{W_{\tilde{\mathcal{A}},l}}+\sum_{l=1}^{q}\mathcal{QW}^2_{W_{\tilde{\mathcal{A}},l}}-\sum_{l=1}^{q}\mathcal{DW}_{W_{\tilde{\mathcal{A}},l}}\cdot\mathcal{QW}_{W_{\tilde{\mathcal{A}},l}}}\right\} \tag{5}$$

$$ENT(W_{\tilde{\mathcal{A}}})=\frac{1}{\log_2 q}\cdot\sum_{l=1}^{q}(-P_{W_{\tilde{\mathcal{A}},l}}\cdot\log_2 P_{W_{\tilde{\mathcal{A}},l}}) \tag{6}$$

$$P_{W_{\tilde{\mathcal{A}},l}}=\mathcal{DW}_{W_{\tilde{\mathcal{A}},l}}/\sum_{l=1}^{q}\mathcal{DW}_{W_{\tilde{\mathcal{A}},l}} \tag{7}$$

其中$\mathcal{QW}_{W_{\tilde{\mathcal{A}},l}}$为查询关键词 $W_{\tilde{\mathcal{A}},l}$的 QKW. 正如式(5),多属性上的相似度是单属性相似度的简单计算. 对于$\mathcal{Q}$中任何属性$\tilde{\mathcal{A}}$上的查询关键词 $W_{\tilde{\mathcal{A}}}$,在计算其与索引$\mathcal{I}_{\tilde{u}}$对应密文数据$\mathcal{C}_{\tilde{u}}$的相似度时,引入熵

$ENT(W_{\tilde{\mathcal{A}}})$作为调整因子，用于衡量$W_{\tilde{\mathcal{A}}}$中被$\mathcal{I}_{\tilde{u}}$正确命中的关键词的分布均匀度. 换句话说，同等条件下，查询关键词被命中越多，熵越大，其相似度就倾向于越大. 该熵仅适用于“OR”查询逻辑，“AND”处理相对更简单. 为了确保熵不至于超过 PDKW 而过度支配相似度值，我们控制其值域为$[0,1]$，见式(6)中的$\frac{1}{\log_2 q}$. 而式(7)中，关键词$W_{\tilde{\mathcal{A}},l}$被正确命中的概率依赖于$W_{\tilde{\mathcal{A}},l}$的 PDKW 占$W_{\tilde{\mathcal{A}}}$中 PDKW 总和的比率.

4.3　RQED 索引机制

基于上述 RQED 框架和 RQED 排序方法，我们设计了 RQED 索引机制，如算法 1 和算法 2 所示. RQED 索引机制示例见图 2. 该机制中：(1) 通过计数布隆过滤器技术来实现索引关键词的部分 PDKW 的存储和获取；(2) 通过动态布隆过滤器索引进一步提升索引空间效率，见算法 1 中 BuildIndex；(3) 通过构建层次动态布隆过滤器索引结构，使查询能够快速定位而进一步提高查询执行时间效率，见算法 1 中 OptiIndex 和 Query.

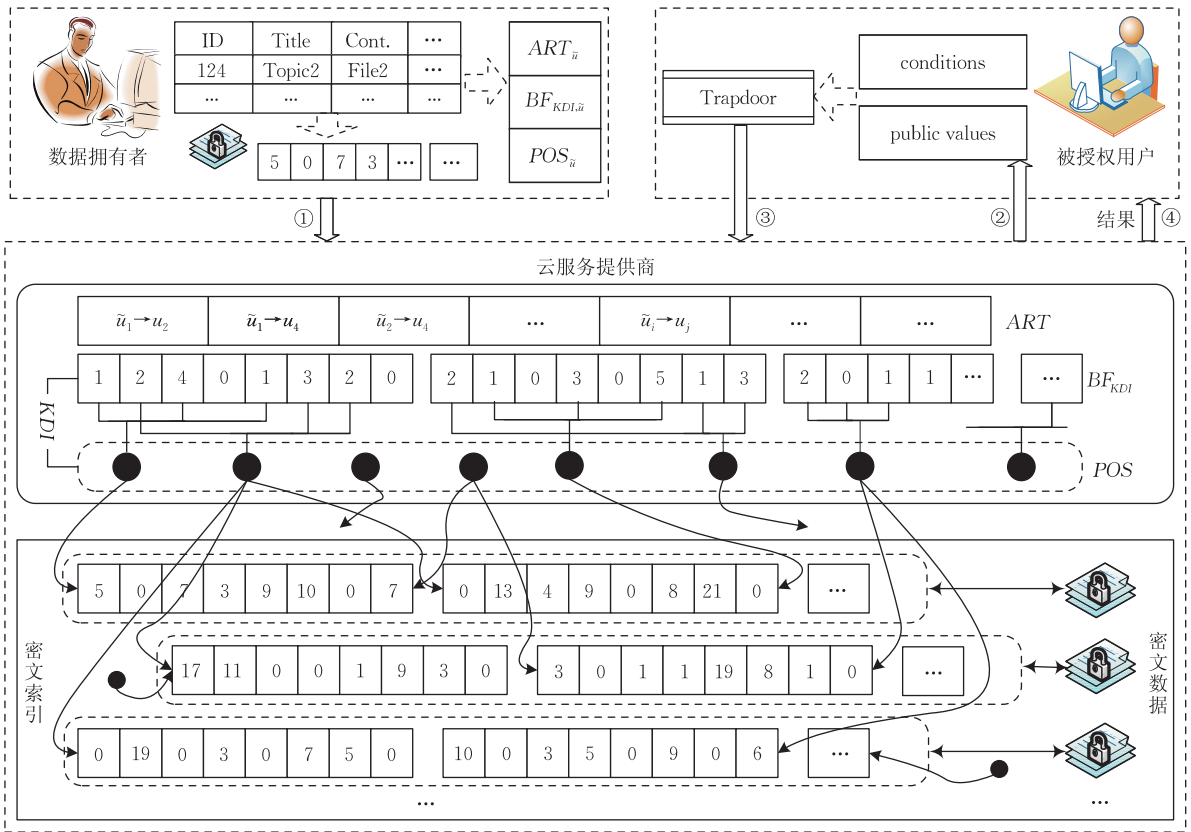


图 2　RQED 索引机制示例

算法 1.　RQED 索引机制.

1. Setup(),KeyGen()：除 RQED 框架中所有处理外，CSP 还需初始化 $KDI=\{BF_{KDI},POS\}$和 ART；
2. PreProcData()：对于$\forall W_{\mathcal{A},l}$，计算$\mathcal{PW}_{W_{\mathcal{A},l}}$；　//式(2)
3. PKES()：加密$\mathcal{D}_{\tilde{u}}$中所有索引关键词为 EW；
4. BuildIndex($EW,R_{\tilde{u}}$)：由数据拥有者$\tilde{u}$执行：
4.1. 初始化私有 KDI，即$(BF_{KDI,\tilde{u}}^{\ell=0},POS_{\tilde{u}}[K]_{j=0})$；
4.2. 初始化一个动态计数布隆过滤器索引$\mathcal{I}_{\tilde{u}}$；
4.3. For $\forall EW_{\mathcal{A},l}\in EW$
4.3.1. $POS_{\tilde{u}}[K]_{j++}=H_{i=[1,K]}^{BF}(EW_{\mathcal{A},l})$；
4.3.2. InsertEW($BF_{KDI,\tilde{u}},EW_{\mathcal{A},l}$)；　//见算法 2
4.3.3. InsertEW($\mathcal{I}_{\tilde{u}},EW_{\mathcal{A},l},\mathcal{PW}_{W_{\mathcal{A},l}}$)；　//见算法 2
4.3.4. 生成 $ART_{\tilde{u}}$用于记录$\tilde{u}$授权的所有用户；
4.3.5. $\tilde{u}$外包给 CSP $\{\mathcal{C}_{\tilde{u}},\mathcal{I}_{\tilde{u}},BF_{KDI,\tilde{u}},POS_{\tilde{u}},ART_{\tilde{u}}\}$；
5. OptiIndex()：接收到$\{\mathcal{C}_{\tilde{u}},\mathcal{I}_{\tilde{u}},BF_{KDI,\tilde{u}},POS_{\tilde{u}},ART_{\tilde{u}}\}$后，CSP 执行如下处理：
5.1. 分别将 $ART_{\tilde{u}}$和 $POS_{\tilde{u}}[K]_{j=1}^{n_{u_i}}$添加到 ART 和 POS，并确保 ART 和 POS 中无相等值的冗余项；
5.2. MergeDBF($BF_{KDI,\tilde{u}},BF_{KDI}$)；　//见算法 2
5.3. 构建或更新系统层次动态布隆过滤器索引结构：
For 每一个密文索引$\mathcal{I}\in\mathbb{I}$
For 每一个位游标 $POS[K]_i\in POS$
For $\mathcal{I}$ 中每一个标准布隆过滤器$\mathcal{I}^{\iota}\in\mathcal{I}$
For $\forall BF_{KDI}^{\ell}\in BF_{KDI}$
If $BF_{KDI}^{\ell}[POS[K]_i]$和$\mathcal{I}^{\iota}[POS[K]_i]$中所有值都不为 0

Then 连接 $BF^{\ell}_{KDI}[POS[K]_i]$到$\mathcal{I}^{\iota}$的指针；

6. PreProcQuery()：对于$\forall W_{\tilde{\mathcal{A}},l} \in W_{\tilde{\mathcal{A}}}$，赋予$\mathcal{QW}_{W_{\tilde{\mathcal{A}},l}}$；

7. Trapdoor()：生成查询 Q 的 Trapdoor $\mathcal{T}_Q$；

8. Query()：CSP 按如下执行查询处理：

8.1. 定义$\mathbb{I}_{\text{hit}}$用于存储被$\mathcal{T}_Q$查询命中的所有密文索引；

8.2. For $\forall \mathcal{T}_{\tilde{u},W_{\tilde{\mathcal{A}},l}} \in \mathcal{T}_Q$

$\ell_{\tilde{u},W_{\tilde{\mathcal{A}},l}}[K]=H^{BF}_{i=[1,K]}(\mathcal{T}_{\tilde{u},W_{\tilde{\mathcal{A}},l}})$；

If $\exists BF^{\ell}_{KDI} \in BF_{KDI}$，

使得 $BF^{\ell}_{KDI}[\ell_{\tilde{u},W_{\tilde{\mathcal{A}},l}}[K]]$中所有值都不为 0

Then 获取由$BF^{\ell}_{KDI}[\ell_{\tilde{u},W_{\tilde{\mathcal{A}},l}}[K]]$指针指向的所有动态布隆过滤器索引，并将其添加进$\mathbb{I}_{\text{hit}}$；

8.3. For 每一个被$\mathcal{T}_Q$命中的密文索引$\mathcal{I}(\mathcal{I}\in\mathbb{I}_{\text{hit}})$

QueryT($\mathcal{I},\mathcal{T}_Q$)：利用$\mathcal{I}$和$\mathcal{T}_Q$计算其对应密文数据与查询之间的相似度； //见算法 2

8.4. Return 按相似度排序的所有或 Top-k 结果密文数据.

RQED 索引机制中，CSP 需要维护一个系统授权关系表 ART 和系统密文关键词典索引 KDI. ART 用于记录系统所有用户间的授权关系，但与具体授权数据无关. KDI 是由以下两部分组成的数据结构：(1) 用于编码系统所有密文索引词的动态布隆过滤器 BF_{KDI}；(2) 用于记录所有密文索引词在 BF_{KDI} 中映射位置的位游标表 $POS=\{POS[K]_i \mid i\in[1,Z^*_P]\}$(其中 Z^*_P 为系统唯一密文索引词个数，每个密文索引词经 Hash 函数簇 $H^{BF}_{i=[1,K]}$ 映射到 K 个位游标 $POS[K]_i$). 由于无法枚举系统所有的索引词及其授权，也就无法枚举系统所有的密文索引词及其位游标. 因此，我们初始化 BF_{KDI} 和 POS 为空，并在每次用户发布数据时执行同步更新.

为此，对于$\forall \tilde{u}\in OU$，当其执行数据外包并授权时，需要同时构建一个用户私有 ART(定义为 $ART_{\tilde{u}}$)和用户私有 KDI. 前者用于记录$\tilde{u}$所要授权的所有用户；后者同样由动态布隆过滤器 $BF_{KDI,\tilde{u}}$ 和位游标表 $POS_{\tilde{u}}$ 组成，用于编码$\tilde{u}$所要授权数据的所有密文索引词及其位游标. 为了节省空间开销，BF_{KDI} 和 $BF_{KDI,\tilde{u}}$ 采用动态布尔型布隆过滤器[17]索引，即所有位值都为 0 或 1. 而对于任何用户数据的密文索引($\forall \mathcal{I}_{\tilde{u}}\in\mathbb{I}_{\tilde{u}}$)和系统任何密文索引($\forall \mathcal{I}\in\mathbb{I}=\{\mathbb{I}_{\tilde{u}} \mid \tilde{u}\in OU\}$)，涉及记录索引关键词的部分 PDKW，则采用动态计数型布隆过滤器[17]构建和维护.

本质上，所有动态布隆过滤器是具有相同特征的标准布隆过滤器序列[17]. 我们有如下定义：对于$\forall \mathcal{I}\in\mathbb{I}$，有$\mathcal{I}=\{\mathcal{I}^{\iota} \mid \iota\in[1,Z^*_P]\}$；对于$\forall \mathcal{I}_{\tilde{u}}\in\mathbb{I}_{\tilde{u}}$，有$\mathcal{I}_{\tilde{u}}=\{\mathcal{I}^{\iota}_{\tilde{u}} \mid \iota\in[1,Z^*_P]\}$；$BF_{KDI}=\{BF^{\iota}_{KDI} \mid \iota\in[1,Z^*_P]\}$且$\forall BF_{KDI,\tilde{u}}=\{BF^{\iota}_{KDI,\tilde{u}} \mid \iota\in[1,Z^*_P]\}$. 同时有$POS=POS[K]^{Z^*_P}_{i=1}=\{POS[K]_i \mid i\in[1,Z^*_P]\}$，且对于$\forall POS_{\tilde{u}}$有$POS_{\tilde{u}}=POS_{\tilde{u}}[K]^{Z^*_P}_{j=1}$. 动态布隆过滤器根据实际需要动态添加标准布隆过滤器成员，以此避免预置统一位空间造成的无效开销. 其每一个标准布隆过滤器成员(即$\mathcal{I}^{\iota}$、$\mathcal{I}^{\iota}_{\tilde{u}}$、$BF^{\iota}_{KDI}$、$BF^{\iota}_{KDI,\tilde{u}}$)都有 3 个属性：$\{mp,ct,st\}$. 其中 mp 指在限定最高查询假阳性率[17]条件下其所能编码的最多索引关键词数；ct 指其已经编码存储了的索引关键词数；st 指其激活状态，若 $ct=mp$，则 $st=$false 且说明其已饱和，否则 $st=$true 且说明其处于活跃状态.

RQED 索引机制中，动态布隆过滤器主要涉及以下 3 类操作：索引关键词的插入 InsertEW()、动态布隆过滤器的合并 MergeDBF()以及 Trapdoor 的查询执行 QueryT()，见算法 2. InsertEW()中涉及索引关键词 $W_{\mathcal{A},l}$的部分 PDKW 的存储，见算法 2 中式(8). QueryT()中则涉及查询关键词 $W_{\tilde{\mathcal{A}},l}$的部分 PDKW 的获取，见算法 2 中式(9). 尽管由于布隆过滤器的假阳性会导致有一定概率存在：当 $W_{\tilde{\mathcal{A}},l}=W_{\mathcal{A},l}$时，$\mathcal{PW}_{W_{\tilde{\mathcal{A}},l}}\neq\mathcal{PW}_{W_{\mathcal{A},l}}$，但通过文献[16-17]的分析，该概率可以通过调整布隆过滤器索引参数而达到可忽略.

算法 2. 动态布隆过滤器索引操作.

InsertEW($DBF,EW_{\mathcal{A},l},*$)：新增 $EW_{\mathcal{A},l}$ 以及 $W_{\mathcal{A},l}$ 的部分 PDKW ($\mathcal{PW}_{W_{\mathcal{A},l}}$)到动态布隆过滤器 DBF

Input1：$DBF=BF_{KDI,\tilde{u}}$，$*=$null

Input2：$DBF=\mathcal{I}_{\tilde{u}}$，$*=\mathcal{PW}_{W_{\mathcal{A},l}}$

1. 找出 DBF 中首个非饱和标准布隆过滤 DBF^{ι}；

2. 重置 DBF^{ι}中位游标为 $H^{BF}_{i=[1,K]}(EW_{\mathcal{A},l})$的所有值：

If 若 DBF^{ι}为布尔型，则设置所有值为 1；

$DBF^{\iota}[H^{BF}_{i=[1,K]}(EW_{\mathcal{A},l})]=\{1,1,\cdots,1\}$；

If 若 DBF^{ι}为计数型，则所有值都加上$\mathcal{PW}_{W_{\mathcal{A},l}}$；

$$DBF^{\iota}[H^{BF}_{i=[1,K]}(EW_{\mathcal{A},l})]+=\mathcal{PW}_{W_{\mathcal{A},l}} \quad (8)$$

3. $DBF^{\iota}.ct=DBF^{\iota}.ct+1$；

4. If $DBF^{\iota}.ct=DBF^{\iota}.mp$ Then $DBF^{\iota}.st=$false；

5. If DBF 中不存在非饱和标准布隆过滤器

6. 新建一个标准布隆过滤器 $DBF^{\iota}(\iota=|DBF|+1)$；

7. 重置 DBF^{ι}中位游标为 $H^{BF}_{i=[1,K]}(EW_{\mathcal{A},l})$的所有值，与上述第 2、3 行相同处理；

MergeDBF(DBF_1,DBF_2)：合并 DBF_1到 DBF_2；

Input：$DBF_1=BF_{KDI,\tilde{u}}$，$DBF_2=BF_{KDI}$

1. For each 标准布隆过滤器 $DBF^{\iota}_1(\iota=[1,|DBF_1|+1])$

2. If 存在一个标准布隆过滤器 $DBF^{\ell}_2\in DBF_2$

使得 $DBF^{\ell}_2.ct+DBF^{\iota}_1.ct\leq DBF^{\ell}_2.mp$

3. Then $DBF^{\ell}_2=DBF^{\iota}_1 \mid DBF^{\ell}_2$；

4. Else 直接顺序添加 DBF^{ι}_1到 DBF_2序列末尾；

QueryT($\mathcal{I},\mathcal{T}_Q$)：利用$\mathcal{T}_Q$在索引$\mathcal{I}$上执行查询；

1. For $\forall \mathcal{T}_{\tilde{u},w_{\tilde{A},l}} \in \mathcal{T}_Q$
2. $t_{\tilde{u},w_{\tilde{A},l}}[K]=H_{i=[1,K]}^{BF}(\mathcal{T}_{\tilde{u},w_{\tilde{A},l}})$;
3. If $\exists \mathcal{I}^k \in \mathcal{I}$
使得$\mathcal{I}^k[t_{\tilde{u},w_{\tilde{A},l}}[K]]$中所有值都不为 0.
4. Then $\mathcal{PW}_{w_{\tilde{A},l}}=\min(\mathcal{I}^k[t_{\tilde{u},w_{\tilde{A},l}}[K]])$　　(9)
5. 计算 $W_{\tilde{A},l}$的 PDKW，即$\mathcal{DW}_{w_{\tilde{A},l}}$;　　//见式(3)
6. 计算$\mathcal{I}$对应密文数据与 Q 的相似度.　　//见式(5)

5　安全性与实验分析

5.1　安全性分析

RQED 机制的安全性等价于在 3.2 节的攻击模型下，对 RQED 隐私需求的保障.

数据隐私保障：数据本身隐私保障主要体现在加密算法及基于密钥管理的访问控制上，对此目前也有大量的研究[15]，其并不是本文的研究重点.

索引隐私保障：(1) 基于无证书认证的 PKES 安全保障为后面的隐私保障奠定基础. 我们基于无证书公钥思想[13-14]改进传统 PKES[8-9]以实现其不依赖于可信第三方和安全传输信道的公钥认证，在算法设计上简化文献[14]中公私钥组成并继承了文献[14]的安全性，因此采用文献[14]中相同方法可证明能有效抵御外部攻击者，甚至是 KGC 通过直接攻击用户私钥或替换用户公钥所进行的适应性选择消息和身份标识 ID 攻击. 而 PKES 加密本身，其计算与文献[8]基本相同，其安全性保障也同等. 由于篇幅限制，本文不再重复详述. (2) 索引隐私信息私密性保障，体现在索引词及词频的私密性. RQED 机制通过融入被授权用户公钥和数据拥有者私钥的 PKES 加密保障了索引关键词私密性，且通过布隆过滤器 Hash 映射进一步隐藏密文索引词，一定程度上避免了已知密文索引词的统计攻击. 同时，RQED 机制中索引关键词的 PDKW 仅仅是相关于数据最大词频的相对值(见式(1))，而非直接的词频，此方法有效抵御了在 3.2 节攻击模型下对词频的统计攻击，避免猜测和推理出部分的数据隐私信息. (3) 索引无关联保障. RQED 机制通过 PKES 技术和布隆过滤器索引中位与索引词的多对多映射特征(即同一索引词映射多个位且同一位映射多索引词)，有效防止了攻击者利用多个索引间关联攻击其数据间隐私关联信息(如包含相同主题等). 同时，若 Hash 函数足够随机，对于被授予相同用户但来自不同数据拥有者、或者来自同一数据拥有者但被授予不同用户的相同索引关键词，都将生成不同的布隆过滤器索引映射位，进一步保证索引无关联性.

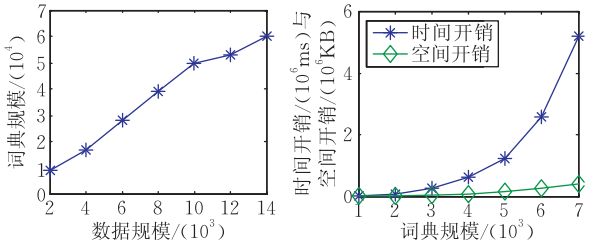
查询隐私保障：(1) 查询可获得性保障，即选择性授权查询控制. 显然，RQED 机制通过 PKES 有效实现数据拥有者对查询执行有效性的控制，即任何用户的查询 Trapdoor 能在某个布隆过滤器索引上正确执行查询，当且仅当用户被授予了该索引对应数据的访问权限. 同时也避免了基于对称密钥加密方法实现该保障时所带来的复杂密钥管理问题. (2) Trapdoor隐私信息私密性保障. RQED 机制利用查询用户私钥和所有授权用户公开值，本地生成云端，可直接执行布隆过滤器索引查询的 Trapdoor，使得在 3.2 节的攻击模型下确保 Trapdoor 对隐私请求信息的无泄漏. 当然，数据拥有者有能力通过与 CSP 的合谋来分析出他已授权的用户在他自己所拥有数据上的查询信息，这不可避免，且我们认为是合理的. (3) Trapdoor 无关联保障. RQED 机制中，对于同一用户的不同查询词、同一用户针对不同数据拥有者的同一查询词、不同用户的任何查询词，其 Trapdoor 不同且无关联性. 这就意味着即使数据拥有者与 CSP 合谋进行已知背景攻击，也无法获取用户在其他数据拥有者所拥有数据上的任何隐私查询信息. 对于同一用户在不同查询中的相同关键词，尽管文献[3]通过在 Trapdoor 生成中引入随机数以实现其 Trapdoor 的无关联性，但却以牺牲排序查询精确性为代价. 并且事实上，文献[3]中内部用户完全可以根据自己所掌握的密钥推理出其他用户的查询请求隐私. RQED 机制对此暂无特殊处理也是可接受的，因为任何攻击者最大限度能获取到的信息也就是用户提交相同查询请求的次数，而不能推理出查询请求的任何隐私信息.

5.2　实验分析

实验系统采用 Java 语言实现，执行在 Intel Core i3-2330M 2.20GHz 处理器及 Win7 64 位操作系统平台上. 实验采用真实数据集：安然公司邮件数据集(Enron Email Dataset, EED)①，其中每一封邮件即作为一条多属性数据(主题、正文等)，每一次邮件发送即作为发件人外包数据并对收件人授权访问. 实验是针对现有典型的多关键词密文排序查询方法进行对比分析，包括文献[3]中 MRSE-1、MRSE-2 和文献[6]中 KNN (KNN 查询能够应用于多关键词排序查询). 文献[3,6]都需预置系统关键词词典，且仅对数据集中包含于预置词典中的关

① Cohen W W. Enron email dataset, http://www.cs.cmu.edu/~enron/

键词执行实验，而其余关键词则被遗弃.但一定程度上，预置系统所有关键词是比较困难的(特别是领域术语或缩写词等).图 3 展示了 EED 中不同规模数据所包含的唯一关键词总数以及随关键词词典规模增大而致使文献[3,6]中产生的庞大密钥矩阵计算和存储开销(关键是这些开销需要用户去承担).



(a) 不同规模数据中关键词词典规模　(b) 不同词典规模产生的客户端密钥矩阵计算和存储开销

图 3

考虑到上述局限，即与文献[3,6]的对比实验中无法执行大规模词典及数据.因此，为了实验具有可比性和更客观地展示 RQED 机制的优势，我们的实验存在预置系统词典和不预置系统词典两类数据集.从 EED 抽取这两类实验数据集的方法如下：(1) PreKWDic，针对 RQED 机制与文献[3,6]的对比实验，主要是为了具有可比性.首先从 EED 随机抽取规模 M 的数据集，再从中随机选取 N 个关键词作为规模 N 的词典，并去除不包含于词典中的数据关键词.根据图 3 中结果及文献[3]中的实验设计，我们取与其相同规模的数据集执行实验(M，$N<12\times10^3$)，并同样不考虑多拥有者数据上的查询.(2) NoKWDic，针对 RQED 机制自身优势的评估，但由于论文篇幅限制，这里仅展示用该数据集进行查询执行效率的评估.无需预置关键词词典，只需从 EED 随机抽取规模为 M 的数据集，实验执行于所选取数据集的所有关键词.

令以下实验中所有给定的查询为 Q 且其总共包含了 q 个查询关键词.

实验 1. 查询假阳性率.

本实验主要考虑 RQED 机制的查询假阳性率，按 NoKWDic 方式抽取数据集且 $M=1\times10^4$.通过设置不同的布隆过滤器位长度 m 和 Hash 函数个数 K 构建密文索引，分析其查询假阳性率变化，结果如图 4.其中假阳性率取值为所有关键词查询假阳性率的平均值，且布隆过滤器位长度是指动态布隆过滤器索引总的位长度.结果可以看出通过调整 m，K，其假阳性率可以忽略，且容忍一定假阳性率情况下，可以节省较大的位空间.

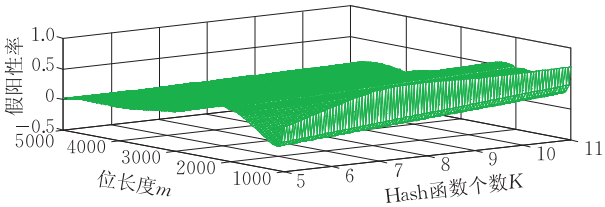
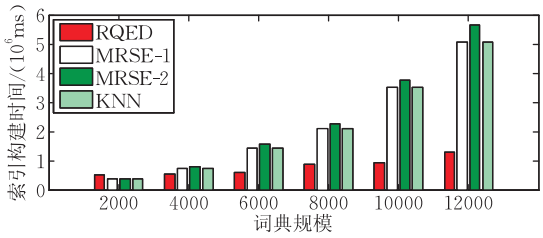


图 4 RQED 机制查询假阳性率分析
(其中 m，K 变化，总数据规模为 1×10^4)

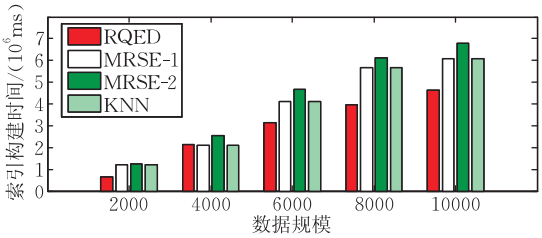
为了对比更清晰合理，在下面的所有实验中，确保 RQED 机制中查询假阳性率小于 2%.

实验 2. 索引构建时间效率.

按 PreKWDic 方式抽取实验数据集并分别根据 M 值和 N 值的变化执行实验，对比分析索引构建时间代价，结果如图 5(a)、(b).RQED 机制索引构建主要包括索引词的 PKES 加密及布隆过滤器索引构建，MRSE-1、MRSE-2 和 KNN 则主要包括一次密钥矩阵的转置、取逆计算和数据索引向量的构建.事实上，预置词典致使数据中除去了一部分索引关键词而使 RQED 代价较小，然而增加词典规模却同样会使 MRSE 和 KNN 计算代价陡增，因此，图 5 展示出无论随词典或数据规模增长，RQED 机制较之 MRSE-1、MRSE-2 和 KNN 都有较大优势.



(a) $M=1000$，N值变化



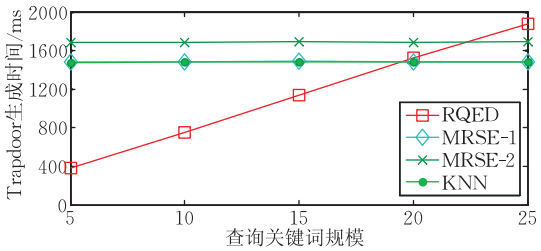
(b) $N=4000$，M值变化

图 5 密文索引构建时间代价对比分析

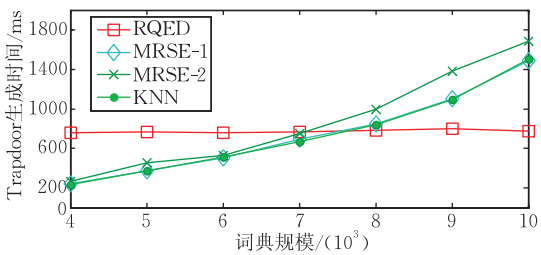
实验 3. Trapdoor 生成时间效率.

客户端代价主要包括 Trapdoor 生成的时间开销及用户索引私钥存储的空间开销，本实验仅分析前者.按 PreKWDic 方式抽取实验数据集执行对比分析，显然 Trapdoor 生成与 M 值无关，但与用户查询请求的复杂性有关.因此根据 N 值和 q 值的变化执行实验，结果如图 6(a)、(b).RQED 机制将查询关键词的 PKES 加密引入 Trapdoor 生成，以求获取

更强的查询隐私保护，其中的 Trapdoor 生成时间仅与 q 值相关，而无关于 N 值. 事实上，实际云系统中 N 值会相当大，而 q 值一般不超过 25 就能够充分表达用户的查询需求. 并且常规用户行为分析表明，当用户有目的查询时，能容忍的等待查询响应时间在 5 s～10 s. MRSE-1、MRSE-2 和 KNN 的 Trapdoor 生成时间虽无关于 q 值，但完全依赖 N 值. 同时，他们还涉及密钥矩阵的逆运算和存储，当 N 较大时，其开销对于普通客户端将难以承受. 因此，结果明显看出 RQED 机制的 Trapdoor 生成时间效率优势.



(a) N=10 000, q值变化



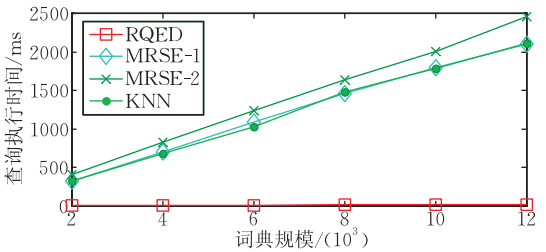
(b) q=10, N值变化

图 6　Trapdoor 生成时间代价对比分析

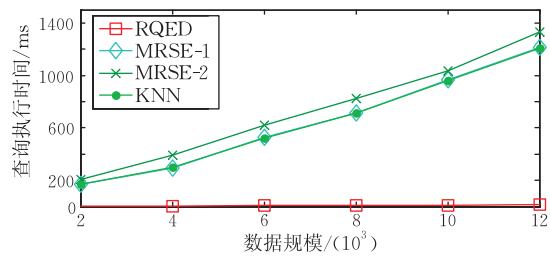
实验 4.　查询执行时间效率.

查询执行效率是任何应用系统的首要考虑因素，本实验用以分析索引查询执行及其结果排序处理的时间开销：(1) 按 PreKWDic 方式抽取数据集，并分别按照 M 值、N 值和 q 值的不同变化执行对比分析，结果如图 7. 相对于现有方法，RQED 机制的查询执行时间与 N 值无关，更切合实际应用. 现有方法的查询时间与 M 值同比增长，而 RQED 则采用层次布隆过滤器索引使查询执行仅作用于命中数据集而不产生无用开销，使查询时间基本不受 M 值的影响，从而适用于云环境下大规模密文数据查询. 较之现有方法，RQED 机制查询执行效率会受 q 值的影响，然而实验 3 中分析可知实际应用中（如 Google、百度、邮件系统等），包含 25 个关键词的查询请求足以表达用户的查询需求或兴趣.

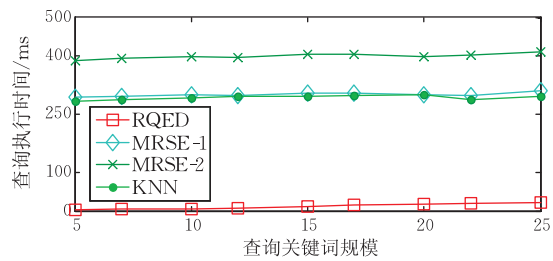
(2) 为使实验更充分体现 RQED 机制的优势，按 NoKWDic 方式抽取实验数据集，并分别根据 M 值和 q 值的变化执行，RQED 机制的查询执行效率评估. 并为了体现大规模密文数据查询需求，实验设置数据规模 M 为 4×10^6 到 24×10^6，结果如图 8 所示.



(a) M=10 000, q=10, N值变化

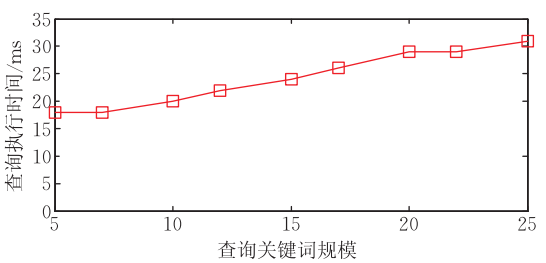


(b) N=4000, q=10, M值变化

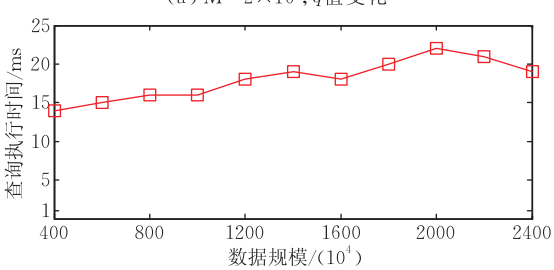


(c) M=5000, N=4000, q值变化

图 7　查询执行时间代价对比分析



(a) $M=2\times10^7$, q值变化



(b) q=10, M值变化

图 8　RQED 机制的查询执行时间效率评估

实验 5.　空间效率.

空间开销最终表现为网络传输和存储代价，由于各方法对于密文数据的空间开销一般无差别，则重点表现在密文索引和用户查询私钥两方面. 首先

按 PreKWDic 方式抽取数据集，并执行相关实验对比分析密文索引的空间开销. 所有方法基本上与 M 值呈同比增长，因此这里只根据 N 值变化执行实验，结果如表 1，可见 RQED 机制中动态布隆过滤器索引大大降低了索引空间开销. 对于索引私钥，RQED 机制中用户只需存储一对公私钥，现有方法则需存储密钥矩阵且随着关键词词典规模增大而带来巨大空间开销，并且当涉及多数据拥有者选择性查询授权时，客户端更是无法承受，对于该明显的对比，本文不再作实验分析.

表 1 RQED 机制与 MRSE-1、MRSE-2 和 KNN 的索引空间代价对比（$M=1000$，N 值变化）

词典规模	空间代价/(10^3KB)			
	RQED	MRSE-1	MRSE-2	KNN
4000	0.986	31.2	32.5	31.2
6000	1.85	46.8	48.1	46.8
8000	3.03	62.5	63.8	62.5
10000	3.85	78.1	79.4	78.1
12000	4.76	93.7	95.0	93.7

实际应用中，用户一般不会采用轻量级客户端发布或更新规模较大数据，而查询则不同. 总结所有实验结果得出结论，RQED 机制较之现有多关键词密文排序查询方法有明显的时空效率优势.

6 结 论

本文针对云环境下多数据拥有者安全数据外包及其选择性查询控制等特征，提出了一种支持大规模云数据上隐私保护的多关键词高效密文排序查询方法 RQED. 通过设计基于无证书认证 PKES 及其 Trapdoor 完全本地生成的 RQED 框架增强查询隐私保护；通过设计隐私保护的多属性多关键词排序函数实现更合理的隐私保护密文排序查询；通过设计基于动态计数型布隆过滤器的层次索引结构提高密文查询时空效率. 理论和实验分析表明 RQED 机制不仅满足云环境下用户个性化查询需求，也在确保查询强隐私保护和高准确性的同时，具有明显的时空效率优势.

同时，我们基于前期可信云存储系统关键技术的研究方法和 Hadoop 研发了可信 Web 电子邮件系统和可信云存储平台. 本文方法在其中实现并得到较好应用. 更重要的是，本文方法基于动态布隆过滤器的更新操作，能够高效支持隐私保护的密文索引动态更新，由于篇幅限制，我们将在下一个工作中详述. 我们下一步也将结合云平台底层计算和存储优势对方法进一步优化.

参 考 文 献

[1] Vaquero L M, Rodero-Merino L, Caceres J, Lindner M A. A break in the clouds: Towards a cloud definition. ACM SIGCOMM Computer Communication Review, 2009, 39(1): 50-55

[2] Kamara S, Lauter K. Cryptographic cloud storage//Proceedings of the Financial Cryptography Workshops. Tenerife, Canary Islands, Spain, 2010: 136-149

[3] Cao Ning, Wang Cong, Li Ming, Ren Kui, Lou Wenjing. Privacy-preserving multi-keyword ranked search over encrypted cloud data//Proceedings of the 30th IEEE International Conference on Computer Communications (INFOCOM 2011). Washington, USA, 2011: 829-837

[4] Hacıgümüş H, Iyer B, Mehrotra S. Providing database as a service//Proceedings of the 18th International Conference on Data Engineering (ICDE 2002). San Jose, CA, USA, 2002: 29-40

[5] Song D, Wagner D, Perrig A. Practical techniques for searches on encrypted data//Proceedings of the IEEE Symposium on Security and Privacy (S&P 2000). Berkeley, California, USA, 2000: 44-55

[6] Wong W K, Cheung D W, Kao B, Mamoulis N. Secure kNN computation on encrypted databases//Proceedings of the 35th International Conference on Management of Data (SIGMOD 2009). Providence, Rhode Island, USA, 2009: 139-152

[7] Goh E-J. Secure indexes. Cryptology ePrint Archive, 2003. http://eprint.iacr.org/2003/216

[8] Boneh D, Crescenzo G, Ostrovsky R, Persiano G. Public key encryption with keyword search//Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques (EUROCRYPT 2004). Interlaken, Switzerland, 2004: 506-522

[9] Boneh D, Kushilevitz E, Ostrovsky R, Skeith III W E. Public key encryption that allows PIR queries//Proceedings of the 27th Annual International Cryptology Conference (CRYPTO 2007). Santa Barbara, CA, USA, 2007: 50-67

[10] Curtmola R, Garay J A, Kamara S, Ostrovsky R. Searchable symmetric encryption: Improved definitions and efficient constructions//Proceedings of the 13th ACM Conference on Computer and Communications Security (CCS 2006). Alexandria, VA, USA, 2006: 79-88

[11] Wang C, Cao N, Li J, Ren K, Lou W. Secure ranked keyword search over encrypted cloud data//Proceedings of the International Conference on Distributed Computing Systems (ICDCS 2010). Genova, Italy, 2010: 253-262

[12] Li J, Wang Q, Wang C, Cao N, Ren K, Lou W. Fuzzy keyword search over encrypted data in cloud computing//Proceedings of the 29th IEEE International Conference on Computer Communications (INFOCOM 2010 Mini-Conference). San Diego, CA, USA, 2010: 441-445

[13] Al-Riyami S, Paterson K. Certificateless public key crypto-

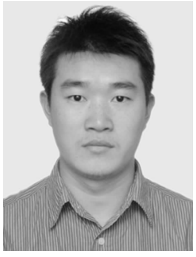
graphy//Proceedings of the 9th International Conference on the Theory and Application of Cryptology and Information Security (ASIACRYPT 2003). Taipei, China, 2003: 452-473

[14] Goyal V. Reducing trust in the PKG in identity based cryptosystems//Proceedings of the 27th Annual International Cryptology Conference (CRYPTO 2007). Santa Barbara, CA, USA, 2007: 430-447

[15] Sabrina De Capitani di Vimercati, Sara Foresti, Sushil Jajodia, Stefano Paraboschi, Pierangela Samarati. Encryption policies for regulating access to outsourced data. ACM Transactions on Database Systems, 2010, 35(2): 12:1-12:46

[16] Bloom B H. Space/time trade-offs in hash coding with allowable errors. Communications of the ACM (CACM), 1970, 3(7): 422-426

[17] Guo Deke, Wu Jie, Chen Honghui, Yuan Ye, Luo Xueshan. The dynamic bloom filters. IEEE Transactions on Knowledge and Data Engineering (TKDE), 2010, 22(1): 120-133



CHENG Fang-Quan, born in 1985, Ph. D. candidate. His current research interests include database, trusted data management.

PENG Zhi-Yong, born in 1963, Ph. D., professor and Ph. D. supervisor. His main research interests include complex data management, Web data management, and trusted data management.

SONG Wei, born in 1978, Ph. D., lecturer. His research interests include distributed system, trusted data management.

WANG Shu-Lin, born in 1987, M. S. candidate. His current research interest is trusted data management

CUI Yi-Hui, born in 1981, Ph. D. candidate. His current research interest is trusted data management

Background

Cloud computing, which has been considered as successful paradigm of X-as-a-service using the pay-as-you-go model, motivates both individuals and enterprises to outsource their local complex data and data management tasks to cloud. Such shift towards cloud computing provides the data owner with significant IT agility, efficiency, and cost savings. As a consequence of outsourcing, the sensitive data stored in the cloud is no longer under the data owner's physical control and is put at risk. To protect data privacy, a straightforward and effective solution is to encrypt the sensitive data locally before outsourcing. However, data encryption makes data utilization a challenging task. Therefore, there is a great need to explore efficient privacy-preserving query service over encrypted cloud data.

Querying over encrypted cloud data brings up many requirements: the multiple data owners; the query privacy preserving; the high time/space efficiency. However, none of existing works can adequately meet all the above-mentioned requirements including data usability, privacy preservation and performance efficiency. To fill this gap, we propose an efficient privacy-preserving rank query over encrypted cloud data (RQED). We establish the RQED framework through the improved searchable public key encryption (PKES). Based on the RQED framework, we design a more sound and privacy-preserving rank function, and propose a hierarchical index based on dynamic Bloom filters. The theoretical analysis and experimental evaluation show that the proposed solutions indeed achieve powerful privacy guarantee, efficient query performance, low communication overhead and effective query authorization control.

This work is supported by the National Natural Science Foundation of China (Grant Nos. 61070011, 61100019, 61202034), the Innovation Foundation of Huawei Corporation (Grant No. YJCB201001078), Wuhan Science and Technology Bureauin China (Grant No. 201150530139), the Ph. D. Programs Foundation of Ministry of Education of China (Grant No. 20110141120033), Fund of State Key Laboratory of Software Engineering (Wuhan University) (Grant No. SKLSE2010-08-20), Natural Science Foundation of Hubei Province in China. These projects focus on the research of the trusted data management in an untrusted cloud environment. Under the projects, several papers have been published, mainly on encrypted data querying, key management, privacy protection etc. And we have developed two prototypes, a trusted Web email service system and a trusted cloud storage service system.