

PDR-SFP: 一种快速高效的私有文档检索方法

董方明^{1,2}, 梅迪菲^{4,5,6}, 安宝宇^{4,5,6}, 周 川^{2,3}, 周 强^{4,5,6},
畅东升^{4,5,6}, 王平辉^{2,7}

¹ 西安交通大学自动化科学与工程学院 西安 中国 710049

² 西安交通大学智能网络与网络安全教育部重点实验室 西安 中国 710049

³ 西安交通大学软件学院 西安 中国 710049

⁴ 中移雄安信息通信科技有限公司 西安 中国 710000

⁵ 中移系统集成有限公司 西安 中国 710000

⁶ 中移信息系统集成有限公司 西安 中国 710000

⁷ 西安交通大学网络空间安全学院 西安 中国 710049

摘要 文档检索是信息系统中的一项目基本功能,旨在从数据库中检索出与用户查询最相关的一个或多个文档。在传统的查询模式中,待查询的原始数据与查询索引全部存放在本地,当数据积少成多,在本地进行检索会带来高昂的本地存储以及计算成本。随着云计算和类似技术的出现,越来越多的计算任务被转移到云服务器上。这一转变引发了重大的隐私问题,因为用户查询文档中的敏感信息可能会暴露给服务器,导致严重后果。为了解决这一问题,保护用户查询隐私变得至关重要。此外,在以往的工作中,研究者使用如全同态加密等密码学技术,协议完成一次查询的成本极高。为了解决在文档搜索过程中的隐私问题以及效率问题,本研究提出了一种旨在保护用户隐私的文档检索方法。利用 Regev 同态加密来保护查询文档的隐私,确保查询内容即使在服务器上处理也能保持机密。此外,本方法采用 Sign-Full 随机投影来压缩数据库存储并减少计算开销,从而使检索过程更加高效。与现有方法相比,本方法在速度和存储效率上都有显著改进。具体而言,它的查询时间大约快了 101 倍,存储空间需求减少了约 30 倍,同时保持了相当的查询准确性。这些改进使该方法在隐私和效率至关重要的实际应用中非常适用。采用这种方法可以显著提高云环境中文档检索系统的安全性。

关键词 私有文档检索; 隐私查询; 同态加密; 随机投影; 私有信息检索

中图法分类号 TP311 DOI 号 10.19363/J.cnki.cn10-1380/tn.2026.05.04

PDR-SFP: A Fast and Efficient Protocol for Private Document Retrieval

DONG Fangming^{1,2}, MEI Difei^{4,5,6}, AN Baoyu^{4,5,6}, ZHOU Chuan^{2,3}, ZHOU Qiang^{4,5,6},
CHANG Dongsheng^{4,5,6}, WANG Pinghui^{2,7}

¹ School of Automation Science and Engineering, Xi'an Jiaotong University, Xi'an 710049, China

² Key Laboratory for Intelligent Networks and Network Security Ministry of Education, Xi'an Jiaotong University, Xi'an 710049, China

³ School of Software Engineering, Xi'an Jiaotong University, Xi'an 710049, China

⁴ China Mobile Xiong'an Information and Communication Technology Co., Ltd., Xi'an 710000, China

⁵ China Mobile System Integration Co., Ltd., Xi'an 710000, China

⁶ China Mobile Information System Integration Co., Ltd., Xi'an 710000, China

⁷ School of Cyber Security, Xi'an Jiaotong University, Xi'an 710049, China

Abstract Document retrieval is a fundamental function in the field of information systems, designed to retrieve one or more documents most relevant to a user's query from a separate database. In traditional query models, both the raw data to be queried and the query index are stored locally. As the data grows over time, performing retrieval locally incurs high costs in terms of storage and computational resources. With the advent of cloud computing and similar technologies, an increasing number of computational tasks have been offloaded to cloud servers. This shift has raised significant privacy concerns, as sensitive information in the documents queried by users could potentially be exposed to servers, leading to severe consequences. Consequently, protecting the privacy of user queries has become a critical issue.

Moreover, in previous studies, researchers have employed cryptographic techniques, such as fully homomorphic encryption, to address privacy concerns; however, the cost of executing a single query with such protocols is prohibitively

通讯作者: 王平辉, 博士, 教授, Email: phwang@mail.xjtu.edu.cn。

本课题得到西安交通大学-中国移动数字政府联合研究院 2023 年“揭榜挂帅”数字政府领域课题(No. XJTU-CMCC-YF202301010)资助。

收稿日期: 2024-11-13; 修改日期: 2025-02-08; 定稿日期: 2026-04-07

high. To resolve the privacy and efficiency issues in the document retrieval process, this study proposes a method designed to protect user privacy during document retrieval. The method utilizes Regev homomorphic encryption to safeguard the privacy of queried documents, ensuring that the query content remains confidential even when processed on the server. Additionally, the method employs Sign-Full random projection to compress database storage and reduce computational overhead, thereby making the retrieval process more efficient. Compared to existing methods, this approach demonstrates significant improvements in both speed and storage efficiency. Specifically, the query time is approximately 101 times faster, and the storage space requirement is reduced by about 30 times, while maintaining comparable query accuracy. These advancements make the method highly applicable to practical scenarios where privacy and efficiency are crucial. Adopting this approach can substantially enhance the security of document retrieval systems in cloud environments.

Key words private document retrieval; private query; homomorphic encryption; random projection; private information retrieval

1 引言

在信息时代,数据的价值日益凸显,如何有效地管理和利用海量的数据资源成为各行各业面临的重要课题。作为信息检索领域的重要分支,文档检索在各类信息系统中举足轻重。然而,传统文档检索方法却常常忽视了用户隐私保护这一关键诉求,由此带来了隐私泄露的隐患。鉴于此,研究并设计一种兼顾检索功能与隐私安全的文档检索方法,不仅具有深远的理论意义,更有广阔的实际应用前景。

从理论研究角度来看,保护查询隐私的文档检索方法研究是隐私计算领域和信息检索领域交叉融合的新方向。传统的隐私保护技术如数据加密、匿名化等并不能完全适用于文档检索场景,需要针对文档数据的特点和检索需求进行专门的研究。同时,文档检索过程中涉及的用户查询、相关性反馈等环节也对隐私保护提出了新的挑战。深入研究隐私保护机制与文档检索应用的融合,有助于丰富和发展相关理论,推动隐私计算、安全多方计算等领域的进步。

该研究具有广泛的应用背景和重要的现实意义。在当今大数据时代,云计算和云存储服务得到了飞速发展和广泛应用。例如像 Dropbox 这样的基于云的数据存储服务,以其便捷、经济、易扩展等特点,成为个人和企业进行文档存储和管理的首选方案。用户可以通过这些服务将海量文档数据外包存储到远程云服务器,节省宝贵的本地存储资源。然而,随着文档数据外包存储的普及,隐私泄露问题日益突出,已然成为阻碍用户使用云存储服务的主要障碍之一^[1-2]。

除了云存储场景,隐私保护的文档检索在信息检索领域也有广泛的应用需求^[3]。如今,通过搜索引擎进行网络信息检索已经成为人们获取信息和知识的主要途径。例如,某人希望去往 A 城市旅游,他希望通过网络中获取有关于 A 城市的相关信息,于是他

在搜索引擎中输入“A”以检索相关的内容;再比如,某人参加体检,被告知可能患有某种疾病 B,为了更好地预防以及治疗,他在搜索引擎中输入“B 预防”来获取有关该疾病的预防措施等内容。在此场景下,由于数据库为公开的,无须考虑数据库的隐私。然而,用户在使用搜索引擎时,往往需要向搜索引擎的服务器输入大量的查询关键字,这些查询词可能包含用户的隐私信息,如位置、身体状况、关注点等敏感数据^[4]。搜索引擎服务商可能在数据泄露时无意间披露这些信息,或者出于盈利目的有意收集、分析这些隐私数据,用于用户画像、定向广告投放等商业目的。即使用户通过 Tor 等匿名工具隐藏 IP 地址,查询本身所包含的个人信息,以及查询之间的相似性分析,也可能导致用户身份被重新识别出来。

因此,在搜索引擎的文档检索过程中,亟须引入隐私保护机制,在保证检索效果的同时,最大限度地减少隐私数据的泄露。然而,提供强大查询隐私保护的加密方案通常要求服务器对每个查询响应扫描整个数据集^[5],否则,服务器将了解到数据集的哪些部分与查询不相关。对于互联网规模的搜索而言,这是一个巨大的挑战,因为在每次查询中扫描每个抓取的网页将变得非常昂贵。

除了以上场景,隐私保护文档检索在教育、医疗、司法等领域^[6-7]也有广泛应用前景。这些领域涉及大量敏感数据和文档,对隐私保护和可用性均提出了极高要求。

本文研究的问题即为在公开的文档集中,查找与给定查询最相关的文档,同时保护查询内容不被文档集持有方知晓(正式问题定义见 4.1 节)的问题。我们称之为私有文档检索问题(Private Document Retrieval Problem)。现有的实现私有文档检索的方案主要使用的同态加密^[5-8]来处理查询信息,实现对查询内容隐私的保护。然而,由于忽视了对服务器持有的数据集的处理,这些方案或有着极高的通信开销;或有着不可接受的查询响应时间。为了解决以上问

题, 本文将研究如何在保护查询隐私的条件下实现快速高效的文档检索。综上, 本文贡献点如下:

(1) 本文针对数据库存储开销大的问题提出了基于随机投影的数据库编码策略。该策略可以在维护数据库本身可查询性质的前提下, 将数据库进行压缩, 便于节省云上存储资源。

(2) 本文设计了新颖的私有文档检索方法, 在保护查询隐私的前提条件下, 快速高效地从编码数据库中检索得到与用户查询内容最相关的文档。与已有方法相比, 该方法大大降低了用户进行在线检索时的响应时间。

(3) 本文开展了具有说服力的实验对提出的私有文档检索方法的有效性进行验证。在常用的文档检索数据集上本文提出的方法在检索时间上相较于主流方法有了至少 101 倍的提升。

本文第 2 节介绍文档查询相关方法和研究现状。第 3 节介绍本文所需的基础知识。第 4 节首先定义了本文所研究的私有文档检索问题本身, 接着介绍了本文所提出的快速高效的私有文档检索方法思路。第 5 节通过对比实验验证了所提方法的有效性。最后总结全文。

2 相关工作

本节主要分析隐私查询的研究现状。按照应用场景的不同, 保护隐私的文档查询的研究可以分为以下两类:

1) 在加密的私有文档上进行隐私查询

此类工作对应的场景为数据为用户个人所有, 并外包到服务器上进行存储。外包至服务器存储的数据需经过加密等手段处理, 以保护用户的个人数据。在此基础上, 查询协议在保护用户查询隐私的前提下在文档集中进行检索。可搜索加密(SE)技术是实现此类安全搜索的关键加密机制之一, 它允许用户在不泄露查询关键词的前提下对密文文档进行搜索, 保护了文档和搜索者的隐私。近年来, 围绕 SE 技术, 研究者提出了多种结合复杂数据结构的对称可搜索加密(SSE)方法, 旨在提高计算及通信效率^[9-10]。此外, 为了支持更加多样化的功能, 如动态关键词更新、联合查询、析取查询、布尔查询、排名查询以及范围查询, 已经有诸多 SE 技术的变体被相继提出^[11-12]。

然而, 尽管这些安全搜索方法在一些场景下颇为实用, 但它们对于恶意攻击的抵御能力有限, 特别是那些能够利用搜索模式和访问模式进行泄露的攻击。典型的攻击方法之一是文件注入攻击^[13], 即攻

击者通过将特定设计的文件注入客户的外包文档库中, 通过分析客户的访问模式来揭露客户的查询内容。其他形式的攻击则尝试利用搜索模式的泄露, 研究表明, 即便攻击者仅凭文档库的统计信息, 也能以较高的成功率推测出用户的查询关键词^[14-15]。

为应对这些威胁, 通常会将差分隐私(DP)、不经意随机访问机制(ORAM)^[16]和私有信息检索(PIR)^[17]等技术融合进安全搜索方法中, 每种技术都各有优劣。本文主要关注 PIR 技术, 尤其是那些基于多服务器 PIR(MPIR)的研究工作^[18-19], 因为它们与本文的研究目标紧密相关。MPIR 技术最早由 Chor 等人提出^[20], 其核心思想是通过秘密共享机制, 在多个云服务器间分布式存储复制的文档库, 而客户的搜索查询则被转化为基于文档索引的二进制共享形式, 以此来掩盖文档获取的具体情况。MPIR 的安全基础是建立在去中心化信任之上的, 即假设至少有一个服务器不会泄露信息给对手^[21-22]。

通过结合对称加密技术, MPIR 能够以本地化的方式推广至云存储的安全关键词搜索, 其中云服务器外包的文档索引采用的是加密后的基于位图的关键词索引形式^[23]。虽然基于 MPIR 的关键词搜索方案在客户端和服务端都显示出了良好的计算效率, 但其高昂的通信成本成为限制其成功应用的一大障碍。值得庆幸的是, Gilboa 等人^[24]提出了一项名为分布式点函数(DPF)的新技术, 并由 Boyle 等人^[25]进一步加以完善。与传统 MPIR 相比, 基于 DPF 的 PIR 方案不仅保持了高计算效率, 而且大幅降低了通信成本, 使其成为安全关键词搜索方法的一个实用且有效的构建块。

Gilboa 等人^[24]和 Boyle 等人^[25]的工作中简要讨论了如何利用基于位图的关键词索引来实现此类构建。在此基础上, Dauterman 等人引入布隆过滤器数据结构, 进一步提升了基于 DPF 的安全关键词搜索方案的性能, 他们的方法通过将外包的关键词索引压缩为等长的布隆过滤器编码形式, 显著降低了计算成本、通信开销以及存储成本^[18]。Huang 等人^[26]使用混淆布隆过滤器和布谷鸟过滤器进行编码, 进一步提升了查询的性能, 代价是额外的构建索引的时空开销。

2) 在公开文档上进行隐私查询

与上文所述的场景不同的是, 此类工作面向的场景多为数据非个人所有, 用户希望从庞大的知识库中检索最相关的文档, 同时保护自己的查询兴趣不被泄露。典型的场景为用户在搜索引擎中查询感兴趣的内容。目前, 有三种流行的非加密方法被用于

网络搜索的隐私保护。第一种方法是通过匿名代理, 如 Tor^[26]或混合网络, 将搜索查询发送到传统的搜索引擎。第二种方法使用虚假查询或混淆查询。然而, 这两种技术仍然向搜索引擎泄露了某种版本的客户查询, 允许搜索引擎链接查询并对用户进行去匿名化。第三种是使用可信硬件^[27], 但其提供的安全保证仅与底层硬件一样强^[28-29]。由于可信执行环境会泄露内存访问模式, 基于可信硬件的方法需要使用遗忘 RAM 来隐藏内存访问模式, 这会带来额外的开销^[30]。

Coeus 是第一个支持在大规模公共文档和不可信基础设施上进行隐私保护排序检索的端到端系统。它采用 TF-IDF 编码查询, 能够有效处理精确字符串匹配, 尽管如此, 它并不支持非文本搜索, 且查询的计算量和通信量极高, 不适用于网络规模的数据集。最近的一些工作展示了如何构建单服务器亚线性时间信息检索协议。然而, 这些方法在实践中仍然具有较高的单次查询开销^[31-32], 或需要向客户端流式传输整个文档库^[33-34], 对于定期变化的大型文档库来说, 这是不切实际的。

在多项先前研究中, 将文本(或图像)搜索问题转化为私有最近邻搜索问题是一个常见的做法, 以此来利用语义嵌入技术。虽然这些研究构建了私有最近邻搜索的协议, 但通常依赖于多个非串谋服务提供商^[35-36], 并使用了计算开销较大的加密工具, 这限制了它们在实际应用中能处理的语料库规模, 通常仅限于几千个条目^[37]。此外, 这些方案可能会向服务器泄露关于客户查询的一些信息。最近的 Tiptoe 方法只需要一个服务提供商, 且支持网络规模的文档集合查询, 尽管如此, 其每次查询的耗费的成本依然较高。

3 预备知识

本章旨在详细描述本文研究内容以及技术路线中所涉及的关键理论以及必要概念。主要包括同态加密、嵌入表示学习、k 均值聚类以及 Sign-Full 随机投影^[38]。

3.1 Regev 同态加密

同态加密(Homomorphic Encryption, HE)是一类允许在加密数据上执行特定计算的加密技术, 其特点在于, 这些计算的结果, 在解密之后, 将与在未加密数据上执行同一计算所得到的结果相同。如果加密算法满足公式(1), 那么 $E(\cdot)$ 称其满足加法同态性。

$$E(m_1) + E(m_2) = E(m_1 + m_2), \forall m_1, m_2 \in M, \quad (1)$$

式中: $E(\cdot)$ 表示加密算法, M 为明文空间。

在基于容错学习(Learning with Errors, LWE)的同态加密方案中, LWE 参数由安全参数 $\lambda \geq 1$, 样本数量 $m \geq 1$, 模数 $q \geq 2$, 以及定义在 $\mathbb{Z}_q$ 上的一个误差分布 χ 组成。LWE 问题定义为, 对于矩阵 $A \in \mathbb{Z}_q^{m \times \lambda}$ 、秘密向量 $s \in \mathbb{Z}_q^\lambda$ 、根据分布 χ 选取的误差向量 $e \in \mathbb{Z}_q^m$, 以及随机向量 $r \in \mathbb{Z}_q^m$, 以下分布在计算上是无法区分的:

$$\{(A, As + e)\} \stackrel{c}{\approx} \{(A, r)\} \quad (2)$$

Regev^[39]提出了基于 LWE 的加密方案。给定 LWE 参数 (λ, q, χ) 和一个明文模数 p , 随机均匀选取向量 $a \in \mathbb{Z}_q^\lambda$, 生成 Regev 密钥 $s \in \mathbb{Z}_q^\lambda$ 。对于消息 $\mu \in \mathbb{Z}_p$ 的 Regev 加密是

$$(a, c) = \left(a, a^T s + e + \frac{q}{p} \cdot \mu \right) \quad (3)$$

公式(3)中, 当误差分布 χ 中采样的误差绝对值小于 $\frac{1}{2}q/p$ 时, 加密过程就能保证解密正确性。具体来说, 拥有密钥 s 的实体可以通过计算 $c - a^T s \bmod q$ 并将结果四舍五入到最接近的 q/p 的倍数, 从而解密出 μ 的值。

此外, Regev 加密支持加法同态性, 即给定两个密文 (a_1, c_1) 和 (a_2, c_2) , 他们在加密下的和 $(a_1 + a_2, c_1 + c_2)$ 经过解密可以得出明文的和。

3.2 嵌入表示学习

嵌入表示学习(Embedding Representation Learning)是自然语言处理和机器学习领域中一种重要的技术, 其核心思想是将离散的语言单元如词语、句子等映射到连续的低维向量空间中, 从而更好地表示它们之间的语义关系。在这个过程中, 语言单元被转化为实值向量, 其中每个维度都捕捉了一定的语义信息。通过嵌入表示, 将人类语言转化为机器学习算法可以处理的数值形式, 使得计算机能够更好地理解和处理自然语言。

随着深度学习技术的发展, 基于预训练语言模型的嵌入表示学习方法逐渐成为主流。与传统的词嵌入模型不同, 预训练语言模型如 BERT、RoBERTa 等使用多层的 Transformer 网络在大规模无监督语料上进行自监督学习, 通过掩码语言建模等任务来学习词语和句子的上下文表示。这种预训练范式可以充分利用无监督数据, 从海量语料中学习到丰富的

语言知识, 再通过迁移学习的方式应用到下游的自然语言处理任务中, 大大提高了模型的性能。

值得一提的是, 嵌入表示学习的应用不仅局限于文本数据, 还可以扩展到图像、音频等其他模态的信息。例如, 卷积神经网络可以将图像映射为固定长度的向量表示, 捕捉图像的视觉特征; 而 wav2vec 等模型^[40]则可以将音频信号转化为嵌入向量, 学习音频的语音特征。通过多模态嵌入表示学习, 可以实现跨模态的信息检索和理解, 如以图搜图、以文搜图等任务。此外, 多模态嵌入还为多模态机器学习提供了便利, 能够将不同模态的数据映射到同一个语义空间中, 实现模态之间的信息融合和交互。

3.3 k 均值聚类算法

k 均值聚类(k-means clustering)是一种经典的无监督机器学习算法, 用于将数据集划分为 k 个簇。该算法的目标是最小化簇内部数据点与簇质心之间的平方距离之和, 以实现簇的紧凑性和独立性。

k 均值聚类算法的基本步骤如下:

- 1) 随机选择 k 个数据点作为初始的簇质心。
- 2) 对于数据集中的每个数据点, 计算其与 k 个簇质心的距离, 并将其分配到距离最近的簇中。
- 3) 对于每个簇, 计算其内部所有数据点的均值, 并将均值作为新的簇质心。
- 4) 重复步骤 2) 和步骤 3), 直到簇质心不再发生变化或达到预设的迭代次数。

数学上, k 均值聚类算法可以表示为公式(4)所示的优化问题, 该优化问题的目标是最小化所有数据点与其所属簇质心之间的平方距离之和。

$$\min_{\{C_1, \dots, C_k\}} \sum_{i=1}^k \sum_{x \in C_i} \|x - \mu_i\|^2 \quad (4)$$

式中: $\{C_1, \dots, C_k\}$ 表示 k 个簇; μ_i 表示第 i 个簇质心; x 表示簇 C_i 中的数据。

k 均值聚类算法简单、易于理解和实现。对于大数据集, 该算法能够保证较快的收敛速度。在各个领域都有广泛的应用, 如客户细分、图像分割、文本聚类等。在自然语言处理领域, k 均值聚类可以用于词语聚类、文档聚类、主题发现等任务。通过将词语或文档映射到嵌入空间, 并对嵌入向量进行聚类, 可以发现语义相似的词语或主题相关的文档, 为进一步的分析和应用提供基础。

3.4 Sign-Full 随机投影

Sign-Full 随机投影技术作为处理大数据集并进行高效搜索中的一种方法, 是在 Sign-Sign 随机投影技术基础上的进一步发展。具体来说, 通过对两个 d

维的实数向量 $u, v \in \mathbb{R}^d$ 进行如公式(5)所示的随机投影, 生成的投影实数 x 和 y 的乘积的期望值 $E(xy)$ 即为向量 u, v 的内积, 如公式(6)所示。

$$x = \sum_{i=1}^d u_i r_i \quad (5)$$

$$y = \sum_{i=1}^d v_i r_i$$

$$E(xy) = u \cdot v = \sum_{i=1}^d u_i v_i \quad (6)$$

式中: r_i 表示服从独立同分布的标准正态分布。

因此在实际应用中, 通过独立生成 k 次 x 和 y , 内积可以通过公式(7)近似计算。并且随着投影次数 k 的增加, 计算结果也越逼近真实值。

$$u \cdot v \approx \frac{1}{k} \sum_{j=1}^k x_j y_j \quad (7)$$

而 Sign-Sign 随机投影^[41]则是上述的一个变种, 它基于以下碰撞概率结果:

$$P(\text{sgn}(x) = \text{sgn}(y)) = 1 - \frac{\arccos^{-1} \rho}{\pi} \quad (8)$$

$$\rho = \frac{\sum_{i=1}^d u_i v_i}{\sqrt{\sum_{i=1}^d u_i^2} \sqrt{\sum_{i=1}^d v_i^2}} \quad (9)$$

式中: $\text{sgn}(\cdot)$ 是符号函数, ρ 是向量 u, v 的余弦相似度。

值得注意的是, 若仅使用投影数据的符号, 就会丢失原始向量 u, v 的范数信息。因此, 可以在不失一般性的前提下假设原始数据向量已经标准化, 如公式(10)所示。这意味着, 投影后的数据 x 和 y 可以被认为符合标准正态分布, 即 $x \sim N(0, 1), y \sim N(0, 1)$ 。

$$\sum_{i=1}^d u_i^2 = \sum_{i=1}^d v_i^2 = 1 \quad (10)$$

综上所述, Sign-Sign 随机投影技术可以通过将符号函数应用到投影数据上, 即 $\text{sgn}(x)$ 和 $\text{sgn}(y)$, 并通过简单的符号来估计向量 u, v 间的余弦相似度 ρ , 该方法在搜索相关问题中已被广泛应用^[42]。

在此基础上, Li^[38]对 Sign-Sign 随机投影进行了扩展, 提出了 Sign-Full 随机投影估计器。具体而言, 他通过证明公式(11)成立, 从而提供了一种更准确的估算余弦相似度的方法, 如公式(12)。

$$E(\text{sgn}(x)y) = \sqrt{\frac{2}{\pi}} \rho \quad (11)$$

$$\hat{\rho} = \sqrt{\frac{\pi}{2}} \left(\frac{\sum_{j=1}^k \text{sgn}(x_j) y_j}{\sqrt{k} \sqrt{\sum_{j=1}^k y_j^2}} \right) \quad (12)$$

式中: $\hat{\rho}$ 表示估算的向量 u, v 的余弦相似度。

Sign-Full 随机投影通过保留 y 的值, 显著增强

了相似度估计的精确性。这一方法允许将两个需要计算余弦相似度的高维实数向量通过随机投影降低维度, 并将其中任意一个转换为符号向量, 最终根据公式(12)估算余弦相似度。具体而言, 在选择投影次数 k 为 200 时, 该估算的余弦相似度误差不到 10^{-3} 。

4 基于 SF 投影的私有文档检索方法

4.1 问题定义

概括地讲, 私有文档检索问题(Private Document Retrieval)是在保护客户端查询隐私的条件下, 从远程数据库中检索得到与客户端提交的查询最相关的文档的问题。该问题中定义了两个参与方, 客户端 $\mathcal{C}$ 与服务器 $\mathcal{S}$: 客户端持有查询文档 q , 而服务器持有一组公开文档数据库 DB 。数据库 $\text{DB} = \{(\text{pid}_1, \text{doc}_1), \dots, (\text{pid}_N, \text{doc}_N)\}$, 即数据库中的每个成员都由文档位置索引 pid_i , 以及文档具体内容 doc_i 组成, 其中 $i = 1, \dots, N$ 。客户端希望在不向服务器泄露自身查询文档 q 的隐私的前提下, 得到与 q 最相关的 K 个文档中的一个或多个。即在完整地执行了一次查询后, 服务器 $\mathcal{S}$ 既不知道查询文档 q 的内容, 也不知道文档数据库中哪些文档与 q 的关联度比较高; 同时客户端能够正确地得到其想要的查询结果。余弦相似度(cosine similarity)是一种常用的比较向量相似性的度量方式, 其定义如式(9)所示。本文拟采用余弦相似度来衡量文档间的相关性, 即两文档间相关性越强, 其向量表示的余弦相似度越趋近于 1; 反之, 则越趋近于 0。

解决该问题的一个自然方法是采用一个两阶段的协议。在第一阶段, 客户端私密地将查询 q 与文档库 DB 中的每个文档进行相似度计算并返回得分。通过对返回相似度的比较, 客户端可以得到与查询文档 q 最相关的文档位置索引 pid_q ; 然后, 在第二阶段, 不经意地检索 DB 中位置索引为 pid_q 的对应文档。如果服务器无法获知任何关于查询的信息, 则必须对整个文档库进行处理, 因为如果服务器能够排除掉某些不在查询范围内的文档, 那么它就能够利用一些攻击手段推断出关于查询的信息。

在私有文档检索问题中, 性能瓶颈与研究重点主要集中在第一阶段, 即如何在保证安全性的前提下, 计算查询与文档库中每个文档之间的相似度, 并根据相似度对文档进行排序。本章将该阶段的问题定义为私密文档排名问题, 所提出的方法也仅考

虑了如何实现第一阶段的检索。事实上, 如果想要实现第二阶段的检索, 用户或客户端可以根据第一阶段获得的文档得分和排名信息, 选择想要查询的文档索引。此时, 基于这些文档索引进行的不经意检索可以转化为一个标准的私有信息检索(PIR)问题。针对 PIR 问题, 目前已有众多成熟的解决方案^[43]。

4.2 方法概述

本章所提的私有文档检索方法(Private Document Retrieval using Sign-Full Projection, PDR-SFP)采用 Sign-Full 随机投影技术, 将文档库的嵌入表示进一步投影成符号(即正号或负号)表示。通过这种方式, PDR-SFP 在查询耗时、通信量、计算量、存储成本以及排名准确性上相对 Tiptoe 均有提升, 改善了成本昂贵的问题。

在 PDR-SFP 中, 查询对应 Sign-Full 随机投影的向量 y , 文档集中的文档则对应向量 x , 即本章将文档集中的文档表示为符号向量, 查询表示为对应的随机投影向量。并且通过观察公式(2-12)可以发现, 对于一次查询, 除了分子部分的内积外, 其余各部分对于所有文档而言是相同的。由于所需的仅仅是查询与文档之间的余弦相似度排名, 因此可以使用查询的随机投影向量与文档的随机投影符号向量之间的内积(即 $\sum_{j=1}^k \text{sgn}(x_j) y_j$)作为它们的相似度得分^[38]。

Sign-Full 随机投影技术为 PDR-SFP 带来了多个显著优势。首先, 该技术允许使用低维投影向量高精度估计余弦相似度, 实现了对 d 维实数向量的紧凑表示。具体而言, 本章利用 k 维随机投影向量来表示 d 维实数向量, 这样做只需牺牲极少的排名准确性, 便可将存储、通信以及计算量降低到原来的 d/k 倍(d 一般为 768, 而 k 仅为 200)。其次, 对文档集合的符号表示进行二值化编码, 即将负号编码为 0、正号编码为 1, 可以进一步大幅压缩存储空间, 这使得用一个服务器保存拥有亿级文档数的数据库成为可能。最后, 在两个向量计算相似度得分而进行内积运算时, 可以简化为加法和减法的组合, 无须复杂的乘法运算。

综上所述, 本章采用 Sign-Full 随机投影方式表示文档, 显著降低了存储、通信以及计算的成本。通过对文档的符号表示进行进一步压缩, 成功将亿级别的文档库容纳于单一服务器内, 并实现了在 GPU 上进行压缩的符号矩阵与加密向量的矩阵乘法, 以计算查询和文档的相似度, 从而将查询的时间降低了近百倍。具体来说, 对于一个包含 3 亿文档的数据集, 目前最先进的方法 Tiptoe^[8]将文档排名服务部署

在 45 个内存优化型的 r5.xlarge AWS 服务器上, 一次查询耗时为 145 个核心秒。相比之下, 本章所提方法仅需部署在一台配备单个 V100 GPU 的服务器上, 单次查询的时间不超过 2s。

PDR-SFP 方法的框架图如图 1 所示, 其主要包含查询优化与数据预处理模块、文档更新与矩阵维护模块以及安全相似度查询模块三个部分。

其中黄色箭头代表了查询优化与数据预处理模

块的执行流程。首先, 该模块接受文档集合作为输入并通过嵌入表示学习生成嵌入向量, 通过对文档的嵌入表示进行聚类生成簇质心的嵌入表示, 该簇质心的嵌入表示被用于查询过程中, 以便快速定位到最近的簇索引, 从而降低查询的通信开销。此外, 该模块还构造了两个矩阵——矩阵 M 为一个紧凑的文档集合符号矩阵, 而矩阵 H 则作用于加速查询的解密过程。

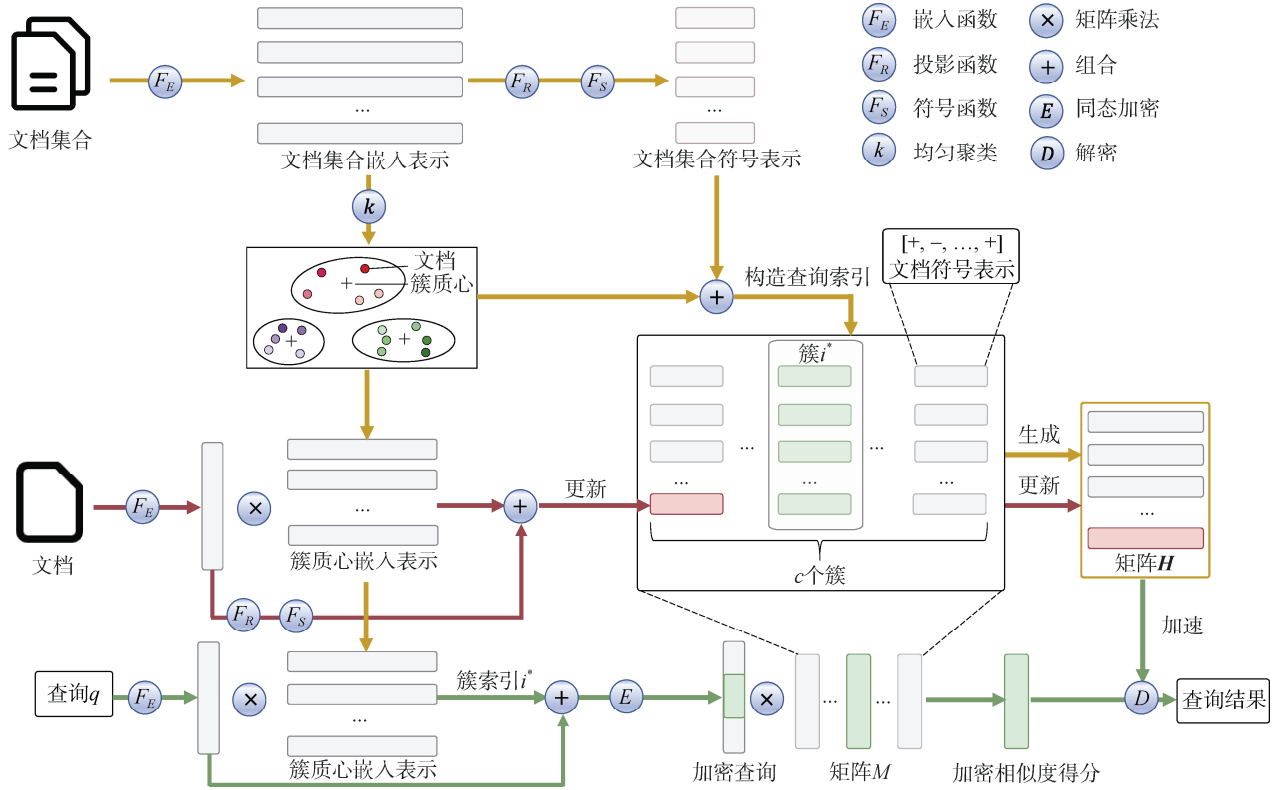


图 1 PDR-SFP 方法框架示意图

Figure 1 Overview of our method PDR-SFP

红色箭头代表文档更新与矩阵维护模块的流程, 当有新文档更新时, 通过该模块对矩阵 M 和矩阵 H 进行相应的更新。绿色箭头表示加密相似度查询模块的流程, 其通过簇质心的符号表示找到查询感兴趣的簇索引 i^* 。接着, 它将簇索引 i^* 与查询结合并进行加密以构造出具有同态加密性的查询向量, 通过在加密状态下进行与矩阵 M 的乘法, 得到查询与簇 i^* 中所有文档的相似度得分。最终用矩阵 H 来加速解密过程, 得到查询结果。

4.3 降低通信开销

如果每次提交查询文档 q , 返回文档库 DB 中所有文档与该查询的相似度, 假设文档库中总共有 N 个文档, 那么查询过程中的通信量为 $O(N)$, 这在实际应用中是很难接受的。为了降低客户端与服务器

之间的通信开销, 本文受 Tiptoe 的启发, 采用分组和组内不经意查询的策略。具体地, 本文将文档库 DB 的嵌入向量表示分为 c 个簇, 并将 c 个簇的质心嵌入表示在查询前发送给客户端存储, 此部分通信开销为 $O(c)$ 。在查询时, 客户端首先确定其查询文档属于哪个簇, 而后构造加密查询向量, 在不泄露查询隐私的情况下查询计算对应簇内文档的相似度, 客户端上传的通信量为 $O(dc)$, 下载的通信量为 $O\left(\frac{N}{c}\right)$, 其中 d 为嵌入向量的维度。在聚类生成 c 个

簇时, 本文将 c 设定为 $\sqrt{N}$, 即共生成 $\sqrt{N}$ 个簇, 每个簇内包含约 $\sqrt{N}$ 个文档。通过这种方法, 本文可以将通信量从 $O(N)$ 降低到 $O(\sqrt{N})$, 这大大提高了方法的可扩展性。

本小节使用一个 k -means 算法的变体来实现这个聚类。该算法输入为 N 个文档的嵌入表示, 首先在文档集的一个子集上执行 k -means 以计算出簇质心, 随后将每份文档分配给其最近的质心对应的簇。为了确保簇的相对平衡, 该算法递归地将较大的簇通过 k -means 分割成数个较小的簇, 对于无法进一步使用 k -means 进行分割的类, 则将簇中的元素平均分为多个子簇。进一步地, 为了增强排名准确性, 本小节遵循以往的研究, 如果一个文档位于簇的边缘附近, 它将被分配到多个簇中。最终, 过小的簇会被合并以减少上传的通信量。该算法输出三个结果: 簇的数量 c 、聚类分配结果 Clusters 以及 c 个簇质心的嵌入表示 Centroids。

在聚类以后, 为了只返回该查询感兴趣簇的文档相似度, 并且不泄露该簇的索引从而保证查询隐私, 本小节使用一个 c 列的矩阵 M 来表示文档集合的符号化表示, 其中每列代表一个不同的簇。矩阵中的每个元素包含 k 个符号, 代表一个文档的符号表示。对应地, 查询被表示为一个 c 维向量, 其中每个元素包含 k 个密文。在这个向量中, 只有查询所关注的簇对应的索引位置包含了查询的随机投影密文(如图中的橙色部分), 其余位置则为零向量的密文。通过这种构造方式, 由于查询向量中除了感兴趣簇对应位置包含有效密文外, 其余位置都为零向量的密文, 因此在矩阵乘法运算过程中, 这些零向量密文不会对相似度得分产生任何贡献。换句话说, 该矩阵乘法的结果只包含查询感兴趣簇的文档相似度得分, 而不会包含其他簇的任何信息, 因此通信量也从原来的 N 个密文降低为 $\sqrt{N}$ 个密文。

图 2 展示的是查询第 2 个簇时(图 2 蓝色部分), 一个包含 3 个簇的矩阵 M 与加密查询进行矩阵乘法, 计算得到第 2 个簇所有文档加密相似度得分(图 2 绿色部分)的过程。

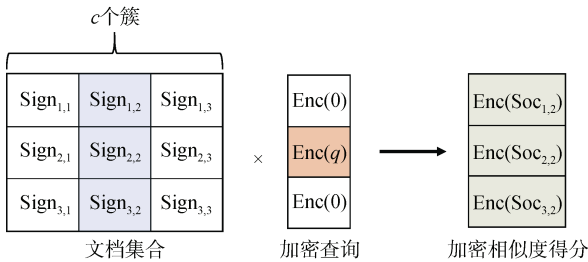


图 2 加密搜索示意图
Figure 2 Diagram of encrypted search

4.4 基于 Sign-Full 随机投影的文档检索

尽管 3.3 节的策略已经显著地降低了客户端与

服务器之间的通信开销, 并且充分利用了 Regev 加密的特点来降低用户提交查询后的在线等待时间。但在处理超大规模文档库时, 在线等待时间仍然存在一定优化的空间。注意到, 经过聚类后尽管通信开销由原来的 $O(N)$ 变为了 $O(\sqrt{N})$, 但计算量仍保持 $O(d*N)$, 也就是说, 计算开销与文档库矩阵的大小线性相关。因此, 为了降低计算开销, 缩短查询时的在线等待时间, 必须对矩阵进行压缩。实际上, 对超大规模文档库进行压缩存储也是应用中的实际需求。例如, Tiptoe 中使用的 Common Crawl 网页数据集中总共包含约 3.64 亿条文档记录, 若不采取任何压缩手段, 仅存储文档库的嵌入向量表示就需要花费超过 850 GB 的存储空间, 带来大量的存储开销。

综上, 本方案需要提出策略在保证文档库矩阵查询功能的前提下, 尽可能地压缩文档库。为此, 本文利用 Sign-Full 随机投影对文档的表示向量进一步压缩。具体地, 对于表示向量 $v_i = [v_1, v_2, \dots, v_d]$, 从标准正态分布 $N(0,1)$ 中均匀随机采样出 d 个样本

$x_1, x_2, \dots, x_d$ 并计算 $\sum_{l=1}^d x_l v_l$, 重复此采样-计算过程 k 次并记录第 j 次的计算结果为 $f_{ij}, j \in 1, \dots, k$, 而后将文档库中的第 i 个文档表示向量替换为如下的符号向量: $[\text{sgn}(f_{i1}), \text{sgn}(f_{i2}), \dots, \text{sgn}(f_{ik})]$ 。替换后的文档库中每个文档对应着 k 比特的存储空间, 完整的文档库总共占用 $N*k$ 比特。

用户确定查询文档 q 后, 原始的查询文档需要经过以下三步的处理再提交给服务器, 分别为向量化表示、随机投影以及同态加密。向量化表示是将原始的文档转化为 d 维的浮点数向量, 并根据该向量表示确定该查询所属的簇(4.3 节); 随机投影过程是将向量化表示得到的 d 维浮点数向量投影成为 k 维的符号向量 $q_s = [q_{s1}, \dots, q_{sk}] \in \mathbb{Z}_p^k$ 。同态加密过程即计算使用 Regev 加密下 q_s 的密文, 即

$$\text{Enc}(q_s) = (A, c) = (A, As + e + \Delta q_s) \in \mathbb{Z}_q^{k \times \lambda} \times \mathbb{Z}_q^k \quad (13)$$

其中, A 为 $\mathbb{Z}_q$ 为 $\left\{-\frac{q}{2}, \dots, -1, 0, 1, \dots, \frac{q}{2}\right\}$ 内的 $k \times \lambda$ 维随机矩阵, s 为 $\mathbb{Z}_q$ 内 λ 维随机向量, e 为均匀采样自离散高斯分布的 k 维随机向量, $\Delta = \left\lfloor \frac{q}{p} \right\rfloor$ 。与

Tiptoe 中的处理方法一样, 用户进行在线查询时, 在线阶段将密文 c 发送至服务器端进行计算。结合离线阶段对密文 A 的计算结果与在线阶段对密文 c 的计

算结果, 用户最终可以解密得到明文文档库符号矩阵与密文查询投影向量的乘积, 即 $\sum_{j=1}^k \text{sgn}(f_{ij}) q_{sj}$ 。由于本文的目标只是获得查询文档与文档库中的成员的相似度排名, 而非得到精确的相似度, 通过式(12), 可以得到若 $\sum_{j=1}^k \text{sgn}(f_{hj}) q_{sj} > \sum_{j=1}^k \text{sgn}(f_{ij}) q_{sj}$, 则 $\hat{\rho}_{hq} > \hat{\rho}_{iq}$ 其中 $\hat{\rho}_{hq}, \hat{\rho}_{iq}$ 分别表示查询文档 q 与文档库中第 h, i 个文档的余弦相似度的估计值。

采用 Sign-Full 随机投影的方法压缩矩阵后, 查询过程的计算复杂度 $O(dN)$ 变为了 $O(kN)$ 。当 $k < d$ 时, 算法的计算复杂度下降为原来的 $\frac{k}{d}$ 。同时服务器的存储开销下降为原来的 $\frac{k}{32d}$ 。在实验部分, 本文评估了改变 k 值对方法表现的影响。

4.5 查询优化

当面对大规模的数据集时, 这种矩阵乘法所需的计算量极其庞大, 这使得 GPU 成为处理此类任务的理想选择。得益于符号向量的有效压缩, 该计算过程不会受到 GPU 显存及主内存大小的限制。因此, 本小节基于 CUDA 实现了压缩矩阵与加密向量的矩阵乘法。具体来说, 本小节使用 64 位整数变量压缩表示 64 位符号, 通过遍历该整数的每一位对加密查询向量进行加减法, 从而避免乘法操作。

4.6 协议分析

4.6.1 安全性

我们首先对查询隐私进行规范化的定义, 如下:

定义 1. 查询隐私. 对于一个在客户端 C 与服务器 S 定义下的协议, 如果其满足对于任意多项式敌手模型 $\mathcal{S}^*$, 客户端输入任意两个查询字符串 q_0, q_1 , 有两个分布如下, 若该分布是计算不可区分的, 有

$$\{\text{View}_{\mathcal{S}^*}[C(q_0)]\} \approx^c \{\text{View}_{\mathcal{S}^*}[C(q_1)]\}$$

则称该协议保护了客户端的查询隐私。

通俗地讲, 上述的定义描述的是, 在任意的服务器敌手视角下, 客户端对于任意两条不同的记录, 其发送的查询内容在多项式时间内是无法区分的。也就是说服务器无法通过客户端发送的查询内容推断出其真正的查询意图, 也就是保护了客户端的查询隐私。

定理 1. SFP-PDR 协议保护了查询隐私。

证明. 在整个查询过程中, 客户端只与服务

器交互一轮, 其发送的信息即为 $C(q_0)$ 。考虑到该同态加密方法是语义安全的(semanticly secure), 即服务器无法从发送的密文中获取到任何有关其明文的信息, 因此本协议实现了对用户查询的隐私保护。

4.6.2 正确性

如同之前保护查询隐私的信息检索协议一样, 协议需要保证查询的正确性。协议的正确性包含两个方面, 1) 同态计算的正确性 2) 检索的正确性。对于前者, 我们通过调整加密参数, 控制单次密文计算中产生的噪声, 可以保证同态计算的结果完全正确。但对于后者, 由于在本文的场景下, 查询结果依赖于服务器端数据的分布, 因此我们无法如保护隐私的精确查找工作一样, 定义查询的正确性。在接下来的章节中, 我们使用实验结果来说明协议的查询正确性。

5 实验评估

5.1 实验数据集

本文使用 MS MARCO^[44](Microsoft Machine Reading Comprehension)数据集对文档排名准确性进行评估, MS MARCO 是一个由微软提供的大规模语言理解数据集。该数据集包含大约 300 万个文档, 其中 MS MARCO 文档排名"dev"子集包含 519300 个真实 Bing 搜索查询及其人工挑选的答案。

对于查询时间, 本文使用和对比方法相同的数据集, 其中 Coeus 使用的是大约 500 万个文档的维基百科文本数据集, 它是一个从维基百科网页中提取而来的大规模结构化文本数据集, 涵盖了广泛的主题, 质量较高, 广泛应用于自然语言处理和信息检索等领域。Tiptoe 使用的是具有 3.64 亿个文档的 C4 数据集, 它是 Common Crawl 的清洗版本, Common Crawl 是一个开放的网页数据集, 包含了互联网上数十亿个网页的信息。C4 数据集通过筛选和清理 Common Crawl 的英文网页语料库, 创建了一个更适合文本学习和模型训练的数据集^[45]。为了对存储成本和通信量指标进行评估, 本节使用根据 C4 数据集进行抽取来对不同文档数量的场景进行模拟。

5.2 评价指标及基准方法

本文将采用 Mean Reciprocal Rank at 100 (MRR@100)作为评价指标, 评估所提方法的排名准确性。MRR 是一种常用于衡量搜索算法和推荐系统性能的指标。对于一次查询, Reciprocal Rank (RR)表示第一个相关结果在搜索结果列表中的倒数排名。MRR 则是对多次查询的 RR 值求平均, 反映了整体

的排名准确性。MRR@ k 是 MRR 的一个变体, 只考虑排名在 top- k 位的结果。本文选取 $k=100$, 即只关注排名在前 100 位的搜索结果, 若相关结果排名超过 100 位, 则 RR 值记为 0。MRR@100 的取值范围为 0~1, 数值越大表示排名准确性越高, 排名靠前的相关结果越多。

本文与针对该问题的最新的两个方法进行了对比, 分别为: Coeus 方法与 Tiptoe 方法。Coeus 是第一个支持在大规模公共文档和不可信基础设施上进行隐私保护排序检索的方法。它采用了词项频率-逆文档频率(Term Frequency-Inverse Document Frequency, TF-IDF)技术对关键词和文档进行编码, 并结合全同态加密以及安全的矩阵-向量乘积算法来计算查询与文档间的相似度。相比之下, Tiptoe 首次引入了语义嵌入技术来表征文档和查询, 将文档排名问题转化为最近邻搜索问题。通过简单的拓展, Tiptoe 还支持文本到图像或者音频等其他数据类型的隐私搜索。此外, Tiptoe 采用了一个双层的线性同态加密算法, 确保了搜索过程的安全性, 相对于 Coeus 有效提高了查询效率并降低了通信成本, 使其能够支持网络规模的文档库查询。

5.3 实验平台与设置

本节使用 C++代码在 Microsoft SEAL 库的基础上实现加密算法, 使用 CUDA 代码实现了在 GPU 下加密相似度计算。具体实验环境如表 1。其中对于单个查询的相似度矩阵运算时间, 本文是在单张 V100 进行的实验。同时, 为了探索在实际应用中, 高并发情况下本章节的所有实验结果都是同时执行 100 次查询, 取平均值的结果, 以模拟若干个用户同时提交对应的查询。

本节选择参数以实现 128 位安全性和约 2^{-40} 的正确性误差。对于 LWE 参数, 设置密文模数, 以在硬件中将密文表示为无符号的 64 位整数。安全参数设置为 2048, 并从具有标准差的离散高斯分布中采样误差向量, 这样可以保证维度小于或等于的加密向量的 128 位安全性。同时选择明文模数, 以满足最大的投影次数和 6 位的带符号明文整数。

表 1 实验环境

Table 1 Experimental testbed

项目	参数
CPU	Intel(R) Xeon(R) CPU E5-2690 v4 @ 2.60 GHz
GPU	NVIDIA Tesla V100 PCIe 32 GB
依赖版本	Python 3.8.18 + CUDA 10.2 + GCC 11.3.0 + Microsoft SEAL 4.1.1
预训练模型	msmarco-distilbert-base-tas-b

5.4 排名准确性评估

由于 PDR-SFP 使用 Sign-Full 随机投影来表示文档和查询, 并使用估计器以估计查询和文档之间的相似性。该估计的准确度与随机投影的次数 k 有很大的关系, 因此本小节将首先探讨投影次数 k (也即投影向量的长度)对排名准确性的影响, 并在最后将所提方法与其他方法进行排名准确性的对比。

图 3 展示了 Sign-Full 投影次数对排名准确性的影响。其中, 蓝色实线表示通过估计器计算相似度得分对文档进行穷举搜索(不进行聚类)计算 MRR@100 的结果。黄色实线则表示在进行聚类后, 仅计算查询感兴趣的簇内文档的估计相似度得分所得到的 MRR@100 结果。两根虚线分别代表聚类和不聚类的情况下, 直接使用 768 维的嵌入向量进行余弦相似度计算的结果。

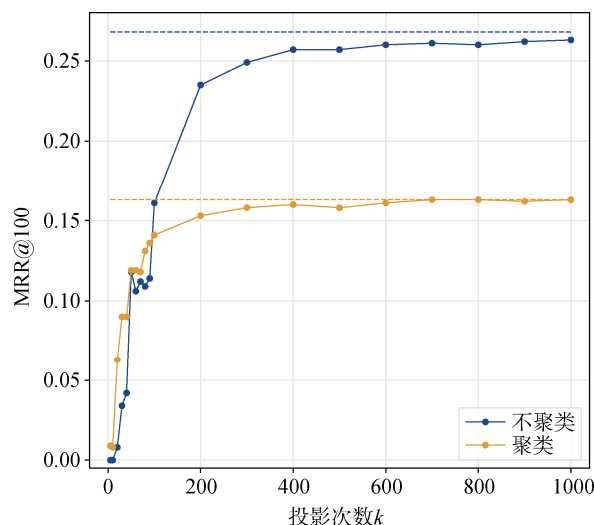


图 3 投影次数 k 对排名准确性的影响

Figure 3 Impact of the number of projections k on ranking accuracy

可以得出结论, PDR-SFP 的排名准确性主要受到聚类的影响, 这种影响是由于部分真实的最佳文档不在查询感兴趣的簇里面。改善排名准确性的一个途径是避免聚类的必要性, 通过聚类可以大幅减少通信成本而在网络规模上运行, 但却占据了排名准确性损失的很大一部分。其次, 当 k 大于 200 时, 采用 Sign-Full 随机投影表示文档和查询对排名准确性的影响就非常小了, 因为当 k 为 200 时, 公式(12)估计余弦相似度的误差^[38]已经不到 10^{-3} 。

接下来我们对三种计算感兴趣簇索引的方式对排名准确性的影响进行了深入的比较和分析, 如图 4 所示。第一种方式是使用高维的嵌入向量计算簇质心索引, 但是使用随机投影在簇内进行最近邻搜索,

也即 PDR-SFP 方法采用的方式, 在图中以橙色实线表示。第二种方式是使用 Sign-Full 随机投影向量计算簇质心索引并进行最近邻搜索, 以绿色实线表示。第三种则为使用主成分分析进行降维之后的嵌入向量计算簇质心并同样使用该嵌入向量进行最近邻搜索, 也即 Tiptoe 的方式, 以灰色虚线表示。蓝色虚线则是直接使用原始的高维嵌入向量计算感兴趣的簇并进行最近邻搜索, 这可以看作聚类后排名准确性的上限。

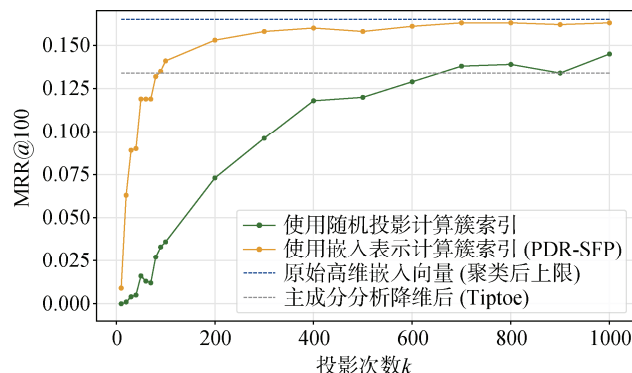


图 4 不同方式计算簇索引的排名准确性对比

Figure 4 Comparison of ranking accuracy for cluster index computed by different methods

通过将第一种方式与其他两种方式进行对比, 可以得出以下结论。首先, 与第二种方式对比进一步证实了提高排名准确性的关键在于如何准确计算簇索引, 因为他们的唯一区别在于计算簇质心索引的方式不同, PDR-SFP 计算簇质心索引的准确度更高, 因此排名准确性也显著提高。其次, 与第三种方式对比发现, PDR-SFP 在选择不到 100 时, 其排名准确性就已经相当于 Tiptoe 的排名准确性。

图 5 比较了多个搜索算法在 MS MARCO 文档排名任务上的排名准确性: 基于深度学习的最先进检索系统 ColBERT^[46]、标准的基于关键词的检索系统 BM25^[47]、TF-IDF 算法(Coeus)、在不聚类的情況下通过余弦相似度得分对嵌入向量进行穷举搜索、Tiptoe 以及 PDR-SFP。

图 5 中, 对于 ColBERT, 本文报告了来自 MS MARCO 排行榜的分数; 对于 BM25, 本文报告了使用默认参数($k_1 = 0.9$, $b = 0.4$)的 Anserini BM25 基线文档排名; 对于 TF-IDF, 本文使用 Gensim 库进行词干提取和构建 TF-IDF 矩阵; 对于嵌入搜索, 本文使用与 PDR-SFP 相同的嵌入函数(但不进行聚类); 对于 Tiptoe, 为了实现公平的比较, 本文使用与 PDR-SFP 相同的嵌入函数及聚类结果, 并将其度量

函数从计算内积换成计算余弦相似度, 同时通过主成分分析对嵌入向量进行降维处理(从 768 维降到 192 维); 对于 PDR-SFP, 本文拟使用参数 $k = 200$ 的结果。

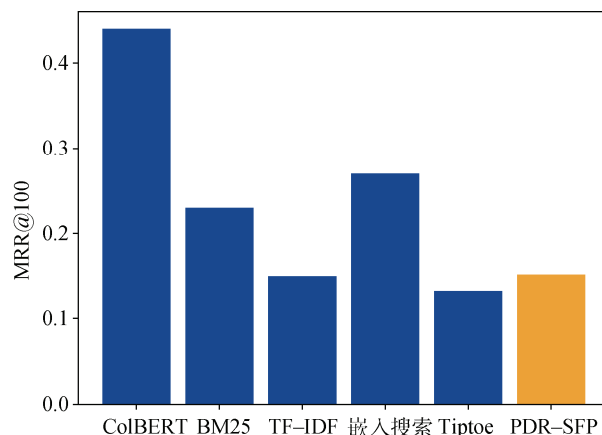


图 5 不同方法的排名准确性对比

Figure 5 Comparison of ranking accuracy across different methods

平均而言, PDR-SFP 的前 100 个搜索结果中, 最佳搜索的平均排名在第 6.53 位。这比 ColBERT、BM25 的排名准确性差, 但是优于具有无限制字典大小的 TF-IDF 以及 Tiptoe。其中, Tiptoe 使用主成分分析对数据进行降维, 以改善通信量和计算量, 但这导致排名准确性相对于仅聚类的方法降低了 0.03。而使用参数的 PDR-SFP, 排名准确性仅降低了 0.01。

图 6 展示了对于几个 MS MARCO 数据集中随机抽样的查询, PDR-SFP 的相似度前三搜索结果。

查询:	central city definition
结果1:	http://www.dictionary.com/browse/central-city
结果2:	https://www.merriam-webster.com/dictionary/central%20city
结果3:	https://www.facebook.com/places/Things-to-do-in-Central-City-Iowa/108122035882157
查询:	government does do
结果1:	https://www.reference.com/government-politics/government-f228d84d469d2f4c
结果2:	https://www.foodsafety.gov/keep/government/index.html
结果3:	http://www.peoplepolitico.com/what-does-our-government-do/
查询:	how to cook string beans
结果1:	http://www.ehow.com/how_5601645_cook-string-beans.html
结果2:	http://www.gardenguides.com/102777-grow-string-beans-seed.html
结果3:	http://www.livestrong.com/article/432484-how-to-cook-frozen-string-beans-in-boiling-water/

图 6 PDR-SFP 搜索结果示例

Figure 6 Example of PDR-SFP search results

5.5 查询成本评估

本节将从文档集合表示占用的存储空间、查询的通信开销以及查询的时间开销等方面, 对 PDR-SFP 与其他保护隐私的文档排名方法进行全面的对比评估。

1) 文档集合表示占用的存储空间评估

本文将对比 Coeus、Tiptoe 以及 PDR-SFP 存储整个文档库表示所需的存储空间。Coeus 使用 TF-IDF 对每个文档进行编码, 每个文档所需要的存储空间取决于语料库中唯一词语的数量、每个权重的存储类型(例如, float32 或 float64 等)以及向量的稀疏性。Coeus 还使用了一种压缩技术, 将多个较小的文档打包到一个对象中。具体来说, 在百万级文档库的情况下, Coeus 的每个对象需要 142 KiB 存储空间, 平均每个对象包含 50 个文档, 因此平均每个文档需要 2908 个字节^[5]。Tiptoe 则使用一个 192 维的整数向量表示一个文档, 因此一个文档需要 768 个字节。

而 PDR-SFP 仅使用 k 个比特来表示一个文档, 当取 k 为 200 时, 一个文档仅需 25 字节。如图 7 展示了三种方法不同文档数量所需的存储空间(单位为 GiB), 其中 1M 代表一百万的文档, Coeus、Tiptoe 以及 PDR-SFP 表示一个一亿文档的文档库分别需要 270 GiB、70 GiB 和 2.32 GiB。此外, PDR-SFP 所需的存储空间是随 k 线性变化的。

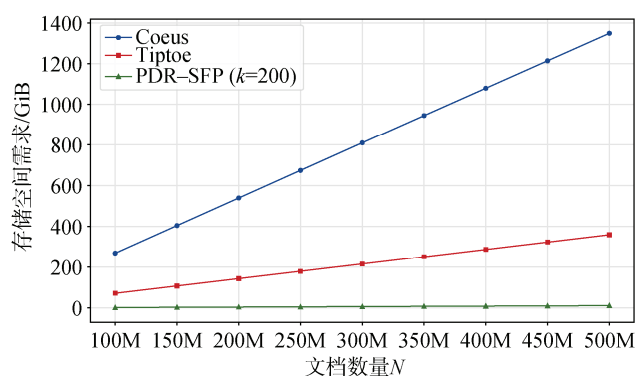


图 7 存储空间对比图

Figure 7 Comparison diagram of storage space

除了上述服务器端的存储需求外, Tiptoe 和 PDR-SFP 还需要在客户端存储额外的信息。在一个有一亿文档的数据集下, Tiptoe 需要下载一个嵌入函数(265 MiB)以及主成分分析降维后的簇质心表示矩阵(大约 7.32 MiB), 总计约 272 MiB。同样地, PDR-SFP 的客户端也需要存储一个嵌入函数(265 MiB)、簇质心嵌入表示矩阵(大约 29 MiB)、一个用

于加密的矩阵 A 的随机种子、一个用于随机投影的矩阵(600 KiB), 以及用于加速解密的矩阵 H (大约 78.12 MiB), 总计约 451 MiB。总的来说, PDR-SFP 以牺牲少量的客户端存储空间为代价, 显著地降低了服务器端的存储空间、计算量和通信, 同时也提高了排名准确性。

2) 查询的通信开销评估

本小节将评估 Coeus、Tiptoe 以及 PDR-SFP 每次查询所需要的通信开销。为了进行比较, 本小节只考虑 Coeus 以及 Tiptoe 查询评分阶段成本。

对于 N 个文档, Coeus 总共需要 $10.66 \cdot N$ 字节的通信量。Tiptoe 采用了一种双层加密算法来实现同态加密, 首先, 它利用 Regev 同态加密方案对查询嵌入进行加密。接着使用基于 Ring-LWE 问题的同态加密方案对 Regev 加密的密钥进行二次加密。加密后的查询和密钥将一并发送给服务器, 使服务器能够在基于 Ring-LWE 的加密保护下执行 Regev 解密。其中该密钥加密的流量可以发生在客户端输入搜索查询之前。本小节将其两层加密安全参数设置为和 PDR-SFP 一致, 并假设他的嵌入向量长度为 d 。具体来说, 对于 N 个文档的每次查询, Tiptoe 查询前发生 30 MiB 的流量, 上传 $d\sqrt{N}$ 个 64 位整数, 下载 $\sqrt{N}/\lambda \cdot \lambda$ 个基于 Ring-LWE 加密的密文。PDR-SFP 的上行通信开销则为 $k\sqrt{N}$ 个 64 位的 Regev 密文, 下行通信开销为 $\sqrt{N}$ 个 64 位的 Regev 密文。这三种方法的通信开销复杂度如表 2 所示, 单位为字节。

表 2 通信开销理论复杂度

Table 2 Theoretical complexity of communication overhead

方法	查询前通信	上行通信开销	下行通信开销	总通信开销
Coeus	0	-	-	$10.66N$
Tiptoe	30×1024	$8d\sqrt{N}$	$26.88 \lceil \sqrt{N}/\lambda \rceil \cdot \lambda$	$30720 + 8\sqrt{N}(3.36 + d)$
PDR-SFP	0	$8k\sqrt{N}$	$8\sqrt{N}$	$8\sqrt{N}(k+1)$

表 3 展示了在五百万和一亿的数据集下, 查询的各阶段通信开销, 单位为 MiB。其中 Coeus 拥有总体最高的通信量。Tiptoe 的大部分通信量都在查询前发生, 其查询期间的通信量略小于 PDR-SFP ($k=200$)。但是总体的通信开销显著大于 PDR-SFP, 且 PDR-SFP 与 Tiptoe 排名准确性一样时($k=100$), 两者的差距更明显, 在五百万数据集下 Tiptoe 的通信量是 PDR-SFP 的 19 倍。

表 3 单次查询通信开销实验结果(MiB)

Table 3 Experimental results of communication overhead for single query (MiB)

设置	查询前通信	上行通信开销		下行通信开销		总通信开销	
文档数 N	-	5×10^6	1×10^8	5×10^6	1×10^8	5×10^6	1×10^8
Coeus	0	-	-	-	-	50.830	1016.61
Tiptoe ^[8]	30	3.275	14.648	0.057	0.256	33.332	44.904
PDR-SFP($k=100$)	0	1.705	7.629	0.017	0.0762	1.723	7.705
PDR-SFP($k=200$)	0	3.411	15.258	0.017	0.0762	3.429	15.335

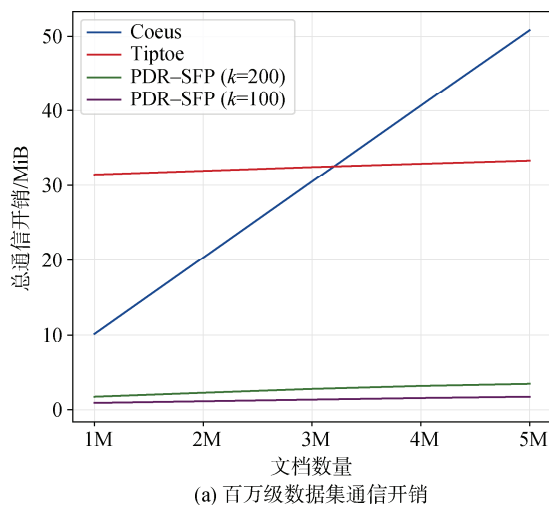
图 8(a)展示了在百万级数据集下, 三种方法的总通信开销对比。可以看出由于 Coeus 的通信量和文档数量呈线性增长, 其不适合较大的数据集。图 8(b)则展示了亿级数据集下 Tiptoe 和 PDR-SFP 的对比, 可以看出 Tiptoe 的通信量仍然是 PDR-SFP ($k=200$) 的两倍左右, 并且 PDR-SFP 在选择更小的 k 值时差距会更明显。

3) 查询的时间开销评估

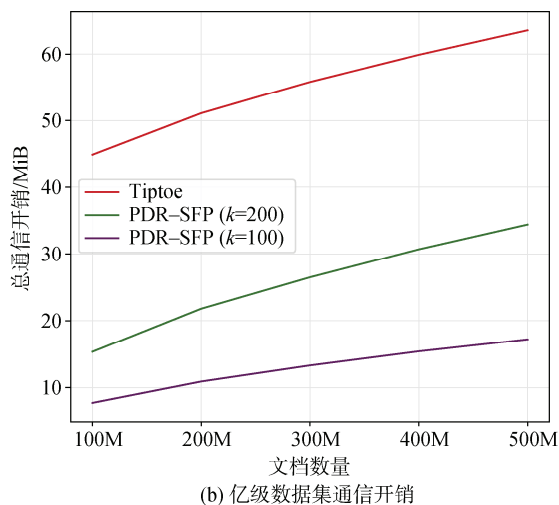
表 4 展示了在维基百科数据集下 Coeus 和 PDR-SFP 查询时间的对比, 以及在 C4 数据集下 Tiptoe 与 PDR-SFP 的对比, 其中单位为核心秒, 是一个衡量 CPU 或 GPU 使用时间的单位, 简单来说, 如果一个 CPU 核心被使用了一秒钟, 那么就产生了 1 个 CPU 核心秒的计算时间。可以看出, PDR-SFP 的

提升非常明显, 相对于 Coeus 有数十万倍的提升, 相对于 Tiptoe 则有近百倍的提升。

PDR-SFP 的性能提升主要归功于以下几点。首先, PDR-SFP 通过随机投影的方法表示文档, 大幅度减少了与 Coeus 相比所需处理的矩阵规模, 降低了数百倍的计算量。其次, 通过符号表示文档向量, 避免了 64 位的乘法操作。然后, 基于预处理的同态加密算法, 使得即使在密文计算的情况下, 相对于明文计算, 计算量的增加仅限于将 32 位运算扩展至 64 位。最后, 最重要的提升在于, 与 CPU 相比, GPU 对于处理这类大规模矩阵运算更为高效。而 PDR-SFP 将文档的符号表示进行二值化编码压缩, 并且其还能够在 GPU 上执行压缩矩阵与密文向量的乘法运算。



(a) 百万级数据集通信开销



(b) 亿级数据集通信开销

图 8 PDR-SFP 查询通信开销随文档数量变化图

Figure 8 Variation of communication overhead with document count

表 4 单次查询所需的核心秒

Table 4 Core seconds required per query

	维基百科(5M 文档)	C4(3.64 亿文档)
Coeus	12900	-
Tiptoe	-	145
Ours($k=200$)	0.038	1.424

图 9 揭示了 PDR-SFP 在处理不同规模数据集时

的查询效率。随着文档数量的增长, 查询时间呈现出亚线性增长的趋势, 这一特点使得其能够在处理较大规模文档集的查询时具有良好的可扩展性。

6 结语

随着云计算技术的兴起, 越来越多的企业、机构、组织选择将其数据存放到云服务器上, 针对云上

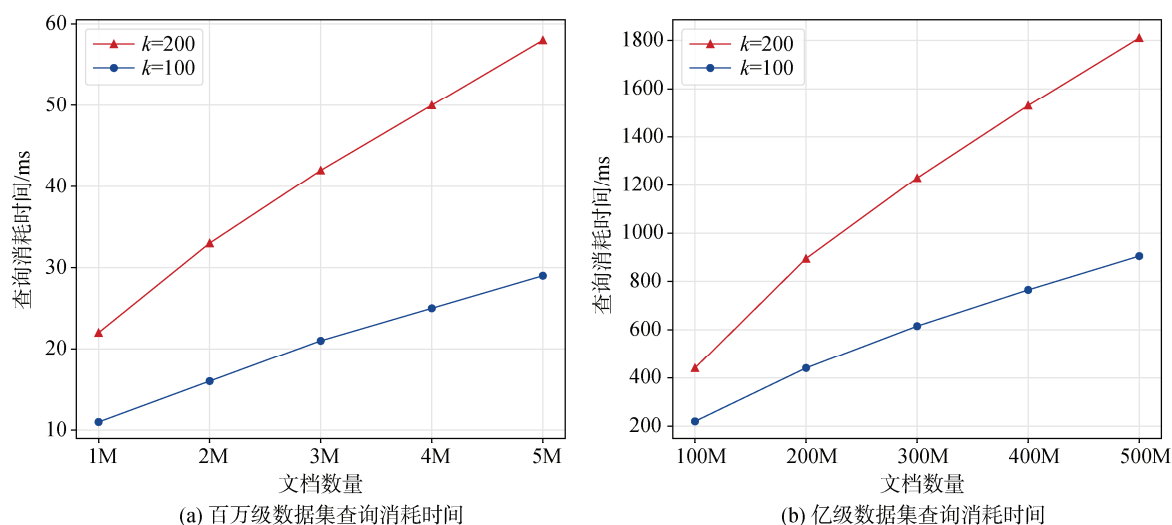


图 9 PDR-SFP 查询时间随文档数量变化

Figure 9 Variation of query time with document count

数据的检索已经拥有了广泛的应用。由于查询可能包含个人敏感信息, 因此在保护查询隐私的条件下实现文档检索成为近年来研究者所重点关注的问题。现有的保护查询隐私的文档检索方法具有计算时间长、通信量大、存储开销高等缺点, 不利于实际应用。本文提出基于 Sign-Full 随机投影的私有文档检索方案, 对云上数据进行预编码。一方面, 在保持查询精确度几乎不变的情况下极大地降低了云存储的成本; 另一方面, 显著降低了用户提交查询后的响应时间, 在极低的计算资源下, 保证了查询响应的实时性, 大大提高了用户的体验。然而, 本方案也存在着一定的局限性。例如, 服务器方的在线计算复杂度仍为 $O(N)$, 计算量随文档集中文档的数量呈线性增长; 对于数据库的动态更新没有较好的优化策略, 使得协议在文档集动态更新频率较快时的场景下表现仍有待提高等, 这些仍存在的问题也为未来的研究提供了研究方向。

参考文献

- [1] Shen X S, Huang C, Liu D X, et al. Data Management for Future Wireless Networks: Architecture, Privacy Preservation, and Regulation[J]. *IEEE Network*, 2021, 35(1): 8-15.
- [2] Deng Hua, Song Fuyuan, Fu Ling, et al. A Review of Data Security and Privacy Protection in Cloud Computing Environments [J]. *Journal of Hunan University (Natural Science Edition)*, 2022, 49(4): 1-10.
- [3] Gui Z C, Paterson K G, Patranabis S. Rethinking Searchable Symmetric Encryption[C]. *2023 IEEE Symposium on Security and Privacy*, 2023: 1401-1418.
- [4] Liu Wenxin, Gao Ying. Research Progress on the Security of Symmetric Searchable Encryption [J]. *Journal of Cyber Security*, 2021, 6(2).
- [5] Ahmad I, Sarker L, Agrawal D, et al. Coeus: A System for Oblivious Document Ranking and Retrieval[C]. *The ACM SIGOPS 28th Symposium on Operating Systems Principles*, 2021: 672-690.
- [6] Zhang Mingwu, Huang Jiajun, Han Liang. A Multi-Keyword Range Search Scheme for Privacy Protection in Medical Big Data [J]. *Journal of Software*, 2021, 32(10): 3266-3282.
- [7] Le Jieyu, Luo Chaoyang, Ding Jingshu, et al. Research on Privacy Protection Mechanisms and Technologies in Educational Big Data [J]. *Big Data*, 2020, 6(6): 0.
- [8] Henzinger A, Dauterman E, Corrigan-Gibbs H, et al. Private Web Search with Tiptoe[C]. *The 29th Symposium on Operating Systems Principles*, 2023: 396-416.
- [9] Peng Y, Liu Q, Tian Y, et al. Dynamic Searchable Symmetric Encryption with Forward and Backward Privacy[C]. *2021 IEEE 20th International Conference on Trust, Security and Privacy in Computing and Communications*, 2021: 420-427.
- [10] Sun S F, Zuo C, Liu J K, et al. Non-Interactive Multi-Client Searchable Encryption: Realization and Implementation[J]. *IEEE Transactions on Dependable and Secure Computing*, 2022, 19(1): 452-467.
- [11] Lin W Y, Cui H L, Li B C, et al. Privacy-Preserving Similarity Search with Efficient Updates in Distributed Key-Value Stores[J]. *IEEE Transactions on Parallel and Distributed Systems*, 2021, 32(5): 1072-1084.
- [12] Miao Y B, Zheng W, Jia X H, et al. Ranked Keyword Search over Encrypted Cloud Data through Machine Learning Method[J]. *IEEE Transactions on Services Computing*, 2023, 16(1): 525-536.
- [13] Wang G L, Cao Z F, Dong X L. Improved File-Injection Attacks on Searchable Encryption Using Finite Set Theory[J]. *The Computer Journal*, 2021, 64(8): 1264-1276.
- [14] Oya S, Kerschbaum F. Hiding the access pattern is not enough: Exploiting search pattern leakage in searchable encryption[C]. *30th USENIX security symposium*, 2021: 127-142.
- [15] Kornaropoulos E M, Papamanthou C, Tamassia R. The State of the

- Uniform: Attacks on Encrypted Databases beyond the Uniform Query Distribution[C]. *2020 IEEE Symposium on Security and Privacy*, 2020: 1223-1240.
- [16] Wu Z Q, Li R. OBI: A Multi-Path Oblivious RAM for Forward-and-Backward-Secure Searchable Encryption[C]. *Proceedings 2023 Network and Distributed System Security Symposium*, 2023: 1-16.
- [17] Choi S G, Dachman-Soled D, Gordon S D, et al. Compressed Oblivious Encoding for Homomorphically Encrypted Search[C]. *The 2021 ACM SIGSAC Conference on Computer and Communications Security*, 2021: 2277-2291.
- [18] Dauterman E, Feng E, Luo E, et al. {DORY}: An encrypted search system with distributed trust[C]. *14th USENIX Symposium on Operating Systems Design and Implementation*, 2020: 1101-1119.
- [19] de Castro L, Polychroniadou A. Lightweight, Maliciously Secure Verifiable Function Secret Sharing[M]. *Advances in Cryptology – EUROCRYPT 2022*. ChamSpringer International Publishing2022: 150-179.
- [20] Chor B, Kushilevitz E, Goldreich O, et al. Private Information Retrieval[J]. *Journal of the ACM*, 1998, 45(6): 965-981.
- [21] Cheng W, Sang D Z, Zeng L F, et al. Tianji: Securing a Practical Asynchronous Multi-User ORAM[J]. *IEEE Transactions on Dependable and Secure Computing*, 2023, 20(6): 5143-5155.
- [22] Hoang T, Yavuz A A, Guajardo J. A Multi-Server ORAM Framework with Constant Client Bandwidth Blowup[J]. *ACM Transactions on Privacy and Security*, 2020, 23(1): 1-35.
- [23] Zhang X, Huang C, Su Y, et al. Divertible Searchable Symmetric Encryption for Secure Cloud Storage[C]. *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, 2022: 3785-3790.
- [24] Gilboa N, Ishai Y. Distributed Point Functions and Their Applications[M]. *Advances in Cryptology – EUROCRYPT 2014*. Berlin, HeidelbergSpringer2014: 640-658.
- [25] Boyle E, Gilboa N, Ishai Y. Function Secret Sharing: Improvements and Extensions[C]. *The 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016: 1292-1303.
- [26] Huang C, Liu D X, Yang A J, et al. Multi-Client Secure and Efficient DPF-Based Keyword Search for Cloud Storage[J]. *IEEE Transactions on Dependable and Secure Computing*, 2024, 21(1): 353-371.
- [27] Li M, Zhu J, Zhang T, et al. Bringing decentralized search to decentralized services[C]. *15th {USENIX} Symposium on Operating Systems Design and Implementation*, 2021: 331-347.
- [28] Chen Z, Vasilakis G, Murdock K, et al. {VoltPillager}: Hardware-based fault injection attacks against Intel {SGX} Enclaves using the {SVID} voltage scaling interface[C]. *30th USENIX Security Symposium*, 2021: 699-716.
- [29] Murdock K, Oswald D, Garcia F D, et al. Plundervolt: Software-Based Fault Injection Attacks Against Intel SGX[C]. *2020 IEEE Symposium on Security and Privacy*, 2020: 1466-1482.
- [30] Dauterman E, Fang V, Demertzis I, et al. Snoopy: Surpassing the Scalability Bottleneck of Oblivious Storage[C]. *The ACM SIGOPS 28th Symposium on Operating Systems Principles*, 2021: 655-671.
- [31] Corrigan-Gibbs H, Henzinger A, Kogan D. Single-Server Private Information Retrieval with Sublinear Amortized Time[M]. *Advances in Cryptology – EUROCRYPT 2022*. ChamSpringer International Publishing2022: 3-33.
- [32] Lin W K, Mook E, Wichs D. Doubly Efficient Private Information Retrieval and Fully Homomorphic RAM Computation from Ring LWE[C]. *The 55th Annual ACM Symposium on Theory of Computing*, 2023: 595-608.
- [33] Mughees M H, Sun I, Ren L. Simple and practical amortized sub-linear private information retrieval[J]. *Cryptology ePrint Archive*, 2023.
- [34] Zhou M X, Park A, Zheng W T, et al. Piano: Extremely Simple, Single-Server PIR with Sublinear Server Computation[C]. *2024 IEEE Symposium on Security and Privacy*, 2024: 4296-4314.
- [35] Chen H, Chillotti I, Dong Y, et al. {SANNS}: Scaling up secure approximate {k-Nearest} neighbors search[C]. *29th USENIX Security Symposium*, 2020: 2111-2128.
- [36] Servan-Schreiber S, Langowski S, Devadas S. Private Approximate Nearest Neighbor Search with Sublinear Communication[C]. *2022 IEEE Symposium on Security and Privacy*, 2022: 911-929.
- [37] Zuber M, Sirdey R. Efficient Homomorphic Evaluation of k -NN Classifiers[J]. *Proceedings on Privacy Enhancing Technologies*, 2021, 2021(2): 111-129.
- [38] Li P. Sign-Full Random Projections[J]. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019, 33(1): 4205-4212.
- [39] Regev O. On Lattices, Learning with Errors, Random Linear Codes, and Cryptography[J]. *Journal of the ACM*, 2009, 56(6): 1-40.
- [40] Baeviski A, Zhou Y, Mohamed A, et al. wav2vec 2.0: A framework for self-supervised learning of speech representations[J]. *Advances in neural information processing systems*, 2020, 33: 12449-12460.
- [41] Charikar M S. Similarity Estimation Techniques from Rounding Algorithms[C]. *The Thiry-Fourth Annual ACM Symposium on Theory of Computing*, 2002: 380-388.
- [42] Manzoor E, Milajerdi S M, Akoglu L. Fast Memory-Efficient Anomaly Detection in Streaming Heterogeneous Graphs[C]. *The 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016: 1035-1044.
- [43] Henzinger A, Hong M M, Corrigan-Gibbs H, et al. One Server for the Price of Two: Simple and Fast {Single-Server} Private Information Retrieval[C]. *32nd USENIX Security Symposium*, 2023: 3889-3905.
- [44] Nguyen T, Rosenberg M, Song X, et al. MS MARCO: A Human Generated MACHine Reading COMprehension Dataset[J]. *CoRR*, 2016.
- [45] Colin R, Noam S, Adam R, et al. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer[J]. *Journal of Machine Learning Research*, 2020, 21(140): 1-67.
- [46] Khattab O, Zaharia M. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT[C]. *The 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020: 39-48.
- [47] Ramesh D, Sharma R P, Edla D R. HHDSSC: Harnessing Healthcare Data Security in Cloud Using Ciphertext Policy Attribute-Based Encryption[J]. *International Journal of Information and Computer Security*, 2020, 13(3/4): 322.



董方明 于 2021 年在四川大学自动化专业获得学士学位。现于西安交通大学控制科学与工程专业攻读博士学位。研究领域为隐私计算、信息检索。Email: dfming@stu.xjtu.edu.cn



梅迪菲 现任中移雄安信息通信科技有限公司科技创新与规划部副总经理。研究领域为网络流量分析。Email: meidifei@cmict.chinamobile.com



安宝宇 于 2012 年在北京邮电大学信息安全专业获得博士学位。现任中国移动通信集团中移系统集成有限公司产品事业部副总经理, 高级工程师。主要研究领域为网络安全、数据安全。Email: anbaoyu@cmict.chinamobile.com



周川 于 2024 年在西安交通大学软件学院获得硕士学位。现就职于湖南电网。主要研究领域为网络安全、数据安全。Email: 2428077412@qq.com



周强 现任中移系统集成有限公司数字政府事业部副总经理, 主要研究领域为网络安全、数据安全。Email: zhouqiang@cmict.chinamobile.com



畅东升 现就职于中移雄安信息通信科技有限公司。主要研究领域为信息安全。Email: changdongsheng@cmict.chinamobile.com



王平辉 于 2012 年在西安交通大学获得博士学位, CCF 高级会员。现任西安交通大学网络空间安全学院副院长。主要研究领域为机器学习与数据挖掘, 移动互联网安全, 隐私计算。Email: phwang@mail.xjtu.edu.cn