计 算 机 研 究 与 发 展
Journal of Computer Research and Development
https://crad.ict.ac.cn
62(11): 2855−2869, 2025

基于区块链索引的可验证布尔搜索加密方案

李宇溪[1] 陈晶晶[1] 周福才[2] 徐 剑[2] 纪 东[3]
[1](东北大学计算机科学与工程学院 沈阳 110819)
[2](东北大学软件学院 沈阳 110819)
[3](工业智能与系统优化国家级前沿科学中心(东北大学) 沈阳 110819)
(liyuxi@mail.neu.edu.cn)

Verifiable Boolean Searchable Encryption Scheme Based on Blockchain Index

Li Yuxi[1], Chen Jingjing[1], Zhuo Fucai[2], Xu Jian[2], and Ji Dong[3]
[1] (*School of Computer Science and Engineering*, *Northeastern University*, *Shenyang* 110819)
[2] (*Software College*, *Northeastern University*, *Shenyang* 110819)
[3] (*National Frontiers Science Center for Industrial Intelligence and Systems Optimization* (*Northeastern University*), *Shenyang* 110819)

Abstract To address the limitations in search expressiveness and the inadequacy of verification mechanisms in existing searchable encryption methods, we propose a verifiable Boolean searchable encryption scheme based on blockchain index (VBSE-BI). The scheme first constructs a security model supporting verifiable Boolean search, and based on this model, designs an incremental secure index construction method utilizing blockchain storage structures and achieves efficient search while ensuring the tamper-proof nature of the index structure. Moreover, the scheme introduces an efficient dynamic update mechanism for the secure index, effectively avoiding the significant storage and update overhead caused by auxiliary update structures. To meet the integrity verification requirements of Boolean searches, the scheme defines the unforgeability of Boolean search results and proposes a Boolean operation integrity verification algorithm based on bilinear map accumulators and the extended Euclidean algorithm. Security analysis demonstrates that VBSE-BI scheme can resist dynamic chosen keyword attacks in the random oracle model and satisfy unforgeability under the bilinear q-strong Diffie-Hellman assumption. Compared with similar schemes, VBSE-BI not only supports more expressive Boolean search statements but also significantly reduces the user's computational complexity to log n (where n is the number of keywords). Experimental results show that by optimizing the verification algorithm, the scheme keeps the user's verification time consistently low (1.0−1.8 s), accounting for only 9.98%−14.03% of the server-side computation time. These findings indicate that VBSE-BI is highly suitable for resource-constrained mobile devices, providing a solid theoretical foundation and efficiency assurance for the practical application of searchable encryption.

Key words searchable encryption; blockchain; secure index; Boolean search; unforgeability

摘 要 针对现有可搜索加密方法中搜索表达能力不足以及验证机制不完善的问题,提出了一种基于区块链索引的可验证布尔搜索加密(verifiable Boolean searchable encryption based on blockchain index, VBSE-BI)方案.该方案首先构建了支持可验证布尔搜索的安全模型,并在此基础上设计了一种基于区块链存储结构的增量型安全索引构造方法,在实现高效搜索的同时,确保索引结构的不可篡改性.此外,方案还

收稿日期: 2024-06-06; **修回日期**: 2025-02-25
基金项目: 中央高校基本科研业务费专项资金(N2316007); 辽宁省自然科学基金项目(2023-BS-053)
This work was supported by the Fundamental Research Funds for the Central Universities (N2316007) and the Natural Science Foundation of Liaoning Province (2023-BS-053).
通信作者: 纪东 (jidong@mail.neu.edu.cn)

设计了安全索引的高效动态更新机制,有效避免了因构建辅助更新结构而导致的索引存储和更新开销过大的问题.针对布尔搜索的完整性验证需求,方案定义了布尔搜索结果的不可伪造性,并提出了一种基于双线性映射累加器和扩展欧几里得算法的布尔运算完整性验证算法.安全性分析表明,VBSE-BI方案在随机预言模型下可抵御动态选择关键字攻击,并在双线性q阶强Diffie-Hellman假设下满足不可伪造性.与同类方案相比,VBSE-BI方案不仅支持更丰富的布尔搜索语句,还将用户端计算复杂度显著降低至$\log n$(n为关键字个数).实验结果显示,通过优化验证算法,该方案使用户端验证时间始终保持在较低水平(1.0～1.8 s),仅为服务器端计算时间的9.98%～14.03%.上述结果表明,VBSE-BI方案非常适用于资源受限的移动设备,为可搜索加密的实际应用提供了坚实的理论基础和高效性保障.

关键词 可搜索加密;区块链;安全索引;布尔搜索;不可伪造性

中图法分类号 TP309

DOI: 10.7544/issn1000-1239.202440501 **CSTR:** 32373.14.issn1000-1239.202440501

当前,移动互联、万物互联以及数字化转型所带动的海量数据流量致使人们对存储的需求呈现爆发式增长.海量数据所增长的储存和处理需求直接拉动云环境下外包数据存储市场高速增长.然而,在人们享受云存储带来的便利的同时,数据安全性的问题随之而来.为了防止云服务提供商和黑客未经授权访问用户的外包数据,用户可以将敏感和隐私数据进行加密再外包至云服务器.然而,传统加密方法会改变数据的原始特征,云服务器无法对密文进行搜索和处理,严重影响外包数据的可用性.为了解决上述问题,研究者们提出了可搜索加密技术,该技术在保障用户安全方便地使用外包数据存储服务的同时,使云服务器端能够在不解密数据的前提下对密文进行搜索,从而保证了外包数据的机密性与可用性.近年来,可搜索加密技术得到了广泛且深入的研究,但仍面临许多挑战.

首先是动态更新方面.现实场景中,企业及个人用户外包数据很少是一成不变的,经常需要频繁更新.已有可搜索加密方案中利用前向索引构造的方案[1]支持直接文件更新,动态性较好,易于实际开发应用,但是前向索引结构泄露了和明文相关的信息,容易被敌手进行统计攻击,而且搜索效率不尽如人意.倒排索引结构是实现关键字到文件映射关系的最佳方式,能够实现高效搜索[2],然而,倒排索引动态更新过程较为复杂,添加或删除文件时必须对索引中的所有条目进行线性扫描.现有基于倒排索引的可搜索加密方案的动态更新过程大多需要额外更新结构进行辅助,这些更新结构通常过于复杂,导致更新效率非常低,并且只能允许有限的更新操作.除此之外,已有的动态更新方法在更新过程中往往会向服务器泄露较多信息,例如关于更新关键字的关联信息等,即敌手可将更新的文件与之前的历史搜索结果建立联系,进而使方案无法抵抗文件注入攻击.

其次,为了更好地提高用户的搜索体验,可搜索加密的一个重要研究方向是提高搜索语句表达性,例如支持多关键字布尔搜索[3].严格意义上的布尔搜索是指利用布尔逻辑运算符连接各个搜索关键字,然后由计算机进行相应逻辑运算,以找出所需信息的方法.布尔搜索在所有搜索功能中使用面最广、使用频率最高.但是,布尔搜索不仅需要云服务器进行关键字密文搜索,还需要对搜索结果进行布尔运算,过程中更容易由于节点崩溃或外部攻击而返回不正确的布尔搜索结果.因此为了应对上述问题,亟需引入搜索结果验证机制来保证搜索结果的正确性.一些可搜索加密方案通过数据认证技术,如MAC认证结构[4]、Merkle哈希树[5]和布隆滤波器[6]等,实现对搜索结果的验证.然而上述验证方法仅能验证搜索语句表达性较差的搜索请求,甚至单一关键字搜索请求.对于多关键词布尔搜索,目前的验证方法主要使用基于数据持有性证明技术中随机抽取的思想来实现[7],面临误差较大以及结果验证计算负载较大等缺陷,不能适应实际广泛的应用场景.

最近几年,随着加密货币和去中心化应用的流行,区块链技术受到了各行业极大的关注.区块链数据存储成为构建分布式存储平台的关键技术.例如BigchainDB[8]和Bluzelle[9],其通过共识协议确保数据存储的一致性和不可篡改性,为解决上述这些问题提供了可能.近年来,很多方法将密码学加密技术与区块链结构相融合,一定程度上实现了链上数据的隐私保护[10-11].一些可搜索加密方案以智能合约代替传统的可信第三方来提供搜索公平性的保证[12-14].然而,上述方案中安全索引仍然部署于服务器端,而且

受制于区块链固有的分叉问题导致方案的可用性受到了影响，并未发挥出区块链数据存储固有的防篡改和增量更新性等优势.

本文针对上述问题展开研究，提出基于区块链索引的可验证布尔搜索加密(verifiable Boolean searchable encryption based on blockchain index, VBSE-BI)方案. 该方案将可搜索加密中的安全索引结构与区块链数据存储进行深度融合，限制服务器由于单点故障原因对索引结构进行破坏，实现索引的存储机密性、动态更新性以及搜索正确性. 同时方案在支持对多关键字布尔搜索的同时，保证了布尔搜索结果的不可伪造性. 本文的主要工作有 3 个方面：

1)提出基于区块链存储结构的增量链式安全索引构造方法. 首先构建“关键字-文件”链式索引结构，将索引拆分为数据块的形式存储在区块链结构的交易信息中并利用指针进行链接；引入同态加密机制代替标准加密方法来加密索引信息，在不解密索引的情况下实现高效搜索. 此方法以区块链的不可篡改性来提供搜索的正确性和可靠性，并且避免了构建辅助更新索引结构导致的索引存储及更新开销大的问题.

2)提出针对布尔搜索结果的完整性验证方法. 针对服务器端布尔运算完整性验证需求，给出 VBSE-BI 的搜索结果不可伪造性定义，基于双线性映射累加器和扩展欧几里得算法设计面向布尔搜索的完整性证明以及验证方法，并且，基于双线性 q 阶强 Diffie-Hellman 假设，对该方法的不可伪造性进行了严格的安全性证明.

3)对 VBSE-BI 方案的每个算法和协议进行详细构建，并从存储开销、计算开销以及通信开销 3 个维度对比分析方案性能；对 VBSE-BI 方案进行了编程实现，并通过对算法的性能测试来分析方案在实际应用中的性能表现. 性能分析结果显示，本方案降低了服务器端存储以及搜索开销的同时，降低了用户计算复杂度，满足实际云环境下资源受限本地设备的应用需求.

1 相关工作

近几年，学者们对可搜索加密方案的各方面问题进行了深入的研究. 针对可搜索加密的动态更新问题，2012 年，Kamara 等人[2] 提出了基于正反双链表结构的动态可搜索加密方案，但是在更新文件的时候会造成较大信息泄露，无法抵抗文件注入攻击. 为了降低更新泄露，Hahn 等人[15] 提出一种新的动态可搜索加密方案，主要思想是根据用户的搜索历史构造前向索引和倒排索引 2 个索引结构，云服务器不会得到除了搜索令牌之外的任何用户信息，但是该方案仅在文件数据庞大且关键字数目不多的场景下具有高的搜索效率. 2016 年，Bost[1] 构造了一个满足前向安全性的动态可搜索加密方案，前向安全性保证了更新文件时敌手无法将其关键字信息与之前的搜索历史进行匹配.2018 年，Zuo 等人[16] 基于二进制树和同态加密算法构建了 2 个动态可搜索加密方案，并支持范围查询，第 1 个方案拥有前向安全，第 2 个方案拥有前向安全和部分后向安全，但对加密文档的数量有限制.

在搜索结果的可验证性方面，2012 年 Chai 等人[17] 首先提出并研究了支持搜索结果验证可搜索加密问题. 2015 年，Wang 等人[18] 提出了一种在混合云中支持容错性和可验证性的可搜索加密方案，通过基于字典的索引构建方法，生成模糊关键字字典，实现搜索结果的动态排序和可验证性. Kurosawa 等人[19] 利用 RSA 累加器生成搜索标签，从而实现搜索结果的验证并支持对文件的动态更新. 2015 年，Ameri 等人[20] 利用布隆过滤器和多层级访问控制策略构造了一种通用的多层级可验证关键字搜索方案. 2017 年，Jiang 等人[21] 基于特殊倒排索引结构 QSet、二进制矢量以及 MAC 函数设计了支持多关键字排序的可验证可搜索加密方案. 2018 年，Wang 等人[22] 基于 OXT 的连接关键词检索方法，采用聚合器和采样检测技术提出了公开可验证的连接关键词检索方案. 已有的支持搜索结果验证的方案或多或少会牺牲搜索性能. 2020 年，闫玺玺等人[23] 提出了一种可验证属性基搜索加密方案，该方案结合区块链技术解决半诚实且好奇的云服务器模型下返回搜索结果不正确的问题. 2021 年，Najafi 等人[24] 针对实际应用场景中对多关键字可搜索加密的需求，提出多关键字动态可搜索加密及搜索结果完整性验证方法. 2024 年，Xie 等人[25] 提出了一种新的可验证中文多关键字模糊排序可搜索加密方案，使用认证标签和概率陷门提高安全性、准确性和效率，解决了现有模糊关键词搜索方法的存储和计算资源消耗问题.

基于上述研究现状可以看出，现有的支持验证的可搜索加密方法中，数据拥有者端通常需要存储庞大的辅助结构，导致本地存储负载较大，而且大多可验证算法包含复杂的通信与计算代价，在实际应用中具有局限性. 考虑到资源受限的移动设备数量

急剧增加,用户端的结果验证代价应尽可能小,否则将没有实用价值.因此,提高搜索语句的表达性并且降低用户验证代价是可验证可搜索加密中亟待解决的关键问题.

2 VBSE-BI 模型

VBSE-BI 方案主要考虑 2 类实体:用户 U 和云服务器 S. 方案模型架构如图 1 所示.用户 U 通过对本地文件集合进行信息提取来构建搜索索引,将索引利用密码学算法进行结构化加密,生成安全链式索引,并将链式索引拆分为交易值的形式存储于区块链索引中,根据对应的交易地址以及附加辅助验证元素生成以关键字为键的查找表,将查找表同样利用密码学算法进行结构化加密,生成密态查找表发送给云服务器 S. 区块链的增量存储特性使得索引能够满足文件的动态更新需求,即用户 U 可以随时添加或删除文件,并同时更新索引和查找表结构.随后,用户 U 发起生成布尔搜索请求,利用密码学工具将布尔表达式中的关键字信息进行加密,生成搜索令牌并发送给云服务器 S. 云服务器 S 在收到搜索令牌后,在密态查找表中定位到区块链索引中的搜索关键字对应的链式索引交易地址,进而在区块链安全索引中进行搜索并对搜索结果进行布尔运算.最后,云服务器 S 将搜索结果以及对于布尔搜索结果正确性的证明返回给用户 U. 用户 U 在收到搜索结果和相关的证明之后,在本地验证搜索结果的正确性,验证通过则可判定云服务器 S 正确执行了布尔搜索请求.值得说明的是,区块链只负责存储链式索引结构,而真实的数据以加密的形式存储在云服务器 S 中.

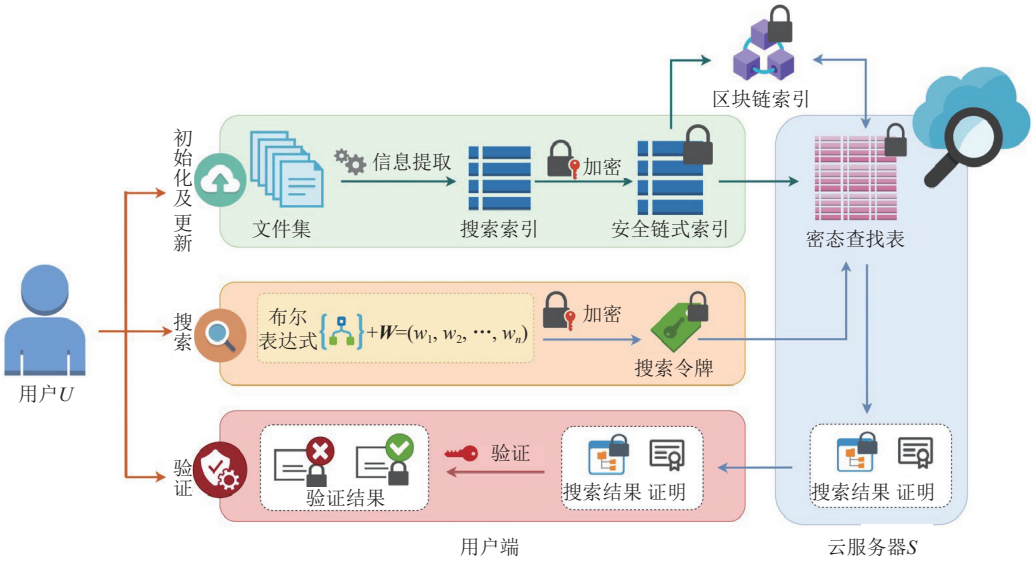


Fig. 1 VBSE-BI model architecture

图 1 VBSE-BI 模型架构

2.1 符号描述

本节列出了论文中涉及的符号及其对应的含义,如表 1 所示.

2.2 形式化定义

定义 1. VBSE-BI 方案主要由 5 个多项式时间算法和协议组成,即 *KeyGen*, *Setup*, *Search*, *Verify*, *Update*,具体描述如下:

1) $(K, PP) \leftarrow KeyGen(1^k)$. 密钥生成算法,由用户 U 执行.输入安全参数 1^k,输出密钥 K 和公共参数 PP.

2) $(T, BC) \leftarrow Setup(K, f)$. 初始化算法,由用户 U 执行.输入密钥 K、文件集合 f,输出区块链索引 BC 以及密态查找表 T.

3) $(R_{\phi(W)}, \pi; \perp) \leftarrow Search(U(K, \phi(W)); S(T, BC))$. 搜索协议,由用户 U 和云服务器 S 交互执行,用于搜索文件.用户输入密钥 K、布尔表达式 $\phi(W)$,云服务器 S 输入密态查找表 T 和区块链索引 BC,输出布尔搜索结果 $R_{\phi(W)}$ 以及正确性证明 π.

4) $b \leftarrow Verify(K, R_{\phi(W)}, \pi)$. 验证算法,由用户 U 执行.输入密钥 K、搜索结果 $R_{\phi(W)}$ 以及证明 π,输出 1 或者 0 表示接受或拒绝.

5) $(\perp; (T', BC')) \leftarrow Update(U(K, f); S(T, BC))$. 更新协议,由用户 U 和云服务器 S 交互执行.用户输入密钥 K、更新文件 f,云服务器输入密态查找表 T 和区块链索引 BC,输出更新后的区块链索引 BC' 以及密

Table 1 Symbols Description

表 1 符号描述

符号	含义
1^k	安全参数
f	文件集合
f_{w_i}	包含关键字w_i的文件集合
BC	区块链索引
T	密态查找表
$\phi(\mathcal{W})$	关键字$\mathcal{W}$布尔表达式
$R_{\phi(\mathcal{W})}$	搜索结果
$\mathcal{L}$	泄露函数
π	证据
τ	搜索/更新令牌
F，G，P	伪随机函数
K_1，K_2，K_3	伪随机函数F，G，P的密钥
K	用户密钥
$L_w=\{N_1,\cdots,N_{\#f_w}\}$	BC中关键字w对应的虚拟链表
$addr(N)$	节点N在BC中的地址
$acc(f_{w_i})$	f_{w_i}的双线性累加值
$I/U/D$	交集/并集/差集
$flag$	添加/删除标志

态查找表T'.

定义 2. VBSE-BI 正确性.VBSE-BI 方案的正确性包括搜索正确性以及更新正确性.

1）搜索正确性. 对于所有的安全参数 k、所有的由$KeyGen(1^k)$生成的K、所有的文件f以及所有由$Setup(K,f)$输出的(T, BC)和$Update(U(K,f);S(T,BC))$，即任意顺序的更新操作，用户U与云服务器S交互执行的搜索协议$Search(U(K,f);S(T,BC))$总会得到正确的结果$R_{\phi(\mathcal{W})}$，即用户端的验证算法$Verify(K,R_{\phi(\mathcal{W})},\pi)$输出为 1.

2）更新正确性. 用户U利用更新协议$Update(U(K,f);S(T,BC))$与云服务器S交互添加或删除包含关键字信息的文件f时，协议执行后服务器端更新的密态查找表T'和区块链索引BC'会正确添加或删除文件f的相关信息，且二者仍然保持一致性且完整性.

2.3 安全性定义

在 VBSE-BI 方案的威胁模型中，假设云服务器为恶意敌手，即可能伪造搜索结果的布尔运算过程. 在这种威胁存在的情况下，VBSE-BI 方案应该满足以下安全性定义：

定义 3. 动态 IND-CKA2 安全性. 设$\mathcal{A}$为敌手，$\mathcal{S}im$为模拟器，$\mathcal{L}_1$，$\mathcal{L}_2$，$\mathcal{L}_3$分别为初始化、搜索、更新文件阶段的泄露函数. 定义如下 2 个实验：

1）$Real_{\mathcal{A}}(1^k)$. 为敌手$\mathcal{A}$和挑战者C之间的交互式实验，使用真实的方案算法. 在该实验中，挑战者C运行$KeyGen(1^k)$生成密钥K和公共参数PP. 敌手$\mathcal{A}$输出文件集合f给挑战者C，挑战者C运行初始化算法$Setup(K,f)$，输出(T,BC)给敌手. 敌手$\mathcal{A}$可以向挑战者发起多项式次自适应质询$\{\phi(\mathcal{W}_i),f_i\}_{1\leqslant i\leqslant q}$. 对于每次质询$\{\phi(\mathcal{W}_i),f_i\}$，挑战者$C$为敌手$\mathcal{A}$生成搜索令牌$\tau_{s,i}$. 或者更新令牌$\tau_{u,i}$. 由于$\mathcal{A}$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q次质询结束后，敌手$\mathcal{A}$输出一个比特值b作为该实验的输出.

2）$Ideal_{\mathcal{A},\mathcal{S}}(1^k)$. 为敌手$\mathcal{A}$和模拟器$\mathcal{S}im$之间的交互式实验. 敌手$\mathcal{A}$输出索引结构$\delta$. 根据泄露函数$\mathcal{L}_1$，模拟器$\mathcal{S}im$生成模拟的$(\widetilde{T},\widetilde{BC})$并发送给敌手$\mathcal{A}$. 敌手$\mathcal{A}$可以向模拟器$\mathcal{S}im$发起多项式次自适应质询$\{\phi(\mathcal{W}_i),\delta_{f_i}\}_{1\leqslant i\leqslant q}$. 对于每次质询$(\phi(\mathcal{W}_i),\delta_{f_i})$，模拟器可以访问泄露函数$\mathcal{L}_2$和$\mathcal{L}_3$其中之一，并返回给敌手对应的模拟搜索令牌$\tilde{\tau}_S$. 最终，敌手$\mathcal{A}$返回一个比特值$b$作为该实验的输出.

VBSE-BI 方案满足动态 IND-CKA2 安全性，当且仅当对于所有满足概率多项式时间计算能力的敌手$\mathcal{A}$，存在多项式时间计算能力的模拟器$\mathcal{S}im$，使得

$$|Pr[Real_{\mathcal{A}}(1^k)=1]-Pr[Ideal_{\mathcal{A},\mathcal{S}}(1^k)=1]|\leqslant negl(1^k),$$

其中$negl(1^k)$为以1^k为输入的可忽略函数.

定义 4. 不可伪造性. 设$\mathcal{A}$为一个有状态的敌手，C为挑战者，$\mathcal{L}_1$，$\mathcal{L}_2$，$\mathcal{L}_3$分别为初始化、搜索、更新文件阶段的泄露函数. 考虑游戏$Forge_{\mathcal{A}}(1^k)$中，敌手$\mathcal{A}$输出文件集合f给挑战者C，挑战者C运行初始化算法$(T,BC)\leftarrow Setup(K,f)$，输出区块链索引$BC$以及密态查找表$T$给敌手. 敌手$\mathcal{A}$可以向挑战者发起多项式次自适应质询$\{\phi(\mathcal{W}_i),f_i\}_q$. 在所有$q$次质询结束后，敌手输出包含一组关键字的布尔表达式$\phi(\mathcal{W})$、一个针对该表达式的错误搜索结果$\widetilde{R}_{\phi(\mathcal{W})}$和针对这一错误结果的证明$\pi$. 如果证明可以通过验证算法的验证，即$Verify(K,\widetilde{R}_{\phi(\mathcal{W})},\pi)=1$，那么实验输出 1，否则输出 0.

VBSE-BI 方案满足不可伪造性，当且仅当对于所有满足多项式时间计算能力的敌手$\mathcal{A}$，使得：$Pr[Forge_{\mathcal{A}}(1^k)=1]\leqslant negl(1^k)$，其中$negl(1^k)$为以$1^k$为输入的可忽略函数.

3 VBSE-BI 方案

本节给出 VBSE-BI 方案的设计思路与具体构建方法. 首先通过构建基于区块链数据存储的增量安

全索引结构来实现对密文的搜索以及高效更新,然后设计布尔搜索结果完整性验证方法来实现布尔搜索结果的完整性验证,最后根据设计思路完成方案中每个算法和协议的详细构建.

3.1 设计思路

3.1.1 基于区块链的安全索引构建方法

VBSE-BI 的主要搜索结构包含存储于区块链结构中的增量链式安全索引(简称区块链索引 BC)以及存储于云服务器的密态查找表 T. 令 F, G, P 为 3 个伪随机函数,用户 U 随机选择 3 个长度为 k 比特的比特串 K_1, K_2, K_3作为伪随机函数 F, G, P 的密钥. 用户首先在文件集合 $f=\{f_1,f_2,\cdots,f_{\#f}\}$中提取到关键字信息 $\mathcal{W}=\{w_1,w_2,\cdots,w_{\#\mathcal{W}}\}$, 构建以关键字为引导的进行组织的 $\#\mathcal{W}$ 条虚拟链表 $\{L_1,L_2,\cdots,L_{\#\mathcal{W}}\}$, 每个链表都由 $\#f_w$ 个节点 $\{N_1,N_2,\cdots,N_{\#f_w}\}$构成, 每个节点都存储一个唯一的“文件/关键字”对, 所有链表共同构成链式索引结构. 随后, 用户将链式索引结构以数据条目的形式上传到区块链中, 每个节点存储在 BC 中的一条交易中. 为了能够将区块链索引中一条虚拟链表中的不同节点进行连接, 在每个节点 N_i 内额外添加指针信息 $addr(N_{i-1})$, 指向链表内上一个节点 N_{i-1} 的交易地址, 从而实现整条链表的高效定位与查找. 定义节点 N_i 的结构为 $N_i=\langle f_i,addr(N_{i-1}),flag\rangle$, 其中 f_i 为文件的唯一标识, $addr(N_{i-1})$ 表示链表内上一个节点 N_{i-1} 的交易地址, $flag$ 为增删标识位. 链表 L_w 中的所有节点都使用 $P_{K_3}(w)$ 进行异或加密从而对索引的真实索引信息进行隐私保护, 其中 K_3 为伪随机函数 P 的密钥. 除此之外, 用户构建查找表 T, 它包含了系统中的所有关键字信息, 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, 以指向区块链索引中链表 L_w 的末节点地址 $addr\left(N_{\#f_w}\right)$ 为值, 其中 K_1 为伪随机函数 F 的密钥, $addr\left(N_{\#f_w}\right)$ 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.

搜索关键字 $w_i\in\mathcal{W}$ 包含的文件时, 服务器 S 使用 $F_{K_1}(w_i)$ 作为键在密态查找表 T 中定位密态地址, 使用 $G_{K_2}(w_i)$ 解密得到 L_{w_i} 的末节点地址明文, 并将其链接到区块链索引 BC 中的相关交易, 最后使用 $P_{K_3}(w_i)$ 来解密交易信息, 获得文件标识和前一个节点的地址信息, 进而在区块链索引 BC 中提取链表 L_{w_i} 中所有节点 $\left\{N_1,N_2,\cdots,N_{\#f_{w_i}}\right\}$ 的信息, 得到针对关键字 w_i 的文件标识符集合 f_{w_i}.

图 2 为区块链索引结构示意图. 为了更好地说明方案的索引结构间的关系和搜索过程, 图 2 中索引结构均以明文形式阐述. 在区块链结构中, 一定数量的新数据条目作为交易的形式封装到一个新的块中, 块头包含当前链的最后一个块的哈希值、随机数、交易的 Merkle 哈希根以及时间戳等信息, 块体包含了所有的交易信息, 即存储索引 BC. 例如, 对于关键字 w_2, 在查找表中键 w_2 对应的值为 402, 即指向了区块链中地址为 402 的交易. 地址为 402 的交易信息中存储了一个指向地址为 101 交易的指针, 地址为 101 的交易中存储的指针指向了地址 0, 然后结束遍历. 通过对关键字 w_2 的虚拟链表的遍历, 所有包含关键字 w_2 的文件都可以被找到. 上述过程实现搜索正确性的思想是, 由于区块链结构对网络中的所有对等点都是可审计的, 一旦挖掘出一个块并达成一致意见, 就不再在该块中进一步执行修改操作. 因此区块链索引凭借着安全不可篡改的特性, 能防止已经确认的交易记录被更改, 能够保证每个得到有效性验证的区块数据和交易记录中的节点信息的完整性, 因此其保证了索引存储以及搜索的正确性.

3.1.2 布尔搜索完整性验证方法

为了搜索针对关键字集合 $\mathcal{W}=\{w_1,w_2,\cdots,w_n\}$ 的某个布尔表达式 $\phi(\mathcal{W})$, 服务器 S 需要通过查找表对每个关键字 $w_i\in\mathcal{W}$ 在区块链中定位链表 L_{w_i} 的末节点 $N_{\#f_{w_i}}$ 的地址 $addr\left(N_{\#f_{w_i}}\right)$, 并分别在区块链索引中进行搜索, 得到结果 $R=\left\{f_{w_1},f_{w_2},\cdots,f_{w_{\#\mathcal{W}}}\right\}$ 后进行针对布尔表达式 $\phi(\mathcal{W})$ 的运算. 根据类型不同, 布尔运算正确性满足如下性质:

1)交集. 如果 $I=\{f_1,f_2,\cdots,f_m\}$ 为 $\{f_{w_1},f_{w_2},\cdots,f_{w_n}\}$ 的正确交集, 当且仅当 2 个条件成立: ①子集条件. 作为集合的交集, 集合 I 一定为每一个集合 f_{w_i} 的子集, 即 $\left(I\subseteq f_{w_1}\right)\cap\cdots\cap\left(I\subseteq f_{w_n}\right)$. ② 完备性条件. 集合 I 必须包含集合 $\{f_{w_1},f_{w_2},\cdots,f_{w_n}\}$ 的所有共同关键字, 即 $\left(f_{w_1}-I\right)\cap\cdots\cap\left(f_{w_n}-I\right)=f_{w_m}$.

2)并集. 直观上, $U=\{f_1,f_2,\cdots,f_m\}$ 为 $\{f_{w_1},f_{w_2},\cdots,f_{w_m}\}$ 的正确并集, 当且仅当 2 个条件成立: ①成员条件. 作为集合的并集, 集合 U 中的任何文件一定包含于某个集合 f_{w_i}, 即 $\forall f_k\in U,\exists i\in\{1,2,\cdots,n\}:f_k\in f_{w_n}$. ②超集条件. 任何集合 f_{w_i} 一定是集合 U 的子集, 即 $\left(U\supseteq f_{w_1}\right)\wedge\left(U\supseteq f_{w_2}\right)\wedge\cdots\wedge\left(U\supseteq f_{w_n}\right)$.

3)差集. 给定集合 D 和 2 个集合 f_{w_i},f_{w_j}, D 为 f_{w_i},f_{w_j} 正确差集当且仅当差集条件成立: 集合 D 中的文件包含所有包含关键字 w_i 但不包含关键字 w_j 中的文件, 即 $D=f_{w_i}-f_{w_j}\Leftrightarrow f_{w_i}-\left(f_{w_i}-D\right)=f_{w_i}\cap f_{w_j}$. 如果 $I_{w_i\wedge w_j}$ 为集合 f_{w_i} 和 f_{w_j} 的交集, 即 $I_{w_i\wedge w_j}=f_{w_i}\cap f_{w_j}$, 则集合 D 和 $I_{w_i\wedge w_j}$ 的并集与集合 f_{w_i} 相等: $D\cup I_{w_i\wedge w_j}=f_{w_i}$. 因此, 证明 $D=f_{w_i}-f_{w_j}$ 等价于证明 $f_{w_i}=D\cup I_{w_i\wedge w_j}$, 即证明 f_{w_i}

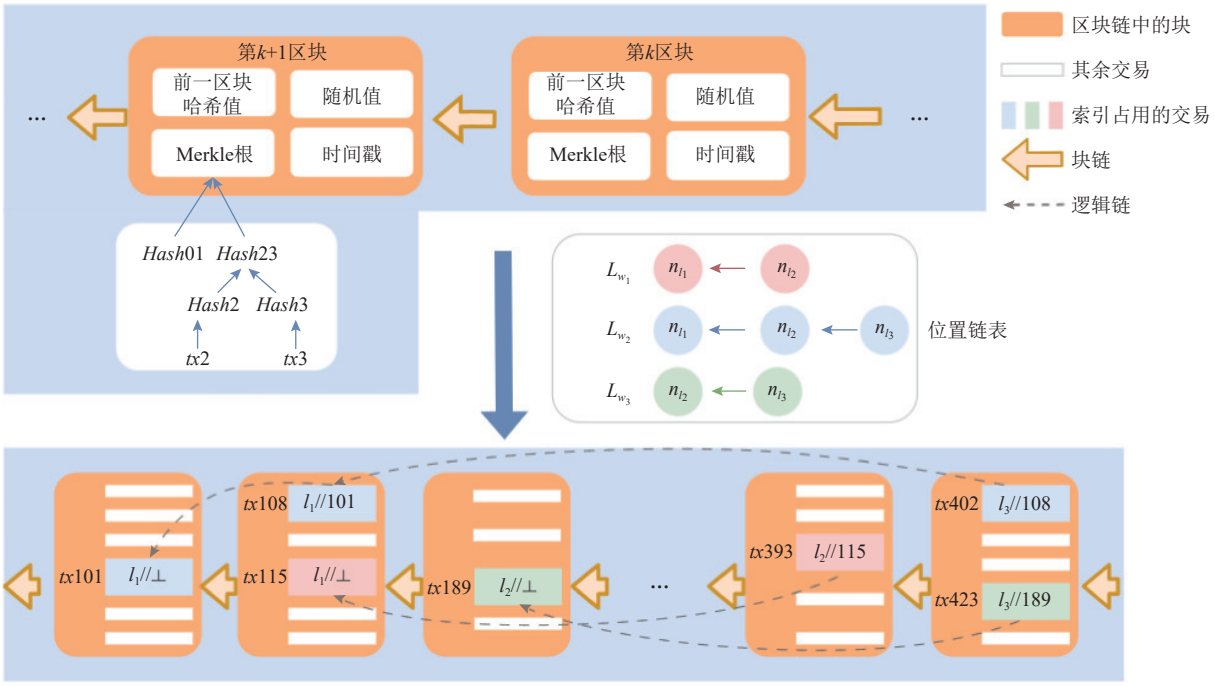


Fig. 2 Schematic diagram of block chain index structure

图 2 区块链索引结构示意图

为D和$I_{w_i \wedge w_j}$的并集.

本文设计融合双线性映射累加器和扩展欧几里得算法的集合运算正确性证明方法.

具体来讲, 为了证明并集搜索结果I的子集条件, 对于所有$w \in \mathcal{W}$, 首先用户分别为$f_{w_1}, f_{w_2}, \cdots, f_{w_n}$计算累加值$acc(f_{w_i})$, 然后便可以为每一个集合$f_{w_i}$生成对应的子集包含证据$(I, \mathcal{W}_{I,f_{w_i}})$. 用户利用$acc(f_{w_i})$和$(I, \mathcal{W}_{I,f_{w_i}})$验证集合$I$和集合$S_i$之间的子集包含关系. 除此之外, 若集合$I$是完备的, 当且仅当存在多项式$q_1(s), q_2(s), \cdots, q_n(s)$, 使得$q_1(s)P_1(s)+q_2(s)P_2(s)+\cdots+q_n(s)P_n(s)=1$, 其中

$$P_i(s)=\prod_{f\in S_i-I}(s+id(f)).$$

$s \in \mathbb{Z}_p^*$作为一个随机的秘密陷门. 因此假设I并不是完备的集合, 那么在多项式$P_1(s), P_2(s), \cdots, P_n(s)$中至少存在 1 个公因式. 在这种情况下, 便不存在多项式$q_1(s), q_2(s), \cdots, q_n(s)$使得$q_1(s)P_1(s)+q_2(s)P_2(s)+\cdots+q_n(s)P_n(s)=1$. 因此如果能找到符合要求的$q_1(s), q_2(s), \cdots, q_n(s)$, 那么就说明$P_1(s), P_2(s), \cdots, P_n(s)$中一定不存在公因式.

为了证明并集搜索结果U的成员条件, 需要为集合U中的每个文件f_j生成针对文件集合f_{w_i}的成员证据$\mathcal{M}_{j,i}$. 验证成员条件通过验证$e\left(\mathcal{M}_{j,i}, g^{f_j} g^s\right) = e\left(acc\left(f_{w_i}\right), g\right)$是否成立. 需要注意的是, 对于集合$U$中的每个文件$f_j$, 只有当对于每个集合$f_{w_i}$上述等式都不成立时才表示元素$f_j$不属于任意集合$f_{w_i}$, 此时认为验证失败. 证明超集条件时, 需要为$f_{w_1}, f_{w_2}, \cdots, f_{w_n}$构造超集证据. 对于$1 \leqslant i \leqslant n$, 计算

$$P_i=\prod_{f\in U-f_{w_i}}(s+f).$$

令$\mathcal{S}=\{\mathcal{S}_i\}_{1\leqslant i\leqslant n}=\{g^{P_i}\}_{1\leqslant i\leqslant n}$为超集证据, 需要验证下列等式是否成立:

$$e\left(\mathcal{S}_i, acc\left(f_{w_i}\right)\right)=e(g^{\prod_{k=1}^{d}(S+f_k)}, g),$$

其中$\mathcal{S}_i$为超集证据$\mathcal{S}$中的元素. 如果所有的验证过程都通过, 那么就可以认为U是正确的, 即的确是由$f_{w_1}, f_{w_2}, \cdots, f_{w_n}$通过并集运算得到的结果. 由于差集证明方法可规约为并集证明, 因此不再过多赘述.

3.1.3 安全索引的动态更新机制

用户U可以随时在方案中添加或删除包含$\mathcal{W}_f=(w_1, w_2, \cdots, w_n)$关键字的文件$f$. 对于$\mathcal{W}_f$中的任意关键字$w_i$, 首先用户$U$和服务器$S$交互获取密态查找表$T$中与$w_i$对应的链表$L_{w_i}$的末节点$N_{\#f_{w_i}}$在区块链索引$BC$中的交易地址$addr(N_{\#f_{w_i}})$以及相关累加值$acc(f_{w_i})$, 并且利用此地址$addr(N_{\#f_{w_i}})$更新链表$L_{w_i}$来增加文件$f$对应的新的末节点$N'_{\#f_{w_i}}$. 之后用户将$N'_{\#f_{w_i}}$作为交易的形式更新到区块链索引$BC$中, 得到$N'_{\#f_{w_i}}$的交易地址$addr(N'_{\#f_{w_i}})$. 除此之外, 用户需要利用原有$f_{w_i}$

对应的累加值$acc(f_{w_i})$和更新的文件f计算新的累加值$acc(f_{w_i})'$. 最后用户将$addr(N'_{\#f_{w_i}})$以及累加值$acc(f_{w_i})'$更新到查找表中w_i键对应的值中, 至此完成安全索引结构的更新. 由于区块链结构为增量数据结构, 且具有防篡改性, 因此无法修改或删除已有交易值信息, 针对上述情况, 将添加和删除文件过程均利用添加节点至区块链来进行, 通过在每个节点内区分$flag$标志位的值来区分添加和删除操作, $flag=1$则表示f为增加的文件, $flag=0$则表示f为删除的文件. 上述动态更新过程中服务器S可以在无需对查找表进行解密的情况下, 使用用户发送的更新令牌中的密文信息直接对密态查找表进行更新.

3.2 详细算法及协议

令$F:\{0,1\}^k\times\{0,1\}^*\to\{0,1\}^k$, $G:\{0,1\}^k\times\{0,1\}^*\to\{0,1\}^*$, $P:\{0,1\}^k\times\{0,1\}^*\to\{0,1\}^k$ 为3个固定长度的伪随机函数, $\sum_{BC}(Init,Put,Get)$为区块链数据存储结构, 具体的算法及协议描述如下.

3.2.1 密钥生成算法

$(K,PP)\leftarrow KeyGen(1^k)$. 进行用户端密钥的生成. 主要步骤为:

1)随机选择3个长度为k比特的比特串K_1,K_2,K_3.

2)选择双线性配对的参数$(p,G,\mathcal{G},e,g)$.

3)选择$s\in\mathbb{Z}_p^*$作为一个随机的秘密陷门.

4)计算$(g,g^s,g^{s^2},\cdots g^{s^q})$作为公开参数.

5)输出密钥$K=(K_1,K_2,K_3,s)$, 并输出公开参数$PP=(g,g^s,g^{s^2},\cdots,g^{s^q})$.

3.2.2 初始化算法

$(T,BC)\leftarrow Setup(K,f)$. 用户$U$初始化索引结构. 主要步骤为:

1)初始化长度为$\#\mathcal{W}$的查找表T.

2)调用$\sum_{BC}.Init(k)$算法, 输入安全参数k, 初始化区块链索引BC.

3)对于每个关键字$w\in\mathcal{W}$

① 构建一个由$\#f_w$个节点$(N_1,N_2\cdots,N_{\#f_w})$构成的链表$L$, 节点的结构定义为

$$N_i=\langle f_i,addr(N_{i-1}),flag\rangle\oplus P_{K_2}(w),$$

其中f_i为文件标识, $addr(N_{i-1})$为节点N_{i-1}的交易地址, $addr(N_1)=\perp$, $flag$为增删标识位.

② 对于链表L_w的所有节点$(N_1,N_2,\cdots,N_{\#f_w})$, 依次调用$\sum_{BC}.Put(N_i)$算法, 将节点$N_i$作为交易值封装进区块链索引$BC$的区块上并获取交易地址$addr(N_i)$, 最终返回$addr(N_{\#f_w})$.

4)对于每个关键字$w\in\mathcal{W}$

① 计算f_w的累加值, 即

$$acc(f_w)=g^{\prod_{f\in f_w}(s+id(f))}.$$

② 输出区块链索引BC和密态查找表T共同作为安全索引结构, 即

$$T\left[F_{K_1}(w)\right]=\langle\theta_1,\theta_2\rangle=\left\langle addr(N_{\#f_w})\oplus G_{K_3}(w),acc(f_w)\right\rangle.$$

3.2.3 搜索协议

$(R_{\phi(\mathcal{W})},\pi;\perp)\leftarrow Search(U(K,\phi(\mathcal{W}));S(T,BC))$. 用户$U$和云服务器$S$交互执行针对关键字集合$\mathcal{W}=\{w_1,\cdots,w_n\}$的布尔表达式$\phi(\mathcal{W})$的搜索过程, 服务器针对搜索结果生成相应布尔运算证明. 主要步骤为:

1)用户U针对每个搜索关键字$w_i\in\mathcal{W}$生成搜索令牌$\tau_i=(\tau_{i,1},\tau_{i,2},\tau_{i,3})=(F_{K_1}(w_i),G_{K_2}(w_i),P_{K_3}(w_i))$. 发送搜索令牌$\tau_s=(\phi(\mathcal{W}),\{\tau_i\}_{i\in\mathcal{W}})$给服务器$S$.

2)服务器S对于每个关键字$w_i\in\mathcal{W}$, 执行以下步骤:

① 使用$\tau_{i,1}$作为键检索密态查找表T中的值$\theta_1\leftarrow T[\tau_{i,1}]$.

② 计算链表头节点在索引BC中的交易地址$\alpha_1\leftarrow\theta_1\oplus\tau_{i,2}$.

③ 在BC中获取地址$addr(N_{\#f_{w_i}})$的交易值$tx_1\leftarrow\sum_{BC}.Get\left(addr_{N_{\#fw}}\right)$.

④ 计算$a_1\|a_2\|a_3\leftarrow tx_1\oplus\tau_{i,3}$.

⑤ 如果$a_3=1$, 则更新$R_{w_i}\leftarrow R_{w_i}\cup\{a_1\}$, 如果$a_3=0$, 则更新$R_{w_i}\leftarrow R_{w_i}/\{a_1\}$.

⑥ 如果$a_2\neq\perp$, 则重复步骤③④, 继续查找BC中地址为a_2的交易值, 直到$a_2=\perp$. 对于所有关键字$w_i\in\mathcal{W}$搜索完毕后, 得到搜索结果$R=\{R_{w_1},R_{w_2},\cdots,R_{w_{\#W}}\}$.

对于布尔表达式$\phi(\mathcal{W})$, 云服务器S进行如下步骤完成集合运算的证明:

1)交集证明

步骤1. 计算交集$R_I=R_{w_1}\cap R_{w_2}\cap\cdots\cap R_{w_n}$, 其中$R_I=\{f_1,f_2,\cdots,f_m\}$.

步骤2. 对于每个集合R_{w_i}, $1\leqslant i\leqslant n$, 执行

①计算多项式$P_i=\prod_{f\in R_{w_i}-R_I}(s+f)$.

②计算集合R_I对于R_{w_i}的子集包含证据$\mathcal{I}_{R_I,R_{w_i}}=g^{P_i}$.

③计算子集证据$\mathcal{I}=\{\mathcal{I}_i\}=\{\mathcal{I}_{I,f_{w_i}}\}_{1\leqslant i\leqslant n}$.

④使用公开参数$(g,g^s,\cdots,g^{s^q})$来计算g^{P_i}的值.

⑤给定$\{P_1,P_2,\cdots,P_n\}$, 使用扩展欧几里得算法找到多项式$\{q_1,q_2,\cdots,q_n\}$, 满足$q_1P_1+q_2P_2+\cdots+q_nP_n=1$, 整合$C=\{g^{q_1},g^{q_2},\cdots,g^{q_n}\}$为完备性证据.

步骤3. 整合交集证据$\pi_I=\{\mathcal{I},C\}$.

2)并集证明

步骤 1. 计算并集 $R_U = R_{w_1} \cup R_{w_2} \cup \cdots \cup R_{w_n}$，其中 $R_U = \{f_1, f_2, \cdots, f_m\}$.

步骤 2. 对于每个集合 R_{w_i}，$1 \leqslant i \leqslant n$，执行

①对于 R_U 中每个文件 f_k，计算多项式

$$F_{k,i} = \prod_{f \in R_{w_i} - f_k} (s + f).$$

②使用公开参数 $(g, g^s, \cdots, g^{s^q})$ 计算 $\mathcal{M}_{k,i} = g^{F_{k,i}}$.

步骤 3. 计算多项式 $P_i = \prod_{f \in R_U - R_i} (s + f)$.

步骤 4. 计算成员证据 $\mathcal{M}=\{\mathcal{M}_{i,j}\}_{1 \leqslant i \leqslant n, 1 \leqslant j \leqslant m}$，超集证据 $\mathcal{S} = \{\mathcal{S}_i\}_{1 \leqslant i \leqslant n} = \{g^{P_i}\}_{1 \leqslant i \leqslant n}$.

步骤 5. 整合并集证据 $\pi_U = \{\mathcal{M}, \mathcal{S}\}$.

3)差集证明

设差集搜索令牌 $\tau_s = (\mathcal{D}(w_i, w_j), \tau_i, \tau_j)$，计算差集 $R_D = R_{w_i} - R_{w_j}$，其中 $R_D = \{f_1, f_2, \cdots, f_m\}$. 之后服务器按照如下步骤计算证据 $\pi_D = \{\mathcal{M}, \mathcal{S}\}$ 以完成证明.

步骤 1. 对于 R_D 中每个文件 f_k，$1 \leqslant k \leqslant m$：

①计算多项式

$F_1 = \prod_{f \in R_D - f_k} (f + s)$，$F_{2,1} = \prod_{f \in R_{w_i} - f_k} (f + s)$，以及 $F_{2,2} = \prod_{f \in R_{w_j} - f_k} (f + s)$.

②使用公开参数 $(g, g^s, \cdots, g^{s^q})$ 来计算

$$\mathcal{M} = \{\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3\},$$

其中 $\mathcal{M}_1 = g^{F_1}$，$\mathcal{M}_2 = g^{F_{2,1}}$，$\mathcal{M}_3 = g^{F_{2,2}}$.

步骤 2. 计算交集 $I_{w_i \wedge w_j} = R_{w_i} \cap R_{w_j}$.

步骤 3. 计算 $acc(I_{w_i \wedge w_j}) = g^{\prod_{f \in I_{w_i \wedge w_j}} (f+s)}$.

步骤 4. 对于集合 R_D 和 $I_{w_i \wedge w_j}$，计算

$$P_1 = \prod_{f \in R_{w_i} - R_D} (f + s),$$

$$P_2 = \prod_{f \in R_{w_i} - I_{w_i \wedge w_j}} (f + s),$$

$$P_3 = \prod_{f \in R_D} (f + s),$$

$$P_4 = \prod_{f \in I_{w_i \wedge w_j}} (f + s).$$

步骤 5. 计算超集证据

$$\mathcal{S} = (\mathcal{S}_1, \mathcal{S}_2, \mathcal{S}_3, \mathcal{S}_4) = (g^{P_1}, g^{P_2}, g^{P_3}, g^{P_4}).$$

步骤 6. 整合差集证据 $\pi_D = \{\mathcal{M}, \mathcal{S}\}$.

随后，云服务器 S 将搜索结果集合 $R_{\phi(\mathcal{W})}$ 和针对该集合的证明 π 返回给用户 U.

3.2.4 验证算法

$b \leftarrow Verify(K, R_{\phi(\mathcal{W})}, \pi)$. 用户 U 验证布尔搜索结果的完整性. 主要步骤为：

1)交集验证

步骤 1. 对于 τ_s 中的每个 τ_i，$1 \leqslant i \leqslant n$，使用 θ_2 通过以下等式来验证子集条件.

$$e(g^{\prod_{k=1}^{m} (s+f_k)}, \mathcal{S}_i) = e(\theta_2, g),$$

其中 $f_1, f_2, \cdots, f_m$ 为交集中的元素，$\mathcal{S}_i = g^{P_i}$ 为子集证据 $\mathcal{S}$ 中的元素.

步骤 2. 通过以下等式来验证完备性条件：

$$\prod_{i=1}^{m} e(\mathcal{S}_i, C_i) = e(g, g),$$

其中 $\mathcal{S}_i = g^{P_i}$ 为子集证据 $\mathcal{S}$ 中元素，$C_i = g^{q_i}$ 为完备性证据 C 中的对应元素.

步骤 3. 如果上述验证过程都通过，则输出“1”代表验证成功；否则输出“0”代表验证失败.

2)并集验证

步骤 1. 对于每个文件 $f_k \in U$，$1 \leqslant k \leqslant m$，使用 θ_2 通过以下等式来验证成员条件：

$$e\left(\mathcal{M}_{k,i}, g^{f_k} g^s\right) = e(\theta_2, g),$$

其中 $\mathcal{M}_{k,i} = g^{F_{k,i}}$ 为 $\mathcal{M}$ 中的元素.

步骤 2. 对于每个集合 R_{w_i}，$1 \leqslant i \leqslant n$，通过以下等式来验证超集条件：

$$e\left(\mathcal{S}_i, acc\left(f_{w_i}\right)\right) = e(g^{\prod_{k=1}^{d} (s+f_k)}, g),$$

其中 $\mathcal{S}_i$ 为超集证据 $\mathcal{S}$ 中的元素.

只有 R_U 中存在的文件对于任意集合 f_{w_i} 的成员条件均不成立时，认为验证失败.

步骤 3. 如果上述验证过程都通过，则输出“1”代表验证成功；否则输出“0”代表验证失败.

3)差集验证

步骤 1. 对于每个文件标识 $f_k \in R_{w_i}$，$1 \leqslant k \leqslant m$，通过以下等式来验证成员条件：

$$e\left(\mathcal{M}_1, g^{f_k} g^s\right) = e(acc(R_D), g),$$
$$e\left(\mathcal{M}_2, g^{f_k} g^s\right) = e\left(acc(R_{w_i}), g\right),$$
$$e\left(\mathcal{M}_3, g^{f_k} g^s\right) = e\left(acc(R_{w_j}), g\right),$$

其中 $(\mathcal{M}_1, \mathcal{M}_2, \mathcal{M}_3) = (g^{F_1}, g^{F_{2,1}}, g^{F_{2,2}})$ 为成员证据 $\mathcal{M}$ 中的元素.

步骤 2. 对于集合 R 和 $I_{w_i \wedge w_j}$，通过以下等式来验证超集条件：

$$e(\mathcal{S}_1, \mathcal{S}_3) = e(g^{\prod_{k=1}^{d} (s+f_k)}, g),$$

$$e(\mathcal{S}_2, \mathcal{S}_4) = e(g^{\prod_{k=1}^{d} (s+f_k)}, g),$$

其中$(\mathcal{S}_1,\mathcal{S}_2,\mathcal{S}_3,\mathcal{S}_4)=(g^{P_1},g^{P_2},g^{P_3},g^{P_4})$为超集证据$\mathcal{S}$中元素.

步骤 3. 当步骤 1 中成员条件 3 个等式都不成立时, 判定验证失败. 如果上述验证过程都通过, 则输出“1”代表验证成功; 否则输出“0”代表验证失败.

3.2.5 更新协议

$(\perp;T';BC')\leftarrow Upate(U(K,f);S(T);BC)$. 用于用户 U 添加或删除包含$\mathcal{W}_f=(w_1,w_2,\cdots,w_d)$关键字的文件$f$时索引结构的更新. 主要步骤如下:

用户 U 计算$(F_{k_F}(w_1),F_{k_F}(w_2),\cdots,F_{k_F}(w_d))$发送至服务器 S; 服务器 S 对于每一个t_{u}^j, 在查找表中获取$T[t_{\mathrm{u}}^j]$, 将$(\theta_1,\theta_2,\cdots,\theta_d)$按序发送回用户 U.

1)用户 U 对于每一个θ_i, 执行以下步骤:

步骤 1. 计算$addr_{N_{\#f_{w_i}-1}}=\theta_i\oplus G_{k_G}(w_i)$.

步骤 2. 如添加文件, 计算

$$N_{\#f_{w_i}}=\left\langle f,addr_{N_{\#f_{w_i}-1}},1\right\rangle\oplus P_{K_P}(w_i),$$

如删除文件, 计算

$$N_{\#f_{w_i}}=\left\langle f,addr_{N_{\#f_{w_i}-1}},0\right\rangle\oplus P_{K_P}(w_i).$$

步骤 3. 调用$\sum_{BC}.Put\left(N_{\#f_{w_i}}\right)$, 将$N_{\#f_{w_i}}$作为交易值封装进区块链索引 BC 的区块上并获取交易地址$addr\left(N_{\#f_{w_i}}\right)$.

步骤 4. 计算更新后f_{w_i}对应的相关累加值

$$\tau_{\mathrm{u},i}^3\leftarrow\theta_2{}^{s+f}.$$

步骤 5. 计算$\tau_i^2\leftarrow G_{K_G}(w_i)\oplus addr\left(N_{\#f_{w_i}}\right)$.

步骤 6. 如果为添加操作, 计算$\tau_i^3\leftarrow\theta_2{}^{s+f}$; 如果为删除操作, 计算$\tau_{\mathrm{u},i}^3\leftarrow\theta_2{}^{1/(s+f)}$.

步骤 7. 计算$\tau_i=\left(\tau_i^1,\tau_i^2,\tau_i^3\right)$.

步骤 8. 对于所有θ_i执行完步骤 1～6 后, 将$\tau_{\mathrm{u}}=\{\tau_1,\tau_2,\cdots,\tau_d\}$作为更新令牌发送给服务器.

2)服务器 S 对于每个τ_i, 执行以下步骤:

步骤 1. 获取$(\theta_1,\theta_2)\leftarrow T[\tau_{i,1}]$.

步骤 2. 使用$\tau_{\mathrm{u},i}^2$对θ_1进行更新: $\theta_1'\leftarrow\tau_{\mathrm{u},i}^1$.

步骤 3. 使用$\tau_{\mathrm{u},i}^3$对θ_2进行更新: $\theta_2'\leftarrow\tau_{\mathrm{u},i}^3$.

步骤 4. 更新查找表 T: $(\theta_1',\theta_2')\leftarrow T[\tau_{i,1}]$.

步骤 5. 输出更新后的索引结构$T';BC'$.

4 安全性分析

4.1 动态 IND-CKA2 安全性

首先定义 3 个泄露函数 $L_1(T,I)$, $\mathcal{L}_2(f,\phi(\mathcal{W}))$, $\mathcal{L}_3(f)$来描述初始化、搜索和更新文件执行期间向服务器泄露的信息. 接下来的定理表明, 给定以上泄露函数的情况下, VBSE-BI 方案在随机预言模型下满足动态自适应 CKA 安全性.

定理 1. 若F,G,P为伪随机函数, 则 VBSE-BI 方案在随机预言模型下针对自适应选择关键字攻击是$(\mathcal{L}_1,\mathcal{L}_2,\mathcal{L}_3)$安全的, 即满足动态 IND-CKA2 安全性.

证明. 在$Ideal_{\mathcal{A},\mathcal{S}}(k)$中, 通过$\mathcal{L}_1$的信息, 模拟器可以确定获知查找表 T 的大小$\#T$、字典大小集合$\#\mathcal{W}$以及集合$[id(w)]_{w\in\mathcal{W}}$. 随后, 它可以使用随机值来构建搜索结构, 并输出作为模拟值$(\widetilde{T},\widetilde{BC})$. 由于真实索引是使用伪随机函数$F,G,P$产生的, 而模拟索引是由随机比特填充的, 因此, 如果函数F,G,P的伪随机性成立, 则对于所有 PPT 敌手$\mathcal{A}$, 无法在多项式时间内区分真实索引和模拟索引.

为了响应敌手的质询, 模拟器需要使用$\mathcal{L}_2$和$\mathcal{L}_3$来构建模拟搜索令牌和更新令牌, 使其与真实令牌无法区分. 具体来说, 泄露函数$\mathcal{L}_2(\delta,f,\phi(\mathcal{W}))$向服务器泄露了所有关键字$w\in\mathcal{W}$的标识$id(w)$, 以及每个关键字标识$id(w)$和包含关键字$w$的文件标识之间的关联. 因为$\tilde{\tau}_{\mathrm{s}}=(\tilde{\tau}_1,\tilde{\tau}_2,\cdots,\tilde{\tau}_n)$, $\tilde{\tau}_i=\left(\gamma(id(w_i)),T\left[\gamma(id(w_i))\right]\oplus iT\left(id(w_i)\right),K_{id(w_i)}\right)$, 而 $\tilde{\tau}_{\mathrm{s}}=(\tilde{\tau}_1,\tilde{\tau}_2,\cdots,\tilde{\tau}_n)$, $\tau_i=\left(F_{K_1}(w_i),G_{K_2}(w_i),P_{K_3}(w_i)\right)$, 根据函数$F$, G, P的伪随机性, 对于所有概率多项式计算能力敌手A, 不能区分搜索令牌和模拟搜索令牌τ_{s}和$\tilde{\tau}_{\mathrm{s}}$. 同理, 泄露函数$L_3(f)$泄露了更新协议中更新文件所包含的关键字的标识符集合$[id(w)]_{w\in\mathcal{W}_f}$, 以及关键字$w$是否是一个新的关键字$\left[apprs(w)\right]_{w\in\mathcal{W}_f}$, 其中$apprs(w)$为一个 1 b 的标志位, 其值为“1”表示关键字w已经存在于索引中; 其值为“0”表示w是一个新关键字. 因此, 基于函数F,G,P的伪随机性, 敌手同样不能区分更新令牌τ_{u}和模拟更新令牌.

综上所述, 对于所有 PPT 敌手$\mathcal{A}$来说, 实验$Real_{\mathcal{A}}(k)$与$Ideal_{\mathcal{A},\mathcal{S}}(k)$的输出是一致的, 除了以可忽略的概率$negl(k)$, 即

$$\left|Pr[Real_{\mathcal{A}}(k)=1]-Pr[Ideal_{\mathcal{A},\mathcal{S}}(k)=1]\right|\leqslant negl(k).$$

因此, VBSE-BI 方案在随机预言模型下满足动态 IND-CKA2 安全性. 证毕.

4.2 不可伪造性

VBSE-BI 方案的不可伪造性指对于所有概率多项式计算能力敌手$\mathcal{A}$, 针对伪造的布尔搜索结果, 无法生成正确有效的证据.

定理 2. 如果双线性q-SDH 假设成立, 那么 VBSE-BI 方案满足不可伪造性.

证明. 如果存在一个概率多项式计算能力敌手

$\mathcal{A}$使得实验$Forge_{\mathcal{A}}(1^k)$的输出为“1”，那么就存在一个模拟器Sim，有能力攻破双线性q-SDH假设. 证明主要针对交集布尔搜索场景的不可伪造性、并集以及差集证明方法与交集不可伪造性证明方法过程类似，不再赘述.

假设$\mathcal{A}$在执行针对集合$\mathcal{W}$的搜索算法Search后，输出一个文件标识符集合$\widetilde{R}_{\phi(\mathcal{W})} \neq R_{\phi(\mathcal{W})}$和一个有效的证明$\pi$. 这意味着该证明$\pi$通过了验证算法Verify中的所有验证，即敌手$\mathcal{A}$成功地进行了伪造. 证明$\pi=\{\mathcal{S},C\}$中的伪造可以分为伪造$\mathcal{S}$和伪造$C$这2种情况. 通过以下分析我们能够得出，模拟器$Sim$能够通过伪造$\mathcal{S}$和伪造$C$来攻破双线性$q$-SDH假设.

给定模拟器Sim双线性q-SDH实例：$(p,G,\mathcal{G},e,g)$以及一个$(q+1)$元组$(g,g^s,\cdots,g^{s^q})$，模拟器Sim和敌手$\mathcal{A}$在实验$Veri_{\mathcal{A}}(1^k)$中交互.

首先，由于模拟器Sim不知道给定实例中的s值，它需要对实验$Veri_{\mathcal{A}}(1^k)$进行修改，重新构造和s值相关的算法：

1）*KeyGen*算法中，模拟器Sim在不知s的情况下直接使用$(g,g^s,g^{s^2},\cdots,g^{s^q})$作为公开参数并发送给敌手$\mathcal{A}$.

2）*Setup*算法中，模拟器Sim利用$(g,g^s,g^{s^2},\cdots,g^{s^q})$计算$acc(f_{w_i})=g^{\prod_{f\in f_w}^{(s+f)}}$. 为了模拟更新令牌，模拟器$Sim$需要构造一个额外辅助结构$\mathcal{N}$，为每个$acc(f_{w_i})$存储一个多项式$n_w=\prod_{f\in f_w}(s+f)$.

3）*Update*协议中，模拟器Sim不能直接生成更新令牌τ_u，首先找到结构$\mathcal{N}$中与$\theta_{i,2}$相等的多项式n_i，然后利用$(g,g^s,g^{s^2},\cdots,g^{s^q})$计算$g^{n_i(s+f)}$，生成新的$\theta_2=g^{n_i(s+f)}$.

模拟器Sim将和s值相关的数值通过上述方式计算，然而由于修改的实验中算法的输出和实验$Veri_{\mathcal{A}}(1^k)$算法的输出是不可区分的，所以在敌手$\mathcal{A}$的视角中，新计算的数值仍然有效. 如果敌手$\mathcal{A}$赢得了游戏，那么意味着敌手$\mathcal{A}$攻破了双线性q-SDH假设，即成功伪造了子集证据$\mathcal{S}$或者成功伪造了完备性证据C.

1）伪造子集证据$\mathcal{S}$. 敌手输出错误的搜索结果$I'=\{f_1,f_2,\cdots,f_m\}$和伪造的子集证据$\mathcal{S}_i=\widehat{g^{P_i}}$. 令$S_i=\{f_1,f_2,\cdots,f_q\}$为包含关键字$w_i$的文件标识符集合，如果$I'\not\subset S_i$，那么一定存在某个$1\leqslant j\leqslant m$，使得$f_j\notin S_i$. 通过验证意味着

$$e(g^{\prod_{f\in I'}^{(s+f)}},\mathcal{S})=e\left(g^{(s+f_1)(s+f_2)\cdots(s+f_q)},g\right),$$

因为$f_j\notin S_i$，因此$(s+f_j)$并不能整除$(s+f_1)\times(s+f_2)\times\cdots\times(s+f_q)$. 所以存在一个次数为$q-1$的多项式$Q(s)$和一个常数$c$，$c\neq 0$，使得$(s+f_1)(s+f_2)\cdots(s+f_q)=Q(s)(s+f_j)+c$. 因此

$$e(g,\widehat{g^{P_i}})^{\prod_{f\in I'-f_j}^{(s+f)}(s+f_j)}=e(g,g)^{Q(s)(s+f_j)+c},$$

经过变形后，最终可以得到

$$e(g,g)^{\frac{1}{s+f_j}}=\left(e(g,\widehat{g^{P_i}})^{\prod_{f\in I'-f_j}^{(s+f)}}e(g,g)^{-Q(s)}\right)^{1/c}.$$

这说明模拟器Sim可以在多项式时间内解决双线性q-SDH问题，这违反了双线性q-SDH假设.

2）伪造完备性证据C. 敌手输出错误的交集搜索$\widetilde{R}_{\phi(\mathcal{W})}=\{f_1,f_2,\cdots,f_m\}$和一个完备性证据$\widehat{C}$. 因为$\widetilde{R}_{\phi(\mathcal{W})}$是$R$的真子集，那么多项式$P_1,P_2,\cdots,P_n$至少存在1个公因式. 令$(s+f)$为$P_1,P_2,\cdots,P_n$的一个公因式，其中$f\notin I'$. 通过验证算法意味着$\prod_{i=1}^{m}e(\mathcal{S},\widehat{C})=e(g,g)$成立. 通过计算$g^{P_i'}=(g^{P_i})^{1/(s+f)}$，将$(s+f)$从每个$P_i$中提取出来，即

$$\prod_{i=1}^{n}e\left(g^{P_i},g^{q_i}\right)=\prod_{i=1}^{n}e((g^{P_i'})^{s+f},g^{q_i})=\left(\prod_{i=1}^{n}e\left(g^{P_i'},g^{q_i}\right)\right)^{s+f}=e(g,g).$$

因此模拟器Sim可以很容易地计算$e(g,g)^{1/(s+f)}=\prod_{i=1}^{n}e\left(g^{P_i'},g^{q_i}\right)$，从而找到双线性$q$-SDH问题的一个实例的解，这同样违反了双线性$q$-SDH假设.

综上所述，如果敌手可以伪造出一个有效的证据，那么就存在一个模拟器Sim，可以攻破哈希函数H_3的抗碰撞性和双线性q-SDH假设的至少1个. 因此，对于所有概率多项式计算能力敌手$\mathcal{A}$来说，赢得实验$Veri_{\mathcal{A}}(1^k)$的概率为：

$$Pr[Forge_{\mathcal{A}}(1^k)=1]\leqslant negl(1^k),$$

其中$negl(1^k)$是一个以安全参数1^k为输入的可忽略函数. 因此VBSE-BI方案具有不可伪造性.

5 性能分析与实验

本节主要分析提出的VBSE-BI方案的性能. 首先从理论层面对方案的各个阶段的性能进行综合对比评估，之后通过设计实验方案的原型系统，对不同阶段的性能进行测试来得到实验结果.

5.1 代价分析

针对模型构建过程中的存储代价、计算代价以及通信代价3个维度分析方案性能. 设N表示关键字集

合的大小, M 为文件总个数, n 为布尔搜索请求中包含的关键字个数, m 为搜索结果中文件个数, d 为更新文件包含的关键字个数. 方案的代价分析如表 2 所示.

存储代价主要考虑云服务器的安全索引存储代价, 设 T 为长度为 N 的查找表, 索引的空间复杂度为 $O(N)$. 在计算代价方面, 构建搜索令牌的计算代价与搜索关键字个数有关, 时间复杂度为 $O(n)$. 服务器对搜索令牌进行搜索操作时的计算复杂度为 $O(mn)$, 进行交集运算的渐近运行时间为 $O(N\log^2 N\log\log N)$, 并集和差集云服务器的渐近运行时间复杂度为 $O(N\log N)$, 用户端完成验证操作的渐近运行时间复杂度为 $O(n)+O(m)$. 搜索通信代价主要由搜索关键字个数以及搜索结果中文件个数决定. 当用户想要搜索 n 个关键字的布尔集合时, 和服务器交互一次, 发送搜索令牌的通信复杂度为 $O(n)$, 云服务器返回搜索结果的通信复杂度为 $O(m)$, 因此通信复杂度为 $O(n+m)$. 与同类型的文献 [23] 方案相比, 本方案在支持更丰富的布尔查询语句的同时, 降低了服务器端索引的存储代价以及搜索、验证、更新的计算代价.

5.2 实验分析

本文对提出的方案进行了编程实现, 然后通过对算法的性能测试来分析方案在实际应用中的表现. 方案部署一个远程服务器来模拟云服务器, 处理器为 Intel Core i7-2600 Quad-Core Processor 3.4 GHz, 内存为 8 GB; 同时部署多台主机模拟用户端, 每台主机模拟为一个和云服务器交互的用户. 此外, 方案利用 Ethereum 仿真平台 Ropsten Testnets 模拟区块链存储平台, 具体来说, 利用 Ethereum 的 web3 API 包[26] 来构建区块链索引, 调用 Metamask[27] 创建本地钱包来实例化 $\sum_{BC}.Init()$ 算法, 调用 signTransaction 函数实例化 $\sum_{BC}.Put()$ 算法来存储链式索引结构中的节点信息; 调用 sendRawTransaction 函数接受已签名交易的同时输出交易的地址; 搜索时, 调用 getTransaction 函数来实例化 $\sum_{BC}.Get()$ 算法将交易的地址作为输入并输出关联的交易内容, 然后读取数据字段以检索链式索引中的节点信息. 本文方案调用 OpenSSL_SHA256-HMAC[28] 算法实现伪随机函数 F, G, P. 使用 Enron 邮件数据集[29] 来进行性能的评估, 从邮件集合中随机选择 1 000 个文件作为测试用的文件集.

根据系统实现的功能, 测试系统执行不同操作时的云服务器端和用户端的计算开销及通信开销, 其中服务器端计算开销主要测试的内容包括搜索时间、生成证明时间, 用户的计算开销主要测试验证时间, 通信开销主要测试云服务器返回搜索结果的数据量. 实验中每个数据均为实验重复 50 次并取平均值后得到的结果.

1)计算开销. 计算开销可以分为区块链索引搜索开销、服务器本地证明开销以及用户端本地验证开销, 如图 3 所示. 当搜索关键字数量为 2, 4, 6, 8, 10 时, 搜索区块链索引的时间开销与关键字个数大致呈线性关系(1.817～5.755 s). 证明过程耗时较搜索过程大, 原因为证明过程包含较多指数运算和双线性配对运算, 尽管随关键字数量变化时间开销相对稳定, 维持在 6.1～6.8 s, 与理论分析一致. 由于这些计算的效率与所选用的曲线参数和配对算法有很大关系, 通过多线程计算和预先计算, 可以显著降低证明阶段的计算时间. 因此随着参数和算法的不断优化, 其效率可以进一步得到提升. 此外, 服务器端的

Table 2 Cost Analysis

表 2 代价分析

复杂度类别	VBSE-BI 方案(本文)		文献 [23] 方案	
	用户端	服务器端	用户端	服务器端
$\mathrm{Stor}_{\mathrm{Index}}$	$O(1)$	$O(N)$	$O(1)$	$O(NM)+O(N)+O(M)$
$\mathrm{Comp}_{\mathrm{Search}}$	$O(n)$	$O(mn)$	$O(n)$	$O(n+mn)$
$\mathrm{Comp}_{\mathrm{ProveI}}$	-	$O(N\log^2 N\log\log N)$	-	$O(N\log^2 N\log\log N)$
$\mathrm{Comp}_{\mathrm{ProveU}}$	-	$O(N\log N)$	-	-
$\mathrm{Comp}_{\mathrm{ProveD}}$	-	$O(N\log N)$	-	-
$\mathrm{Comm}_{\mathrm{Search}}$	$O(n+m)$	$O(n+m)$	$O(n)$	$O(n\log n+m)$
$\mathrm{Comp}_{\mathrm{Verify}}$	$O(n+m)$	-	$O(n\log n+m)$	-
$\mathrm{Comp}_{\mathrm{Update}}$	$O(d)$	$O(d)$	$O(\log N+d)$	$O(d)$
$\mathrm{Comm}_{\mathrm{Update}}$	$O(d)$	$O(d)$	$O(d)$	$O(d)$

注: $\mathrm{Stor}_{\mathrm{Index}}$ 表示存储空间复杂度; $\mathrm{Comp}_{\mathrm{Search}}$ 表示搜索计算复杂度; $\mathrm{Comp}_{\mathrm{ProveI}}$ 表示交集证明计算复杂度; $\mathrm{Comp}_{\mathrm{ProveU}}$ 表示并集证明计算复杂度; $\mathrm{Comp}_{\mathrm{ProveD}}$ 表示差集证明计算复杂度; $\mathrm{Comm}_{\mathrm{Search}}$ 表示搜索通信复杂度; $\mathrm{Comp}_{\mathrm{Verify}}$ 表示验证计算复杂度; $\mathrm{Comp}_{\mathrm{Update}}$ 表示更新计算复杂度; $\mathrm{Comm}_{\mathrm{Update}}$ 表示更新通信复杂度; "-"表示无复杂度体现.

算法还可以进行一些实现方面的优化，从而可以大大降低证明阶段所花费的时间. 例如对集合证据使用多线程来计算，以及使用预先计算好的P_i和g^{P_i}等. 用户计算开销主要测试在不同数目关键字情况下进行搜索时用户完成验证操作的时间，开销与关键字个数无关，维持在 1.0～1.8 s 之间. 验证过程中耗时最多的操作为双线性配对过程，与服务器端的搜索和证明时间相比，用户端计算开销为服务器端计算开销的 9.98%～14.03%. 因此，VBSE-BI 方案在用户端的计算开销相对较低，验证操作高效快速，极大地减少了用户端的计算负担. 这使得 VBSE-BI 方案在实际应用中具有较好的用户体验，特别是在资源受限的移动设备上表现尤为突出.

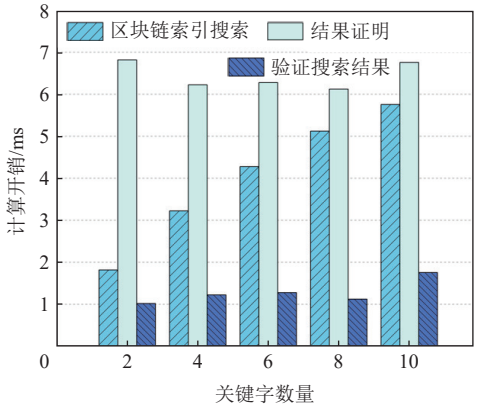


Fig. 3 Relationship between computational cost and the number of keywords

图 3 计算开销与关键字数量之间的关系

2)通信开销. 通信开销主要包含搜索结果的数据量和额外证明的数据量. 图 4 表明了数据量与搜索时布尔表达式中的关键字数量之间的关系. 当搜索时使用的关键字数量递增时，通信开销中搜索结果的数据量从 7 580 亿减少至 2 010 亿，这是因为符合多个关键字布尔表达式的搜索结果减少，传输的数据量也相应减少. 证明数据量从 5 020 亿增加至 18 960 亿，因为更多的关键字需要更多的证明数据来保证搜索结果的正确性和完整性，占总数据量的 36.97%～76.54%. 因此，额外证明的数据量是影响通信开销的主要因素，与理论分析一致. 因此，对于频繁查询的关键字，预先生成并存储其证明数据，减少实时计算和传输的开销，可以显著减少 VBSE-BI 方案中的通信开销，提高整体性能. 特别是在处理较多关键字的搜索时，通过优化证明算法和减少冗余数据，可以有效地控制通信开销的增长，从而提高方案在实际应用中的效率和可扩展性.

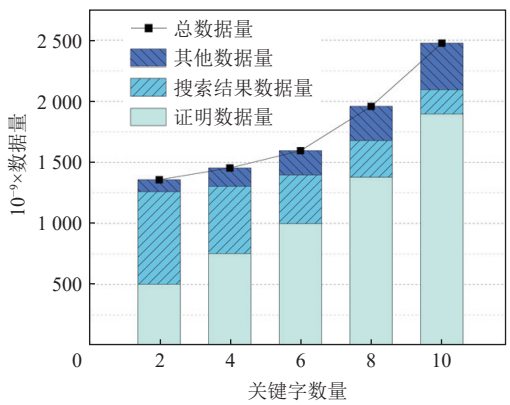


Fig. 4 Relationship between communication cost and the number of keywords

图 4 通信开销与关键字数量之间的关系

6 总 结

本文提出一种基于区块链索引的可验证布尔搜索加密方案. 设计基于区块链的索引构建算法和索引动态更新算法，并利用可证安全的对称加密和同态加密机制对索引进行加密，实现多关键字布尔搜索. 针对布尔搜索的完整性验证需求，给出布尔搜索结果不可伪造性定义，并提出基于双线性映射累加器和扩展欧几里得算法的布尔运算完整性验证算法. 安全性分析表明，本方案在随机预言模型下能够抵抗动态选择关键字攻击，并且在双线性阶强 Diffie-Hellman 假设下满足不可伪造性. 性能分析表明，本方案有效地降低了搜索时的通信代价以及用户端计算复杂度，对提升可搜索加密的可用性和确保用户享受的合法权益不受损害具有重要意义. 本文搜索协议在进行搜索结果证明时的计算代价具有提升空间，因此在未来的研究中重点关注此方面效率的提高. 除此之外，未来考虑使方案满足有隐私保护需求的更多密文搜索场景，例如基于兴趣的推荐服务以及云边协同隐私保护查询等.

作者贡献声明：李宇溪负责模型设计、方案设计、算法设计、方案分析、论文撰写及修改；陈晶晶负责安全性分析、实验环境部署、论文撰写及修改；周福才负责模型设计和技术路线可行性分析；徐剑负责理论基础、整体思路的确定；纪东负责实验分析和论文修改.

参 考 文 献

[1] Bost R. ∑οφος: Forward secure searchable encryption[C]//Proc of

the 2016 ACM SIGSAC Conf on Computer and Communications Security. New York: ACM, 2016: 1143−1154

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

[3] Li Yuxi, Zhou Fucai, Xu Jian, et al. Multiple-keyword encrypted search with relevance ranking on dual-server model[J]. Journal of Computer Research and Development, 2018, 55(10): 2149−2163 (in Chinese)
(李宇溪, 周福才, 徐剑, 等. 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[J]. 计算机研究与发展, 2018, 55(10): 2149−2163)

[4] Othman S B, Trad A, Youssef H, et al. Secure data aggregation with MAC authentication in wireless sensor networks[C]//Proc of the 12th IEEE Int Conf on Trust, Security and Privacy in Computing and Communications. Piscataway, NJ: IEEE, 2013: 188−195

[5] Li Hongwei, Lu Rongxing, Zhou Liang, et al. An efficient Merkle-tree-based authentication scheme for smart grid[J]. IEEE Systems Journal, 2013, 8(2): 655−663

[6] Luo Lailong, Guo Deke, Ma R T. B, et al. Optimizing Bloom filter: Challenges, solutions, and comparisons[J]. IEEE Communications Surveys & Tutorials, 2018, 21(2): 1912−1949

[7] Shi Zhenkui, Fu Xuemei, Li Xianxian, et al. ESVSSE: Enabling efficient, secure, verifiable searchable symmetric encryption[J]. IEEE Transactions on Knowledge and Data Engineering, 2020, 34(7): 3241−3254

[8] McConaghy T, Marques R, Müller A, et al. BigchainDB: A scalable blockchain database [EB/OL]. 2016: 53−72. [2024-07-20]. https://gamma.bigchaindb.com/whitepaper/bigchaindb-whitepaper.pdf

[9] Aslam T, Maqbool A, Akhtar M, et al. Blockchain based enhanced ERP transaction integrity architecture and PoET consensus[J]. Computers, Materials & Continua, 2022, 70(1): 1089−1109

[10] Nóbrega T, Pires C E S, Nascimento D C. Blockchain-based privacy-preserving record linkage: Enhancing data privacy in an untrusted environment[J]. Information Systems, 2021, 102: 101826

[11] Piao Chunhui, Hao Yurong, Yan Jiaqi, et al. Privacy preserving in blockchain-based government data sharing: A service-on-chain (SOC) approach[J]. Information Processing & Management, 2021, 58(5): 102651

[12] Chen Lanxiang, Lee W, Chang C, et al. Blockchain based searchable encryption for electronic health record sharing[J]. Future Generation Computer Systems, 2019, 95: 420−429

[13] Niu Shufen, Xie Yaya, Yang Pingping, et al. Cloud-assisted attribute-based searchable encryption scheme on blockchain[J]. Journal of Computer Research and Development, 2021, 58(4): 811−821 (in Chinese)
(牛淑芬, 谢亚亚, 杨平平, 等. 区块链上基于云辅助的属性基可搜索加密方案[J]. 计算机研究与发展, 2021, 58(4): 811−821)

[14] Tang Qiang. Towards blockchain-enabled searchable encryption [C]//Proc of the 21st ICICS Int Conf on Information and Communications Security. Berlin: Springer, 2020: 482−500

[15] Hahn F, Kerschbaum F. Searchable encryption with secure and efficient updates[C]//Proc of the 2014 ACM SIGSAC Conf on Computer and Communications Security. New York: ACM, 2014: 310−320

[16] Zuo Cong, Sun Shifeng, Liu J K, et al. Dynamic searchable symmetric encryption schemes supporting range queries with forward (and backward) security[C]//Proc of the 23rd European Symp on Research in Computer Security. Berlin: Springer, 2018: 228−246

[17] Chai Qi, Gong Guang. Verifiable symmetric searchable encryption for semi-honest-but-curious cloud servers[C]//Proc of the 2012 IEEE Int Conf on Communications (ICC). Piscataway, NJ: IEEE, 2012: 917−922

[18] Wang Jie, Yu Xiao, Zhao Ming. Fault-tolerant verifiable keyword symmetric searchable encryption in hybrid cloud[J]. International Journal of Network Security, 2015, 17(4): 471−483

[19] Kurosawa K, Ohtaki Y. How to update documents verifiably in searchable symmetric encryption[C]//Proc of the 12th Int Conf on Cryptology and Network Security. Berlin: Springer, 2013: 309−328

[20] Ameri M H, Assar M R, Mohajeri J, et al. A generic construction for verifiable attribute-based keyword search schemes[J]. Cryptology ePrint Archive, 2015

[21] Jiang Xiuxiu, Yu Jia, Yan Jingbo, et al. Enabling efficient and verifiable multi-keyword ranked searchover encrypted cloud data[J]. Information Sciences, 2017, 403: 22−41

[22] Wang Jianfeng, Chen Xiaofeng, Sun Shifeng, et al. Towards efficient verifiable conjunctive keyword search for large encrypted database[C]// Proc of the 23rd European Symp on Research in Computer Security. Berlin: Springer, 2018: 83−100

[23] Yan Xixi, Yuan Xiaohan, Tang Yongli, et al. Verifiable attribute-based searchable encryption scheme based on blockchain[J]. Journal on Communication, 2020, 41(2): 187−198 (in Chinese)
(闫玺玺, 原笑含, 汤永利, 等. 基于区块链且支持验证的属性基搜索加密方案[J]. 通信学报, 2020, 41(2): 187−198)

[24] Najafi A, Javadi H H S, Bayat M. Efficient and dynamic verifiable multi-keyword searchable symmetric encryption with full security[J]. Multimedia Tools and Applications, 2021, 80(17): 26049−26068

[25] Xie Mande, Yang Xuekang, Hong Haibo, et al. A novel verifiable chinese multi-keyword fuzzy rank searchable encryption scheme in cloud environments[J]. Future Generation Computer Systems, 2024, 153: 287−300

[26] Ethereum Docs. Ethereum[EB/OL]. 2015[2024-07-27]. https://www.ethereum.org/

[27] ConsenSys Software Inc. Metamask[EB/OL]. 2020[2024-07-20]. https://metamask.io/

[28] The OpenSSL Project. OpenSSL: The open source toolkit for SSL/TLS[EB/OL]. 2015[2024-06-17]. http://www.openssl.org/

[29] Cohen W W. Enron email dataset[EB/OL]. 2015[2024-06-01]. https://www.cs.cmu.edu/~enron/



Li Yuxi, born in 1990. PhD, lecturer. Member of CCF. Her main research interest includes secure cloud storage.

李宇溪, 1990 年生. 博士, 讲师. CCF 会员. 主要研究方向为安全云存储.



Chen Jingjing, born in 1999. Master candidate. Her main research interest includes spatial data security. (chenjingjing5051@163.com)

陈晶晶, 1999 年生. 硕士研究生. 主要研究方向为空间数据安全.



Zhuo Fucai, born in 1964. PhD, professor, PhD supervisor. Senior member of CCF. His main research interests include cryptography and network security, trusted computing, and critical technologies in electronic commerence. (fczhou@mail.neu.edu.cn)

周福才, 1964 年生. 博士, 教授, 博士生导师. CCF 高级会员. 主要研究方向为密码学与网络安全、可信计算、电子商务关键技术.



Xu Jian, born in 1978. PhD, professor and PhD supervisor. Senior member of CCF. His research interests include cryptography and network security. (jianxu@mail.neu.edu.cn)

徐　剑, 1978 年生. 博士, 教授, 博士生导师. CCF 高级会员. 主要研究方向为密码学、网络安全.



Ji Dong, born in 1989. PhD. His main research interests include timing analysis in embedded systems, and machine learning and its various applications in renewable energy.

纪　东, 1989 年生. 博士. 主要研究方向为嵌入式系统时序分析、机器学习在可再生能源领域的应用.