

基于区块链的数据完整性多方高效审计机制

周家顺, 王娜, 杜学绘

(信息工程大学, 河南 郑州 450001)

摘要: 提出了一种面向大数据环境的基于区块链的数据完整性多方高效审计机制(MBE-ADI)。构建基于数据域的混合 Merkle DAG 结构对数据组织, 实现大数据环境下大量非结构化数据同时验证; 为应对大数据环境下数据量大的问题, 设计基于 BLS 签名多副本确定性验证方法, 实现支持多副本的数据完整性高效验证; 设计基于联盟链的双验证审计架构, 实现去中心化自动审计以及审计历史可信追溯, 同时为数据拥有者和数据使用者提供数据完整性验证服务, 实现数据的可信获取。测试证明系统的可行性以及数据完整性审计的高效性。

关键词: 区块链; 大数据; 完整性审计; Merkle DAG; 数据共享

中图分类号: TP309.2

文献标识码: A

DOI: 10.11959/j.issn.2096-109x.2021107

Multi-party efficient audit mechanism for data integrity based on blockchain

ZHOU Jiashun, WANG Na, DU Xuehui

Information Engineering University, Zhengzhou 450001, China

Abstract: A blockchain-based data integrity multi-party high-efficiency audit mechanism (MBE-ADI) for the big data environment was proposed. A hybrid Merkle DAG structure based on data domain was built to organize data, and realized the simultaneous verification of a large number of unstructured data in the big data environment. In order to deal with the problem of large amount of data in the big data environment, a multi copy deterministic verification method based on BLS signature was designed to realize the efficient verification of data integrity supporting multiple copies. A dual verification audit architecture based on consortium blockchain was designed to realize decentralized automatic audit and audit history credible traceability. At the same time, data integrity verification services were provided for data owners and data users to realize reliable acquisition of data. The test proves the feasibility of the system and the efficiency of data integrity audit.

Keywords: blockchain, big data, integrity audit, Merkle DAG, data sharing

收稿日期: 2021-09-06; 修回日期: 2021-11-05

基金项目: 国家重点研发计划(2018YFB0803603); 国家自然科学基金(61802436)

Foundation Items: The National Key R&D Program of China (2018YFB0803603), The National Natural Science Foundation of China (61802436)

论文引用格式: 周家顺, 王娜, 杜学绘. 基于区块链的数据完整性多方高效审计机制[J]. 网络与信息安全学报, 2021, 7(6): 113-125.

ZHOU J S, WANG N, DU X H. Multi-party efficient audit mechanism for data integrity based on blockchain[J]. Chinese Journal of Network and Information Security, 2021, 7(6): 113-125.

1 引言

大数据技术高速发展,并深入人们的生活。政务大数据^[1]、司法大数据^[2]、医疗大数据^[3]等大数据技术正在逐渐为社会的进步提供有力的数据技术支撑。但是大数据具有与传统数据不同的特点,需要使用适合大数据特点的技术进行处理。

大数据最显著的特点就是数据量大,据 Statista 公司预测 2025 年的全球数据量将达到 175 ZB。数据量的激增促使相应的存储技术的发展,本地存储、分布式存储、云存储等存储技术为大数据存储提供了技术支撑。为保证大数据存储的安全,需要进行数据完整性验证^[4]判断数据是否被篡改或者损坏。

数据完整性验证技术一般通过挑战应答机制实现数据完整性的判断。数据拥有者将数据存储在数据存储系统中,在数据进行存储之前利用 BLS 签名等技术实现存储数据的签名,并生成证明元数据。数据拥有者从证明元数据中选取相应的数据生成挑战,并将挑战发送给数据存储系统。数据存储系统依据收到的挑战,利用存储的数据生成相应的数据完整性的证据。数据完整性审计机构对数据存储系统生成的证据进行判断,确定数据是否与原始数据一致。

对于大数据进行数据完整性审计需要充分考虑大数据的特点。

1) 大数据数据量大的特点要求数据完整性验证技术能够满足审计的效率要求,实现数据完整性的高效判决。

2) 由于大数据的来源广泛,大数据的数据类型变得多样,数据可以分为结构化、半结构化和非结构化 3 种类型^[5]。例如,政务大数据中的数据类型可以分为 4 类:业务数据、民意社情数据、环境数据和分散性公共数据^[6]。大数据环境下的数据主要为非结构化数据,据 IDC 统计显示,大数据中非结构化数据占比 80%,非结构化数据完整性验证成为大数据完整性验证的关键。

3) 同时,信息化时代,数据成为越来越重要的资源。通过共享大数据来打破数据孤岛,成为信息企业面临的重要挑战。数据共享不仅需要实现数据交换,也要保证数据的真实性。例如,对

于政务大数据、司法大数据和医疗大数据这些较为敏感的数据,数据真实性的重要程度表现极为突出。

大数据的完整性验证不仅需要向数据拥有者提供数据未被篡改或者未被破坏的证明,对于共享数据的使用者同样需要进行证明。

中本聪发表文章 *Bitcoin: a peer to peer electronic cash system*^[7],标志着区块链技术的诞生。从此之后,区块链技术以其去中心化,不易篡改等良好特性吸引大批学者研究。数据的完整性验证技术实际上对于数据存储者等多方信用的验证,将区块链技术与数据完整性验证技术结合能够发挥数据完整性验证体系更大的价值。

基于此,本文提出基于区块链的数据完整性多方高效审计机制(MBE-ADI),解决大数据环境下数据完整性的审计问题,主要贡献如下。

1) 提出大数据环境中数据域的概念并构建基于数据域的混合 Merkle DAG 结构,实现对非结构化数据的管理,借助此结构实现证明元数据的生成,解决大数据环境下大量非结构化数据同时验证的问题。

2) 设计基于 BLS 签名多副本确定性验证方法,实现数据完整性的多副本同时确定性验证,满足大数据环境下数据完整性高效验证的需求。

3) 设计基于联盟链的双验证审计架构,相应智能合约以及验证过程元数据上传方法,实现数据完整性去中心化自动审计以及审计历史可信追溯,同时为数据拥有者和数据使用者提供数据完整性验证服务,保证数据进行共享之前的历史一致性,提高数据的可信度。

4) 基于阿里云服务器进行 MBE-ADI 系统部署并进行相关测试,验证系统的可行性以及数据完整性审计的高效性。

2 研究现状

Deswarte 等^[8]提出基于 MAC 码的完整性验证机制,该机制将 MAC 值作为认证元数据实现数据的完整性验证,但存在通信开销大且容易泄露隐私的问题。Ateniese 等^[9]提出数据持有性证明(PDP, provable data possession)机制,将数据分块并使用 RSA 签名机制进行抽样检测数据块的

完整性,提高了检测效率,减少了通信开销。Wang等^[10]提出支持全同态操作的PDP机制,该机制使用Merkle树验证数据块位置的正确性,采用BLS签名对数据块的完整性进行验证。谢四江等^[11]提出使用多分支路径树MBT进行完整性验证的机制,该机制增加了节点的出度,相较于基于Merkle树的完整性验证机制能够验证更大规模的数据,且利用MBT的结构能够较好地实现数据块替换等动态操作。

多副本机制能够提高数据的抗风险能力,即时使用多副本对损坏数据进行修复,对于重要的数据,多副本机制重要性更为突出。刘洪宇等^[12]提出支持动态操作的多副本验证机制,该机制对Merkle树结构进行改造,提出基于等级的Merkle树,实现对动态验证的支持。该机制通过对多个副本进行关联,实现多副本的同步更新。Curtmola等^[13]通过随机掩码技术实现多副本数据的生成,为任意数量的副本使用恒定数量的元数据,可以动态创建新副本而无需再次预处理数据,且多副本完整性验证的时间和单副本数据的验证时间接近。但是多副本机制存在生成的随机数等元数据过多、处理大数据量的文件时会出现元数据管理负担过重的问题,不适合数据量大且包含大量非结构化数据的大数据环境。

以上的数据完整性验证的审计主要使用可信第三方机构,但可信第三方机构寻找困难,容易发生第三方攻击的问题。应用区块链技术进行数据的完整性验证成为新的选择^[14-21]。

Liu等^[18]采用区块链智能合约取代第三方审计机构,并认为在共享数据之前,数据使用者应该对数据的完整性进行验证。为了实现公平的完整性审计,Zhao等^[19]考虑使用区块链技术进行数据完整性的验证。数据拥有者将数据块的签名上传到区块链账本,将加密后的数据上传到云上,验证的时候对数据进行下载,利用区块链账本记录的数字签名对数据的完整性进行验证。Wei等^[20]通过虚拟代理机制实现基于区块链的数据完整性验证,并结合基于角色的访问控制技术对存储的数据进行管控。魏艳等^[21]提出基于以太坊^[22]的数据完整性验证机制,该机制在智能合约中存储数据的哈希值以及数据签名等信息,在进行数据完

整性验证的时候,将现有数据的哈希与智能合约中存储的哈希进行对比实现验证。目前基于区块链的数据完整性验证机制没有考虑数据使用者获取共享数据真实性的需求,仅对数据拥有者提供服务。

目前有学者注意到对于大数据进行完整性验证的问题。Chang等^[23]对外包大数据完整性验证技术进行了总结,但与谭霜等^[4]提出的技术基本一致,未体现大数据完整性验证的特点。Chen等^[24]实现数据块的细粒度更新,采用平衡更新树作为ADS(authenticated data structure)来减少动态更新后的更新验证,从而减少计算和通信资源。Nasonov等^[25]提出基于区块链实现数据交易完整性的分布式大数据平台,重点设计完整性管理器模块,确保数据的真实性与一致性。Lebdaoui等^[26]考虑到大数据来源广泛以及大数据数据量大的特点,并提出数据输入验证模型实现对数据来源的验证,提出连续完整性监测模型实现数据在使用过程中的完整性验证,但仅提出了模型的框架。

综上所述可得,目前仍需要对适合大数据环境的数据完整性验证机制进行研究,对大数据环境下数据量大、包含大量非结构化数据和倾向共享等特点进行充分考虑。

3 基于数据域的混合Merkle DAG结构

由于大数据环境中的数据来源广泛,且大多为结构不一的非结构化数据(例如,数据拥有者获得的一组数据中可能同时包含图像、视频、文档等),实现对这些非结构化数据的有效组织是进行高效验证的前提。本节针对大数据环境数据中的特点,为实现对数据的有效管理,并在此基础上生成数据完整性审计的证明元数据,提出基于数据域的混合Merkle DAG结构。在数据域层面,使用Merkle DAG结构构造非结构化数据间的组织关系;在数据块层面,对单个数据的数据分块构造多分支平衡Merkle树。

本节提出数据域的概念,借助此概念实现对非结构化数据进行组织。这里的域是指对一类关联的数据或者子数据域的包含。对于一批需要进行存储的数据,数据拥有者根据数据的内在联系,

(如数据的来源、获取日期、类别等)关系,对数据进行划分,将一类数据归置于一个域中。从而获得一个包含所有数据和子数据域的最大域。

在该存储结构中使用 Merkle DAG。在 Merkle 树的基础上构建 Merkle DAG,打破了 Merkle 树的子节点个数的限制,无须进行数据的平衡操作,能够根据实际的需要构建更为灵活的数据结构。Merkle DAG 保留 Merkle 树循环计算节点哈希获得 Merkle Root 的特点,父节点的哈希值由子节点的哈希值决定,同时父节点包含指向子节点的信息。在 IPFS^[27]中将 Merkle DAG 作为数据的存储结构,现实分布式文件存储网络。

构建基于数据域的混合 Merkle DAG 结构的过程如下:

- 1) 根据非结构化数据的包含与并列关系,构建包含所有数据的 Merkle DAG 文件结构;
- 2) 对域内的每条数据构建多分支平衡 Merkle 树结构,获取 Merkle DAG 节点信息中的 $node_{id}$ 。

3.1 构建基于数据域的 Merkle DAG 文件结构

针对非结构化数据构建数据域,相关联的数据放在一个域内,数据域中包含子数据域,各级数据域表示数据不同的关联程度,域中同时存放相关联的多条数据,并规定域内至少包含两个数据文件。如图 1 所示,数据域为 $\{A, A_1, A_2\}$ 。 A 域包含 $\{A_1, A_2, d_7, d_8, d_9\}$ 。 A_1 域包含 $\{d_1, d_2\}$ 。 A_2 域包含 $\{d_3, d_4, d_5, d_6\}$ 。

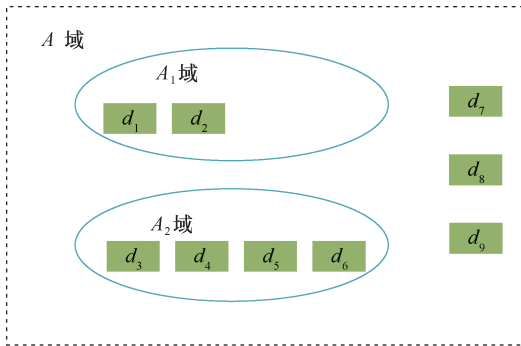


图 1 数据域结构
Figure 1 Data field structure

基于数据域的混合 Merkle DAG 结构包含域节点和数据节点两种节点。为每个域构建域节点标识该域,域节点如图 2 所示, $node_{id}$ 为该域

点的唯一标识信息,能够使用 $node_{id}$ 区分节点; L_r 为右指针,指向同级域中的其他节点; $\{Id_1, Id_2, \dots, Id_i\}$ 为子指针,指向数据节点或者子数据域节点。

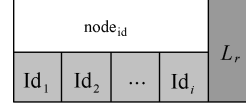


图 2 基于数据域的 Merkle DAG 结构域节点信息
Figure 2 Domain-based Merkle DAG structure domain node information

在基于数据域的混合 Merkle DAG 结构中,每个非结构化数据由一个数据节点标识,数据节点如图 3 所示, $node_{id}$ 为该数据节点的唯一标识信息,能够使用 $node_{id}$ 区分数据节点; L_r 为右指针,指向同级域中的数据节点。对图 1 的数据域可以构建图 4 所示的 Merkle DAG 文件结构。



图 3 基于数据域的 Merkle DAG 结构化数据节点信息
Figure 3 Domain-based Merkle DAG structure data node information

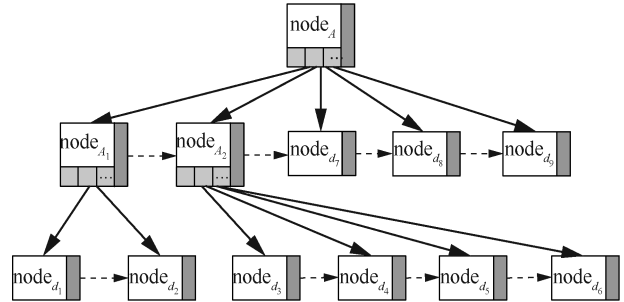


图 4 基于数据域的 Merkle DAG 文件结构
Figure 4 Merkle DAG file structure based on data domain

3.2 获取 Merkle DAG 节点 $node_{id}$ 信息

构建基于数据域的 Merkle DAG 文件结构之后,需要获取各个域节点或者数据节点中的 $node_{id}$ 。本文通过对非结构化数据构建多分支 Merkle 树结构获取标识不同节点的 $node_{id}$ 。设非叶子节点最大的子节点的个数为 N_{max} ;数据块分割大小恒定为 D 。

构建多分支 Merkle 树结构流程如下:

- 1) 数据分片。依据数据块的分片大小对数据文件进行分片,分片之后剩余不足分片大小的数据充当最后一个数据块。设将文件 D_f 分成了 n

块, 则 D_f 可以表示为 $D_f = \{m_1, m_2, \dots, m_n\}$ 。

2) 计算多分支平衡 Merkle 树第一层节点哈希。对数据块 $\{m_1, m_2, \dots, m_n\}$ 分别计算哈希, 获得多分支平衡 Merkle 树第一层 (i 为 1) 哈希 $\{h(m_1), h(m_2), \dots, h(m_n)\}$ 。

3) 对哈希节点分组。依据 $N_{\max}$ 对平衡 Merkle 树的第 i 层哈希层进行分组, 每组的个数为 $N_{\max}$ 。如果最后一组的个数不足 $N_{\max}$, 则将剩余的数据哈希归为一组。

4) 计算 $i+1$ 层哈希。对各组进行 $i+1$ 层哈希节点的计算: $h_i = h(h_1 \parallel h_2 \parallel \dots \parallel h_{N_{\max}})$, h_i 表示 $i+1$ 层节点, $\{h_1, h_2, \dots, h_{N_{\max}}\}$ 表示 i 层某组节点, 各组获得的哈希节点组成平衡 Merkle 树第 $i+1$ 层哈希节点。

5) 循环计算获得新的哈希层。对步骤 3) 和步骤 4) 进行循环, 直到某层哈希节点的个数为 1, 将该哈希节点作为该条数据的 Merkle Root, 计为 node_{d_i} 。

$N_{\max}$ 取 3, 数据 d_1 可以构建如图 5 所示的多分支平衡 Merkle 树结构。该结构能够满足大数据量的表示。设非叶子节点的最大子节点的个数为 $N_{\max}$, 数据块分割大小为 D , 设哈希层数为 k , 则该结构可以表示的数据量为 $N_{\max}^{k-1} \cdot D$ 。当 $N_{\max}$ 取 27, D 取 24 MB, k 取 4, 则表示的数据量为 461 GB。

获得数据 node_{d_i} 之后, 利用域中的各个数据节点的 node_{id} 计算数据域节点的 node_{id} 。例如, node_{A_1} 表示为 $h(\text{node}_{d_1} \parallel \text{node}_{d_2})$ 。获得域节点和数据节点的 node_{id} 信息之后, 便可获得完整的基于

数据域的混合 Merkle DAG 结构。完整的基于数据域的混合 Merkle DAG 结构如图 6 所示。

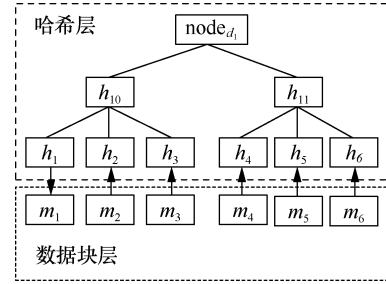


图5 数据 d_1 的多分支平衡 Merkle 树结构

Figure 5 Multi-branch balanced Merkle tree structure of data d_1

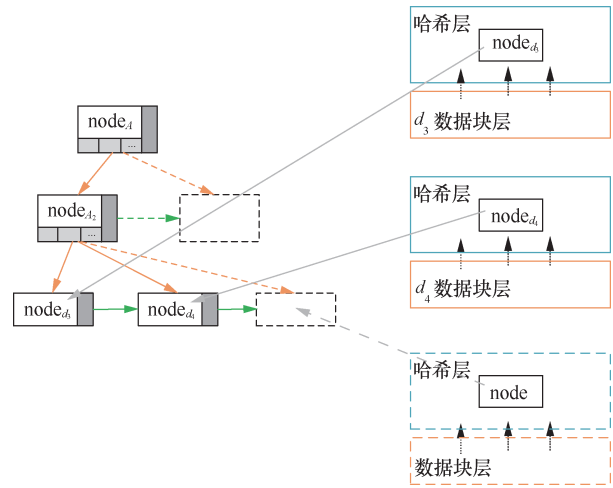


图6 完整的基于数据域的混合 Merkle DAG 结构

Figure 6 Integrate hybrid Merkle DAG structure based on data domain

当数据域中的非结构化数据的数据量小于构建多分支 Merkle 树的分片大小时, 非结构化数据不经过分片直接计算数据的哈希, $\text{node}_{id} = h(d)$ 。图 7 表示非结构化数据较小时, 基于数据域的混合 Merkle DAG 结构, 其构造简单, 能够对数据

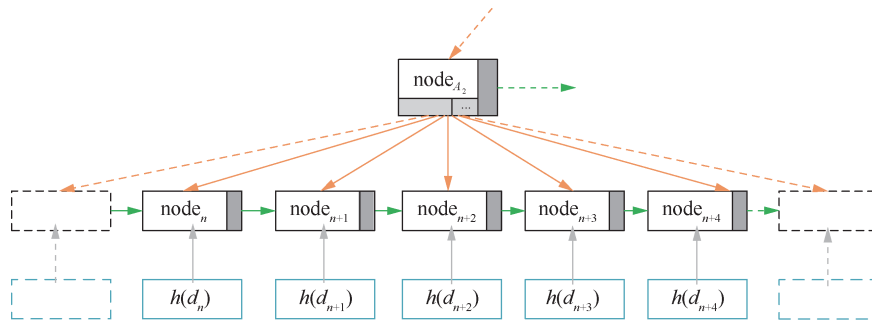


图7 数据较小时, 基于数据域的混合 Merkle DAG 结构

Figure 7 Hybrid Merkle DAG structure based on data domain for small data

个数多但数据量小的非结构化数据进行有效组织,并有利于在此基础上生成完整性审计的证明元数据。

4 方案设计

4.1 MBE-ADI 系统

本文考虑为某类数据相关的各方建立联盟链系统。数据所有者、数据使用者与数据存储系统均会加入区块链系统并且成为区块链系统中的节点。基于联盟链的双验证审计架构如图8所示,DO表示数据所有者,DU表示数据使用者,SP表示数据存储系统。

各个参与方在系统中包含链下基础设施部分

和链上区块链节点部分。链下基础设施包含存储设施和计算设施,完成审计流程主要的存储和计算任务。系统中通过区块链网络连接参与的所有节点,完整性审计的流程通过区块链的元数据上传合约、完整性判决合约和审计历史索引合约实现,审计过程的关键数据通过区块链账本进行记录。

在本文方案中,区块链部分通过 Hyperledger Fabric 平台实现。Hyperledger Fabric 是由 Linux 基金会牵头研发的面向企业应用的联盟链平台^[28]。Hyperledger Fabric 中的账本分为世界状态和区块链两部分。世界状态以键值对的方式存储账本状态的当前值,相较于遍历整个交易日志能够获得

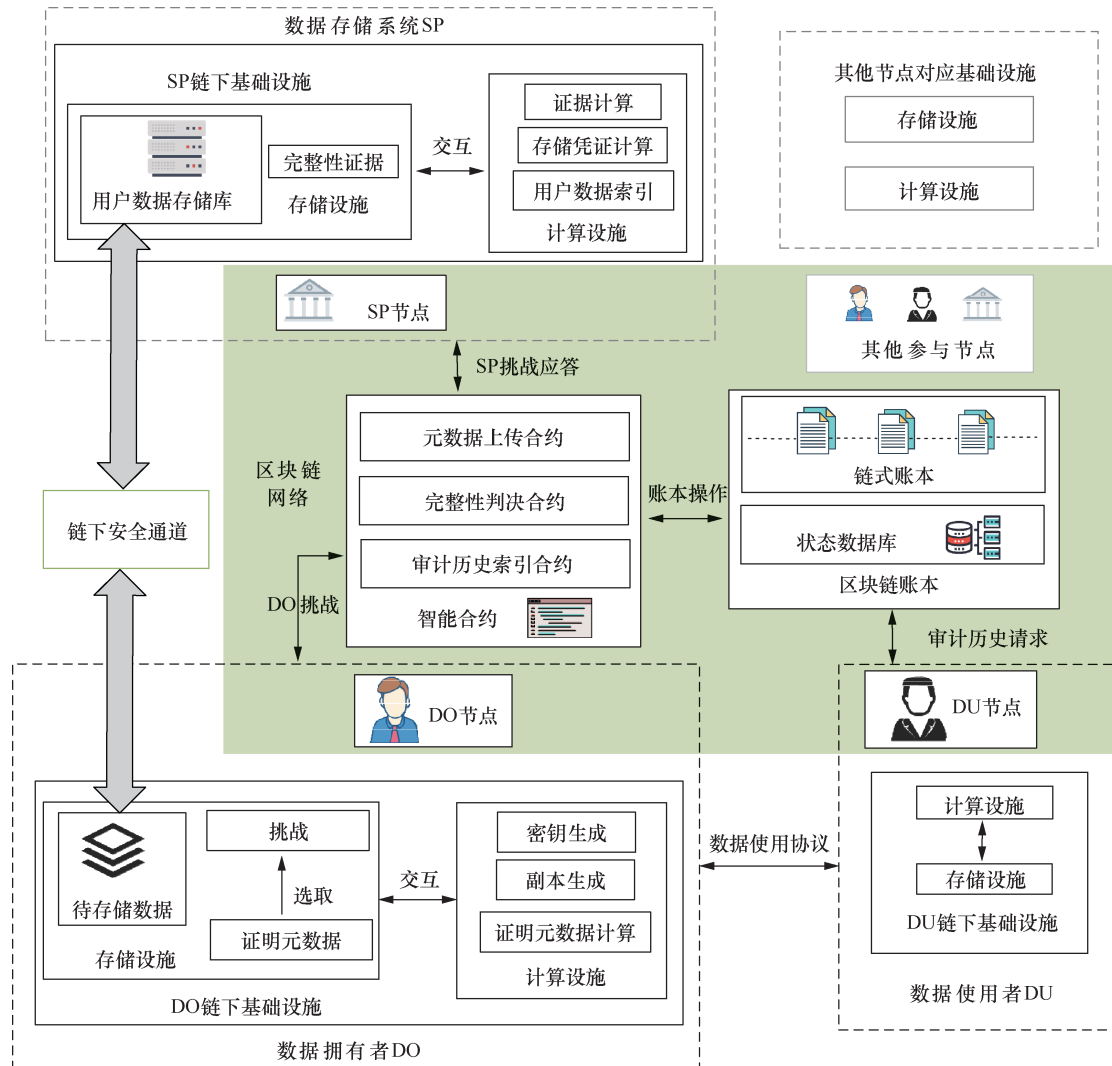


图8 基于联盟链的双验证审计架构
Figure 8 Dual verification audit architecture based on consortium blockchain

更快的访问速度。区块链是交易日志，它记录了区块链账本的完整数据，世界状态的改变由这些交易日志决定。当链上数据的键唯一的时候，世界状态的值就不会被改变，能够稳定记录数据的键值对。

对某个数据域的完整性审计的所有历史记录会在区块链账本上保存，通过检索并审查审计历史实现对数据域历史可信度的增强。为保证对数据完整性审计历史检索的速度，本文对状态数据库进行检索。为实现多类型索引查询操作，状态数据库采用 CouchDB 数据库。各个流程相关信息以 JSON 格式进行存储，存储的数据可以表示为：

```
type Integrity struct {
    ObjectType      string `json:"docType"`
    Timestamp       string `json:"timestamp"`
    Node_block      string `json:"node_block"`
    Data1           string `json:"data1"`
    Data2           string `json:"data2"`
}
```

ObjectType 标识数据类型，分为 data_up、challenge、prove、audit，分别代表上链数据、挑战数据、证明数据、审计数据。时间戳 Timestamp 作为该条状态数据的唯一标识，由区块链智能合约自动生成。Node_block 为需要进行数据完整性审计的数据域 df 节点标识，对于审计数据域，构建基于数据域的混合 Merkle DAG 结构，获得 Node_block=node_{df}。data1 和 data2 为其他信息，由不同的数据类型决定。

本文的完整性验证方案通过 BLS 签名实现。BLS^[29]为数字签名算法，其签名长度较 RSA 和 DSA 签名方案更短。

DH 四元组定义：G 为阶为素数 q 的循环群， $D=(g_1, g_2, u_1, u_2) \in G^4$ 为四元组，其中 g_1, g_2 为 G 的生成元， $u_1 = g_1^\alpha$ ， $u_2 = g_2^\beta$ ($\alpha, \beta \in \mathbb{Z}_q$)。判断 $\alpha = \beta$ 是否成立，如果成立，则称 D 为 DH 四元组。

设 G 为循环群，g 为 G 的生成元， $H: \{0,1\}^* \rightarrow G$ 为全域哈希函数。随机整数 x 为私钥， $y = g^x$ 为公钥。 $M \in \{0,1\}^*$ 为待签名的消息， $h = H(M)$ ，则 $\sigma = h^x \in G$ 为消息 M 的签名。如果四元组 (g, h, y, σ) 为 DH 四元组，则签名判断为

True，否则签名判断为 False。

4.2 基于 BLS 签名多副本确定性验证方法

4.2.1 初始化阶段

对于数据的加密，本文选用 AES 对称加密。该阶段生成数据加密所需的 AES 加密密钥以及 BLS 签名所需的公钥和私钥。该阶段在每个数据拥有者链下计算设施中分别进行。

设需要产生 μ 个数据域副本，生成 μ 个 AES 加密密钥 $\{K_1, K_2, \dots, K_\mu\}$ ，密钥长度为 k bit。g 为生成元，随机选择 $\mathbb{Z}_p^* \leftarrow x$ ，令 x 为数据拥有者 DO 的 BLS 签名私钥，数据拥有者 DO 的 BLS 签名公钥 y 满足 $y = g^x$ 。启动区块链网络，在区块链各个节点部署智能合约。

4.2.2 证明元数据生成阶段

数据拥有者使用 AES 加密密钥 $\{K_1, K_2, \dots, K_\mu\}$ 分别对数据域 df 中每个数据进行加密，获得 μ 个数据域副本 $\{df_1, df_2, \dots, df_\mu\}$ 。为实现对多个副本完整性的同时验证，对多副本拼接获得复合数据域 DF，对 DF 进行数据完整性验证实现对多副本的同时验证。获得数据域证明元数据流程如下。

1) 获取复合数据域 DF。获取复合数据域需要对数据域中的数据进行逐个复合，以实现对多副本数据完整性的同时验证。设原始数据域中的某个数据为 d，数据域副本的个数为 3，则数据 d 有 3 个数据副本 $\{d^1, d^2, d^3\}$ ，数据副本被划分成 n 个数据块，表示为 $d^j = \{m_1^j, m_2^j, \dots, m_n^j\}$ ， $j \in (1, 2, 3)$ 。数据拥有者确定生成证明元数据数量为 k，并使用伪随机数发生器生成 k 个随机序列： $L = \{l_1, l_2, \dots, l_k\}$ ，并为每个随机序列生成一个随机序列拼接策略 $S = \{s_1, s_2, \dots, s_k\}$ 。设随机序列 l_i 对应的拼接策略是 $s_i: \{m_{d^1} \parallel l_i \parallel m_{d^2} \parallel m_{d^3}\}$ ，其中 m_{d^1} 、 m_{d^2} 、 m_{d^3} 分别表示数据副本 d^1 、 d^2 、 d^3 的数据块，不同的拼接策略规定随机序列插入的位置不同。则数据 d 的复合数据 D 可以表示为 $M_t = m_1^1 \parallel l_i \parallel m_2^1 \parallel m_3^1 \parallel m_4^1 \parallel m_5^1 \parallel m_6^1 \parallel m_7^1 \parallel m_8^1 \parallel m_9^1 \parallel m_{10}^1 \parallel m_{11}^1 \parallel m_{12}^1 \parallel m_{13}^1 \parallel m_{14}^1 \parallel m_{15}^1 \parallel m_{16}^1 \parallel m_{17}^1 \parallel m_{18}^1 \parallel m_{19}^1 \parallel m_{20}^1 \parallel m_{21}^1 \parallel m_{22}^1 \parallel m_{23}^1 \parallel m_{24}^1 \parallel m_{25}^1 \parallel m_{26}^1 \parallel m_{27}^1 \parallel m_{28}^1 \parallel m_{29}^1 \parallel m_{30}^1 \parallel m_{31}^1 \parallel m_{32}^1 \parallel m_{33}^1 \parallel m_{34}^1 \parallel m_{35}^1 \parallel m_{36}^1 \parallel m_{37}^1 \parallel m_{38}^1 \parallel m_{39}^1 \parallel m_{40}^1 \parallel m_{41}^1 \parallel m_{42}^1 \parallel m_{43}^1 \parallel m_{44}^1 \parallel m_{45}^1 \parallel m_{46}^1 \parallel m_{47}^1 \parallel m_{48}^1 \parallel m_{49}^1 \parallel m_{50}^1 \parallel m_{51}^1 \parallel m_{52}^1 \parallel m_{53}^1 \parallel m_{54}^1 \parallel m_{55}^1 \parallel m_{56}^1 \parallel m_{57}^1 \parallel m_{58}^1 \parallel m_{59}^1 \parallel m_{60}^1 \parallel m_{61}^1 \parallel m_{62}^1 \parallel m_{63}^1 \parallel m_{64}^1 \parallel m_{65}^1 \parallel m_{66}^1 \parallel m_{67}^1 \parallel m_{68}^1 \parallel m_{69}^1 \parallel m_{70}^1 \parallel m_{71}^1 \parallel m_{72}^1 \parallel m_{73}^1 \parallel m_{74}^1 \parallel m_{75}^1 \parallel m_{76}^1 \parallel m_{77}^1 \parallel m_{78}^1 \parallel m_{79}^1 \parallel m_{80}^1 \parallel m_{81}^1 \parallel m_{82}^1 \parallel m_{83}^1 \parallel m_{84}^1 \parallel m_{85}^1 \parallel m_{86}^1 \parallel m_{87}^1 \parallel m_{88}^1 \parallel m_{89}^1 \parallel m_{90}^1 \parallel m_{91}^1 \parallel m_{92}^1 \parallel m_{93}^1 \parallel m_{94}^1 \parallel m_{95}^1 \parallel m_{96}^1 \parallel m_{97}^1 \parallel m_{98}^1 \parallel m_{99}^1 \parallel m_{100}^1 \parallel m_{101}^1 \parallel m_{102}^1 \parallel m_{103}^1 \parallel m_{104}^1 \parallel m_{105}^1 \parallel m_{106}^1 \parallel m_{107}^1 \parallel m_{108}^1 \parallel m_{109}^1 \parallel m_{110}^1 \parallel m_{111}^1 \parallel m_{112}^1 \parallel m_{113}^1 \parallel m_{114}^1 \parallel m_{115}^1 \parallel m_{116}^1 \parallel m_{117}^1 \parallel m_{118}^1 \parallel m_{119}^1 \parallel m_{120}^1 \parallel m_{121}^1 \parallel m_{122}^1 \parallel m_{123}^1 \parallel m_{124}^1 \parallel m_{125}^1 \parallel m_{126}^1 \parallel m_{127}^1 \parallel m_{128}^1 \parallel m_{129}^1 \parallel m_{130}^1 \parallel m_{131}^1 \parallel m_{132}^1 \parallel m_{133}^1 \parallel m_{134}^1 \parallel m_{135}^1 \parallel m_{136}^1 \parallel m_{137}^1 \parallel m_{138}^1 \parallel m_{139}^1 \parallel m_{140}^1 \parallel m_{141}^1 \parallel m_{142}^1 \parallel m_{143}^1 \parallel m_{144}^1 \parallel m_{145}^1 \parallel m_{146}^1 \parallel m_{147}^1 \parallel m_{148}^1 \parallel m_{149}^1 \parallel m_{150}^1 \parallel m_{151}^1 \parallel m_{152}^1 \parallel m_{153}^1 \parallel m_{154}^1 \parallel m_{155}^1 \parallel m_{156}^1 \parallel m_{157}^1 \parallel m_{158}^1 \parallel m_{159}^1 \parallel m_{160}^1 \parallel m_{161}^1 \parallel m_{162}^1 \parallel m_{163}^1 \parallel m_{164}^1 \parallel m_{165}^1 \parallel m_{166}^1 \parallel m_{167}^1 \parallel m_{168}^1 \parallel m_{169}^1 \parallel m_{170}^1 \parallel m_{171}^1 \parallel m_{172}^1 \parallel m_{173}^1 \parallel m_{174}^1 \parallel m_{175}^1 \parallel m_{176}^1 \parallel m_{177}^1 \parallel m_{178}^1 \parallel m_{179}^1 \parallel m_{180}^1 \parallel m_{181}^1 \parallel m_{182}^1 \parallel m_{183}^1 \parallel m_{184}^1 \parallel m_{185}^1 \parallel m_{186}^1 \parallel m_{187}^1 \parallel m_{188}^1 \parallel m_{189}^1 \parallel m_{190}^1 \parallel m_{191}^1 \parallel m_{192}^1 \parallel m_{193}^1 \parallel m_{194}^1 \parallel m_{195}^1 \parallel m_{196}^1 \parallel m_{197}^1 \parallel m_{198}^1 \parallel m_{199}^1 \parallel m_{200}^1 \parallel m_{201}^1 \parallel m_{202}^1 \parallel m_{203}^1 \parallel m_{204}^1 \parallel m_{205}^1 \parallel m_{206}^1 \parallel m_{207}^1 \parallel m_{208}^1 \parallel m_{209}^1 \parallel m_{210}^1 \parallel m_{211}^1 \parallel m_{212}^1 \parallel m_{213}^1 \parallel m_{214}^1 \parallel m_{215}^1 \parallel m_{216}^1 \parallel m_{217}^1 \parallel m_{218}^1 \parallel m_{219}^1 \parallel m_{220}^1 \parallel m_{221}^1 \parallel m_{222}^1 \parallel m_{223}^1 \parallel m_{224}^1 \parallel m_{225}^1 \parallel m_{226}^1 \parallel m_{227}^1 \parallel m_{228}^1 \parallel m_{229}^1 \parallel m_{230}^1 \parallel m_{231}^1 \parallel m_{232}^1 \parallel m_{233}^1 \parallel m_{234}^1 \parallel m_{235}^1 \parallel m_{236}^1 \parallel m_{237}^1 \parallel m_{238}^1 \parallel m_{239}^1 \parallel m_{240}^1 \parallel m_{241}^1 \parallel m_{242}^1 \parallel m_{243}^1 \parallel m_{244}^1 \parallel m_{245}^1 \parallel m_{246}^1 \parallel m_{247}^1 \parallel m_{248}^1 \parallel m_{249}^1 \parallel m_{250}^1 \parallel m_{251}^1 \parallel m_{252}^1 \parallel m_{253}^1 \parallel m_{254}^1 \parallel m_{255}^1 \parallel m_{256}^1 \parallel m_{257}^1 \parallel m_{258}^1 \parallel m_{259}^1 \parallel m_{260}^1 \parallel m_{261}^1 \parallel m_{262}^1 \parallel m_{263}^1 \parallel m_{264}^1 \parallel m_{265}^1 \parallel m_{266}^1 \parallel m_{267}^1 \parallel m_{268}^1 \parallel m_{269}^1 \parallel m_{270}^1 \parallel m_{271}^1 \parallel m_{272}^1 \parallel m_{273}^1 \parallel m_{274}^1 \parallel m_{275}^1 \parallel m_{276}^1 \parallel m_{277}^1 \parallel m_{278}^1 \parallel m_{279}^1 \parallel m_{280}^1 \parallel m_{281}^1 \parallel m_{282}^1 \parallel m_{283}^1 \parallel m_{284}^1 \parallel m_{285}^1 \parallel m_{286}^1 \parallel m_{287}^1 \parallel m_{288}^1 \parallel m_{289}^1 \parallel m_{290}^1 \parallel m_{291}^1 \parallel m_{292}^1 \parallel m_{293}^1 \parallel m_{294}^1 \parallel m_{295}^1 \parallel m_{296}^1 \parallel m_{297}^1 \parallel m_{298}^1 \parallel m_{299}^1 \parallel m_{300}^1 \parallel m_{301}^1 \parallel m_{302}^1 \parallel m_{303}^1 \parallel m_{304}^1 \parallel m_{305}^1 \parallel m_{306}^1 \parallel m_{307}^1 \parallel m_{308}^1 \parallel m_{309}^1 \parallel m_{310}^1 \parallel m_{311}^1 \parallel m_{312}^1 \parallel m_{313}^1 \parallel m_{314}^1 \parallel m_{315}^1 \parallel m_{316}^1 \parallel m_{317}^1 \parallel m_{318}^1 \parallel m_{319}^1 \parallel m_{320}^1 \parallel m_{321}^1 \parallel m_{322}^1 \parallel m_{323}^1 \parallel m_{324}^1 \parallel m_{325}^1 \parallel m_{326}^1 \parallel m_{327}^1 \parallel m_{328}^1 \parallel m_{329}^1 \parallel m_{330}^1 \parallel m_{331}^1 \parallel m_{332}^1 \parallel m_{333}^1 \parallel m_{334}^1 \parallel m_{335}^1 \parallel m_{336}^1 \parallel m_{337}^1 \parallel m_{338}^1 \parallel m_{339}^1 \parallel m_{340}^1 \parallel m_{341}^1 \parallel m_{342}^1 \parallel m_{343}^1 \parallel m_{344}^1 \parallel m_{345}^1 \parallel m_{346}^1 \parallel m_{347}^1 \parallel m_{348}^1 \parallel m_{349}^1 \parallel m_{350}^1 \parallel m_{351}^1 \parallel m_{352}^1 \parallel m_{353}^1 \parallel m_{354}^1 \parallel m_{355}^1 \parallel m_{356}^1 \parallel m_{357}^1 \parallel m_{358}^1 \parallel m_{359}^1 \parallel m_{360}^1 \parallel m_{361}^1 \parallel m_{362}^1 \parallel m_{363}^1 \parallel m_{364}^1 \parallel m_{365}^1 \parallel m_{366}^1 \parallel m_{367}^1 \parallel m_{368}^1 \parallel m_{369}^1 \parallel m_{370}^1 \parallel m_{371}^1 \parallel m_{372}^1 \parallel m_{373}^1 \parallel m_{374}^1 \parallel m_{375}^1 \parallel m_{376}^1 \parallel m_{377}^1 \parallel m_{378}^1 \parallel m_{379}^1 \parallel m_{380}^1 \parallel m_{381}^1 \parallel m_{382}^1 \parallel m_{383}^1 \parallel m_{384}^1 \parallel m_{385}^1 \parallel m_{386}^1 \parallel m_{387}^1 \parallel m_{388}^1 \parallel m_{389}^1 \parallel m_{390}^1 \parallel m_{391}^1 \parallel m_{392}^1 \parallel m_{393}^1 \parallel m_{394}^1 \parallel m_{395}^1 \parallel m_{396}^1 \parallel m_{397}^1 \parallel m_{398}^1 \parallel m_{399}^1 \parallel m_{400}^1 \parallel m_{401}^1 \parallel m_{402}^1 \parallel m_{403}^1 \parallel m_{404}^1 \parallel m_{405}^1 \parallel m_{406}^1 \parallel m_{407}^1 \parallel m_{408}^1 \parallel m_{409}^1 \parallel m_{410}^1 \parallel m_{411}^1 \parallel m_{412}^1 \parallel m_{413}^1 \parallel m_{414}^1 \parallel m_{415}^1 \parallel m_{416}^1 \parallel m_{417}^1 \parallel m_{418}^1 \parallel m_{419}^1 \parallel m_{420}^1 \parallel m_{421}^1 \parallel m_{422}^1 \parallel m_{423}^1 \parallel m_{424}^1 \parallel m_{425}^1 \parallel m_{426}^1 \parallel m_{427}^1 \parallel m_{428}^1 \parallel m_{429}^1 \parallel m_{430}^1 \parallel m_{431}^1 \parallel m_{432}^1 \parallel m_{433}^1 \parallel m_{434}^1 \parallel m_{435}^1 \parallel m_{436}^1 \parallel m_{437}^1 \parallel m_{438}^1 \parallel m_{439}^1 \parallel m_{440}^1 \parallel m_{441}^1 \parallel m_{442}^1 \parallel m_{443}^1 \parallel m_{444}^1 \parallel m_{445}^1 \parallel m_{446}^1 \parallel m_{447}^1 \parallel m_{448}^1 \parallel m_{449}^1 \parallel m_{450}^1 \parallel m_{451}^1 \parallel m_{452}^1 \parallel m_{453}^1 \parallel m_{454}^1 \parallel m_{455}^1 \parallel m_{456}^1 \parallel m_{457}^1 \parallel m_{458}^1 \parallel m_{459}^1 \parallel m_{460}^1 \parallel m_{461}^1 \parallel m_{462}^1 \parallel m_{463}^1 \parallel m_{464}^1 \parallel m_{465}^1 \parallel m_{466}^1 \parallel m_{467}^1 \parallel m_{468}^1 \parallel m_{469}^1 \parallel m_{470}^1 \parallel m_{471}^1 \parallel m_{472}^1 \parallel m_{473}^1 \parallel m_{474}^1 \parallel m_{475}^1 \parallel m_{476}^1 \parallel m_{477}^1 \parallel m_{478}^1 \parallel m_{479}^1 \parallel m_{480}^1 \parallel m_{481}^1 \parallel m_{482}^1 \parallel m_{483}^1 \parallel m_{484}^1 \parallel m_{485}^1 \parallel m_{486}^1 \parallel m_{487}^1 \parallel m_{488}^1 \parallel m_{489}^1 \parallel m_{490}^1 \parallel m_{491}^1 \parallel m_{492}^1 \parallel m_{493}^1 \parallel m_{494}^1 \parallel m_{495}^1 \parallel m_{496}^1 \parallel m_{497}^1 \parallel m_{498}^1 \parallel m_{499}^1 \parallel m_{500}^1 \parallel m_{501}^1 \parallel m_{502}^1 \parallel m_{503}^1 \parallel m_{504}^1 \parallel m_{505}^1 \parallel m_{506}^1 \parallel m_{507}^1 \parallel m_{508}^1 \parallel m_{509}^1 \parallel m_{510}^1 \parallel m_{511}^1 \parallel m_{512}^1 \parallel m_{513}^1 \parallel m_{514}^1 \parallel m_{515}^1 \parallel m_{516}^1 \parallel m_{517}^1 \parallel m_{518}^1 \parallel m_{519}^1 \parallel m_{520}^1 \parallel m_{521}^1 \parallel m_{522}^1 \parallel m_{523}^1 \parallel m_{524}^1 \parallel m_{525}^1 \parallel m_{526}^1 \parallel m_{527}^1 \parallel m_{528}^1 \parallel m_{529}^1 \parallel m_{530}^1 \parallel m_{531}^1 \parallel m_{532}^1 \parallel m_{533}^1 \parallel m_{534}^1 \parallel m_{535}^1 \parallel m_{536}^1 \parallel m_{537}^1 \parallel m_{538}^1 \parallel m_{539}^1 \parallel m_{540}^1 \parallel m_{541}^1 \parallel m_{542}^1 \parallel m_{543}^1 \parallel m_{544}^1 \parallel m_{545}^1 \parallel m_{546}^1 \parallel m_{547}^1 \parallel m_{548}^1 \parallel m_{549}^1 \parallel m_{550}^1 \parallel m_{551}^1 \parallel m_{552}^1 \parallel m_{553}^1 \parallel m_{554}^1 \parallel m_{555}^1 \parallel m_{556}^1 \parallel m_{557}^1 \parallel m_{558}^1 \parallel m_{559}^1 \parallel m_{560}^1 \parallel m_{561}^1 \parallel m_{562}^1 \parallel m_{563}^1 \parallel m_{564}^1 \parallel m_{565}^1 \parallel m_{566}^1 \parallel m_{567}^1 \parallel m_{568}^1 \parallel m_{569}^1 \parallel m_{570}^1 \parallel m_{571}^1 \parallel m_{572}^1 \parallel m_{573}^1 \parallel m_{574}^1 \parallel m_{575}^1 \parallel m_{576}^1 \parallel m_{577}^1 \parallel m_{578}^1 \parallel m_{579}^1 \parallel m_{580}^1 \parallel m_{581}^1 \parallel m_{582}^1 \parallel m_{583}^1 \parallel m_{584}^1 \parallel m_{585}^1 \parallel m_{586}^1 \parallel m_{587}^1 \parallel m_{588}^1 \parallel m_{589}^1 \parallel m_{590}^1 \parallel m_{591}^1 \parallel m_{592}^1 \parallel m_{593}^1 \parallel m_{594}^1 \parallel m_{595}^1 \parallel m_{596}^1 \parallel m_{597}^1 \parallel m_{598}^1 \parallel m_{599}^1 \parallel m_{600}^1 \parallel m_{601}^1 \parallel m_{602}^1 \parallel m_{603}^1 \parallel m_{604}^1 \parallel m_{605}^1 \parallel m_{606}^1 \parallel m_{607}^1 \parallel m_{608}^1 \parallel m_{609}^1 \parallel m_{610}^1 \parallel m_{611}^1 \parallel m_{612}^1 \parallel m_{613}^1 \parallel m_{614}^1 \parallel m_{615}^1 \parallel m_{616}^1 \parallel m_{617}^1 \parallel m_{618}^1 \parallel m_{619}^1 \parallel m_{620}^1 \parallel m_{621}^1 \parallel m_{622}^1 \parallel m_{623}^1 \parallel m_{624}^1 \parallel m_{625}^1 \parallel m_{626}^1 \parallel m_{627}^1 \parallel m_{628}^1 \parallel m_{629}^1 \parallel m_{630}^1 \parallel m_{631}^1 \parallel m_{632}^1 \parallel m_{633}^1 \parallel m_{634}^1 \parallel m_{635}^1 \parallel m_{636}^1 \parallel m_{637}^1 \parallel m_{638}^1 \parallel m_{639}^1 \parallel m_{640}^1 \parallel m_{641}^1 \parallel m_{642}^1 \parallel m_{643}^1 \parallel m_{644}^1 \parallel m_{645}^1 \parallel m_{646}^1 \parallel m_{647}^1 \parallel m_{648}^1 \parallel m_{649}^1 \parallel m_{650}^1 \parallel m_{651}^1 \parallel m_{652}^1 \parallel m_{653}^1 \parallel m_{654}^1 \parallel m_{655}^1 \parallel m_{656}^1 \parallel m_{657}^1 \parallel m_{658}^1 \parallel m_{659}^1 \parallel m_{660}^1 \parallel m_{661}^1 \parallel m_{662}^1 \parallel m_{663}^1 \parallel m_{664}^1 \parallel m_{665}^1 \parallel m_{666}^1 \parallel m_{667}^1 \parallel m_{668}^1 \parallel m_{669}^1 \parallel m_{670}^1 \parallel m_{671}^1 \parallel m_{672}^1 \parallel m_{673}^1 \parallel m_{674}^1 \parallel m_{675}^1 \parallel m_{676}^1 \parallel m_{677}^1 \parallel m_{678}^1 \parallel m_{679}^1 \parallel m_{680}^1 \parallel m_{681}^1 \parallel m_{682}^1 \parallel m_{683}^1 \parallel m_{684}^1 \parallel m_{685}^1 \parallel m_{686}^1 \parallel m_{687}^1 \parallel m_{688}^1 \parallel m_{689}^1 \parallel m_{690}^1 \parallel m_{691}^1 \parallel m_{692}^1 \parallel m_{693}^1 \parallel m_{694}^1 \parallel m_{695}^1 \parallel m_{696}^1 \parallel m_{697}^1 \parallel m_{698}^1 \parallel m_{699}^1 \parallel m_{700}^1 \parallel m_{701}^1 \parallel m_{702}^1 \parallel m_{703}^1 \parallel m_{704}^1 \parallel m_{705}^1 \parallel m_{706}^1 \parallel m_{707}^1 \parallel m_{708}^1 \parallel m_{709}^1 \parallel m_{710}^1 \parallel m_{711}^1 \parallel m_{712}^1 \parallel m_{713}^1 \parallel m_{714}^1 \parallel m_{715}^1 \parallel m_{716}^1 \parallel m_{717}^1 \parallel m_{718}^1 \parallel m_{719}^1 \parallel m_{720}^1 \parallel m_{721}^1 \parallel m_{722}^1 \parallel m_{723}^1 \parallel m_{724}^1 \parallel m_{725}^1 \parallel m_{726}^1 \parallel m_{727}^1 \parallel m_{728}^1 \parallel m_{729}^1 \parallel m_{730}^1 \parallel m_{731}^1 \parallel m_{732}^1 \parallel m_{733}^1 \parallel m_{734}^1 \parallel m_{735}^1 \parallel m_{736}^1 \parallel m_{737}^1 \parallel m_{738}^1 \parallel m_{739}^1 \parallel m_{740}^1 \parallel m_{741}^1 \parallel m_{742}^1 \parallel m_{743}^1 \parallel m_{744}^1 \parallel m_{745}^1 \parallel m_{746}^1 \parallel m_{747}^1 \parallel m_{748}^1 \parallel m_{749}^1 \parallel m_{750}^1 \parallel m_{751}^1 \parallel m_{752}^1 \parallel m_{753}^1 \parallel m_{754}^1 \parallel m_{755}^1 \parallel m_{756}^1 \parallel m_{757}^1 \parallel m_{758}^1 \parallel m_{759}^1 \parallel m_{760}^1 \parallel m_{761}^1 \parallel m_{762}^1 \parallel m_{763}^1 \parallel m_{764}^1 \parallel m_{765}^1 \parallel m_{766}^1 \parallel m_{767}^1 \parallel m_{768}^1 \parallel m_{769}^1 \parallel m_{770}^1 \parallel m_{771}^1 \parallel m_{772}^1 \parallel m_{773}^1 \parallel m_{774}^1 \parallel m_{775}^1 \parallel m_{776}^1 \parallel m_{777}^1 \parallel m_{778}^1 \parallel m_{779}^1 \parallel m_{780}^1 \parallel m_{781}^1 \parallel m_{782}^1 \parallel m_{783}^1 \parallel m_{784}^1 \parallel m_{785}^1 \parallel m_{786}^1 \parallel m_{787}^1 \parallel m_{788}^1 \parallel m_{789}^1 \parallel m_{790}^1 \parallel m_{791}^1 \parallel m_{792}^1 \parallel m_{793}^1 \parallel m_{794}^1 \parallel m_{795}^1 \parallel m_{796}^1 \parallel m_{797}^1 \parallel m_{798}^1 \parallel m_{799}^1 \parallel m_{800}^1 \parallel m_{801}^1 \parallel m_{802}^1 \parallel m_{803}^1 \parallel m_{804}^1 \parallel m_{805}^1 \parallel m_{806}^1 \parallel m_{807}^1 \parallel m_{808}^1 \parallel m_{809}^1 \parallel m_{810}^1 \parallel m_{811}^1 \parallel m_{812}^1 \parallel m_{813}^1 \parallel m_{814}^1 \parallel m_{815}^1 \parallel m_{816}^1 \parallel m_{817}^1 \parallel m_{818}^1 \parallel m_{819}^1 \parallel m_{820}^1 \parallel m_{821}^1 \parallel m_{822}^1 \parallel m_{823}^1 \parallel m_{824}^1 \parallel m_{825}^1 \parallel m_{826}^1 \parallel m_{827}^1 \parallel m_{828}^1 \parallel m_{829}^1 \parallel m_{830}^1 \parallel m_{831}^1 \parallel m_{832}^1 \parallel m_{833}^1 \parallel m_{834}^1 \parallel m_{835}^1 \parallel m_{836}^1 \parallel m_{837}^1 \parallel m_{838}^1 \parallel m_{839}^1 \parallel m_{840}^1 \parallel m_{841}^1 \parallel m_{842}^1 \parallel m_{843}^1 \parallel m_{844}^1 \parallel m_{845}^1 \parallel m_{846}^1 \parallel m_{847}^1 \parallel m_{848}^1 \parallel m_{849}^1 \parallel m_{850}^1 \parallel m_{851}^1 \parallel m_{852}^1 \parallel m_{853}^1 \parallel m_{854}^1 \parallel m_{855}^1 \parallel m_{856}^1 \parallel m_{857}^1 \parallel m_{858}^1 \parallel m_{859}^1 \parallel m_{860}^1 \parallel m_{861}^1 \parallel m_{862}^1 \parallel m_{863}^1 \parallel m_{864}^1 \parallel m_{865}^1 \parallel m_{866}^1 \parallel m_{867}^1 \parallel m_{868}^1 \parallel m_{869}^1 \parallel m_{870}^1 \parallel m_{871}^1 \parallel m_{872}^1 \parallel m_{873}^1 \parallel m_{874}^1 \parallel m_{875}^1 \parallel m_{876}^1 \parallel m_{877}^1 \parallel m_{878}^1 \parallel m_{879}^1 \parallel m_{880}^1 \parallel m_{881}^1 \parallel m_{882}^1 \parallel m_{883}^1 \parallel m_{884}^1 \parallel m_{885}^1 \parallel m_{886}^1 \parallel m_{887}^1 \parallel m_{888}^1 \parallel m_{889}^1 \parallel m_{890}^1 \parallel m_{891}^1 \parallel m_{892}^1 \parallel m_{893}^1 \parallel m_{894}^1 \parallel m_{895}^1 \parallel m_{896}^1 \parallel m_{897}^1 \parallel m_{898}^1 \parallel m_{899}^1 \parallel m_{900}^1 \parallel m_{901}^1 \parallel m_{902}^1 \parallel m_{903}^1 \parallel m_{904}^1 \parallel m_{905}^1 \parallel m_{906}^1 \parallel m_{907}^1 \parallel m_{908}^1 \parallel m_{909}^1 \parallel m_{910}^1 \parallel m_{911}^1 \parallel m_{912}^1 \parallel m_{913}^1 \parallel m_{914}^1 \parallel m_{915}^1 \parallel m_{916}^1 \parallel m_{917}^1 \parallel m_{918}^1 \parallel m_{919}^1 \parallel m_{920}^1 \parallel m_{921}^1 \parallel m_{922}^1 \parallel m_{923}^1 \parallel m_{924}^1 \parallel m_{925}^1 \parallel m_{926}^1 \parallel m_{927}^1 \parallel m_{928}^1 \parallel m_{929}^1 \parallel m_{930}^1 \parallel m_{931}^1 \parallel m_{932}^1 \parallel m_{933}^1 \parallel m_{934}^1 \parallel m_{935}^1 \parallel m_{936}^1 \parallel m_{937}^1 \parallel m_{938}^1 \parallel m_{939}^1 \parallel m_{940}^1 \parallel m_{941}^1 \parallel m_{942}^1 \parallel m_{943}^1 \parallel m_{944}^1 \parallel m_{945}^1 \parallel m_{946}^1 \parallel m_{947}^1 \parallel m_{948}^1 \parallel m_{949}^1 \parallel m_{950}^1 \parallel m_{951}^1 \parallel m_{952}^1 \parallel m_{953}^1 \parallel m_{954}^1 \parallel m_{955}^1 \parallel m_{956}^1 \parallel m_{957}^1 \parallel m_{958}^1 \parallel m_{959}^1 \parallel m_{960}^1 \parallel m_{961}^1 \parallel m_{962}^1 \parallel m_{963}^1 \parallel m_{964}^1 \parallel m_{965}^1 \parallel m_{966}^1 \parallel m_{967}^1 \parallel m_{968}^1 \parallel m_{969}^1 \parallel m_{970}^1 \parallel m_{971}^1 \parallel m_{972}^1 \parallel m_{973}^1 \parallel m_{974}^1 \parallel m_{975}^1 \parallel m_{976}^1 \parallel m_{977}^1 \parallel m_{978}^1 \parallel m_{979}^1 \parallel m_{980}^1 \parallel m_{981}^1 \parallel m_{982}^1 \parallel m_{983}^1 \parallel m_{984}^1 \parallel m_{985}^1 \parallel m_{986}^1 \parallel m_{987}^1 \parallel m_{988}^1 \parallel m_{989}^1 \parallel m_{990}^1 \parallel m_{991}^1 \parallel m_{992}^1 \parallel m_{993}^1 \parallel m_{994}^1 \parallel m_{995}^1 \parallel m_{996}^1 \parallel m_{997}^1 \parallel m_{998}^1 \parallel m_{999}^1 \parallel m_{1000}^1$

3) 计算 $h = H(\text{node}_{\text{DF}})$, 使用私钥 x 对 h 进行 BLS 签名, $\sigma_i = h^x$, 则数据域 df 的证明元数据可以表示为 $\{\text{node}_{\text{df}}, l_i, s_i, \sigma_i, y\}$ 。

4.2.3 数据副本存储阶段

数据存储系统通过安全通道接收数据所有者需要存储的数据域副本 $\{\text{df}_1, \text{df}_2, \dots, \text{df}_\mu\}$, 并通过签名机制生成数据存储凭证 $\text{storage_credentials}$ 。

在该阶段, 数据所有者上传上链数据, docType 取 data_up , data1 取数据名称 data_name , data2 取数据存储凭证 $\text{storage_credentials}$ 。

4.2.4 挑战阶段

在该阶段, 数据所有者确定进行挑战的数据域 df , 选取 df 对应的证明元数据为挑战数据 $\{\text{node}_{\text{df}}, l_i, s_i, \sigma_i, y\}$, docType 取 challenge , data1 取挑战数据, data2 取数据存储凭证 $\text{storage_credentials}$ 。

4.2.5 证据生成阶段

数据存储系统作为区块链节点, 在线监听数据所有者对自己的挑战, 获取挑战元数据 $\{\text{node}_{\text{df}}, l_i, s_i, \sigma_i, y\}$, 依据 node_{df} , 从存储数据库中获取数据域的副本 $\{\text{df}_1, \text{df}_2, \dots, \text{df}_\mu\}$, 依据随机序列 l_i 和拼接策略 s_i 获得复合数据域 DF' , 对复合数据域 DF 构建基于数据域的混合 Merkle DAG 结构获得证据 $K_i = H(\text{node}_{\text{DF}'})$ 。

在该阶段, 存储系统上传证明数据, docType

取 prove , data1 取证据 K_i , data2 取对应的挑战时间戳 $\text{challenge_timestamp}$ 。

4.2.6 完整性判决阶段

完整性判决合约获得挑战元数据 $\{\text{node}_{\text{df}}, l_i, s_i, \sigma_i, y\}$ 和证据 K_i , 并判断 (g, K_i, y, σ_i) 是否为 DH 四元组, 如果是, 则判断数据完整, $\text{audit_results}=\text{True}$; 如果不是, 则判断数据不完整, $\text{audit_results}=\text{False}$ 。

在该阶段, 数据完整性判决合约上传审计数据, docType 取 audit , data1 取审计结果 audit_results , data2 取对应的证据时间戳 prove_timestamp 。

4.3 数据完整性审计历史可信追溯

为实现对审计历史的高效检索, 创建索引文件并与本文使用的智能合约一起安装到区块链系统节点。索引文件内容如下:

```
{
  "index": {
    "fields": ["node_block"]
  },
  "ddoc": "index1Doc",
  "name": "index1",
  "type": "json"
}
```

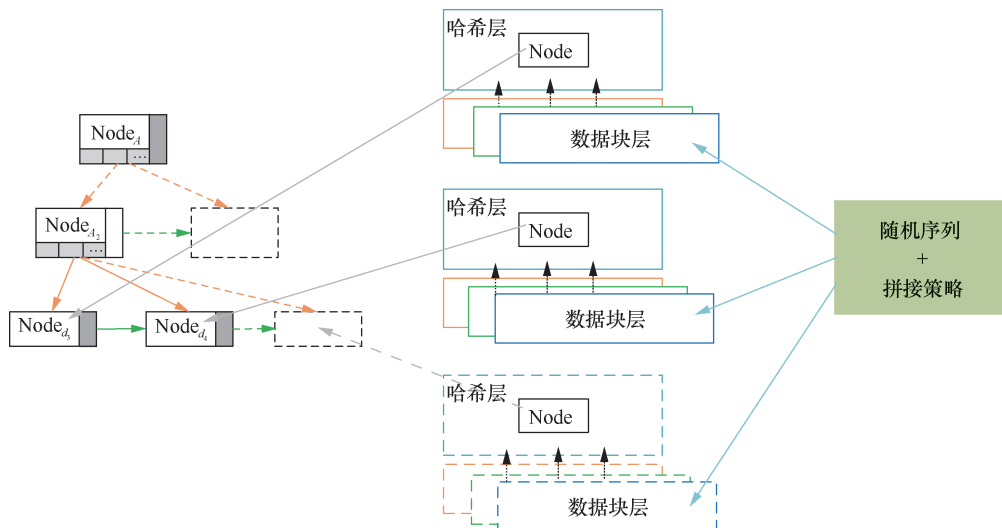


图9 获取证明元数据
Figure 9 Obtain proof metadata

```

    "index":{
        "fields":["docType","node_block"]
    },
    "ddoc":"index2Doc"
    "name":"index2"
    "type":"json"
}

```

上述索引文件创建了两个索引,索引1依据 Node_block 进行索引,能够索引与某个数据域相关的所有信息,包括上链数据、挑战数据、证明数据、审计数据;索引2依据 docType 和 Node_block 进行索引,能够分别索引某个数据域的上链数据、挑战数据、证明数据、审计数据信息。

5 方案分析

5.1 安全性分析

5.1.1 验证流程安全性

完整性验证流程概括为数据副本生成、证明元数据生成、证据生成、证据审计4个阶段。非结构化数据通过基于数据域的 Merkle DAG 文件结构进行组织,节点信息中的 $node_{df}$ 保证节点的唯一性,节点信息中指针信息保证 Merkle DAG 文件结构确定,数据在数据存储系统 SP 中以确定的方式进行存储。副本生成机制保证数据损坏能够即时修复。数据副本生成阶段,通过设置 μ 个不同的 AES 密钥 $\{K_1, K_2, \dots, K_\mu\}$ 对数据加密实现副本生成,防止存储系统进行数据多副本伪存储。该副本生成机制,能够减少副本参数存储,防止海量参数丢失与损坏。在证明元数据生成阶段,使用随机序列拼接方式组合获得复合数据域 $node_{DF}$,对 $node_{DF}$ 进行签名获得证明元数据 $\{node_{df}, l_i, s_i, \sigma_i, y\}$ 。在证据生成阶段,存储系统同样使用随机序列拼接方式获得复合数据域 $node_{DF}$,该方式能够保证元数据生成必须使用完整副本数据块,保证完整性审计的可行性。在证据验证阶段通过检验 (g, K_i, y, σ_i) 是否为 DH 四元组,判断数据是否遭到破坏,保证校验结果的可信度。

5.1.2 区块链账本记录可信度

本文的完整性验证流程通过智能合约实现,验证的相关数据记录在区块链账本上。智能合约

取代可信第三方对证据的审计,这样能够解决防止第三方对验证流程进行攻击的问题。区块链账本能够实现账本数据的安全可信多方存储,验证流程数据记录在区块链账本上,防止各方对验证流程的篡改,保证数据完整性验证历史的真实性。在区块链账本上记录数据在区块链账本上的唯一标识 Node_block,唯一标识的确定能够保证数据存储与完整性验证的一致性,实现对某一数据完整性验证历史的检索。通过数据完整性验证历史的检索,能够保证数据在进行共享之前的历史一致性,保证数据的可信度。

5.2 方案对比

表1为本文方案与现有方案的对比。本文方案包含多个数据拥有者与数据使用者,以及多个数据存储系统,能够实现同领域数据或者联盟成员间数据的多方审计。本文方案选用智能合约作为系统审计机构,能够避免寻找可信第三方。与抽样验证相比,基于数据域的混合 Merkle DAG 结构能够实现多副本数据的确定性验证,提高数据完整性审计的效率与准确度。同时,本文方案设计数据完整性审计历史高效检索机制,实现数据完整性验证历史高效检索以及多方验证,保证数据历史的一致性,增强数据的可信度。本文方案没有实现数据的动态修改,降低了数据存储的灵活性,但可以增加数据的历史一致性,适合本文探讨的倾向数据共享的场景。

6 系统测试

本文实验部署6台阿里云服务器,其功能标识如表2所示。云服务器配置 Intel Xeon Platinum 8269CY @2.5GHz 处理器,256 GB 内存,Ubuntu 16.04 64 位操作系统。通过 Java 实现 CBC 模式 128 bit 密钥 AES 加密。证明元数据生成阶段使用 go 语言编程实现,并利用 go 语言的并发机制加快计算速度,使用 SHA256 算法进行数据摘要提取。BLS 签名验证的部分通过 Java 的 JPBC 库实现。区块链部分基于 Hyperledger Fabric 2.2 实现。区块链系统的背书策略为:将智能合约安装在6台服务器中的 peer 节点上,在进行区块链交易时,6台服务器中的 peer 节点均进行交易背书。

表1 方案对比
Table 1 Scheme comparison

方案	参与方			审计机构	多副本	确定性验证	公开验证	异构多数据	数据动态修改	验证历史检索	多方验证
文献[10]	多个 DO	SP	TPA	TPA	NO	NO	YES	NO	YES	NO	NO
文献[11]	DO	SP	TPA	TPA	NO	NO	YES	NO	YES	NO	NO
文献[12]	DO	SP		DO	YES	NO	NO	NO	YES	NO	NO
文献[14]	DO	SP	多个 TPA	动态 TPA	NO	NO	YES	NO	NO	YES	YES
文献[18]	DO	DU	SP	智能合约	NO	YES	NO	NO	NO	NO	YES
文献[19]	DO	SP		DO/SP/TPA	NO	NO	NO	NO	YES	NO	NO
文献[20]	多个 DO	SP		智能合约	NO	NO	YES	NO	YES	NO	NO
本文方案	多个 DO	多个 DU	多个 SP	智能合约	YES	YES	YES	YES	NO	YES	YES

表2 云服务器功能
Table 2 Cloud server function

云服务器	用户	功能标识
server1	DO_A	Peer1、Org1、Ca1、Kafka1、Zookeeper1、Orderer1
server2	DO_B	Peer2、Org2、Ca2、Kafka2、Zookeeper2
server3	DU_A	Peer3、Org3、Ca3、Kafka3、Zookeeper3、Orderer2
server4	DU_B	Peer4、Org4、Ca4、Kafka4、Zookeeper4
server5	SP_A	Peer5、Org5、Ca5、Kafka5、Zookeeper5、Orderer3
server6	SP_B	Peer6、Org6、Ca6、Kafka6、Zookeeper6

证明元数据生成过程与证据生成过程相似,在本文的测试中,对证明元数据生成效率进行测试。为验证系统对大数据环境中广泛存在的小数据量数据进行完整性验证的效率,测试了包含大量小数据的数据域生成证明元数据的速度。并进行证据校验效率测试。

6.1 证明元数据生成效率测试

对数据量为 1~10 GB 的数据进行证明元数据生成效率的测试如图 10 所示。在该测试中,数据副本的数量为 3,多分支 Merkle 树结构中参数 $N_{\max}$ 取 27,分别构建数据块大小为 16 MB、24 MB、32 MB 的基于数据域的混合 Merkle DAG 结构。图 11 所示的是对 1 GB 以内数据进行证明元数据生成效率的测试。数据副本的数量为 3,多分支 Merkle 树结构中参数 $N_{\max}$ 取 27,分别构建数据块大小为 1 MB、4 MB、8 MB 的基于数据域的混合 Merkle DAG 结构。

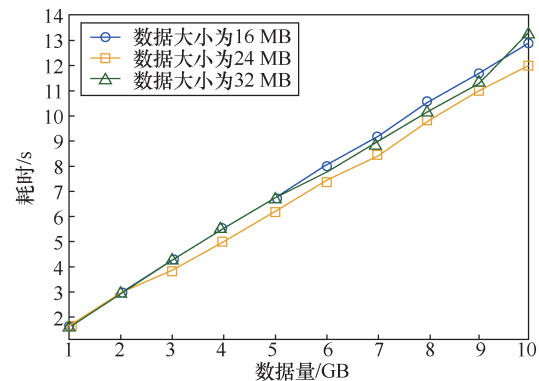


图10 1~10 GB 证明元数据的生成测试
Figure 10 1~10 GB proof metadata generation test

通过两组曲线可以看出,相同的数据量使用不同大小的分块进行处理,耗时基本相同。在实际的应用中,数据分块大小可以根据数据量的大小以及实际的需求进行确定。证明元数据生成的时间与数据量成正比,耗时与数据量的比例关系约为 1.25 s/GB,数据处理效率较高。

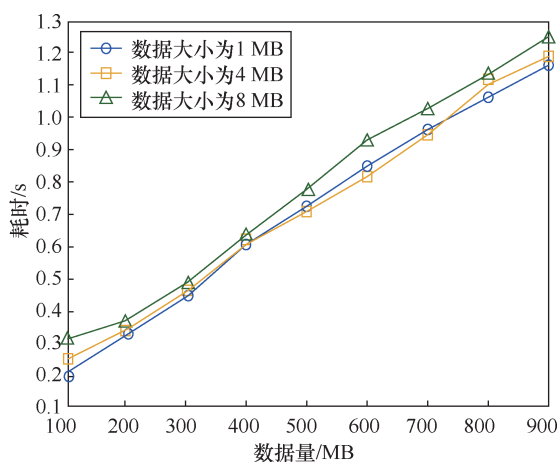


图 11 1 GB 以内证明元数据的生成测试
Figure 11 Data within 1 GB to prove metadata generation test

图 12 所示的是对 1 000~10 000 条非结构化小数据进行证明元数据生成效率的测试。非结构化数据大小为 1 MB 左右,数据副本的数量为 3,多分支 Merkle 树结构中参数取 27,构建数据块大小为 1 MB 的基于数据域的 Merkle DAG 文件结构。从获得的曲线可以看出证明元数据生成的速度与数据条数成正比,约为 870 条/秒,数据处理效率较高。

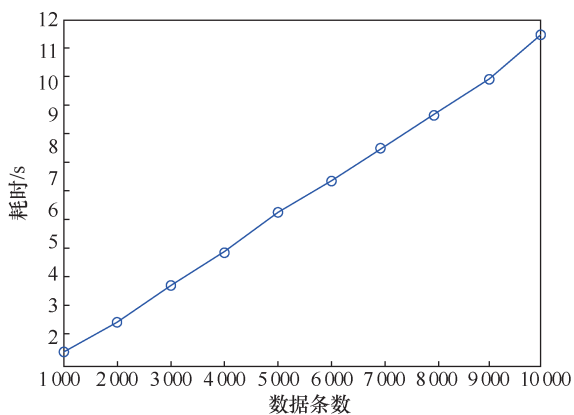


图 12 非结构化小数据证明元数据生成测试
Figure 12 Unstructured small data proof metadata generation test

6.2 证据校验效率测试

图 13 为对数据量 1~10 GB 的数据进行证据真实性校验的效率测试。图 14 为对数据量 100~900 MB 的数据进行证据真实性的效率测试。由结果可知,完整性证据的耗时与数据量的大小无关,证据真实性校验的时间约为 40 ms,能够实现对证据真实性的快速校验。

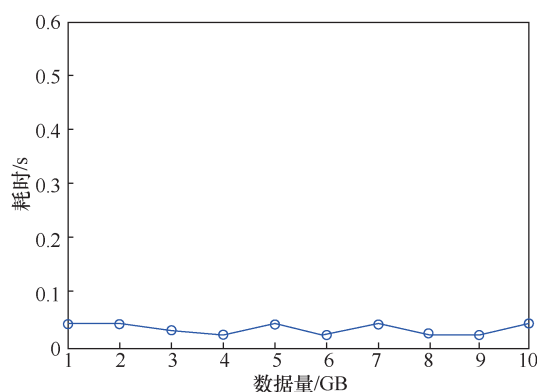


图 13 1~10 GB 数据证据校验测试
Figure 13 1~10 GB data evidence verification test

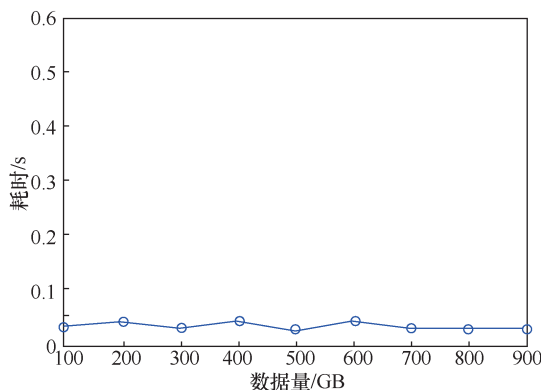


图 14 1 GB 以内数据证据校验测试
Figure 14 Data proof verification test within 1 GB

6.3 性能分析

从实验的结果可以看出数据完整性证明元数据生成以及完整性审计的速度都较高,能够满足数据异构以及数据量大的要求。需要说明的是,以上的测试均是在 3 个数据副本的基础上进行的,测试消耗的时间是对 3 个数据副本同时进行数据完整性证明所用的时间。如果减少副本的数量,则将成比例减少时间消耗。在实际使用中,数据拥有者可以根据数据的重要程度选取合适的副本数或者不使用副本技术。

7 结束语

本文针对大数据环境下数据特点构建基于区块链的数据完整性多方高效审计机制,实现非结构化小数据以及大体积数据的高效多副本审计。通过智能合约实现数据完整性审计过程,对审计历史追溯实现对审计过程多方监督,保证数据共享之前的历史一致性,增加数据可信度。但是本

文方案证明元数据的生成需要在生成随机序列的基础上实现, 下一步计划研究更灵活的证明元数据生成方式。

参考文献:

- [1] AL-SAI Z A, ABUALIGAH L M. Big data and E-government: a review[C]//Proceedings of 2017 8th International Conference on Information Technology (ICIT). 2017: 580-587.
- [2] ZUO W M, WANG C Y. Judicial big data and big-data-based legal research in China[J]. Asian Journal of Law and Society, 2020, 7(3): 495-514.
- [3] RAGHUPATHI W, RAGHUPATHI V. Big data analytics in health-care: promise and potential[J]. Health Information Science and Systems, 2014, 2: 3.
- [4] 谭霜, 贾焰, 韩伟红. 云存储中的数据完整性证明研究及进展[J]. 计算机学报, 2015, 38(1): 164-177.
TAN S, JIA Y, HAN W H. Research and development of provable data integrity in cloud storage[J]. Chinese Journal of Computers, 2015, 38(1): 164-177.
- [5] 魏凯敏, 翁健, 任奎. 大数据安全保护技术综述[J]. 网络与信息安全学报, 2016, 2(4): 1-11.
WEI K M, WENG J, REN K. Data security and protection techniques in big data: a survey[J]. Chinese Journal of Network and Information Security, 2016, 2(4): 1-11.
- [6] 蒋余浩. 开放共享下的政务大数据管理机制创新[J]. 中国行政管理, 2017(8): 42-46.
JIANG Y H. Open data through collaborative sharing: an essay on innovative mechanism of government big data management[J]. Chinese Public Administration, 2017(8): 42-46.
- [7] NAKAMOTO S. Bitcoin: a peer-to-peer electronic cash system[R]. 2009.
- [8] DESWARTE Y, QUISQUATER J J, SAÏDANE A. Remote integrity checking[M]//Integrity and Internal Control in Information Systems VI. Boston: Kluwer Academic Publishers, 2003: 1-11.
- [9] ATENIESE G, BURNS R, CURTMOLA R, et al. Provable data possession at untrusted stores[C]//Proceedings of the 14th ACM conference on Computer and Communications Security. 2007: 598-609.
- [10] WANG Q, WANG C, REN K, et al. Enabling public auditability and data dynamics for storage security in cloud computing[C]//Proceedings of IEEE Transactions on Parallel and Distributed Systems. 2009: 847-859.
- [11] 谢四江, 贾倍, 王鹤, 等. 基于多分支路径树的云存储大数据完整性证明机制[J]. 计算机科学, 2019, 46(3): 188-196.
XIE S J, JIA B, WANG H, et al. Cloud big data integrity verification scheme based on multi-branch Tree[J]. Computer Science, 2019, 46(3): 188-196.
- [12] 刘洪宇, 丁奕文, 陈雷霆. 支持动态操作的多副本数据完整性验证方案[J]. 计算机应用研究, 2019, 36(9): 2778-2782.
LIU H Y, DING Y W, CHEN L T. Multiple-replica data integrity checking protocol with efficient dynamic update in cloud storage[J]. Application Research of Computers, 2019, 36(9): 2778-2782.
- [13] CURTMOLA R, KHAN O, BURNS R, et al. MR-PDP: multiple-replica provable data possession[C]//Proceedings of 2008 The 28th International Conference on Distributed Computing Systems. 2008: 411-420.
- [14] LU N, ZHANG Y X, SHI W B, et al. A secure and scalable data integrity auditing scheme based on hyperledger fabric[J]. Computers & Security, 2020, 92: 101741.
- [15] HANG L, ULLAH I, KIM D H. A secure fish farm platform based on blockchain for agriculture data integrity[J]. Computers and Electronics in Agriculture, 2020, 170: 105251.
- [16] HANG L, KIM D H. Design and implementation of an integrated IoT blockchain platform for sensing data integrity[J]. Sensors (Basel, Switzerland), 2019, 19(10): 2228.
- [17] ALTULYAN M, YAO L N, KANHERE S S, et al. A unified framework for data integrity protection in people-centric smart cities[J]. Multimedia Tools and Applications, 2020, 79(7): 4989-5002.
- [18] LIU B, YU X L, CHEN S P, et al. Blockchain based data integrity service framework for IoT data[C]//Proceedings of 2017 IEEE International Conference on Web Services (ICWS). Piscataway: IEEE Press, 2017: 468-475.
- [19] ZHAO Q Y, CHEN S Y, LIU Z L, et al. Blockchain-based privacy-preserving remote data integrity checking scheme for IoT information systems[J]. Information Processing & Management, 2020, 57(6): 102355.
- [20] WEI P C, WANG D H, ZHAO Y, et al. Blockchain data-based cloud data integrity protection mechanism[J]. Future Generation Computer Systems, 2020, 102: 902-911.
- [21] 魏艳, 毛燕琴, 沈苏彬. 一种基于区块链的数据完整性验证解决方案[J]. 计算机技术与发展, 2020, 30(1): 76-81.
WEI Y, MAO Y Q, SHEN S B. A data integrity verification solution based on Blockchain[J]. Computer Technology and Development, 2020, 30(1): 76-81.
- [22] GAVIN W. Ethereum: A secure decentralized generalised transaction ledger[R]. Ethereum Project Yellow Paper, 2014.
- [23] LIU C, YANG C, ZHANG X Y, et al. External integrity verification for outsourced big data in cloud and IoT: a big picture[J]. Future Generation Computer Systems, 2015, 49: 58-67.
- [24] CHEN X Y, SHANG T, KIM I, et al. A remote data integrity checking scheme for big data storage[C]//Proceedings of 2017 IEEE Second International Conference on Data Science in Cyber-space (DSC). Piscataway: IEEE Press, 2017: 53-59.
- [25] NASONOV D, VISHRATIN A A, BOUKHANOVSKY A. Blockchain-based transaction integrity in distributed big data marketplace[M]//Lecture Notes in Computer Science. Cham: Springer International Publishing, 2018: 569-577.
- [26] LEBDAOUI I, EL HAJJI S, ORHANOU G. Managing big data integrity[C]//Proceedings of 2016 International Conference on Engineering & MIS (ICEMIS). Piscataway: IEEE Press, 2016: 1-6.
- [27] BENET J. IPFS - content addressed, versioned, P2P file system[J].

arXiv: 1407.3561, 2014.

- [28] ANDROULAKI E, BARGER A, BORTNIKOV V, et al. Hyperledger fabric: a distributed operating system for permissioned blockchains[C]//Proceedings of Proceedings of the Thirteenth EuroSys Conference. New York, NY, USA: ACM, 2018: 1-15.
- [29] BONEH D, LYNN B, SHACHAM H. Short signatures from the Weil pairing[J]. Journal of Cryptology: the Journal of the International Association for Cryptologic Research, 2004, 17(4): 297-319.



王娜（1980- ），女，山西临汾人，博士，信息工程大学副教授，主要研究方向为信息系统安全、大数据和区块链安全。

[作者简介]



周家顺（1997- ），男，河南商丘人，信息工程大学硕士生，主要研究方向为信息安全、区块链。



杜学绘（1968- ），女，河南新乡人，博士，信息工程大学教授、博士生导师，主要研究方向为信息系统安全、大数据和区块链安全。