计算机研究与发展
Journal of Computer Research and Development
DOI: 10.7544/issn1000-1239.202440472
61(10): 2540−2553, 2024

面向跨联盟链的隐私保护数据要素交易审计方案

张 川[1] 王子豪[1] 梁晋文[2] 刘梦轩[1] 邓淏天[1] 祝烈煌[1]
[1](北京理工大学网络空间安全学院 北京 100081)
[2](香港理工大学电子计算学系 香港 999077)
(chuanz@bit.edu.cn)

A Privacy-Preserving Data Element Trading Audit Scheme for Cross-Consortium-Blockchains

Zhang Chuan[1], Wang Zihao[1], Liang Jinwen[2], Liu Mengxuan[1], Deng Haotian[1], and Zhu Liehuang[1]
[1] (*School of Cyberspace Science and Technology*, *Beijing Institute of Technology*, *Beijing* 100081)
[2] (*Department of Computing*, *The Hong Kong Polytechnic University*, *Hong Kong* 999077)

Abstract With the development of data assetization in enterprises, consortium blockchains have become the core infrastructure for data transactions within enterprise alliances. Constructing a decentralized, tamper-proof, and access-controlled data trading market using consortium blockchains can enhance the credibility of data transactions and reduce the risk of data leakage. However, with the increase in enterprise alliances, how to promote the circulation of data elements across different enterprise alliances and resist security threats such as data privacy leaks, data fraud, and payment repudiation have become an urgent problem in the field of data circulation. To address this, we propose a privacy-preserving data element transaction audit scheme for cross-consortium-blockchains. This scheme uses consortium blockchains for data attestation and employs smart contracts on a relay chain for data verification and payment management. By combining Pedersen commitments and Shamir's secret sharing techniques, we construct a zero-knowledge proof mechanism to prevent data fraud without disclosing data elements. Additionally, a bilinear mapping-based data element integrity audit mechanism is introduced, allowing for the verification of data element receipts without revealing data privacy, effectively solving the problem of payment repudiation. Through theoretical analysis and experimental evaluation, the proposed scheme's effectiveness and reliability are verified. The experimental results in a local simulation environment demonstrate that the proposed scheme is feasible and efficient.

Key words data elements; consortium blockchain; cross-chain; audit; zero-knowledge proof

摘 要 随着企业数据资产化的发展,联盟链成为了企业联盟内部数据交易的核心基础设施.利用联盟链构建去中心化、不可篡改、访问可控的数据交易市场,可以提高数据交易的可信度,降低数据泄露风险.然而,随着企业联盟的增多,如何促进跨企业联盟的数据要素流通,抵御数据隐私泄露、数据欺诈和支付抵赖等安全威胁,成为了数据流通领域亟待解决的难题.为此,提出了一种面向跨联盟链的隐私保护数据要素交易审计方案.该方案利用联盟链进行数据存证,通过中继链上的智能合约进行数据验证和支付管理.结合 Pedersen 承诺和 Shamir 秘密分享技术,构建零知识证明机制,在不泄露数据要素的情况下防止数据欺诈的发生.此外,还引入了一种基于双线性映射的数据要素完整性审计机制,允许在不透露数据隐私的

收稿日期: 2024-05-31; **修回日期**: 2024-07-18
基金项目: 国家自然科学基金项目(62232002); 北京理工大学青年学者启动计划项目; 中国科协青年人才托举工程项目(2023QNRC001)
This work was supported by the National Natural Science Foundation of China (62232002), the Beijing Institute of Technology Research Fund Program for Young Scholars, and the Young Elite Scientists Sponsorship Program by CAST (2023QNRC001).
通信作者: 梁晋文(jinwen.liang@polyu.edu.hk)

情况下验证数据要素的收据，有效解决支付抵赖问题. 通过理论分析和实验评估，验证了所提方案的有效性和可靠性，在本地模拟实验环境下，实验结果表明所提方案是可行且高效的.

关键词 数据要素；联盟链；跨链；审计；零知识证明

中图法分类号 TP393

数据要素在当今数字化时代的地位和重要性日益凸显. 它构成了数字经济的核心组成部分，对经济、社会和科技领域产生了深远的影响. 数据要素不仅为决策者提供信息支持，帮助其做出明智的决策，而且通过分析市场趋势、客户需求和内部运营情况，提高了企业的生产效率和资源分配效率，从而降低了成本. 此外，数据要素还改善了用户体验，促进了创新的发展，推动了科学研究的进步，并支撑政府决策与治理. 充分利用数据要素不仅可以促进经济增长和产业升级，还可以提升国家和企业的竞争力. 数据要素作为新型生产要素，正在推动数字经济的发展，其交易和共享对于各行业的数据资源整合和价值释放具有重要意义.

数据要素的交易与流通，是企业数据资产化的关键. 联盟链技术因其去中心化、不可篡改和具有访问控制的特点，被广泛应用于构建企业联盟内部的数据市场. 联盟链作为一种去中心化技术，减少了对可信第三方的依赖，降低了单点故障，交易不透明、不可追溯和隐私泄露的风险；通过区块信息不可篡改的特性，确保了数据的完整性和可信度，防止数据被更改或删除；通过引入访问控制，精细化设置数据访问权限，确保只有经过授权的成员才能访问和处理敏感数据，防止未经授权的访问，减少了节点作恶和数据泄露的风险.

联盟链为多种数据流通应用提供了去中心化解决方案. Liu 等人[1] 基于联盟链提出了一种区块链-云透明数据营销方案 Block-DM，解决了集中式数据营销管理缺乏透明性和分布式市场管理，以及对物联网用户（数据卖方）和第三方（数据买方）的营销公平性不足的问题. 王继业等人[2] 针对能源互联网企业，构建了基于区块链的数据安全共享网络体系，促进了企业内部及企业间的数据安全共享. Cui 等人[3] 利用联盟链技术实现了可追踪和匿名的车辆对车辆（V2V）数据共享，有效防止数据的隐私泄露. Zhang 等人[4] 提出了一种基于联盟链的医疗数据共享方案，解决了医疗数据共享和隐私保护问题. Chen 等人[5] 提出了一种基于区块链的物联网数据交易不可抵赖方案，以解决数据共享的可信度和实时性的限制.

然而，随着企业联盟数量的增加，海量基于联盟链的数据市场已经形成. 传统的联盟链技术虽然在确保数据要素在单个联盟链内的高效安全交易方面表现出色，但同时也造成了链与链之间的分立，局限了企业联盟间的数据流通. 因此，如何构建跨联盟链的数据交易市场，促进企业联盟间的数据安全流通，成为了亟待解决的难题.

为了解决上述难题，Jiang 等人[6] 提出了一种跨链框架，用于集成多个区块链，实现高效和安全的物联网数据管理. Geng 等人[7] 提出了一种新颖的跨组织数据交换方法，使不同组织能够追踪未能实时执行的相关请求，从而及时为用户提供一致的查询结果. Liu 等人[8] 提出了一种基于 Hyperledger Fabric 和属性基访问控制（ABAC）的跨域数据安全共享访问控制模型，实现了多级、细粒度和可审计的访问控制，通过自动权限验证确保数据安全. Singh 等人[9] 提出了一种使用多个安全网关的集中式基于云的跨域数据共享平台. Pedreira 等人[10] 针对许可区块链之间的跨链资产转移问题，提出了使用去中心化视图存储和 Polkadot 连接器的 T-ODAP 协议，解决了中心化第三方信任问题，从而增强跨链互操作性的安全性和鲁棒性. Jiang 等人[11] 提出了一种基于跨链技术的多链融合模型，用于在区块链医疗物联网中实现安全的医疗数据共享. 该模型解决了联盟链技术带来的“信息孤岛”问题，并通过跨链网关和交互协议保障跨链交易的安全与稳定. Kannoori 等人[12] 提出了一种基于非交互式零知识证明的创新方法，用于在同构的许可区块链平台（如 Hyperledger Fabric）之间安全地共享保险数据. de Vos 等人[13] 提出了一种名为 XChange 的机制，实现了无需信任第三方的联盟链间资产所有权交换. XChange 通过增量结算和限制义务策略降低了交易风险，确保了资产交换的安全性和有效性. Zhao 等人[14] 提出了一种跨联盟链的医疗数据访问控制模型，旨在解决医疗数据共享与用户隐私之间的矛盾.

然而，现有的研究主要关注跨链数据交易的隐私保护、数据所有权转移等问题，无法有效应对跨联盟链交易过程中可能出现的数据欺诈和支付抵赖等

问题. 具体来说，文献 [8–9, 12, 14] 聚焦于跨链的数据共享和访问控制，但未涉及数据交易中的支付与追责. 文献 [10, 13] 关注的是联盟链上数据资产的所有权跨链转移，而非数据内容的交易. 对于隐私数据，由于数据明文不能上链，联盟链上存储的是数据的密文或哈希，所提方案只涉及链上存储的数据的所有权转移，并未给出交付隐私数据的明文的方法. 文献 [11] 中，跨链交易的数据均处于密文状态，因此中继链和网关无法判断密文对应的数据明文内容是否真实有效.

基于以上理由，本文提出了一种面向跨联盟链的隐私保护数据要素交易审计方案. 该方案结合跨链技术、零知识证明和智能合约，充分利用联盟链和中继链的特性，确保数据交易过程中的隐私保护和交易公平性. 具体而言，方案通过在联盟链上存储数据存证，利用中继链上执行的智能合约进行数据验证和支付管理，结合 Pedersen 承诺[15] 和 Shamir 秘密分享技术[16] 构建零知识证明，在保护数据隐私的同时审计数据交易的真实性，防止数据欺诈的发生. 此外，本文引入了一种基于双线性映射的数据要素完整性审计机制[17]，允许在不透露数据隐私的情况下验证数据要素的收据，有效解决支付抵赖问题.

本文的主要贡献有 3 点：

1)设计并提出了一种基于 Pedersen 承诺和 Shamir 秘密分享的数据要素分享机制. 该机制结合联盟链上存储数据的存证信息，利用零知识证明技术确保数据要素的真实性，在保护隐私的同时防止数据欺诈的发生.

2)引入了一种基于双线性映射的数据要素完整性审计机制. 该机制允许在不透露数据隐私的情况下验证数据要素的收据，解决购买方支付抵赖的问题.

3)对所提出的方案进行了理论分析和实验评估. 从理论上证明了方案的安全性和有效性，并通过实验展示了方案的高效性.

1 预备知识

1.1 Pedersen 承诺

Pedersen 承诺(Pedersen commitment)是一种基于密码学的承诺方案，广泛应用于需要隐私保护和数据完整性的加密协议中. 承诺者能够在不暴露实际数据的情况下生成一个承诺值，并且无法在不更改承诺值的情况下更改实际数据.

具体来说，Pedersen 承诺在一个循环群 G 中进行运算，该群具有阶 p. 选择 2 个生成元 g 和 h. 给定一个要承诺的值 m 和一个随机数 r，承诺值 CM 按式(1)计算.

$$CM = g^m h^r, \tag{1}$$

其中，m 是承诺的值，而 r 是随机选取的隐藏因子. 即使知道 g 和 h，由于 r 的存在和离散对数问题的困难性，无法从 CM 推断出 m. 当承诺方决定揭示承诺内容时，可以公开 m 和 r，并让验证方通过计算 $g^m h^r$ 来验证承诺的真实性.

Pedersen 承诺的两大核心性质是隐藏性(hiding)和绑定性(binding). 隐藏性保证了在揭示之前，承诺的内容对观察者是不可见的，且观察者无法从承诺中推断出承诺的内容. 绑定性保证了承诺方在生成承诺后无法更改承诺的值. 这种特性使得 Pedersen 承诺在零知识证明、电子投票和数字签名等领域有广泛应用.

1.2 Shamir 秘密分享

Shamir 秘密分享方案是一种密码学技术，用于将一个秘密分割成多个部分，每个部分被称为一个“秘密分片”，以便分发给不同的参与者. 该方案的独特之处在于只有集合中达到某个阈值数量的分片时，才能恢复原始的秘密；少于阈值数量的分片无法提供任何关于秘密的信息.

Shamir 秘密分享基于多项式插值原理. 具体来说，首先选择一个大素数 q 和一个随机的 $t-1$ 次多项式 $f(x) = a_0 + a_1 x + \cdots + a_{t-1} x^{t-1}$，使得 $f(0) = S$，其中 S 是秘密. 为了生成 n 个秘密分片，计算多项式在 n 个不同点的值，每个分片为 $(x_i, f(x_i))$. 之后将不同的秘密分片发送给不同的参与者.

当需要恢复秘密时，至少 t 个参与者可以将他们的分片提交，通过拉格朗日插值法重构多项式 $f(x)$ 并计算 $f(0)$ 来恢复秘密 S.

Shamir 秘密分享的安全性依赖于多项式插值的唯一性和多项式系数的随机性，使得少于 t 个分片无法提供关于秘密 S 的任何信息. 这种方案广泛应用于密钥管理、分布式计算和容错系统中.

1.3 双线性群与双线性映射

双线性群和双线性映射在现代密码学中具有重要地位，特别是在构建如身份基加密、群签名和配对加密等高级密码学协议时. 一个典型的双线性群系统可由五元组 (s, G_1, G_2, G_T, e) 来描述. 其中包括 3 个阶为 s 的群 G_1、G_2 和 G_T，函数 e 将来自 G_1 的元素和来自 G_2 的元素作为输入，并映射到 G_T 中的一个元素.

函数e被称为双线性映射.

双线性映射用e表示，具有以下性质：

1）双线性（bilinearity）. 对于任意2个群元素g_1和g_2，其中$g_1 \in G_1$，$g_2 \in G_2$，我们有：

$$e(g_1^a, g_2^b) = e(g_1, g_2)^{ab} \in G_T. \tag{2}$$

2）非退化性（non-degeneracy）. 至少存在群元素g_1和g_2，其中$g_1 \in G_1$，$g_2 \in G_2$，满足$e(g_1, g_2) \neq 1_{G_T}$，其中1_{G_T}是群G_T的单位元.

3）可计算性（computability）. 对于任意的2个群元素$g_1 \in G_1, g_2 \in G_2$,可以在多项式时间内计算$e(g_1, g_2)$.

1.4 困难问题

本文方案安全性主要基于的困难问题有2个：

1）Strong Diffie-Hellman（SDH）问题[18]. 给定$\left(g_1, g_1^x, g_1^{x^2}, \cdots, g_1^{x^s}, g_2, g_2^x\right) \in G_1^{s+1} \times G_2^2$，计算$\left(c, g_1^{\frac{1}{x+c}}\right) \in \mathbb{Z}_s \times G_1$.

2）Double pairing (DP) 问题[19]. 给定$(g_a, g_b) \in_R G_1^2$，计算一个非平凡的$(a, b) \in G_2^2$，满足$e(g_a, a) \cdot e(g_b, b) = 1_{G_T}$. 其中$1_{G_T}$表示群$G_T$的单位元.

1.5 区块链

区块链[20]是一种去中心化的分布式账本技术，最初因比特币[21]而闻名，但其应用已扩展到众多领域，包括金融[22]、供应链管理[23]、医疗[24]等. 区块链的核心特点包括去中心化、透明性和不可篡改性. 每个区块包含一个时间戳、一组交易数据及前一个区块的哈希值，这种结构确保了数据的完整性和安全性. 根据其参与节点的权限和管理方式，可以分为3种主要类型：

1）公有链

公有链（public blockchain）是一种完全去中心化的区块链，任何人都可以参与和访问网络，其交易记录对所有人公开透明. 通过工作量证明（PoW）或权益证明（PoS）等共识机制，确保网络的安全和防篡改性. 公有链具有高度公开性，任何人都可以自由加入和退出，常见的例子包括比特币和以太坊.

2）私有链

私有链（private blockchain）是由单一组织或机构控制的区块链，只有授权的节点才能参与和访问网络. 它提供了更强的隐私保护，交易记录和数据仅对授权用户可见. 由于参与节点较少，私有链的交易处理速度快且能耗低，适用于企业的内部管理系统或银行和金融机构的内部结算系统.

3）联盟链

联盟链（consortium blockchain）是由多个组织或机构共同管理的区块链，每个参与方需要获得许可才能加入网络. 通常采用高效的共识机制，如拜占庭容错（PBFT）或权益授权证明（DPoS），确保多个参与方的共同决策. 联盟链在隐私和透明性之间找到平衡，既能在联盟成员间共享交易记录，又能对外部隐藏部分信息，适用于企业间的供应链管理系统、金融行业的联合征信系统和医疗行业的联合数据管理系统.

1.6 跨链技术

随着区块链技术的发展，不同区块链之间的互操作性成为一个重要问题. 跨链技术[25]旨在实现不同区块链之间的数据交换和资产转移，打破“信息孤岛”，促进更广泛的区块链生态系统发展. 常见的跨链工具的包括以下4个：

1）Polkadot. Polkadot[26]是由Web3基金会开发的跨链协议，旨在实现不同区块链之间的互操作. Polkadot使用中继链（relay chain）和平行链（parachain）架构，通过中继链来协调不同区块链的通信和数据交换. 中继链负责整个网络的安全性和共识，而平行链可以独立运行并与中继链进行通信. 这种架构确保了高扩展性和共享安全性，同时允许灵活的跨链通信.

2）Cosmos. Cosmos[27]是由Tendermint团队开发的跨链生态系统，旨在创建一个由独立区块链组成的网络. Cosmos通过Tendermint共识协议和IBC（inter-blockchain communication）协议来实现区块链之间的互操作. Tendermint提供高性能的共识机制，而IBC则实现了不同区块链之间的安全消息传递和数据交换. 这种方法使得Cosmos能够连接各种异构区块链，形成一个互联的区块链网络.

3）Wanchain. Wanchain[28]是一个致力于实现不同区块链之间资产和数据跨链交易的平台. Wanchain通过锁定账户机制和分布式密钥生成技术来实现跨链资产转移. 具体而言，Wanchain会在来源链和目标链上创建锁定账户，并使用多方安全计算（MPC）生成分布式密钥，确保跨链转移的安全性和去中心化. 这个过程通过跨链桥（Bridge）实现，保证了多种资产在不同区块链之间的流通.

4）BitXHub. BitXHub[29]是由趣链科技公司开发的跨链平台，旨在实现不同区块链之间的互操作. BitXHub采用了多层架构设计，通过中继链和跨链网关（cross-chain gateway）实现跨链通信和数据交换. 中继链作为核心组件，负责协调和验证跨链交易，确保跨链操作的安全性和一致性. 跨链网关则作为接口，连接不同的区块链网络，支持多种跨链操作，包括资产转移、智能合约调用和数据共享. BitXHub的设计

使其能够高效地处理跨链交易，并且具有良好的可扩展性，适用于多种跨链应用场景.

2 数据要素交易审计方案概述

2.1 系统模型与工作流程

本文方案的主要应用场景为跨联盟链的数据交易，如图 1 所示，方案涉及 5 个实体，其中 4 个主要实体分别是：数据要素分享者、数据要素购买者、分享者所在联盟链(记为“联盟链 A”)，以及中继链. 购买者所在的联盟链(记为“联盟链 B”)并不是主要实体，不会直接参与方案.

1)联盟链 A. 联盟链 A 是一条存储了其用户数据要素凭证的许可链. 对于一个用户，其存储的数据要素 D 的凭证是三元组：$(ID_D, Hash(D), Sig(Hash(D)))$，其中，$ID_D$是数据要素 D 的标识，具有唯一性. $Sig(Hash(D))$是用户通过私钥进行签名后的数据，可用用户公钥验证. 联盟链 A 之外的用户不能直接访问联盟链 A 并获取其上存储的数据信息.

2)中继链. 中继链是一条公开的非许可链，中继链可通过由联盟链 A 维护的跨链网关访问联盟链 A 的账本，并调取存储信息.

3)数据要素分享者. 数据要素分享者是联盟链 A 里的一名用户，他是数据要素 D 的所有者，并在联盟链 A 上留有数据要素 D 的存证. 分享者持有数据要素 D 的所有权，可以出售数据要素 D.

4)数据要素购买者. 数据要素购买者是联盟链 B 上的一名用户，他本身不是联盟链 A 的用户，不能直接访问联盟链 A. 购买者想要购买联盟链 A 上的数据要素 D.

在本文提出的方案中，由于联盟链 A 是一条许可链，其外部用户(包括联盟链 B 的用户)无法直接访问其存储的数据. 这一限制使得数据要素 D 的隐私和安全需通过联盟链 A 上的存证和签名机制来保障. 购买者必须信任联盟链 A 的安全性，以确保数据要素 D 存证的真实性. 此外，方案依赖于中继链系统的功能和安全性. 中继链系统必须能够确保数据在传输过程中的一致性和完整性，防止数据丢失或篡改. 因此，一个安全高效的中继链系统是该方案成功实施的关键因素之一.

下面结合图 1 概述系统工作流程：

步骤 1. 分享者在联盟链 A 上留存数据要素 D 的存证. 跨链工具是中继链的组成部分，用于监听、访问、查询联盟链 A 的账本.

步骤 2. 分享者和购买者根据安全参数生成一致的公共参数 .

步骤 3. 分享者和购买者在中继链上部署方案中智能合约的代码. 分享者部署质押追责合约、跨链查询合约、审计合约的代码，购买者部署支付合约代码.

步骤 4. 分享者计算数据要素 D 的 Pedersen 承诺并上传到中继链.

步骤 5. 分享者使用 Shamir 秘密分享算法将数据要素 D 分成 2 个秘密分片，同时生成秘密分片的标签.

步骤 6. 分享者在链下将秘密分片、加密秘密分片以及秘密分片标签发送给购买者.

步骤 7. 购买者验证接收到的秘密分片信息是否与链上承诺一致.

步骤 8. 购买者生成接收到秘密分片信息的收据证明，并将收据输入到分享者在中继链上发布的审计合约实例中，然后等待分享者验证.

步骤 9. 当分享者完成验证后，在中继链上发布质押追责合约实例，并向合约中质押代币.

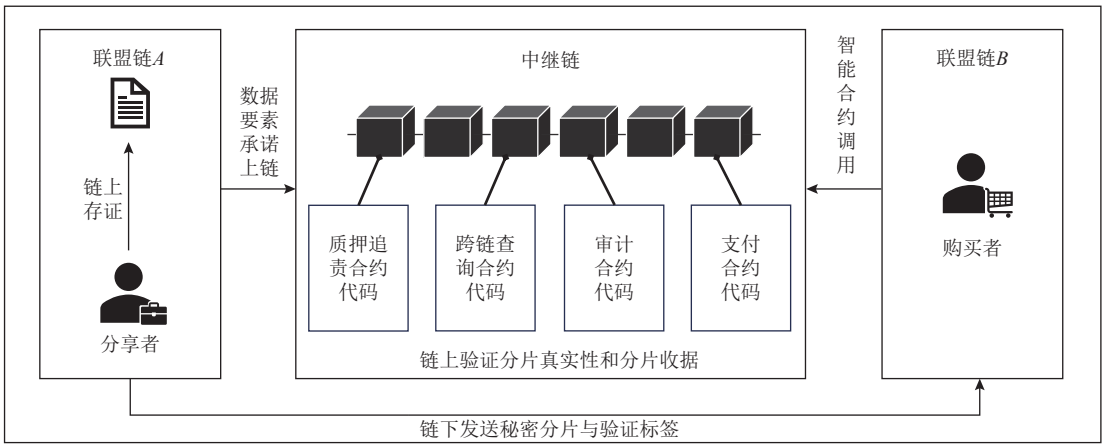

Fig. 1 System model

图 1 系统模型

步骤 10. 购买者验证质押追责合约实例内容合理无误后，在中继链上发布支付合约实例并存入支付代币.

步骤 11. 分享者验证支付合约实例内容合理无误后，输入加密秘密分片的明文. 购买者得到明文后结合另一秘密分片恢复数据要素 D，之后调用中继链上的跨链查询合约，查询数据要素 D 的存证.

步骤 12. 购买者根据查询结果验证数据要素 D 的正确性. 若不正确，购买者调用质押追责合约获取分享者的质押代币.

2.2 威胁模型

本节定义了每个实体的可能行为，并考虑了针对方案的可能攻击及其目的.

1）分享者被认为是潜在的恶意方，可能会故意发送错误或无用的数据要素 D' 以欺骗购买者并收取金额.

2）购买者被认为是潜在的恶意方，可能试图在交易进行过程中提前获取交易的数据要素 D 并且支付较少金额或不支付金额. 另外，在交易完成后，购买者可能会否认收到完整或正确的数据要素 D 以试图收回已支付的资产.

3）中继链被定义为可以信任的一方，中继链上的用户可以通过跨链网关获取联盟链 A 上存储的数据要素凭证信息 . 购买者和分享者可以自由地访问中继链.

4）联盟链 A 被定义为可以信任的一方，链上存储的数据要素凭证(ID_D, $Hash(D)$, $Sig(Hash(D))$)具有真实性.

假设以上任何一方都不会做出损害其利益的行为. 此外，假设中继链上的智能合约正确运行，并且联盟链 A 和中继链上的交易信息记录准确.

2.3 算法与智能合约描述

首先，我们说明本文方案主要的参数符号定义，具体如表 1 所示. 然后，本文方案包含的 10 个算法描述为：

1）Pedersen 承诺初始化 $PedSetup(1^\lambda)\rightarrow(G, \lambda, p, q, g, h)$. 分享者和购买者执行该算法. 该算法以安全参数 1^λ 为输入，输出 Pedersen 承诺公共参数 (G, λ, p, q, g, h). 具体计算过程为：

选定用于确定安全性的安全参数 λ. 选定位长为 λ 位的大素数 p，选定另一个大素数 q，并且 q 是 $p-1$ 的一个大素数因子. 选定循环群 G 中的一个生成元 g，其中 G 为模 p、阶为 q 的循环子群. h 是 G 中的另一个生成元，满足 $h = g^x$，其中 x 是在 $\mathbb{Z}_q$ 中随机选择的一个非零秘密值.

Table 1 Description of Parameter Symbol

表 1 参数符号说明

参数符号	参数说明
λ	安全参数
g/g_i	群G/G_i的生成元
h	群 G 的另一生成元
q	群 G 的阶
s	群G_i的阶
e	定义在$G_1\times G_2\rightarrow G_T$上的双线性映射
H_i	定义在不同域上的哈希函数
U	进行分片收据验证的公钥
α	进行分片收据验证的私钥
D	数据要素
CM	数据要素的 Pedersen 承诺
r	生成 Pedersen 承诺所需的随机数
f_i	定义在模 q 整数环上的 1 阶多项式函数
P_i	数据分片
EP	数据分片 P_2 的密文分片
$t/\hat{t}$	时间戳和时间戳的哈希
σ_i	数据分片的标签
π	秘密分片收据

2）完整性审计初始化 $AudSetup(1^\lambda)\rightarrow(pp)$. 分享者和购买者执行该算法. 该算法以安全参数 1^λ 为输入，输出完整性审计公共参数集 pp. 具体计算过程为：

选定循环群 G_1, G_2 的生成元 g_1, g_2，e 是定义在 $G_1\times G_2\rightarrow G_T$ 上的双线性映射，s 是循环群 G_1, G_2, G_T 的阶. 选择密码学哈希函数 $H_1:\{0,1\}^*\rightarrow G_1$ 和 $H_2:\{0,1\}^*\rightarrow\mathbb{Z}_s$.

3）公私钥对生成 $KeyGen(1^\lambda, pp)\rightarrow(U,\alpha)$. 分享者执行该算法. 该算法以安全参数和完整性审计公共参数集 pp 为输入，输出用于进行完整性审计的公私钥对 (U,α)，具体计算过程为：

选定 $\alpha\in\mathbb{Z}_s$，然后计算 $U = g_2^\alpha$.

4）Pedersen 承诺生成 $PedGen(D,r)\rightarrow(CM)$. 分享者执行该算法. 该算法以数据要素 D 和随机数 r 为输入，输出数据要素 D 的 Pedersen 承诺 CM. 具体计算过程为：

分享者将数据要素 D 映射到 $\mathbb{Z}_q$ 上，并选择随机数 $r\in\mathbb{Z}_q$，生成 Pederson 承诺：

$$CM = g^D h^r \bmod p. \tag{3}$$

5）Shamir 秘密共享数据分片 $ShaCut(D, r, f_1(x), f_2(x))\rightarrow(P_1, P_2)$. 分享者执行该算法. 该算法以数据要素 D、随机数 r、1 阶多项式 $f_1(x)$ 和 $f_2(x)$ 为输入，输出数据要素 D 和随机数 r 的 Shamir 秘密共享的分片 P_1，

P_2. 具体计算过程为：

对于数据要素 D, 随机数 r, 分享者在$\mathbb{Z}_q$上定义 2 个不同的 1 阶函数：

$$f_1(x)=ax+D \bmod q, \tag{4}$$

$$f_2(x)=bx+r \bmod q, \tag{5}$$

分别在$f_1(x), f_2(x)$上各取 2 个点，生成 2 组秘密分片：

$$P_1=(1,f_1(1),f_2(1)), \tag{6}$$

$$P_2=(2,f_1(2),f_2(2)), \tag{7}$$

即，在 $f_1(x)$上取点 $(1,f_1(1))$，$(2,f_1(2))$，在 $f_2(x)$上取点 $(1,f_2(1))$，$(2,f_2(2))$.

6)数据分片加密 $ShareEnc(P_2)\rightarrow(EP)$. 分享者执行该算法. 该算法以 Shamir 秘密共享的分片 P_2 为输入，输出 P_2 的秘文分片 EP. 具体计算过程为：

计算分片 P_2 的密文值 E_2：

$$E_2=g^{f_1(2)}h^{f_2(2)}, \tag{8}$$

得到密文分片 EP：

$$EP=(2,E_2). \tag{9}$$

7)秘密分片标签生成 $TagGen(P_1,EP,\alpha,t)\rightarrow(\sigma_1,\sigma_2,\hat{t})$. 分享者执行该算法. 该算法以数据分片 P_1、密文分片 EP、私钥α和时间戳 t 为输入，生成标签σ_1、σ_2和时间戳因子$\hat{t}$. 具体流程为：

随机选择$u\in G_1$，计算$\hat{t}=H_2(t)$. 然后计算标签

$$\sigma_1=\left(H_1(1)\cdot u^{H_2(f_1(1)+f_2(1))}\right)^{\frac{1}{\alpha+\hat{t}}}, \tag{10}$$

$$\sigma_2=\left(H_1(2)\cdot u^{H_2(E_2)}\right)^{\frac{1}{\alpha+\hat{t}}}. \tag{11}$$

8)数据分片恢复 $ShaRec(P_1, EP)\rightarrow(CM')$. 购买者执行该算法. 该算法以数据分片 P_1 和密文分片 EP 为输入，输出还原的 Pedersen 承诺 CM'. 具体计算公式为：

$$CM'=g^{f_1(1)\times 2}h^{f_2(1)\times 2}\times E_2^{-1}. \tag{12}$$

9)分片收据生成 $ProofGen(P_1,EP,\eta_i,\sigma_i,\hat{t})\rightarrow(\pi)$. 购买者执行该算法. 该算法以数据分片 P_1、密文分片 EP、随机系数$\eta_i\in\mathbb{Z}_s$、标签σ_i和时间戳因子$\hat{t}$为输入，输出分片收据$\pi=(E_c,E_d,E'_c,E'_d,\mu,\kappa,\zeta)$. 具体计算过程为：

首先生成随机数$c,d\in\mathbb{Z}_s$，然后计算

$$E_c=g_2^c, E'_c=g_1^{c^{-1}}, \tag{13}$$

$$E_d=g_2^d, E'_d=g_1^{d^{-1}}, \tag{14}$$

$$\mu=c+H_2(f_1(1)+f_2(1))\eta_2+H_2(E_2)\eta_1, \tag{15}$$

$$\kappa=d+\hat{t}\eta_1\eta_2, \tag{16}$$

$$\zeta=\sigma_1^{\eta_1^{-1}}\sigma_2^{\eta_2^{-1}}. \tag{17}$$

10)分片收据验证 $ProofVer(\pi)\rightarrow(0/1)$. 分享者执行该算法，该算法以分片收据$\pi$为输入，输出验证结果. 具体计算过程为：

$$e(E'_c,E_c)=e(g_1,g_2), \tag{18}$$

$$e(E'_d,E_d)=e(g_1,g_2), \tag{19}$$

$$e(H_1(1)^{\eta_2}H_1(2)^{\eta_1}u^{\mu},g_2)=e(u,E_c)\cdot e\left(\zeta,\frac{g_2^{\kappa}}{E_d}U^{\eta_1\eta_2}\right), \tag{20}$$

全部正确返回 1，否则返回 0.

除了以上 10 个算法外，本文方案还设计并使用了 4 种智能合约，具体介绍为：

1)审计合约. 该合约由分享者发布、购买者调用，用于生成随机系数并接受购买者生成的分片收据π.

审计合约逻辑为：

①随机选择$\eta_1,\eta_2\in\mathbb{Z}_s$，购买者根据随机系数计算分片收据$\pi=(E_c,E_d,E'_c,E'_d,\mu,\kappa,\zeta)$并提交.

②等待分享者验证分片收据.

③分享者返回验证结果.

2)跨链查询合约. 该合约由购买者发布，被质押追责合约和购买者调用，用于查询分享者在联盟链 A 上存储的对应于标识 ID_D的数据要素凭证$(ID_D, Hash(D), Sig(Hash(D)))$.

跨链查询合约逻辑为：

①输入目标联盟链标识 CID, 查询参数 $Query=ID_D$(数据要素 D 在联盟链上的存证标识). 通过与标识为 CID 的联盟链相连接的跨链网关，查询该联盟链的账本，根据查询参数 $Query$ 查找数据要素 D 在联盟链上的存证$(ID_D, Hash(D), Sig(Hash(D)))$.

3)质押追责合约. 该合约由分享者发布、购买者调用，用于当分享者分享的 D'与其在联盟链 A 上存储的数据要素 D 的凭证不符时，对分享者进行惩罚.

质押追责合约逻辑为：

①合约发布者(即分享者)向合约中存入与数据要素 D 价格等值的代币(数额为 M)并锁定这笔质押.

②调用跨链查询合约，输入ID_D以获得数据要素 D 的联盟链存证 $Hash(D)$ 与数字签名$Sig(Hash(D))$.

③若赔付条件满足，质押代币转账到购买者账户(地址). 否则，等待到时刻 T_1 后，锁定的质押代币退回给分享者.

赔付条件为合约发布后的时刻 T_1 前，合约接收到来自购买者的输入(D', r')，同时满足 2 个条件：

①$g^{D'}h^{r'}=CM$(CM 为分享者发布在中继链的数据要素 D 的承诺值)；

② $Hash(D')\neq Hash(D)$.

4)支付合约. 该合约由购买者发布、分享者调用，用于购买者接收 Shamir 秘密共享分片并支付代币.

支付合约逻辑为：

①合约发布者(即购买者)向合约中存入与数据要素 D 价格等值的购资代币(数额为 M)并锁定这笔购资代币；

②若预支付条件满足，等待到时刻 T_3 后，将锁定的购资代币转账给分享者账户(地址). 否则，等待到时刻 T_2 后，锁定的购买代币退回给购买者.

预支付条件为合约发布后的时刻T_3前，合约接收到来自分享者的输入 (y, r_y), 同时满足 $g^y h^{r_y} = E_2$, E_2 为购买者在承诺证明验证过程中收到的密文分片 EP 中的分片承诺值：

$$E_2 = g^{f_1(2)} h^{f_2(2)}, \tag{21}$$

且时间锁关系满足 $T_3 \ll T_2 \ll T_1$.

3 数据要素交易审计方案

本文方案分为 4 个阶段，即初始化阶段、预计算阶段、审计阶段和支付阶段.

3.1 初始化阶段

在初始化阶段，分享者和购买者将生成对应的参数和公私钥.

1)分享者和购买者调用算法 $PedSetup(1^\lambda)\rightarrow(G, \lambda, p, q, g, h)$ 生成 Pedersen 承诺公共参数 (G, λ, p, q, g, h).

2)分享者和购买者调用算法 $AudSetup(1^\lambda)\rightarrow(pp)$ 生成完整性审计公共参数集 pp.

3)分享者调用算法 $KeyGen(1^\lambda, pp)\rightarrow(U,\alpha)$. 生成用于进行分片收据验证的公私钥对.

3.2 预计算阶段

在预计算阶段，主要实现了分享者对数据要素 D 的 Pederson 承诺与 Shamir 秘密分享，以及对 Shamir 秘密分享分片的加密和标签生成，之后通过链上和链下的方式发送对应的信息.

1)分享者调用算法 $PedGen(D, r)\rightarrow(CM)$ 计算数据要素 D 的 Pedersen 承诺.

2)分享者调用算法 $ShaCut(D, r, f_1(x), f_2(x))\rightarrow(P_1, P_2)$ 把数据要素 D 用 Shamir 秘密共享方法进行切割.

3)分享者调用算法 $ShareEnc(P_2)\rightarrow(EP)$ 对分片 P_2 进行加密.

4)分享者调用算法 $agGen(P_1, EP, \alpha, t)\rightarrow(\sigma_1, \sigma_2, \hat{t})$ 对 P_1, EP生成标签.

5)分享者在中继链上将承诺值 CM 发布.

6)分享者将明文秘密分片 P_1、密文分片 EP 作为承诺 CM 的证明，在链下发送给购买者.

7)分享者在中继链上发布审计合约，部分参数由分享者在发布时填写完成.

3.3 审计阶段

在审计阶段，主要实现了购买者对 Pederson 承诺的验证，对 Shamir 秘密分享分片收据的生成.

1)购买者调用算法 $ShaRec(P_1, EP,)\rightarrow(CM')$ 根据接收到的明文分片 $P_1=(1, f_1(1), f_2(1))$、密文分片 $EP_2=(2, E_2)$，计算承诺 CM'.

2)购买者验证 CM'是否等于中继链上分享者的承诺 CM. 若相等，则证明购买者收到的数据要素 D 的明文分片 P_1、密文分片 EP 与承诺 CM 相符，交易继续；若不等，则证明分享者提供了假的分片信息，交易终止. 具体理由为：

$$\begin{aligned} CM' =& g^{f_1(1)\times 2} h^{f_2(1)\times 2} \times E_2^{-1} = g^{f_1(1)\times l_1(0)} h^{f_2(1)\times l_1(0)} \times E_2^{l_2(0)} = \\ & g^{f_1(1)\times l_1(0)} h^{f_2(1)\times l_1(0)} \times \left(g^{f_1(2)} h^{f_2(2)}\right)^{l_2(0)} = \\ & g^{f_1(1)\times l_1(0)+f_1(2)\times l_2(0)} h^{f_2(1)\times l_1(0)+f_2(2)\times l_2(0)} = g^{D'} h^{r'}, \end{aligned} \tag{22}$$

其中，D', r'为根据拉格朗日插值法还原的 D, r.

因为分片时在 $f_1(x)$与 $f_2(x)$上取点的横坐标相同(同为 1 和 2)，所以 $f_1(x)$上的点 $(1, f_1(1))$与 $f_2(x)$上的点 $(1, f_2(1))$有相同的拉格朗日基函数：

$$l_1(x) = \frac{x-2}{1-2}. \tag{23}$$

点 $(2, f_1(2))$与点 $(2, f_2(2))$有相同的拉格朗日基函数：

$$l_2(x) = \frac{x-1}{2-1}. \tag{24}$$

当 $x=0$时，$l_1(0) = \dfrac{0-2}{1-2} = 2$, $l_2(0) = \dfrac{0-1}{2-1} = -1$.

因此购买者可以在不了解数据要素 D 的情况下通过 CM'与 CM 的关系推断出分享者是否提供了假的 Shamir 秘密共享分片信息.

3)购买者查看审计合约获取计算所需参数，然后调用算法 $ProofGen(P_1, EP, \eta_i, \sigma_i, \hat{t})\rightarrow(\pi)$ 计算秘密分片收据 $\pi=(E_c, E_d, E'_c, E'_d, \mu, \kappa, \zeta)$. 计算完成后购买者调用审计合约输入分片收据$\pi$.

4)购买者输入分片收据π后，分享者调用算法 $ProofVer(\pi)\rightarrow(0/1)$ 验证购买者提交的分片收据π.

3.4 支付阶段

在支付阶段，主要完成分享者对数据要素 D 的分享、购买者代币的支付，以及分享者进行作恶行为后的追责.

1)购买者在中继链上发布跨链查询合约.

2)分享者在中继链上观察到跨链查询合约发布后，检查合约内容，无误则发布质押追责合约并质押

代币,等待购买者发布支付合约.

3)购买者在中继链上观察到质押追查合约发布后,检查合约内容,无误则发布支付合约并存入代币,等待分享者输入秘密分片 EP.

4)当支付合约满足预支付条件后,购买者可还原出 D' 与 r',购买者调用跨链查询合约,本地计算 $Hash(D')$ 进行验证,若 $Hash(D')\neq Hash(D)$,调用质押追责合约以获取分享者的质押.否则,交易完成.

4 理论分析

4.1 数据要素隐私性分析

在分享者调用支付合约之前,购买者可以得到分享者发送的明文分片 P_1 与密文分片 EP,其中 P_1 中包含常数项为数据要素 D 的 1 阶函数 $f_1(x)$上的一个点$(1,f_1(1))$,而购买者若想还原出 1 阶函数 $f_1(x)$,还需要 $f_1(x)$上第 2 个不同的点.密文分片 EP 中包含的密文 $E_2=g^{f_1(2)}h^{f_2(2)}$, E_2是一个 G上的 Pedersen 承诺值,Pedersen 承诺的隐藏性确保了购买者无法通过承诺值 E_2 找到承诺的原像与随机值.即,无法找到 E_2 中隐藏的 $f_1(x)$上的第 2 个点的值 $f_1(2)$.从而保证购买者无法仅通过 P_1 和 EP 恢复出数据要素 D.

在分享者调用支付合约后,由于明文分片 P_1 由分享者链下发送给购买者,因此只有购买者拥有明文分片 P_1.在预支付条件达成后,只有购买者同时拥有 2 个分片的明文,因此只有购买者可以恢复出数据要素 D.对于中继链上的其他用户,仅可以通过支付合约查询到一个明文分片 P_2,无法获取 P_1,不能还原出数据要素 D.

4.2 数据要素正确性分析

由于购买者在审计阶段可以对数据要素的承诺进行验证,因此当分享者在中继链上做出了数据要素 D 的承诺后,分享者不能发送错误的分片,使得购买者在可以成功验证承诺 CM 的同时无法恢复数据要素 D.

若分享者在最初对错误的数据要素 D'做出了承诺,购买者在发布支付合约后,分享者将假数据要素 D'的密文分片信息输入到支付合约中.由于中继链上的承诺与假数据要素 D'相符,所以该输入可以满足支付合约中的预支付条件,等待时刻 T_3 后,即可收到购买者的代币.

然而,在时刻 T_3 前,预支付条件达成后(支付合约由购买者发布的,因此 T_3 是由购买者设置的),购买者得到 D'的密文分片后,可以还原出 D'与随机数 r',并可调用跨链查询合约获得真实数据要素 D 的联盟链存证 $Hash(D)$ 与数字签名 $Sig(Hash(D))$,通过对比 $Hash(D')$ 与 $Hash(D)$,购买者确认自己拿到的数据要素 D'是否有误.若有误,购买者可将 D'与随机数 r' 输入到质押追责合约中,当满足质押追责中的赔付条件时,购买者可获得质押追责合约中的质押代币.

在此过程中,分享者从支付合约中获得了购买者锁定的购资代币,购买者通过质押追责合约获得了分享者锁定的质押代币,由于设定的质押代币与购资代币等值,均等于数据要素 D 的价格,因此购买者没有因为分享者作弊而受到损失.若要惩罚作弊者,可要求质押代币数额大于数据要素 D 的价格.

4.3 智能合约时间锁分析

1) 质押追责合约中的时刻 T_1 虽然由分享者设定,但若分享者故意将时刻 T_1 设置不合理(如太小),购买者检查质押追责合约后就不会发布支付合约,分享者则无法获得购买者的支付代币.

2)支付合约中的时刻 T_2 由购买者设定,若购买者故意将时刻 T_2 设置不合理,分享者就不会输入密文分片的解,购买者则无法获得数据要素 D.

3)支付合约中的时刻 T_3 由购买者设定,用于购买者验证分享者在中继链承诺的数据要素 D 是否与联盟链 A 上的存证相符以及不符时的追责.而若购买者故意将时刻 T_3 设置不合理(如太接近 T_2),分享者就不会输入密文分片的解,购买者则无法获得数据要素 D.

4.4 分片收据有效性分析

1)审计合约的有效性

首先说明验证公式的正确性:

$$e(E_c',E_c)=e(g_1^{c^{-1}},g_2^c)=e(g_1,g_2),\tag{25}$$

$$e(E_d',E_d)=e(g_1^{d^{-1}},g_2^d)=e(g_1,g_2),\tag{26}$$

$$e(u,E_c)\cdot e\left(\zeta,\frac{g_2^{\kappa}}{E_d}U^{\eta_1\eta_2}\right)=e(u,g_2^c)\cdot e\left(\sigma_1^{\eta_1^{-1}}\sigma_2^{\eta_2^{-1}},g_2^{\hat{\eta}_1\eta_2}g_2^{\alpha\eta_1\eta_2}\right)=$$
$$e(u^c,g_2)\cdot e\left(\left(\sigma_1^{\eta_1^{-1}}\sigma_2^{\eta_2^{-1}}\right)^{(\hat{\iota}+\alpha)\eta_1\eta_2},g_2\right)=$$
$$e\left(u^c\left(H_1(1)u^{H_2(f_1(1)+f_2(1))}\right)^{\eta_2}\left(H_1(2)u^{H_2(E_2)}\right)^{\eta_1},g_2\right)=$$
$$e(H_1(1)^{\eta_2}H_1(2)^{\eta_1}u^{\mu},g_2).\tag{27}$$

分享者需要正确的分片收据以防购买者抵赖未收到秘密分片,因此分享者会根据购买者提交的分片收据诚实地返回验证结果.

2)分片收据的不可伪造性

在双线性群中,若满足 SDH 假设和 DP 假设,则购买者在没有完整数据要素分片的情况下,无法伪造分片收据.

引理 1. 在双线性群中，若 SDH 问题是一个困难问题，则在审计阶段，购买者难以找到一个收据$\pi'=(E_c,E_d,E'_c,E'_d,\mu,\kappa,\zeta')$，其中$\zeta'\neq\zeta$，使得审计合约输出验证成功.

证明. 假设购买者可以提出一个分片收据$\pi'=(E_c,E_d,E'_c,E'_d,\mu,\kappa,\zeta')$，其中$\zeta'\neq\zeta$，使得审计合约输出验证成功.

当分享者获得一个 SDH 问题$\left(g_1,g_1^x,g_1^{x^2},\cdots,g_1^{x^s},g_2,g_2^x\right)$时，其目标是计算$\left(c,g_1^{\frac{1}{x+c}}\right)\in\mathbb{Z}_s\times G_1$. 为计算 SDH 问题，分享者将进行如下操作：

分享者设G_1,G_2的生成元为g_1,g_2，选择$x\in\mathbb{Z}_s$作为私钥，将$U=g_2^x$设置为公钥，并将公钥返回给购买者. 设置$u=g_1^{\gamma},\gamma\in\mathbb{Z}_s$，为每个分片随机选择$r_i\in\mathbb{Z}_s$，并令：

$$H_1(1)=\frac{g_1^{r_1\left(f(x)+f(\hat{t})\right)}}{g_1^{\gamma H_2(f_1(1)+f_1(2))}},\quad H_1(2)=\frac{g_1^{r_2\left(f(x)+f(\hat{t})\right)}}{g_1^{\gamma H_2(E_2)}},\tag{28}$$

其中$f(\cdot)$是q次多项式，对于每个分片，计算：

$$\sigma_1=\left(H_1(1)u^{H_2(f_1(1)+f_1(2))}\right)^{\frac{1}{x+\hat{t}}}=g_1^{r_1F(x)},\tag{29}$$

$$\sigma_2=\left(H_1(2)u^{H_2(E_2)}\right)^{\frac{1}{x+\hat{t}}}=g_1^{r_2F(x)},\tag{30}$$

其中$F(x)=\dfrac{f(x)+f(\hat{t})}{x+\hat{t}}$是一个次数为$q-1$的多项式.

为了通过审计合约，购买者响应分片收据$\pi'=(E_c,E_d,E'_c,E'_d,\mu,\kappa,\zeta')$，其中$\zeta'\neq\sigma_1^{\eta_1^{-1}}\sigma_2^{\eta_2^{-1}}$. 设：

$$E_c=g_2^{c+xc'},\quad E'_c=g_1^{\frac{1}{c+xc'}},\tag{31}$$

$$E_d=g_2^{d+xd'},\quad E'_d=g_1^{\frac{1}{d+xd'}},\tag{32}$$

若c'或d'不为 0，则$(c/c',E_c'^{c'})$或$(d/d',E_d'^{d'})$是给定 SDH 问题实例的解. 若$c'=d'=0$，令：

$$A=\gamma(\mu-c-(\eta_2H_2(f_1(1)+f_1(2))+\eta_1H_2(E_2))).\tag{33}$$

由式(20)可得：

$$e(\zeta'^{\eta_1\eta_2},g_2)=e\left(g_1^{\frac{\eta_2r_1\left(f(x)+f(\hat{t})\right)+\eta_1r_2\left(f(x)+f(\hat{t})\right)+A}{x+(\kappa-d)/\eta_1\eta_2}},g_2\right).\tag{34}$$

进一步有：

$$\zeta'^{\eta_1\eta_2}=g_1^{\frac{(\eta_2r_1+\eta_1r)\left(f(x)+f(\hat{t})\right)+A}{x+(\kappa-d)/\eta_1\eta_2}}=g_1^{(\eta_2r_1+\eta_1r)F(x)+\frac{(\eta_2r_1+\eta_1r)\left(f(\hat{t})-f((\kappa-d)/\eta_1\eta_2)\right)+A}{x+(\kappa-d)/\eta_1\eta_2}}.\tag{35}$$

因此，可以计算给定 SDH 问题的解：

$$\left(\frac{\kappa-d}{\eta_1\eta_2},\left(\frac{\zeta'^{\eta_1\eta_2}}{g_1^{(\eta_2r_1+\eta_1r)F(x)}}\right)^{\frac{1}{(\eta_2r_1+\eta_1r)\left(f(\hat{t})-f((\kappa-d)/\eta_1\eta_2)\right)+A}}\right).\tag{36}$$

这与 SDH 在双线性群中是一个困难问题矛盾，因此假设不成立. 证毕.

引理 2. 在双线性群中，若 DP 问题是一个困难问题，则在审计阶段，购买者难以找到一个收据$\pi'=(E_c,E_d,E'_c,E'_d,\mu',\kappa,\zeta)$，其中$\mu'\neq\mu$，使得审计合约输出验证成功.

证明. 假设购买者可以提出一个分片收据$\pi'=(E_c,E_d,E'_c,E'_d,\mu',\kappa,\zeta)$，其中$\mu'\neq\mu$，使审计合约输出验证成功.

当分享者获得一个 DP 问题$(g_r,g_t)\in G_1^2$时，其目标是找到满足$e(g_r,R)e(g_t,T)=1$的(R,T). 为计算 DP 问题，分享者将进行如下操作：

分享者设G_1,G_2的生成元为g_1,g_2，选择$x\in\mathbb{Z}_s$作为私钥，将$U=g_2^x$设置为公钥，并将公钥返回给数据接收者. 设置$u=g_r^{\gamma}g_t^{\iota}$. 对于每个分片，计算：

$$\sigma_1=\left(H_1(1)\cdot u^{H_2(f_1(1)+f_2(1))}\right)^{\frac{1}{\alpha+\hat{t}}},\tag{37}$$

$$\sigma_2=\left(H_1(2)\cdot u^{H_2(E_2)}\right)^{\frac{1}{\alpha+\hat{t}}}.\tag{38}$$

为了通过审计合约，购买者响应分片收据$\pi'=(E_c,E_d,E'_c,E'_d,\mu',\kappa,\zeta)$. 由式(20)可得：

$$\begin{aligned}1=&e\left(\zeta^{-1},\frac{g_2^{\kappa}}{E_d}U^{\eta_1\eta_2}\right)\cdot e\left(H_1(1)^{\eta_2}H_1(2)^{\eta_1}u^{\mu'},g_2\right)\cdot e(u,E_c^{-1})=\\&e\left(\zeta^{-1},g_2^{(x+\hat{t})\eta_1\eta_2}\right)\cdot e\left(H_1(1)^{\eta_2}H_1(2)^{\eta_1}u^{\mu'},g_2\right)\cdot e(u,g_2^{-c})=\\&e\left(u^{\mu'-c-(H_2(f_1(1)+f_2(1))\eta_2+H_2(E_2)\eta_1)},g_2\right).\end{aligned}\tag{39}$$

令：

$$R=g_2^{\gamma(\mu'-c-H_2(f_1(1)+f_2(1))\eta_2-H_2(E_2)\eta_1)},\tag{40}$$

$$T=g_2^{\iota(\mu'-c-H_2(f_1(1)+f_2(1))\eta_2-H_2(E_2)\eta_1)},\tag{41}$$

有$1=e(g_r,R)e(g_t,T)$.

因此，可以计算出给定 DP 问题实例的解(R,T). 注意$\mu'\neq\mu$，即：

$$\mu'-c-H_2(f_1(1)+f_2(1))\eta_2-H_2(E_2)\eta_1\neq0.\tag{42}$$

从而(R,T)是非平凡的，这与 DP 问题在双线性群中是一个困难问题矛盾，假设不成立. 证毕.

结合引理 1 和引理 2，在双线性群中满足 SDH 和 DP 假设的情况下，购买者不能在没有完整数据要素分片的情况下伪造分片收据.

5 实验评估

5.1 实验环境设置

本文的实验是在本地主机上开发完成的. 在本地搭建 2 条以太坊[30]私有链，并采用 BitXHub 方案实现跨链通信. 在搭建的 2 条区块链中，一条模拟联盟链 *A*，存储数据要素 *D* 的存证；另一条与 BitXHub 中继链共同形成中继系统，实现了方案所需智能合约的部署与调用. 表 2 展示了本文的整体实验环境，

Table 2 Experimental Environment

表 2 实验环境

配置项目	型号/大小
CPU	Intel Core i7-10750H CPU @ 2.60 GHz
RAM	16.00 GB
OS	64 位 Ubuntu 20.04（VM 虚拟机）
语言	Java 22（JDK 22）
库	JPBC-2.0.0

包括主机的硬件配置和软件环境.

5.2 计算开销

首先对本文方案的计算开销进行分析. 由于跨链操作和智能合约的部署与调用均在本地主机上实现, 与实际环境差异较大, 本文仅测量了方案其余部分的时间开销. 图 2 展示了本文方案各个阶段的平均时间开销, 图 3 展示了方案中各算法在不同运行次数下的总时间开销变化情况.

如图 2 所示, 在初始化阶段, Pedersen 承诺的公共参数选取会导致较大的时间开销. 这是因为在选取大素数 p 的同时, 还需要选取 p–1 的一个大素数

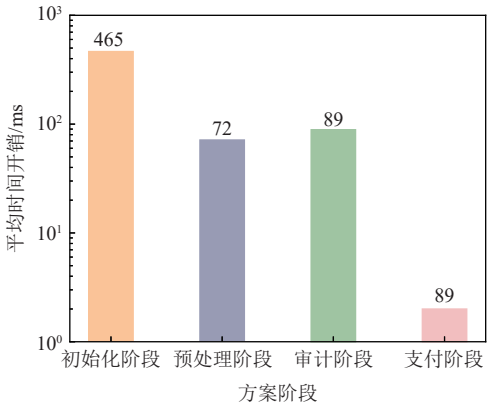

Fig. 2 Time overhead in each stage

图 2 各阶段时间开销

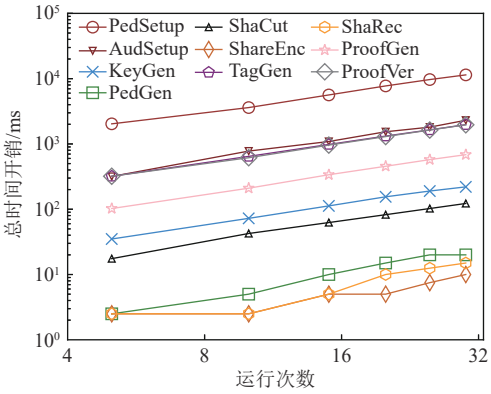

Fig. 3 Variation of time overhead of different algorithms

图 3 不同算法时间开销变化

因子 q, 这使得选取合适的 p 和 q 需要一定的计算量. 在预处理阶段, 主要的计算开销来自 Shamir 分片的标签生成. 这是因为在计算标签时, 需要进行多次群上的指数运算和乘法运算, 而 Pedersen 承诺和 Shamir 秘密分片的计算都是在有限域上进行的, 计算开销较小. 由于标签数量(即 Shamir 秘密共享的分片数量)为 2, 因此预处理阶段的计算开销不大. 同样地, 在审计阶段, 主要的计算开销来自分片收据的生成与验证. 其他算法是在有限域上进行的, 计算开销较小. 最后, 在支付阶段, 大多数操作通过智能合约实现, 数据要素 D 的恢复操作在有限域上进行, 计算开销较小. 此外, 图 3 展示了各算法在运行 5, 10, 15, 20, 25, 30 次情况下的总时间开销. 各个算法的总时间开销与运行次数呈线性关系.

从整体角度来看, 本文方案计算开销相对较小. 在不计入跨链网络的延迟与智能合约的部署与调用的情况下, 单次运行时间开销约为 600 ms, 主要取决于 Pedersen 承诺参数 p 和 q 的寻找速度. 相比于实际跨链场景中的网络延迟以及中继链上的出块确认时间开销(为了安全性考虑, 在以太坊中交易通常需要等待大约 12 个区块才能被确认, 每个区块间隔约 15 s, 总计约 180 s), 本文的跨链数据要素交易审计方案单次运行的时间开销微乎其微.

5.3 链上通信开销

考虑到链上通信代价昂贵, 我们测量了在方案中进行链上信息传输所需的通信开销. 链上通信开销主要包括承诺 CM 和分片收据 π, 表 3 展示了不同安全参数下 CM 和 π 的大小. 在 $\lambda=256$ 的情况下, 链上通信开销总计约为 0.04 KB, 由此可见, 在实际应用中, 本文所提方案的链上通信开销较低.

Table 3 Communication Overhead

表 3 通信开销

λ	CM/KB	π/KB
64	0.01	0.07
128	0.02	0.16
256	0.04	0.33

5.4 智能合约 Gas 成本

我们对方案中的智能合约的 Gas 成本进行了测量. 本文方案包括 4 个智能合约, 分别是审计合约、跨链查询合约、质押追责合约和支付合约. 我们使用 Solidity 语言编写合约, 通过 Remix 进行合约的编译, 并在本地搭建的中继系统上进行了部署和测试, 结果如图 4 所示.

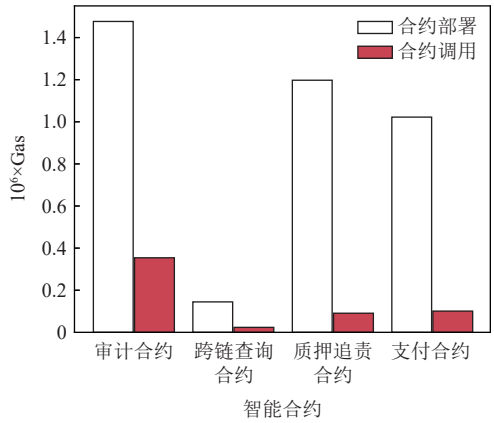

Fig. 4 Gas cost of smart contract

图 4 智能合约 Gas 成本

我们测试了审计合约、跨链查询合约、质押追责合约和支付合约的 Gas 成本. 从图 4 中可以看出这 4 个合约中, 跨链查询合约的 Gas 成本最小. 通过分析合约逻辑, 跨链查询合约仅实现了跨链消息的查询, 因此 Gas 成本较低; 质押追责合约和支付合约的逻辑较为相似, 都需要设置时间锁并进行 2 次群的指数运算和 1 次群的乘法运算, 因此 Gas 成本相近. 而审计合约需要传输较多的计算参数, 所以 Gas 成本较高.

部署以上 4 个智能合约总计消耗了约为 3.8×10^6 Gas. 根据 2024 年 6 月在以太坊的区块链浏览器 Etherscan 上提出的 Gas 价格为 12 Gwei, 其中 Gwei 是以太坊网络中用于表示 Gas 价格的单位, 1 Gwei 表示单位 Gas 的价格是 1 Wei. 部署合同成本约为 4.56×10^{-2} ETH. 后续进行调用时, Gas 成本明显下降.

6 结 论

本文提出了一种面向跨联盟链的隐私保护数据要素交易审计方案, 通过结合跨链技术、零知识证明和智能合约, 实现了数据交易过程中的隐私保护和公平性. 方案通过在联盟链上存储数据存证, 利用中继链上执行的智能合约进行数据验证和支付管理, 结合 Pedersen 承诺和 Shamir 秘密分享技术构建零知识证明, 在保护数据隐私的同时, 审计数据交易的真实性, 防止数据欺诈的发生. 此外引入了一种基于双线性映射的数据要素完整性审计机制, 在不透露数据隐私的情况下验证数据要素的收据, 有效解决支付抵赖问题. 本文方案解决了跨联盟链数据交易中数据购买者和数据分享者之间的信任问题, 从而推动了数据要素的自由流通和利用, 进一步激活数据要素的潜在价值.

未来, 在本文的基础上, 我们考虑从 2 方面进行下一步的工作:

1)数据要素的全生命周期管理. 未来的研究将着眼于数据要素从产生到销毁的全生命周期管理. 这包括数据要素的可追溯性和合规性管理, 以确保数据要素交易的可信性和透明性, 满足数据要素交易的法规要求. 使数据要素交易更加便捷、安全和可靠, 促进数据要素交易的广泛应用和发展.

2)跨链技术与智能合约的进一步研究和优化. 随着区块链技术的不断发展, 跨链技术也在不断演进. 未来将探索更高效、更安全的跨链方案, 以便更好地支持数据要素的流通和交易. 同时考虑进一步优化智能合约, 降低合约 Gas 成本, 增强方案实用性.

作者贡献声明: 张川提出算法思路和实验方案; 王子豪完成实验设计与论文撰写; 梁晋文讨论研究思路、协助论文撰写并参与论文修改; 刘梦轩协助完成实验和论文修改; 邓淏天协助完成实验和论文修改; 祝烈煌提供指导意见.

参 考 文 献

[1] Liu Dongxiao, Huang Cheng, Ni Jianbing, et al. Blockchain-cloud transparent data marketing: Consortium management and fairness[J]. IEEE Transactions on Computers, 2022, 71(12): 3322−3335

[2] Wang Jiye, Gao Lingchao, Dong Aiqiang, et al. Block chain based data security sharing network architecture research[J]. Journal of Computer Research and Development, 2017, 54(4): 742−749 (in Chinese)
(王继业, 高灵超, 董爱强, 等. 基于区块链的数据安全共享网络体系研究[J]. 计算机研究与发展, 2017, 54(4): 742−749)

[3] Cui Jie, Ouyang Fenqiang, Ying Zuobin, et al. Secure and efficient data sharing among vehicles based on consortium blockchain[J]. IEEE Transactions on Intelligent Transportation Systems, 2022, 23(7): 8857−8867

[4] Zhang Duo, Wang Shangping, Zhang Yinglong, et al. A secure and privacy-preserving medical data sharing via consortium blockchain[J]. Security and Communication Networks, 2022, 2022: 2759787

[5] Chen Fei, Wang Jiahao, Jiang Changkun, et al. Blockchain based non-repudiable iot data trading: Simpler, faster, and cheaper[C]//Proc of IEEE INFOCOM 2022-IEEE Conf on Computer Communications. Piscataway, NJ: IEEE, 2022: 1958−1967

[6] Jiang Yiming, Wang Chenxu, Wang Yawei, et al. A cross-chain solution to integrating multiple blockchains for IoT data management[J]. Sensors, 2019, 19(9): 2042

[7] Geng Qian, Ziang C, Jin Jian. Cross-organizational data exchange

based on consortium blockchain with consistency guarantee[J]. The Journal of Supercomputing, 2024, 80: 18199−18236

[8] Liu Yang, Yang Weidong, Wang Yanlin, et al. An access control model for data security sharing cross-domain in consortium blockchain[J]. IET Blockchain, 2023, 3(1): 18−34

[9] Singh P, Masud M, Hossain M S, et al. Cross-domain secure data sharing using blockchain for industrial IoT[J]. Journal of Parallel and Distributed Computing, 2021, 156: 176−184

[10] Pedreira C, Belchior R, Matos M, et al. Securing cross-chain asset transfers on permissioned blockchains[J]. Authorea Preprints, 2023

[11] Jiang Bohao, Li Chaoyang, Tang Yu, et al. Secure Cross-chain transaction for medical data sharing in blockchain-based internet of medical things[C]//Proc of the Int Conf on Frontiers in Cyber Security. Berlin: Springer, 2023: 3−18

[12] Kannoori H, Balaraju P, Harika C, et al. Privacy preservation of insurance data sharing across permissioned blockchains[C]//Proc of the Int Conf on Frontiers in Computing and Systems. Berlin: Springer, 2023: 275−293

[13] de Vos M, Ileri C U, Pouwelse J. XChange: A universal mechanism for asset exchange between permissioned blockchains[J]. World Wide Web, 2021: 1−38

[14] Zhao Fangxin, Yu Jiguo, Yan Biwei. Towards cross-chain access control model for medical data sharing[J]. Procedia Computer Science, 2022, 202: 330−335

[15] Pedersen T P. Non-interactive and information-theoretic secure verifiable secret sharing[C]//Proc of Annual Int Cryptology Conf. Berlin: Springer, 1991: 129−140

[16] Shamir A. How to share a secret[J]. Communications of the ACM, 1979, 22(11): 612−613

[17] Zhang Chuan, Xuan Haojun, Wu Tong, et al. Blockchain-based dynamic time-encapsulated data auditing for outsourcing storage[J]. IEEE Transactions on Information Forensics and Security, 2023, 19: 1979−1993

[18] Boneh D, Boyen X. Short signatures without random oracles[C]// Advances in Cryptology-EUROCRYPT 2004: Int Conf on the Theory and Applications of Cryptographic Techniques. Interlaken, Switzerland, Proceedings 23, 2004: 56−73

[19] Groth J. Homomorphic trapdoor commitments to group elements[R]. Cryptology ePrint Archive, 2009

[20] Si Bingru, Xiao Jiang, Liu Cunyang, et al. Survey on blockchain network[J]. Journal of Software, 2024, 35(2): 773−799 (in Chinese)
(司冰茹, 肖江, 刘存扬, 等. 区块链网络综述[J]. 软件学报, 2024, 35(2): 773−799)

[21] Nakamoto S. Bitcoin: A peer-to-peer electronic cash system[EB/OL]. White paper, 2008[2024-05-30]. http://bitwin.org/bitcoin.pdf

[22] Schär F. Decentralized finance: On blockchain- and smart contract-based financial markets[J]. FRB of St. Louis Review, 2021, 153−174

[23] Rejeb A, Rejeb K, Simske S, et al. Exploring blockchain research in supply chain management: A latent Dirichlet allocation-driven systematic review[J]. Information, 2023, 14(10): 557

[24] Liu P T S. Medical record system using blockchain, big data and tokenization[C]//Proc of the 18th Int Conf on Information and Communications Security (ICICS 2016). Singapore: Springer International Publishing, 2016: 254−261

[25] Li Fang, Li Zhuoran, Zhao He. Research on the progress in cross-chain technology of blockchains[J]. Journal of Software, 2019, 30(6): 1649−1660 (in Chinese)
(李芳, 李卓然, 赵赫. 区块链跨链技术进展研究[J]. 软件学报, 2019, 30(6): 1649−1660)

[26] Wood G. Polkadot: Vision for a heterogeneous multi-chain framework[R]. White paper, 2016, 21(2327): 4662

[27] Kwon J, Buchman E. Cosmos whitepaper: A network of distributed ledgers[R]. White paper, 2019, 27: 1−32

[28] Wanchain. Wanchain whitepaper: A distributed financial infrastructure for digital assets[R]. White paper, 2017

[29] Ye Shaojie, Wang Xiaoyi, Xu Caichao, et al. BitXHub: Side-relay chain based heterogeneous blockchain interoperable platform[J]. Computer Science, 2020, 47(6): 294−302 (in Chinese)
(叶少杰, 汪小益, 徐才巢, 等. BitXHub: 基于侧链中继的异构区块链互操作平台[J]. 计算机科学, 2020, 47(6): 294−302)

[30] Buterin V. A next-generation smart contract and decentralized application platform[R]. White paper, 2014, 3(37): 2−1

Zhang Chuan, born in 1991. PhD, assistant professor. Member of IEEE. His main research interests include cloud computing, applied cryptography, machine learning, and blockchain.
张 川, 1991 年生. 博士, 助理教授. IEEE 会员. 主要研究方向为云计算、应用密码学、机器学习、区块链.

Wang Zihao, born in 2001. Bachelor. His main research interest includes blockchain.
王子豪, 2001 年生. 学士. 主要研究方向为区块链.

Liang Jinwen, born in 1992. PhD, postdoc. Member of IEEE. His main research interests include applied cryptography, AI security, blockchain, and database security.
梁晋文, 1992 年生. 博士, 博士后. IEEE 会员. 主要研究方向为应用密码学、人工智能安全、区块链、数据库安全.

Liu Mengxuan, born in 1993. Master. His main research interests include blockchain and distributed systems.
刘梦轩, 1993 年生. 硕士. 主要研究方向为区块链、分布式系统.

Deng Haotian, born in 1994. Master. Student member of IEEE. His main research interests include blockchain, IoT security, and applied cryptography.

邓淏天, 1994 年生. 硕士. IEEE 学生会员. 主要研究方向为区块链、物联网安全、应用密码学.

Zhu Liehuang, born in 1976. PhD, professor. Senior member of IEEE. His main research interests include security protocol analysis and design, group key exchange protocols, wireless sensor networks, and cloud computing.

祝烈煌, 1976 年生. 博士，教授. IEEE 高级会员. 主要研究方向为安全协议分析与设计、群密钥交换协议、无线传感器网络、云计算.