计算机科学
COMPUTER SCIENCE

基于联盟区块链的数据可信共享方案

刘漳辉，林哲旭，陈汉林，马新建，陈星

引用本文

刘漳辉，林哲旭，陈汉林，马新建，陈星．基于联盟区块链的数据可信共享方案[J]．计算机科学，2025，52(11)：398-407.

LIU Zhanghui, LIN Zhexu, CHEN Hanlin, MA Xinjian, CHEN Xing. Data Trusted Sharing Scheme Based on Consortium Blockchain [J]. Computer Science, 2025, 52(11): 398-407.

相似文章推荐（请使用火狐或 IE 浏览器查看文章）

Similar articles recommended (Please use Firefox or IE to view the article)

利用环盲签名+仲裁的认证混币方案

Using Ring Blind Signature＋Arbitration Authentication Mixed Coin Scheme

计算机科学, 2025, 52(11): 390-397. https://doi.org/10.11896/jsjkx.241000048

SCDDA:基于SCA和Dinkelbach的空-天-地一体化网络无人机轨迹与计算卸载优化方法

SCDDA:SCA and Dinkelbach-based Approach for UAV Trajectory and Computation Offloading in Space-Air-Ground Integrated Networks

计算机科学, 2025, 52(11): 270-279. https://doi.org/10.11896/jsjkx.241100163

区块链共识算法综述

Review of Blockchain Consensus Algorithm

计算机科学, 2025, 52(11): 255-269. https://doi.org/10.11896/jsjkx.241100140

基于区块链的轻量级可验证数据管理方法

Approach for Lightweight Verifiable Data Management Based on Blockchains

计算机科学, 2025, 52(10): 348-356. https://doi.org/10.11896/jsjkx.250200001

多权威可撤销密文策略属性基加密数据共享方案

Multi-authority Revocable Ciphertext-policy Attribute-based Encryption Data Sharing Scheme

计算机科学, 2025, 52(9): 388-395. https://doi.org/10.11896/jsjkx.240700066

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.241000169

基于联盟区块链的数据可信共享方案

刘漳辉[1,2]　林哲旭[1,2]　陈汉林[1,2]　马新建[3,4]　陈　星[1,2]
1 福州大学计算机与大数据学院　福州 350116
2 福建省网络计算与智能信息处理重点实验室(福州大学)　福州 350116
3 数据空间技术与系统全国重点实验室　北京 100195
4 北京大数据先进技术研究院　北京 100195
(lzh@fzu.edu.cn)

摘　要　随着大数据时代的来临,如何在开放、动态、难控的互联网中实现安全可信的数据共享已成为亟待解决的问题。区块链技术以其去中心化、不可篡改等特性,为构建数据可信共享机制提供了技术思路。为此,提出了一种基于联盟区块链的数据可信共享方案。首先,定义了一种基于联盟区块链的数据架构范式,并通过标准化注册流程,高效整合异源、异域、异构的数据资源;同时,设计并实现了数据可信追溯机制,通过数据共享全过程链上留痕的方式,来保证数据需求方、计算节点和数据提供方之间数据流动的安全性和完整性;此外,设计了一种数据处理即服务(Data Processing-as-a-Service,DPaaS)的数据可信共享框架,来支撑数据共享的关键步骤,即需求匹配、数据共享、满意度评价,以应对数据共享过程中的信任挑战。实验结果表明,与传统数据共享方案相比,随着数据集的增大,拟议方案的额外时间占比可降至数据共享总时间开销的30%以内;智能合约查询平均时延能够稳定在0.12～0.2s,智能合约写入平均时延能够稳定在3～5s。

关键词:区块链;智能合约;Hyperledger Fabric;数据共享;可信凭证
中图分类号　TP311

Data Trusted Sharing Scheme Based on Consortium Blockchain

LIU Zhanghui[1,2],LIN Zhexu[1,2],CHEN Hanlin[1,2],MA Xinjian[3,4] and CHEN Xing[1,2]
1 College of Computer and Data Science,Fuzhou University,Fuzhou 350116,China
2 Fujian Key Laboratory of Network Computing and Intelligent Information Processing(Fuzhou University),Fuzhou 350116,China
3 National Key Laboratory of Data Space Technology and System,Beijing 100195,China
4 Advanced Institute of Big Data,Beijing,Beijing 100195,China

Abstract　With the advent of the big data era,securing and trustworthily sharing data on an open,dynamic,and difficult-to-control Internet has become an urgent problem to solve. Blockchain can reasonably be introduced into the trust resolution mechanism for data sharing,leveraging its significant advantages in decentralization and tamper resistance. Thus,a data trusted sharing scheme based on consortium blockchain is proposed. Firstly,a consortium blockchain-based data architecture is defined to solve the problem of heterogeneous data sources and domains. Through standardized registration processes,data resources are integrated efficiently. Secondly,a trusted data traceability mechanism is designed and implemented to ensure the security and integrity of data flow among data requesters,compute nodes,and data providers by leaving the traces of the data sharing process on consortium blockchain. In addition,a data-processing-as-a-service data sharing framework is developed to support key steps in data sharing:demand matching,data sharing,and satisfaction evaluation,addressing trust challenges during the data sharing process. The experimental results show that,compared with traditional data sharing schemes,the proportion of additional time in the proposed scheme decreases to less than 30% of the total time cost as the dataset size increases. Additionally,the average latency for querying smart contracts remains stable between 0.12 and 0.2 seconds,while the average latency for writing smart contracts stays consistent at 3 to 5 seconds.

Keywords　Blockchain,Chaincode,Hyperledger Fabric,Data sharing,Trusted credential

到稿日期:2024-10-29　返修日期:2025-03-03
基金项目:国家自然科学基金(62072108);福建省促进海洋与渔业产业高质量发展专项资金(FJHYF-ZH-2023-02);福建省技术创新重点攻关及产业化项目(2024XQ004);数据空间技术与系统全国重点实验室资助项目(QZQC2024007)
This work was supported by the National Natural Science Foundation of China(62072108),Special Funds for Promoting High-quality Development of Marine and Fishery Industries in Fujian Province(FJHYF-ZH-2023-02),Fujian Key Technological Innovation and Industrialization Projects(2024XQ004) and National Key Laboratory of Data Space Technology and System(QZQC2024007).
通信作者:马新建(maxj@aibd.ac.cn)

1 引言

随着大数据时代的到来，以“数据互联、应需调度”[1]为特征的数字经济正深刻改变着各行各业的商业模式。为了提高生产力及经济效益，近年来出现了许多集中式数据共享平台，如 GXSChain、京东万象、Terbine 等。集中式数据共享平台的涌现，不仅推动了异源、异域、异构数据资源的整合和流通，而且加速了对数据资源的高效处理和共享，极大程度上满足了人们对数据资源共享的效率及成本效益的需求。不过，这些平台多数基于数据托管即服务模式（Data-as-a-Service，DaaS），一旦数据监管力度薄弱，就会存在数据虚假及平台恶意采集并转售用户数据资源的隐患。此外，检测用户数据是否被平台篡改或转售，以及权衡数据共享过程中的责权分配问题，是十分困难的。

区块链技术[2]以其去中心化、不可篡改等特性，为构建数据共享信任机制提供了技术思路。区块链还能够清晰地界定数据的权属，其中涵盖了数据的隐私保护权和所有权等，从而激励数据提供方积极参与到区块链网络中。但是，比特币[3]、以太坊[4]这类公链普遍存在吞吐量低、隐私性差等问题，在大规模数据处理中的应用受到限制。联盟链作为一个很好的解决方案，以 Hyperledger Fabric 为代表，通过灵活地定制和集成各种组件，很好地满足了当前大数据场景下的业务需求。

目前的研究主要集中在利用联盟链构建无第三方介入的数据共享平台，旨在让数据资源能够在不同数据系统之间互联互通。但与此同时，由于区块链采用多副本存储模式，随着网络节点数递增，数据备份和存储成本呈指数增长，这将导致不可容忍的性能下降问题。鉴于链上存储压力和对系统吞吐量的考量，一些学者通过结合 IPFS[5]和 DOIP[6]等标准化传输协议，在链外数据仓库中对链上数据副本进行存储优化，从而更好地实现异构、异域、异主数据的互联、互通、互操作[7]。

以上方案大多聚焦于确保数据交付的完整性，没有考虑到对数据提供方的数据所有权的保护。为此，本文设计了一种基于联盟链的数据可信共享方案，将传统的数据托管即服务模式转变为数据处理即服务模式，即数据提供方并非将原始数据在第三方托管以供数据需求方使用，而是在数据需求方不接触原始数据的前提下为其提供数据资源分析处理的结果，从而实现数据的“可用不可见”，重塑了数据共享过程中双方之间的信任关系。

本文的主要贡献如下：

1）定义了一种基于联盟区块链的数据架构范式（Consortium Blockchain-based Data Architec-ture，CBDA）来标准化数据注册流程，在高效整合和互通数据资源的同时，缓解“数据孤岛”问题。

2）提出了一种基于联盟区块链的数据可信追溯机制，利用数据共享参与方的密钥对数据共享全过程记录签名确责、链上留痕，并利用哈希摘要技术保证了数据共享的完整性。同时，链上仅存储关键注册信息和数据共享记录，减轻了链上存储压力。

3）提出了一种数据处理即服务的数据可信共享框架，通过数据共享关键步骤即需求匹配、数据共享、满意度评价，将数据资源所有权归还给数据提供方，以应对数据共享过程中的信任挑战。

2 相关工作

2.1 联盟区块链和 Hyperledger Fabric

早期设计的传统公链具备记录数据主体的数字化资产在各方之间流动、共享的完整过程的特性。然而，在对等互联的区块链网络中，所有链上信息开放透明，这类方案无法满足企业在真实应用场景下的协助需求。出于对安全性和用户隐私性的考量，联盟链不失为一种卓越的解决方案。联盟链[8]作为传统公链的延伸，旨在通过限制参与者和访问权限，提供更高的安全性、性能和隐私保护。在联盟链中，参与者能够根据业务需求对网络的规则、更新协议和共识算法进行定制和扩展，以适应不同规模和复杂度的应用场景。联盟链借助准入机制，保护多方协作责权在智能合约层面的共识，实现了对区块链网络成员的可控；同时，还能够依据具体应用场景定制化部署不同的隐私保护策略，在保证区块链网络安全性的同时，具备更灵活的可扩展性。

Hyperledger Fabric[9]被视为联盟链的典型应用，是一款由 Linux 基金会发起创建的开源联盟链框架。它提供了一个模块化架构，允许用户灵活地定制和集成各种组件，以满足不同的企业需求。Hyperledger Fabric 被广泛应用于多个场景，如医疗保险[10]、物联网[11]、政务系统[12]、供应链管理[13]等。其主要由 Peer 节点、排序服务节点、客户端、证书机构（Certificate Authority，CA）等组成。其中，Peer 节点分为背书节点和提交节点，负责存储区块链数据，运行维护链码并参与共识过程；排序服务节点负责接收、排序和打包共享，生成区块并将其分发给 Peer 节点；客户端作为用户和 Fabric 网络的媒介，旨在简化用户与 Fabric 网络的交互和操作；CA 是 Hyperledger Fabric 的数字证书认证中心，用于核验区块链网络参与方的证书及身份信息。图 1 展示了 Hyperledger Fabric 的完整工作流程。

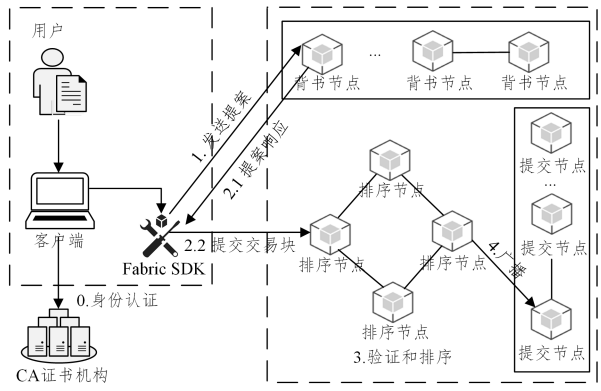


图 1 Fabric 工作流程

Fig. 1 Workflow of Fabric

2.2 研究现状

随着数字经济的发展，许多研究人员通过区块链技术对数据共享方案进行研究。Chen 等[14]提出一种基于区块链的可信 SOA 架构，其通过区块链来实现完整且安全的服务调用过程。Zheng 等[15]提出了一个基于区块链的去中心化数据

共享平台，数据提供方可以通过加密的方式来控制数据的共享；并设计了智能合约来保证数据奖励分配的合理性。然而，现有研究大多侧重于数据共享的安全性和隐私性，并没有充分考虑到区块链网络的存储压力问题，大规模密文上链将导致严重的存储资源消耗和性能瓶颈问题。

一些研究人员舍弃了链上大规模数据存储的方式，转而使用链外存储结合硬件的方式来解决数据传输的安全性问题。Dai 等[16]结合 Intel SGX 和以太坊实现了一个基于区块链的数据共享生态系统，但是，该系统对于硬件部署环境的依赖性太强，无法满足更广泛的应用场景下的扩展性需求，同时针对 SGX Enclave 中原始数据的安全性还有待加强。Wang 等[17]提出一种基于区块链的安全数据共享框架，通过硬件可信执行环境 TEE(Trusted Execution Environment)加密用户数据，同时采用链外存储机制，大幅减少了系统成本开销。

数据的隐私保护同样值得重视，数据所有者应当始终拥有对其数据资源的所有权。Truong 等[18]提出了一个适用于欧盟保护法律的数据管理方案，确保只有指定方可以对数据进行操作，任何违规行为都将被永久记录上链。然而，该方法假设数据代理节点作为可信的第三方，无法真正地应用于开放、动态、难控的互联网，可能带来隐私泄露的风险。

3 数据可信共享方案设计

3.1 框架概述

本文提出了一个基于联盟区块链的数据可信共享框架(Data TrustedSharing Framework，DTSF)，选用 Hyperledger Fabric 作为底层平台来搭建联盟链网络。如图 2 所示，该框架共分为 5 层。其中，数据注册层负责对异源、异域、异构的数据资源进行标准化注册；数据可信层通过链上留痕的方式支持可信追溯操作，并防止恶意参与方篡改；数据共识层通过共识节点共同维护 Fabric 网络上数据的一致性；数据处理层负责处理数据共享过程的流程把控；数据交付层支撑最终数据处理结果的可信交付及评价。

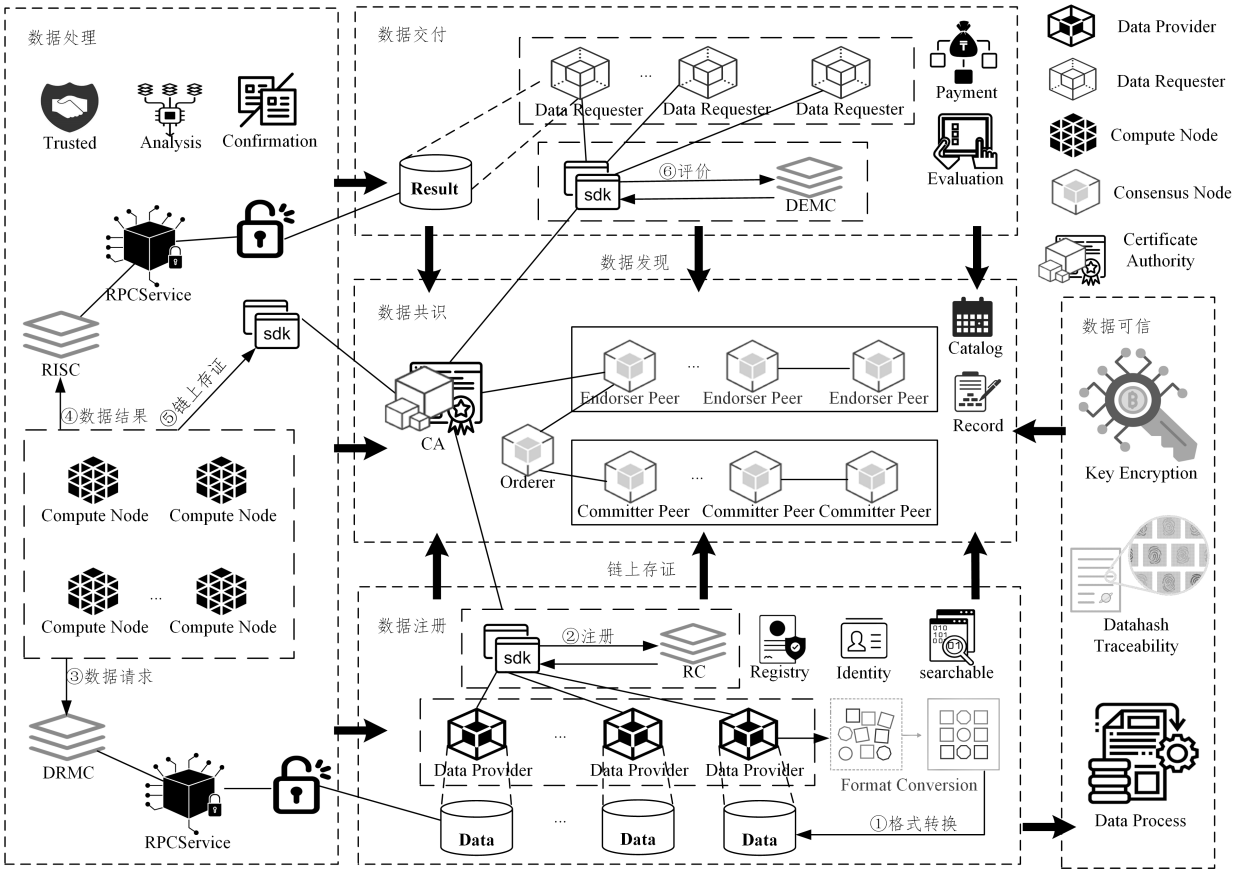


图 2 数据可信共享框架概览

Fig. 2 Overview framework of data trustedsharing

下面介绍 DTSF 的主要角色组成。

1)数据需求方(DR)：数据资源的潜在客户可以通过 Fabric SDK 接入 DTSF，并通过注册表合约(RC)向 Fabric 网络匹配感兴趣的数据提供方。

2)数据提供方(DP)：在 DTSF 中注册数据资源的用户能够为匹配成功的数据需求方提供满足既定需求的数据资源。

3)计算节点(CN)：对于满足数据需求方既定计算需求的节点，假设其安全可信，并能正确执行对应的计算任务。

4)共识节点：共识节点由 DTSF 中的所有成员节点共同组成，维护 Fabric 网络中各节点数据副本的共识。

5)证书机构：DTSF 中权威可信的数字证书认证中心，负责为接入 DTSF 的参与方发放在 Fabric 网络中通信的证书及密钥。

在 DTSF 中的数据共享流程如下：DP 预先对其数据资源注册并上链。接着，DR 通过 Fabric 网络中的注册表合约匹配到合适的 DP 及 CN，然后向 DTSF 联盟链网络发送携带具体需求信息的数据共享请求。联盟链网络广播该请求给对应的 CN，CN 通过解析并进一步构造数据获取请求来向 DP 申

请数据资源的使用权，之后在 CN 上执行数据计算任务并返回最终的结果数据给 DR。符合计算需求的 CN 将由智能合约自动把包含数据共享参与方签名的数据共享记录存证于联盟链网络，同时 DR 可以对本次数据共享流程进行满意度评价。本文涉及的主要符号说明如表 1 所列。

表 1 符号表

Table 1 Symbol table

符号名称	符号含义
DP_i	数据提供方
DR_i	数据需求方
CN_i	计算节点
$CN_{\text{Info}}^{PK_i}$	计算节点的状态信息
$D_i^{PK}{}_j$	数据资源
UID_i	数据资源的唯一身份标识
UI_i	数据资源的标识
CD_i	数据资源的特征数据
DE_i	数据资源的数据实体
$CBDAO_i$	数据架构对象
$UUID_i$	数据共享的唯一标识
RC	注册表合约
DRMC	数据请求管理合约
RISC	结果信息存储合约
DEMC	数据评价管理合约
SK_i	私钥
PK_i	公钥
$Cert_i$	数字证书
K_i	对称密钥
$Sig()$	短签名函数
$dataProof_i$	数据共享记录
ω	满意度评价指标

3.2 智能合约模块设计

本节介绍数据可信共享框架 DTSF 在拟议 Hyperledger Fabric 联盟链网络中的智能合约模块设计。这些模块包含了数据共享的功能需求，即数据注册、需求匹配、数据共享、满意度评价，其分别以注册表合约（RC）、数据请求管理合约（DRMC）、结果信息存储合约（RISC）和数据评价管理合约（DEMC）的形式呈现。具体的功能模块设计如图 3 所示。

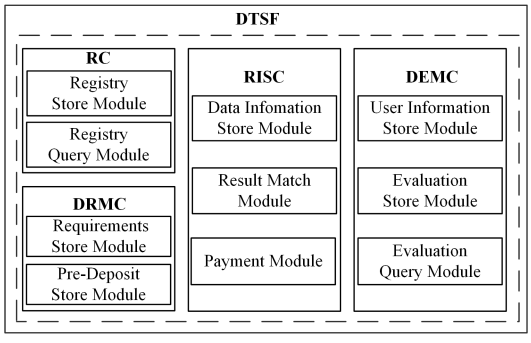


图 3 智能合约模块设计

Fig. 3 Module design for smart contract

1）RC：RC 通过注册表存储模块存储并管理所有注册上链的数据资源，为不同来源的数据资源提供统一的接入方式；同时，RC 的注册表查询模块为数据需求方提供了查询接口，供数据需求方匹配合适的数据资源。

2）DRMC：DRMC 的请求信息存储模块负责存储并管理数据共享过程中的所有数据请求，包括数据需求方发起的数据共享请求、拟议框架处理后的数据计算请求、计算节点发起的数据获取请求；DRMC 的预付款存储模块要求数据需求方在进行需求匹配后预存付款来支付给数据提供方和计算节点。

3）RISC：RISC 通过共享信息存储模块存储并管理数据共享过程中的关键存证信息，包括数据提供方的数据应答信息、计算节点上链的最终数据共享可信凭证；结果匹配模块对计算节点的数据分析处理结果进行统计，相同结果最多的即为最终结果；支付模块负责依据最终一致的结果数据，向数据提供方和计算节点的共享地址发放付款。

4）DEMC：DEMC 通过用户信息存储模块存储所有数据共享参与方的身份信息；评价存储模块用于存储每次数据共享流程结束后，数据需求方对数据提供方和计算节点的满意度评价；评价查询模块可以帮助数据需求方更好地选择合适的数据提供方和计算节点，以进行后续的需求匹配。

4 数据可信共享框架

4.1 数据架构

传统数据架构主要构建在关系型数据库之上，随着社会数据的“爆炸式”增长。其运维和存储的成本也将呈指数级增长。同时，传统数据架构无法很好地解决“数据孤岛”问题，分散在不同系统中的数据难以整合和共享，缺乏灵活性和拓展性。

为了让数据提供方规范注册和发布其数据资源，并弥补传统数据架构的缺陷，本文提出了一种基于联盟区块链的数据架构范式（Consortium Blockchain-based Data Architecture，CBDA）。该架构旨在提供一种统一的方式来持久性地标识、描述、定位和管控异源、异域、异构数据资源，无论它们的位置、类型或格式如何，都能很好地解决“数据孤岛”问题。将拟议数据架构应用于 Hyperledger Fabric 分布式联盟链网络后，能够在提升拓展性的同时，达到“数据互联”的目的。

本节将以转换 DP_j 的数据资源 $D_i^{PK_j}$ 为例，来介绍 CBDA 的数据架构模型。首先使用定义 1－4 对其中的关键概念进行描述。

定义 1 UI_i 是 $D_i^{PK_j}$ 在 CBDA 中的标识部分，表示为如下四元组：

$$UI_i=(UID_i,tStamp,PK_j,Sig_{(SK_j)}(M)) \tag{1}$$

其中，UID_i 是 $D_i^{PK_j}$ 的标识部分，$tStamp$ 是标识生成的具体时间，PK_j 和 SK_j 是 DP_j 的公私密钥对，M 是 DP_j 自定义的用于加密的字符串，$Sig_{(SK_j)}(M)$ 是 DP_j 使用其私钥 SK_j 对 M 签名后的密文。

定义 2 CD_i 是 $D_i^{PK_j}$ 在 CBDA 中的特征数据部分，类似于元数据，具体表示为如下四元组：

$$CD_i=(UID_i,Meta_i,Sig_{(SK_j)}(UID_i),apiInfo_i) \tag{2}$$

其中，$Meta_i$ 是 $D_i^{PK_j}$ 的描述信息，$Sig_{(SK_j)}(UID_i)$ 是 DP_j 使用其私钥 SK_j 对 $D_i^{PK_j}$ 的 UID_i 签名后的密文，$apiInfo_i$ 是 $D_i^{PK_j}$ 的访问接口信息。

定义 3 DE_i 是数据资源 $D_i^{PK_j}$ 在 CBDA 中的数据实体部分，能够在减轻拟议数据架构的运维和存储成本的同时，将原始数据的所有权交还给数据提供方，表示为如下五元组：

$$DE_i=(UID_i,AD_i,dl,AS,Mp_{(K_{\text{DTSP}})}(AD_i)) \tag{3}$$

其中，AD_i 表示 $D_i^{PK_j}$ 的合约地址，dl 表示 $D_i^{PK_j}$ 的有效期限，AS 是 $D_i^{PK_j}$ 的状态标识，$Mp_{(K_{DTSP})}(AD_i)$ 是通过对称密钥 K_{DTSP} 对 AD_i 加密后的密文。

定义 4 $CBDAO_i$ 是 $D_i^{PK_j}$ 经 CBDA 数据架构模型转换后，最终在 DTSF 上的格式化存储形式，表示为如下三元组：

$$CBDAO_i=(UI_i,CD_i,DE_i) \tag{4}$$

其中，UI_i，CD_i，DE_i 分别是 $D_i^{pk_j}$ 在 CBDA 数据架构模型中的标识、特征数据、数据实体部分。

CBDA 数据转换流程如下：DTSF 为每个数据资源分配 UI_i（定义 1）作为其对应 CBDA 对象的唯一标识；同时，构建与 UI_i 对应的 CD_i（定义 2）；最后，按照 DE_i（定义 3）的格式封装对应的数据调用接口信息。例如，将上海数据交易所的医械行业标讯数据信息所对应的数据集应用于 CBDA 数据架构模型中，得到了如图 4 所示的转换结果，成功实现了 CBDA 模型转换的预期目标。

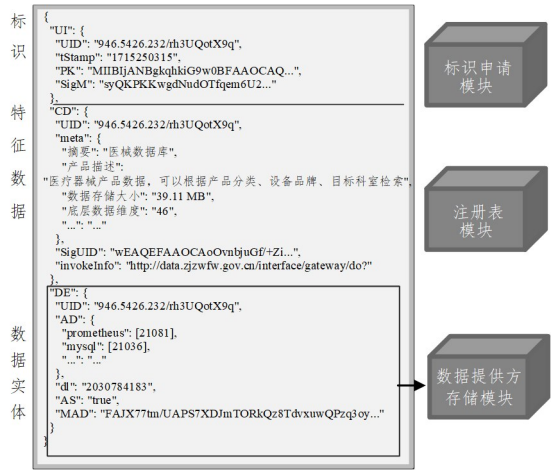


图 4 数据转换模型

Fig. 4 Data conversion model

4.2 数据追溯机制

本节将具体介绍一种基于联盟区块链的数据可信追溯机制，通过 Hyperledger Fabric 基础框架设计并构建拟议追溯机制来管控数据共享的全过程，其中包括数据注册、共享记录存证两大模块，旨在支撑数据使用可审计、数据来源可追溯、数据共享可确责的特性。数据注册模块对应的注册表合约 *RC* 负责管理数据资源的元数据信息，并对外提供数据发现服务；共享记录存证模块对应的结果信息存储合约 RISC 负责持久化存储数据共享存证信息，以确保数据共享过程的透明可追溯。

以数据提供方 DP_j 注册其数据资源 $D_i^{PK_j}$ 为例，本节将对数据可信追溯机制中具体模块所对应的流程及算法进行详细介绍。

4.2.1 数据注册

算法 1 中定义了必要的参数和函数。其中，*enrollCheck* 用于判断是否满足数据注册的条件；*checkId*() 函数用于验证角色证书及密钥信息；*decode*() 函数用于解密操作；*formatValid*() 函数用于验证数据资源的合规性；$DataHashList_i$ 集合记录 $D_i^{PK_j}$ 所有共享流程所对应的哈希摘要；*isDuplicate*() 函数用于判断是否重复注册；*enroll*() 函数是注册函数；*update*() 函数用于更新 *DTSF* 中智能合约的状态数据库；*enrollProof* 是 $D_i^{PK_j}$ 在 DTSF 联盟链网络中的注册证明，通过 *getData*() 函数获取。

算法 1 的具体流程如下：首先，注册表合约 *RC* 通过 *checkId*() 函数和 *decode*() 函数验证角色身份并解密出待注册的 $CBDAO_i$，之后使用 *formatValid*() 函数和 *dataCheck*() 函数检查其格式以及是否重复注册。若通过检测，则注册 $CBDAO_i$ 并返回本次注册所对应的注册证明。

算法 1 数据注册

输入：UID_i，PK_j，$Sig_j(CBDAO_i)$

输出：enrollInfo

1. enrollCheck←true;
2. if(!checkId(UID_i,PK_j))then
3. enrollCheck←false;
4. end if
5. $CBDAO_i$←decode($Sig_j(CBDAO_i)$,PK_j);
6. if(enrollCheck==true)then
7. if(formatValid($CBDAO_i$))then
8. if(! isDuplicate($CBDAO_i$))then
9. $DataHashList_i$←null;
10. enroll($CBDAO_i$);
11. end if
12. update(registerInfo,$CBDAO_i$);
13. end if
14. enrollProof←getData(UID_i,$CBDAO_i$,PK_j);
15. update(enrollInfo,enrollProof);
16. end if
17. return enrollInfo.

4.2.2 数据共享记录存证

算法 2 中定义了一些必要的参数及函数。其中，*shareCheck* 表示是否达到进行存证的要求，$Sig_j(trainProof_i)$ 是 $D_i^{PK_j}$ 最终在 Fabric 网络中存证的密文信息，*decompose*() 函数用于解析密文中的数据结果哈希摘要，*isExist*() 函数用于验证数据哈希值的唯一性。

算法 2 介绍了数据共享记录存证流程。结果信息存储合约 RISC 首先通过 *verifyId*() 函数和 *matchStyle*() 函数核实角色证书及密钥信息，并对待存证的共享记录的格式进行审查。之后，结果信息存储合约 RISC 将使用 *analysis*() 函数和 *isExist*() 函数解析出 $Sig_j(trainProof_i)$ 所对应的数据哈希值 $dataHash_i$，并检查是否重复存证，若为首次存证，则在存证信息中加入 DTSF 的签名并存入 $DataHashList_i$ 中。最后，通过 *update*() 函数写入此条数据共享记录，并返回数据共享记录存证成功的凭证信息。

算法 2 数据共享记录存证

输入：UID_i，$UUID_i$，PK_i，PK_j，K_{DTTP}，Sig_j(trainProof)

输出：dataSharingRecord

1. shareCheck←true;
2. if(checkId(UID_i,PK_j))then
3. shareCheck←false;
4. end if
5. if(shareCheck==true)then
6. if(formatValid(Sig_j(trainProof$_i$)))then

```
7.      dataHash_i ← decompose(Sig_j(trainProof));
8.      if(isExist(DataHashList_i, dataHash_i))then
9.        return dataSharingRecord;
10.     end if
11.     DataHashList_i. Add(UUID_i, dataHash_i, K_DTTP);
12.     update(dataSharingRecord, Sig_j(trainProof));
13.   end if
14. end if
15. return dataSharingRecord.
```

4.3 数据可信共享框架

本节将详细介绍数据可信共享框架 DTSF，主要包括需求匹配、数据共享、满意度评价关键步骤。

4.3.1 参数定义

为满足本文拟议数据共享框架的安全可信要求，需要重新构造数据共享过程参数模型，具体如定义 5－定义 10。

定义 5 数据处理代码的调用过程 *Train_Process* 是数据需求方对其选中的计算节点的部署参数，表示为如下五元组：

$$Train\_Process=(UUID_i, tStamp, Desc, checksum, PK_k) \tag{5}$$

其中，*tStamp* 表示数据处理代码的部署时间，*Desc* 表示数据处理代码的调用过程说明，*checksum* 表示数据处理代码的哈希校验和。

定义 6 数据处理代码 *Process_Code* 是计算节点用于处理不同数据共享需求的工具，具体表示为如下四元组：

$$Process\_Code=(CClass, CUnit, CText, CTask) \tag{6}$$

其中，*CClass* 是 *Process_Code* 的参数类型，*CUnit* 是 *Process_Code* 的单元部件，*CText* 是 *Process_Code* 的文档资料，*CTask* 是 *Process_Code* 的具体数据计算任务。

定义 7 数据共享请求信息 *Train_Request* 是数据需求方经过需求匹配后发起的数据共享请求，表示为如下四元组：

$$Train\_Request=(UUID_i, tStamp, nodeInfo, Train\_Info) \tag{7}$$

其中，*tStamp* 是数据共享请求的发起时间；需求匹配所选中的计算节点为 $nodeInfo=\{nodeInfo_{CN_1}, nodeInfo_{CN_2}, \cdots, nodeInfo_{CN_l}\}$，任意一个 *nodeInfo* 表示为 $nodeInfo=(UUID_i, nodeName, nodeParameters)$，其中 *nodeName* 表示计算节点标识名称，*nodeParameters* 表示计算节点的算力参数；*Train_Info* 表示需求匹配所选中的数据资源的参数信息。

定义 8 数据共享参数信息 *Train_Info* 是发送给所选中的计算节点的具体共享参数信息，表示为如下六元组：

$$Train\_Info=(UUID_i, dataInfo, Desc, dl, PK_k, PK_j) \tag{8}$$

其中，数据资源的定位信息为 $dataInfo=\{dataInfo_{D_1^{PK_j}}, dataInfo_{D_2^{PK_j}}, \cdots, dataInfo_{D_n^{PK_j}}\}$，任意一个 *dataInfo* 表示为 $dataInfo=(UUID_i, dataName, dataParameters, dataProcess)$，其中 *dataName* 表示数据资源名称，*dataParameters* 表示数据资源的方法地址的定位参数，*dataProcess* 表示数据资源的调用过程参数；*Desc* 是数据资源的明文描述信息；*dl* 表示数据共享请求的有效截止时间。

定义 9 数据获取请求信息 *Data_Request* 是计算节点向数据提供方获取共享数据资源的请求，具体表示为如下七元组：

$$Data\_Request=(UUID_i, tStamp, dataInfo, RInfo, Sig_{\{SK_k\}}(Train\_Request), PK_k, PK_l) \tag{9}$$

其中，*tStamp* 是 *Data_Request* 的时间戳，*RInfo* 是 *Data_Request* 的摘要，$Sig_{\{SK_k\}}(Train\_Request)$ 是含数据需求方签名的加密数据共享请求。

定义 10 数据评价信息 *Train_Evaluation* 是每次数据共享后数据需求方对数据提供方及参与本次流程的计算节点的评价信息和执行信息，表示为如下五元组：

$$Train\_Evaluation=(UUID_i, tStamp, \omega_{DP}, \omega_{node}, Sig_{\{SK_l\}}(Train\_resultHash)) \tag{10}$$

其中，*tStamp* 是数据评价的发起时间，ω_{DP} 表示数据需求方对数据提供方的满意度评价信息，ω_{node} 表示数据需求方对计算节点的满意度评价信息，$Sig_{\{SK_l\}}(Train\_resultHash)$ 是带有计算节点签名的加密结果数据哈希值。

4.3.2 需求匹配

需求匹配是数据共享的关键前置步骤，不仅可以帮助数据需求方找到合适的数据提供方和可信计算节点，还可以保护数据提供方的权益，实现出于预期目的的数据资源共享，做到“应需调度”。同时，拟议框架通过收集汇总链上的用户共享满意度评价信息，能够帮助数据需求方更好地了解数据提供方的信誉和服务质量、计算节点的性能等。

用户注册信息分为用户私有信息和用户公共信息两个部分，是数据共享参与方的唯一身份证明，本文使用定义 11 和定义 12 对其进行描述。

定义 11 用户私有信息 δS_k 是数据共享参与方注册后返回的私人身份信息，具体表示为如下七元组：

$$\delta S_k=(U_k, tStamp, M, PK_k, SK_k, Role_k, \mathcal{Z}_k) \tag{11}$$

其中，U_i 是用户身份标识；*tStamp* 表示用户的注册时间；*M* 表示用户键入的随机字符串，用于加密生成身份密钥；PK_k 和 SK_k 表示 CA 为用户生成的公私密钥对；$Role_k$ 表示用户的身份状态信息；$\mathcal{Z}_k$ 表示用户的合约账户地址。

定义 12 用户公共信息 δP_k 是数据共享参与方注册后在链上存证的公开身份信息，具体表示为如下四元组：

$$\delta P_k=(U_k, tStamp, PK_k, \mathcal{Z}_k) \tag{12}$$

其中，与用户私有信息 δS_k 不同的是，δP_k 是用户私有信息 δS_k 的脱敏版本，隐藏了相关的机密身份信息。

以数据需求方 DR_k 进行需求匹配为例，完整的需求匹配流程如下：数据需求方 DR_k 通过拟议框架提供的注册程序获得用户身份信息 δS^{DR_k}（定义 11），联盟链网络将其用户公共信息 δP^{DR_k}（定义 12）同步到链上数据评价管理合约 DEMC。注册成功后，数据需求方 DR_k 通过 Fabric SDK 接入 DTSF，并通过注册表合约 RC 找到感兴趣的数据资源 $D_i^{PK_j}$ 的 CBDA 格式信息 $DCAO_i$（定义 4）。此外，数据需求方还需要选择合适的计算节点并部署数据分析代码来执行处理共享数据资源 $D_i^{PK_j}$。依据数据评价管理合约 DEMC 记录的评价信息进行评判，从而选择合适的数据提供方和计算节点。不仅如此，需求方还可以进一步查询联盟链上特定共享流程的存证记录，验证其信息的真实性。一旦数据共享双方就数据共享服务和

付款达成一致，数据需求方DR_k就可以将预付款发送到结果信息存储合约 RISC，并以可执行二进制文件的形式在所选中的计算节点上部署并签名防伪。

4.3.3 数据共享

图 5 展示了数据共享的详细工作流，使用〈PK，SK〉来表示数据共享参与方的公私密钥对，计算节点个体的集合表示为$CN_B=\{CN_{B1},\cdots,CN_{Bi},\cdots,CN_{Bn}\}$，$i\in n$，为方便表述，下述计算节点用$CN_B$统称。以数据需求方$DR_A$获取数据提供方$DP_C$的数据资源$D_i^{PK_C}$处理后的数据结果为例，下面给出数据共享的具体步骤。

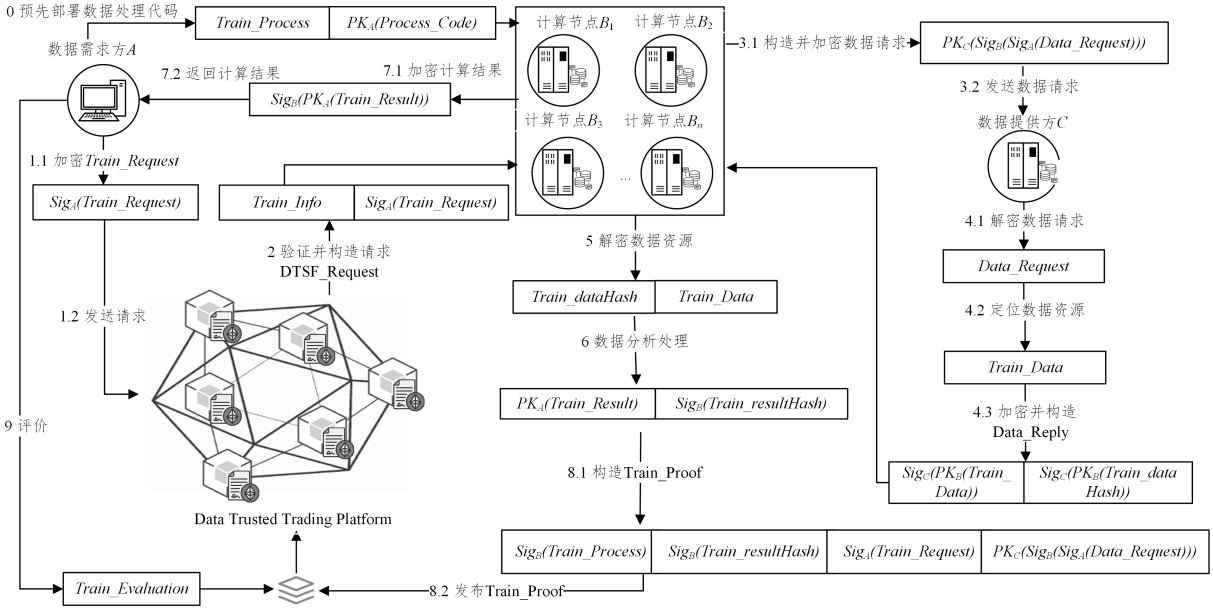


图 5 数据共享流程

Fig. 5 Process of data sharing

步骤 1 DR_A进行需求匹配后，使用其私钥SK_A对本次数据共享的数据请求 $Train\_Request$ 加密，然后将得到的加密值$Sig_{\{SK_A\}}(Train_Request)$发送给数据请求管理合约 DRMC。

步骤 2 联盟链网络接收到加密的数据共享请求$Sig_{\{SK_A\}}(Train_Request)$后，验证其是否注册在籍，使用$DR_A$的公钥$PK_A$解析出数据共享参数信息 $Train\_Info$(定义 8)，识别出本次数据共享所选中的CN_B，并在此基础上进一步构造数据共享请求的参数$DTSF_Request_{PK_A}$二元组。该二次元组包括以下内容：第一部分是解析出的数据共享参数信息 $Train\_Info$，第二部分是步骤 1 中得到的带有DR_A签名的加密请求$Sig_{\{SK_A\}}(Train_Request)$。最后，将所构造的二元组发送给本次数据共享流程中所有被选中的CN_B。

步骤 3 CN_B依据 $Train\_Info$ 找到参与本次数据共享的DP_C的地址消息，构造并加密数据获取请求 $Data\_Request$(定义 9)，得到$PK_C(Sig_{\{SK_B\}}(Sig_{\{SK_A\}}(Data_Request)))$，将其发送给对应的$DP_C$。

步骤 4 DP_C向联盟链网络获取DR_A及CN_B的公钥PK_A和PK_B，并解密验证$PK_C(Sig_{\{SK_B\}}(Sig_{\{SK_A\}}(Data_Request)))$，核验并确认$DR_A$及$CN_B$的身份后得到数据获取请求 $Data\_Request$，并根据 $dataInfo$ 定位到请求的具体数据资源 $Train\_Data$。之后，在此基础上进一步构造数据应答$Data_Reply_{\{PK_C\}}$二元组，其中包括以下内容：第一部分是加密后的数据资源$Sig_{\{SK_C\}}(PK_B(Train_Data))$，第二部分是加密后的数据资源哈希值$Sig_{\{SK_C\}}(PK_B(Train_dataHash))$。

步骤 5 CN_B使用注册在籍的DP_C的公钥PK_C和自身私钥SK_B验证并解密$Data_Reply_{PK_C}$，解析得到数据资源 $Train_Data$和数据资源的防伪哈希值 $Train\_dataHash$。

步骤 6 CN_B验证数据资源 $Train\_Data$ 后，利用预先部署的数据处理代码对其进行分析处理，得到结果数据 $Train\_Result$ 及其对应的哈希摘要 $Train\_resultHash$，并通过DR_A的公钥PK_A加密得到$PK_A(Train_Result)$和$Sig_{\{SK_B\}}(Train_resultHash)$。

步骤 7 CN_B使用SK_B加密步骤 6 中得到的密文$PK_A(Train_Result)$，得到$Sig_{\{SK_B\}}(PK_A(Train_Result))$，之后通过远程过程调用将其返回给$DR_A$。

步骤 8 CN_B在返回$PK_A(Train_Result)$给DR_A的同时，需要构造数据共享可信凭证 $Train\_Proof$。$Train\_Proof$ 由 4 个部分组成，分别是包含CN_B签名的数据处理代码的调用过程$Sig_{\{SK_B\}}(Train_Process)$、包含$CN_B$签名的数据处理结果哈希值$Sig_{\{SK_B\}}(Train_resultHash)$、包含$DR_A$签名的共享请求$Sig_{\{SK_A\}}(Train_Request)$，以及包含本次数据共享所有参与方签名的数据获取请求$PK_j(Sig_{\{SK_i\}}(Sig_{\{SK_i\}}(Data_Request)))$。之后，$CN_B$通过结果信息存储合约 RISC 把 $Train\_Proof$ 发布到 DTSF 联盟链网络中，同时把得到的$Sig_{\{SK_B\}}(Train_resultHash)$存到$DataHashList_i$中，结果信息存储合约 RISC 核验最终结果的有效性。若数据参与方认为最终交付的结果$Sig_{\{SK_B\}}(PK_A(Train_Result))$与其预期大相径庭，可以通过 DTSF 联盟链网络中三方达成共识的 $Train\_Proof$ 进行核验，以发现问题根源。

步骤 9 DR_A可以在完成本次数据共享后进行满意度评价，将数据评价信息 $Train\_Evaluation$(定义 10)以交易的方式提交给数据评价管理合约 DEMC 进行链上存证。

4.3.4 满意度评价

本文使用数据处理结果的哈希值作为数据共享满意度

评价[19]的唯一依据。也就是说，尽管一个共享可能对应多个共享记录，但最终的结果哈希只有一个。这也避免了恶意数据需求方进行多次无效的异常评价。算法 3 对若干关键参数和函数进行了定义。其中，*evaluateCheck* 表示是否满足进行满意度评价的要求；*check*()函数用于检测当前数据共享流程的所有计算节点的执行状态，以及指示任务是否执行完毕；*evaluate*()函数依据满意度评价指标 ω 对本次数据共享进行评价，如定义 13 所示。

定义 13 满意度评价指标 ω 是对参与数据共享的数据提供方及计算节点的综合评价得分，表示为如下三元组：

$$\omega=(Accuracy,Score,Efficiency) \tag{13}$$

其中，$Accuracy$ 是对数据结果的准确性评价分数，$Score$ 是数据需求方对本次数据共享的主观评价分数，$Efficiency$ 是对数据交付耗时的评价分数。对于一次数据共享，为避免数据需求方做出不合实际的恶意评价，满意度评价指标应该在一个可控的范围内，即 $\omega_i=(Accuracy_i,Score_i,Efficiency_i)\in((-\xi,\xi),(-\xi,\xi),(-\xi,\xi))$，$\xi$ 为防止恶意数据需求方进行异常评价的阈值。

算法 3 展示了满意度评价的流程。数据评价管理合约 DEMC 首先使用 *verifyId*()函数核验数据提供方的证书及身份信息，之后检查是否所有计算节点都已经执行完毕。如果同时满足上述条件，*analysis*()函数则会解析出结果数据的哈希值，并在 $DataHashList_i$ 中进行匹配，若匹配成功，则可以利用 *evaluate*()函数进行满意度评价，数据需求方将依据满意度评价指标 ω(定义 13)对本次数据共享流程进行客观评价。评价后，从 $DataHashList_i$ 中删除本条数据哈希值记录，确保评价的正确性和唯一性。同时，数据需求方将本次数据共享流程的满意度评价信息记录在链上，为后续参与数据共享的数据需求方在需求匹配阶段提供更准确的参考。

算法 3 满意度评价

输入：UID_i，PK_j，$CN_{Info}^{PK_j}$，U UID_i，Sig_j(resultHash)，$DataHashList_i$

输出：evaluateInfo

```
1. evaluateCheck←true;
2. if(!checkId(UID_i,PK_j) ‖ check(CN_state^PK_j)) then
3.     evaluateCheck←false;
4. end if
5. if(evaluateCheck==true)then
6.     resultHash←decompose(Sig_j(resultHash));
7.     if(!isExist(DataHashList_i,resultHash))then
8.         return errorInfo;
9.     end if
10.    evaluate(UUID_i,Sig_j(resultHash),CN_Info^PK_j,ω);
11.    DataHashList_i.Remove(resultHash);
12. end if
13. return evaluateInfo.
```

4.4 安全性分析

1)身份认证。数据共享参与方 *User* 接入联盟链网络前需要进行身份认证，CA 机构通过参与方随机键入的字符串 M 为其生成专属公私密钥对 $\{PK_{user},SK_{user}\}$ 和数字证书 $Cert_{user}$ 来接入联盟链网络，所有注册参与方的用户公共信息 δP_{user} 将存储在联盟链网络中，用户私有信息 δS_{user} 则由参与方自己保管。只有经过授权的参与方才能获得对 Hyperledger Fabric 网络中资源的访问权限，维护了联盟链安全隐私的特性。

2)数据自治性。DP 可以依据 CN 的加密数据获取请求 $PK_{DP}(Sig_{\{SK_{CN}\}}(Sig_{\{SK_{DR}\}}(\text{Data\_Request})))$ 中的具体参数信息，决定是否响应该数据获取请求，极大程度地避免了不公平、不合理的数据共享。不同于传统数据托管即服务模式，本文通过构建数据可信共享机制，CN 需要正确转发 DR 加密后的请求 $Sig_{\{SK_{DR}\}}$(Train_Request)才能获得数据资源，使得其他 DR 及 CN 无权越过 DP 进行不合规的数据流通。同时，数据计算任务由 CN 处理，DTSF 中只记录结果数据的哈希值 $Sig_{\{SK_{CN}\}}(Train\_resultHash)$，DR 无法获取未经处理的数据资源。

3)数据完整性和安全性。原始数据资源将被 CN 的公钥 PK_{CN} 和 DP 的私钥 SK_{DP} 进行加密以确保其在数据共享过程的安全性，与此同时，其他参与方无法获取数据的明文。此外，数据资源的流动始终依托于数字签名和哈希摘要技术，以此确保数据共享的每一步操作都有明确的记录存证，最终通过数据共享可信凭证 *Train_Proof* 来体现，从而实现数据共享全过程的监控与管理，进而保证共享数据的完整性。

4)针对不诚实数据提供方 DP 的安全性。数据共享过程中可能出现不诚实的 DP 最终交付不完整或不相关数据的问题。DP 和 CN 需要对其提供的共享数据及其哈希摘要通过 SK_{DP} 和 PK_{CN} 进行签名确责，得到加密后的密文 $Sig_{\{SK_{DP}\}}(PK_{CN}(Train\_Data))$ 和 $Sig_{\{SK_{Dp}\}}(PK_{CN}(Train\_dataHash))$，并构造数据应答 $Data\_Reply_{PK_{Dp}}$ 以响应数据获取请求。DR 可基于此哈希摘要验证所获取的数据资源的真实性，以防止数据提供方在产生争议时发生否认或抵赖的情况。此外，DR 可以在进行需求匹配时从数据评价管理合约 DEMC 中获得 DP 的相关评价信息 ω 进行进一步筛选。

5)针对不诚实计算节点 CN 的安全性。不诚实的 CN 可能伪造数据请求 *Data_Request* 向 DP 获取数据资源，DP 能够通过数据请求管理合约 DRMC 来鉴别该请求是否为异常请求，只有匹配到 DR 签名的请求，证明确实是对应的 DR 发起的，DP 才会将数据资源交付给 CN，避免了 CN 恶意获取数据资源。为避免 CN 不按照预定要求处理数据以节约算力并依旧可以获得预期收益，CN 交付数据处理结果后必须将数据共享可信凭证 *Train_Proof* 上链才能获得预期奖励，该凭证中含有 CN 对该结果数据的哈希摘要的签名值 $Sig_{\{SK_{DP}\}}(PK_{CN}(Train\_dataHash))$，CN 如若作恶，能够被轻易追溯并采取惩戒措施，极大程度地避免了 CN 监守自盗的问题。

6)针对不诚实数据需求方 DR 的安全性。DR 不仅可能伪造其他 DR 的身份来发送虚假请求，还可能拒绝确认本次数据共享结果。DTSF 在处理请求时，会通过 Fabric CA 来对 DR 的 $Cert_{DR}$ 进行核验，有效防止了恶意 DR 冒充其他 DR 的身份。此外，为了保护 DP 的声誉和利益，DTSF 中的 RISC 合约将记录每个数据共享流程中包含所有参与方签名的可信凭证信息 *Train_Proof*，从而显著降低恶意 DR 拒绝承认其提供的数据共享结果的隐患。

5 实验分析

本章通过对比本文提出的数据可信共享框架与传统数据

共享方案的执行耗时，来对拟议框架的执行效率进行验证；并依据不同量级的并发请求来测试拟议框架中智能合约的读写性能，以评估本方案的有效性。

5.1 实验环境

本框架以 Go 语言为基础，基于 Hyperledger Fabric v2.3 框架搭建联盟链网络，并使用 3 台机器来分别模拟数据需求方、数据提供方和计算节点，以模拟真实的数据共享场景。其中网络配置包括 3 个 Order 节点和 3 个 Org 组织，每个 Org 配置 4 个 Peer 节点，节点间采用 Raft 共识机制，并在所有 Peer 节点上部署拟议框架所需的所有智能合约，包括注册表合约、数据请求管理合约、结果信息存储合约和数据评价管理合约，设备的详细配置及所需测试工具如表 2 和表 3 所列。

表 2 设备配置

Table 2 Device configuration

设备	DP	DR	CN
CPU/GHz	3.1	3.2	3.1
RAM/GB	8	8	8
Network(Mb/s)	100	100	100
OS	Ubuntu	macOS	Ubuntu

表 3 测试工具

Table 3 Test tools

工具名称	版本号	说明
Hyperledger Fabric	2.3	超级账本
Docker	20.10	容器化管理工具
Go	1.18	编程语言
Go-Gin	v1.5.0	HTTP-Web 框架
Docker-compose	1.24	容器部署工具

5.2 执行效率分析

针对实际大规模的数据共享场景的需求，本文对比了 DTSF 和传统数据共享方案的总时间开销，其中不包含参与方在数据共享流程外所耗费的时间(如数据需求方从 DEMC 获得数据提供方评价信息后的决策时间)，重复执行 1000 次来计算系统所需的总时间，进而计算出平均时延。其中，每个测试数据集包含 500 个固定数量的键值对，在 10～200 内调整测试数据集的数量。如图 6 所示，由于 DTSF 数据共享流程最终链上存证的信息的大小变化不大，因此接入 DTSF 耗费的额外时间成本所占总时间开销的比例将随着数据集数量的增加而逐渐降低，可降至数据共享流程总时间开销的 30% 以内。

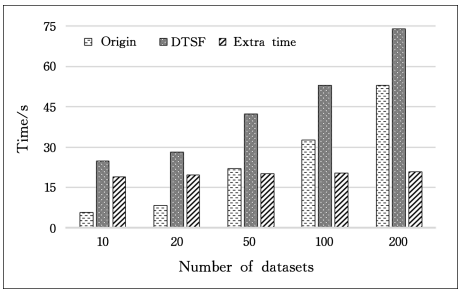


图 6 数据共享时间开销

Fig. 6 Time cost of data sharing

5.3 智能合约性能评估

本节通过模拟 20～500 个虚拟 Hyperledger Fabric Client 代理来对拟议框架的所有智能合约分别进行读写性能测试。其中，采用 Hyperledger Foundation 研发的性能基准测试工具 Caliper[20]，以及交互工具 Fabric-Nodejs-SDK[21]，以每秒 10 笔链上交易请求的频率，对所有智能合约进行平均执行耗时的性能测试。测试共计 5 轮，测试的虚拟客户端的并发数从 20 逐步增加至 500。如图 7 和图 8 所示，随着系统吞吐量的逐步增加，智能合约的读写操作耗时也将逐步增加，最终能够稳定在一个可接受的波动范围内，其中智能合约查询平均时延能够稳定在 0.12～0.2 s，智能合约写入平均时延能够稳定在 3～5 s。

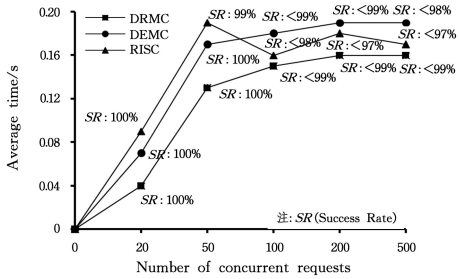


图 7 查询性能测试结果

Fig. 7 Performance test results of read

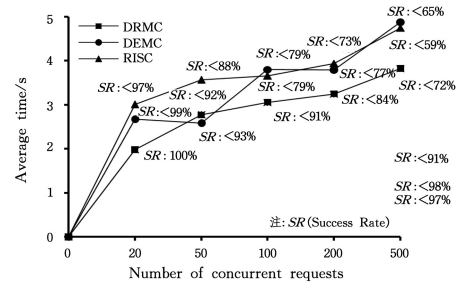


图 8 写入性能测试结果

Fig. 8 Performance test results of write

5.4 有效性分析

虽然本文提出的基于联盟区块链的数据可信共享方案能够解决现阶段数据共享的一些技术难题，但其有效性在实践中是有限的。

互联网的开放性、动态性与不可控性构成了对外部有效性的主要威胁，面对更大数据请求并发量的场景，拟议方案的有效性可能达不到理想效果。例如，在 5.3 节的智能合约性能测试中，当虚拟客户端数量达到 500 时，拟议方案所设计的智能合约写入交易成功率不到 75%；同时，随着并发请求数的增加，时间开销增长速度加快。目前，拟议方案仅在 3 台机器中进行小规模场景测试，无法充分还原真实的数据共享场景，未来工作将考虑利用更多的物理设备来部署集群，进一步验证拟议方案的分布式性能和可靠性。

内部有效性的威胁源于本方案的不确定性，这可能导致每次数据共享所耗费的时间有差异，具体体现在计算节点的计算任务分配上。在 5.3 节的数据共享时间开销测试中，每个测试数据集均包含 500 个键值对，虽然拟议实验方案通过重复执行 1000 次并取平均时间开销来减少这种威胁，但是，不同的计算节点不可避免地存在性能上的差异，未来工作将考虑调整每个计算节点的计算任务占比，以最大化降低这种不确定性。

结束语 本文提出了一种基于联盟区块链的数据可信共享方案，首先，定义了一种新的数据架构范式 CBDA 来规范化数据注册流程；此外，设计了一种数据可信追溯机制，并依据数据共享全过程即需求匹配、数据共享、满意度评价，来构建数据可信共享框架，以保证数据共享的安全可信。通过实验分析评估，验证了本方案的有效性。

下一步研究将细化数据共享流程中对计算节点的计算任务分配，针对计算节点的性能差异，调整每个计算节点的计算任务占比，最大化数据计算效率；同时，选取更大规模的场景进行并发测试。

参 考 文 献

[1] LUO C, MA Y, JING X, et al. Internet of data: a solution for dataspace infrastructure and its technical challenges [J]. Big Data Research, 2023, 9(2): 110-121.

[2] ZHENG Z, XIE S, DAI H N, et al. Blockchain challenges and opportunities: A survey [J]. International Journal of Web and Grid Services, 2018, 14(4): 352-375.

[3] NAKAMOTO S. Bitcoin: A peer-to-peer electronic cash system [J/OL]. Decentralized Business Review, 2008: 21260. https://www.bitcoin.org/bitcoin.pdf.

[4] CHEN H, PENDLETON M, NJILLA L, et al. A survey on ethereum systems security: Vulnerabilities, attacks, and defenses [J]. ACM Computing Surveys, 2020, 53(3): 1-43.

[5] DANIEL E, TSCHORSCH F. IPFS and friends: A qualitative comparison of next generation peer-to-peer data networks [J]. IEEE Communications Surveys & Tutorials, 2022, 24(1): 31-52.

[6] VASILE A, GUILLAUME S, AOUINI M, et al. Le Digital Object Identifier, une impérieuse nécessité? L'exemple de l'attribution de DOI à la Collection Pangloss, archive ouverte de langues en danger [J]. I2D-Information, Données & Documents, 2020(2): 155-175.

[7] ZHANG N, LIU Y, MA X J, et al. Identifier Resolution Technology for Human-cyberphysical Ternary Based on Internet of Data [J]. Journal of Software, 2024, 35(10): 4681-4695.

[8] ZHONG B, WU H, DING L, et al. Hyperledger fabric-based consortium blockchain for construction quality information management [J]. Frontiers of Engineering Management, 2020, 7(4): 512-527.

[9] ANDROULAKI E, BARGER A, BORTNIKOV V, et al. Hyperledger fabric: a distributed operating system for permissioned blockchains[C]// Proceedings of the thirteenth EuroSys Conference. 2018: 1-15.

[10] ANTWI M S, ADNANE A, AHMAD F, et al. The case of HyperLedger Fabric as a blockchain solution for healthcare applications [J]. Blockchain: Research and Applications, 2021, 2(1): 100012.

[11] HONAR PAJOOH H, RASHID M, ALAM F, et al. Hyperledger fabric blockchain for securing the edge internet of things [J]. Sensors, 2021, 21(2): 359.

[12] HAO Y, PIAO C, ZHAO Y, et al. Privacy preserving government data sharing based on hyperledger blockchain[C]// Advances in E-Business Engineering for Ubiquitous Computing: Proceedings of the 16th International Conference on E-Business Engineering(ICEBE 2019). Springer, 2020: 373-388.

[13] RAVI D, RAMACHANDRAN S, VIGNESH R, et al. Privacy preserving transparent supply chain management through Hyperledger Fabric [J]. Blockchain: Research and Applications, 2022, 3(2): 100072.

[14] CHEN Y, LIN B, CHEN X, et al. Blockchain-based Trusted Service-oriented Architecture [J]. Computer Science, 2023, 50(1): 342-350.

[15] ZHENG S, PAN L, HU D, et al. A blockchain - based trading platform for big data [C]// IEEE INFOCOM 2020 — IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). IEEE, 2020: 991-996.

[16] DAI W, DAI C, CHOO K K R, et al. SDTE: A secure blockchain-based data trading ecosystem [J]. IEEE Transactions on Information Forensics and Security, 2019, 15: 725-737.

[17] WANG Y, SU Z, ZHANG N, et al. SPDS: A secure and auditable private data sharing scheme for smart grid based on blockchain [J]. IEEE Transactions on Industrial Informatics, 2020, 17(11): 7688-7699.

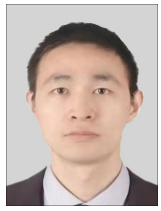
[18] TRUONG N B, SUN K, LEE G M, et al. Gdpr-compliant personal data management: A blockchain-based solution [J]. IEEE Transactions on Information Forensics and Security, 2019, 15: 1746-1761.

[19] TIAN Z, ZHONG R Y, VATANKHAH B A, et al. A blockchain-based evaluation approach for customer delivery satisfaction in sustainable urban logistics [J]. International Journal of Production Research, 2021, 59(7): 2229-2249.

[20] CHOI W, HONG J W K. Performance evaluation of ethereum private and testnet networks using hyperledger caliper[C]// 2021 22nd Asia-Pacific Network Operations and Management Symposium(APNOMS). IEEE, 2021: 325-329.

[21] ZHAI Z, SHEN S, MAO Y. A Toolbox for Migrating the Blockchain-Based Application From Ethereum to Hyperledger Fabric [J]. The Computer Journal, 2024, 67(4): 1309-1323.

LIU Zhanghui, born in 1971, master, associate professor, master's supervisor, is a member of CCF(No. 79769M). His main research interests include big data technology and intelligent computing.



MA Xinjian, born in 1987, Ph.D, associate researcher, is a member of CCF (No. C5681M). His main research interests include information security and distributed system.

（责任编辑：柯颖）