Computer Science and Application 计算机科学与应用, 2021, 11(12), 2900-2912
Published Online December 2021 in Hans. http://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2021.1112294

Hans汉斯

基于区块链的网络协同设计资源共享追溯方法

林　翰

天津工业大学，天津

收稿日期：2021年11月8日；录用日期：2021年12月6日；发布日期：2021年12月13日

摘　要

集团型企业内部在开展网络化协同研发设计过程中，高价值资源共享使用所产生的数字痕迹存在易被篡改、易泄露、信息真实度存疑等安全问题，导致资源供需双方共识度低、可信度差，严重影响资源共享意愿和共享效果。为此，本文提出了一种基于区块链的集团企业研发设计资源共享追溯方法及解决方案，利用区块链中的联盟链模型、加密算法、分布式账本、共识机制等，实现集团企业在网络化协同研发设计过程中资源共享、数字痕迹安全保密、防篡改、去中心化存储等目的；提出基于数字痕迹的共享链路模型，支持历史痕迹多维度追溯查询。同时针对Hyperledger Fabric框架内部存储与查询方式存在查询效率随着数据量变多而明显下降这一问题，提出了基于多维度的索引表数据切分策略，通过数据切分阈值与索引表，显著提升追溯查询效率与查询速度。通过原型系统的搭建及实验验证，本文所提方案能够支撑网络化协同设计中资源共享行为数字痕迹可靠存储和高效率追溯查询。

关键词

区块链，数字痕迹，共享链路，共享追溯，多维度追溯

Blockchain-Based Network Collaborative Design Resource Sharing and Tracing Method

Han Lin

Tiangong University, Tianjin

Received: Nov. 8th, 2021; accepted: Dec. 6th, 2021; published: Dec. 13th, 2021

Abstract

In the process of networked collaborative R & D and design within group enterprises, the digital

文章引用: 林翰. 基于区块链的网络协同设计资源共享追溯方法[J]. 计算机科学与应用, 2021, 11(12): 2900-2912.
DOI: 10.12677/csa.2021.1112294

traces generated by the shared use of high-value resources have security problems such as easy tampering, easy leakage, and doubts about the authenticity of information, resulting in low consensus and credibility between the resource supply and demand parties, poor degree, seriously affecting the willingness and sharing effect of resource sharing. To this end, this article proposes a blockchain-based method and solution for group enterprise R & D and design resource sharing and traceability, using the alliance chain model, encryption algorithm, distributed ledger, consensus mechanism, etc. in the blockchain to realize group enterprise in the process of networked collaborative R & D and design, resource sharing, digital trace security, anti-tampering, decentralized storage, etc.; proposes a shared link model based on digital traces to support multi-dimensional traceability query of historical traces. At the same time, in view of the problem that the query efficiency of the internal storage and query mode of the Hyperledger Fabric framework decreases significantly as the amount of data increases, a multi-dimensional index table data segmentation strategy is proposed, and the data segmentation threshold and index table are used to significantly improve traceability, query efficiency and query speed. Through the construction of the prototype system and experimental verification, the solution proposed in this paper can support the reliable storage of digital traces of resource sharing behaviors and high-efficiency traceability query in networked collaborative design.

Keywords

Blockchain, Digital Trace, Shared Link, Shared Traceability, Multi-Dimensional Traceability

1. 引言

随着工业互联网、智能工厂等智能制造技术的不断发展，网络化协同制造、云制造等新型制造模式不断涌现，极大地促进了集团型制造企业内部制造资源——特别是信息资源——在产品全生命周期内的传递、转移、共享和协作，进而形成以分布式协同研发设计、协同生产计划、协同供应链等为代表的协同制造、管理及运营体系[1] [2]。其中，产品研发设计资源是集团企业核心制造资源之一，具有高价值、高保密性、企业独有等特征，通常是集团下属企业的核心资产[3]。在集团企业分布式协同研发设计过程中，把由各集团内部企业分散所有的、具有共享使用价值的、虚拟化的研发设计资源在集团内部高效共享，对于集团企业开展协同研发设计意义重大[4]。但是在开放网络环境下，如何保证高价值的资源在共享使用过程中不被恶意破坏、滥用，所产生的使用(交易)记录真实可靠、不被单方面或第三方随意更改、全程可追溯，进而保证资源在共享使用过程中资源供需双方高共识度和高可信度，直接影响到资源的共享意愿和共享效果。

在传统的信息系统架构下，信息资源及其访问记录(操作日志)通常采用基于数据库或文件系统的集中存储模式。这种方式存在单点故障、易被篡改、无法保证数据安全性等问题，使得存储记录出现问题后，各单位无法对中心化存储下的共享记录达成共识[5]。2008 年末中本聪[6]首次提出了区块链这一概念，区块链的产生实现了真正意义上的去中心化可信任存储系统。区块链的本质是分布式账本系统，结合密码学、共识机制、点对点通信等技术实现安全、防篡改、防丢失、保密、可追溯等效果[7]，并在食品、农产品、金融交易、医疗等领域形成一定的研究成果和解决方案[8]-[13]，但在网络化协同设计及制造领域鲜有研究。此外，现有研究所使用的区块链系统框架存在追溯维度单一、追溯效率低等问题[5]，无法满

足网络化协同环境下高效追溯查询和多维度复杂追溯查询的实际需求。

为此，本文以集团型制造企业产品协同研发设计为背景，根据集团企业对研发设计资源共享的客观需求，提出基于区块链的集团企业研发设计资源共享追溯设计，并构建了基于该设计的追溯系统。通过设计智能合约，保证区块链中数据自主生成无人为干预，使得区块链中数据安全可靠。同时，针对现有区块链追溯模型在追溯方面存在维度单一，追溯效率低，追溯查询输入缺少灵活性，不适用于共享资源这种大数据量的追溯等问题，提出新的模型，进行优化改进，实现对共享资源的全过程高效率追溯。

2. 相关工作

近些年，由于区块链在数据安全以及数据信任方面的特殊地位，被应用到各种场景，国内外学者对区块链的不同方面展开不断研究。在数据访问控制以及安全方面，Ding 等人[14]根据大部分企业信息和数据库系统都是基于角色访问控制技术，提出了一种基于 RBAC (Role Based Access Control，基于角色的访问控制)与 ABAC (基于属性的访问控制)的安全访问控制模型解决该问题。Amofa 等人[15]通过将用户生成的可接受使用策略与智能合约配对，提出了一个区块链支持的体系结构框架，用于在健康信息交换中安全的控制个人数据。邹秀清[16]等将区块链应用到水质信息管理中，提出了基于区块链的水质信息存证系统，以解决中心化系统中存在的数据易篡改，不公开透明以及安全性难以保证等问题。

此外，在数据私密保护方面，Wang [17]等人通过利用区块链独特的去中心性、匿名性、不可伪造性和可验证性，提出了一种基于区块链的安全和隐私保护的 EHR 共享协议。在共识算法方面，早期的 Bitcoin [6] (比特币)，以太坊采用 POW (Percent of Volume，工作量)算法[18]，这类算法的对算力的要求极为苛刻。苛刻的要求同时带来巨大的资源消耗与浪费问题，后来随着不少学者的不断研究，共识算法呈现出多样化，例如：POS (Proof of Stake，股权证明)算法[19]，PBFT (Practical Byzantine Fault Tolerance，实用拜占庭容错)算法[20] [21]，DPOS (Delegated Proof of Stake，代理权益证明)算法[22]，Paxos 算法[23]，Raft (Reliable, Replicated, Redundant, And Fault-Tolerant)算法[24]等，这些算法各有优缺，适用于不同场景。在区块链框架方面有 Bitcoin (比特币)，以太坊[25]，Hyperledger Fabric [26] (超级账本)等，其中 Bitcoin 与以太坊都采用公链模型，以太坊相比于 Bitcoin 不同之处在于创造性地引入了智能合约，使得计算机可以以数字化方式达成共识、履约、监控履约过程并验证履约结果，极大地扩展了区块链的功能。而相较于前两者 Hyperledger Fabric 是一种企业级的区块链框架，本身框架支持组件的可拔插，可根据不同应用场景自定义结构。Hyperledger Fabric 针对于 Bitcoin，以太坊中的用户身份可能会出现匿名信任问题，采用许可的联盟链模型，通过授权机制，保证了区块链网络中的用户身份，解决了信任问题。同时延用智能合约，降低了引入恶意代码的风险，使系统的健壮性增强。Hyperledger Fabric 的共识机制为 Raft 算法，相比于比特币的 POW 算法，以太坊的 POS 算法，省去了“挖矿”这一步骤，对用户机器的算力要求低很多，并且不需要额外的激励机制，避免了为系统在使用过程中的管理增加额外的负担，节省了企业大量的资源。此外每个节点上的“账本”，不仅仅是区块链，还结合数据库中记录的当前最新的状态下的所有数据组成的世界状态，由这两者来共同维护。世界状态针对于每一条记录都会有最新的状态与之匹配，数据按照大小顺序排列，大大提高了交易提交的速度和交易吞吐量。Hyperledger Fabric 对于世界状态的存储目前只支持非关系型数据库，并且对于查询的方式也有诸多限制。

在追溯领域方面，仵冀颖等[27]将区块链技术应用到食品追溯体系，利用区块链网络的特点，有效整合涉及食品安全的各个参与主体数据，为监管机构提供实时、可靠、不可篡改的产品信息。针对于区块链追溯系统模型方面，一些研究结合了隐藏网络、物联网等技术[8] [10] [28] [29]，满足了信息安全、可追溯的需求。

3. 基于区块链的网络协同设计资源共享追溯设计

以集团型制造企业产品协同研发设计为例，集团下属制造企业分布在不同地域，所有下属企业通过集团共享资源平台统一地共享研发设计资源。使用共享研发设计资源产生共享记录，共享记录永久存储。考虑到集团企业内对研发设计资源共享记录的保密，安全，防篡改，可追溯等方面的需求，区块链中的许可的联盟链模型具有去中心化、分布式存储、安全、防篡改、保密等特点，相较于公链模型可以更好地用于此场景。

本文以共享资源平台为数据来源，应用 Hyperledger Fabric 搭建基于数字痕迹与共享链路的集团型制造企业研发设计资源共享追溯系统，系统架构如图 1 所示。

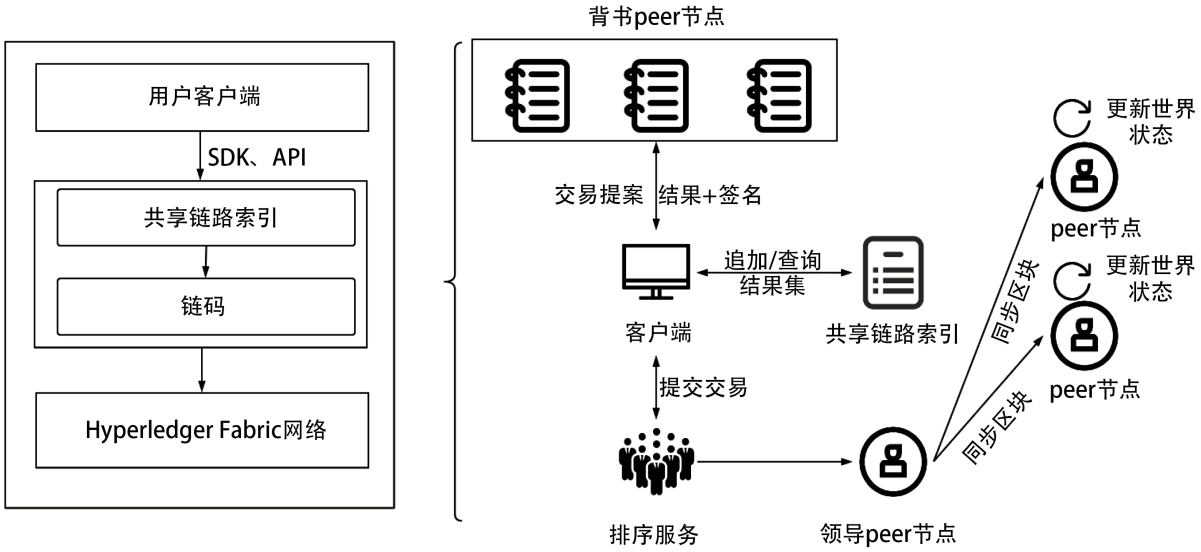


Figure 1. Blockchain-based network collaborative design resource sharing traceability design
图 1. 基于区块链的网络协同研发设计资源共享追溯系统架构

整体架构整体是由资源共享平台、索引表和区块链组成。资源共享平台是与使用者交互的唯一途径，其中嵌入研发设计资源共享操作获取模块与共享资源追溯模块。操作获取模块监听集团企业中的用户使用研发设计资源的操作。共享资源追溯模块，提供与管理着交互的追溯功能。索引表根据共享链路的数据段建立，实现数据切分，并且对追溯请求进行响应。区块链内自成网络体系对外透明，peer 节点负责存储，orderer 节点负责排序，组织负责划分，智能合约负责实现存储共享链路模型与响应追溯请求。

架构中的上链流程：

步骤一：监听用户操作通过 API、SDK 封装成请求提交给区块链网络中的 peer 节点；

步骤二：peer 节点根据智能合约自动执行，将操作生成数字痕迹，然后进行背书；

步骤三：背书后的结果，经过验证满足背书策略，则递交给 orderer (排序)服务，通过共识机制达成共识，否则提交失败，返回用户操作请求失败；

步骤四：达成共识后，结果被打包成区块，通过 Gossip 协议分发给所有 peer 节点；

步骤五：peer 节点验证区块，并将区块上链到区块链，同时根据标签创建/追加共享链路，修改本地数据库中的世界状态；

步骤六：到此共享链路创建/追加成功，随后进行创建或修改索引表；

步骤七：区块链、索引表都提交成功后，平台执行用户操作，以确保所有的数字痕迹都会被记录。

系统架构中的追溯流程：

步骤一：在追溯模块输入追溯字段，经过索引表不完全匹配，向客户端返回索引集；

步骤二：客户端根据索引集逐条形成请求通过 API、SDK 并发地发送给区块链网络中的 peer 节点；

步骤三：peer 节点根据智能合约自动执行相关逻辑，返回结果响应。

3.1. 数字痕迹与共享链路模型设计

集团企业对共享资源的管理体系有两大关键点，即：1) 同一资源多用户共享使用；2) 同一用户同时使用多个资源。这与现有区块链追溯模型不匹配[4]。为此本文提出基于数字痕迹和共享生命周期的共享链路模型。

定义 1 (数字痕迹)：数字痕迹是指研发设计资源在共享使用过程中由信息系统产生的、直接反映资源供需主体共享行为的、经过规范化处理后的数字记录。

由于共享资源的形式和资源共享使用行为均存在多样性，产生的共享记录也会存在差异，因此需要规范化共享记录中的字段。规范后的数字痕迹仅保留共享记录中的关键因素，一定程度上使资源本身对系统透明。数字痕迹由时间 Time、地点 IP、用户 User、共享资源 Resource、操作 Operation 组成。此外为了增强多追溯维度各主体的唯一性，使其更好的服务于共享链路模型引入标识符 UserID，ResourceID。

虽然数字痕迹的设计规范了基本单元，但数字痕迹生成后仍是简单的追加，物理上的堆积。为此提出共享生命周期这一概念，作为定义共享链路的一个重要依据。

定义 2 (共享生命周期)：共享主体在共享使用过程中所经历的拥有权的交接在逻辑上所形成的全部连续过程称为该共享主体的共享生命周期。

定义 2 中的交接是由于用户使用切换所形成的，具有随机性和不确定性。从一个共享主体被创建象征着生命周期的开始，到共享主体被去除，象征着生命周期的结束，中间穿插着各种相关的用户使用操作。共享生命周期是一个逻辑上的概念，并没有确定的实体。

共享链路将数字痕迹与共享生命周期结合起来，通过链表的方式将同一追溯主体的数字痕迹按照交接逻辑顺序链接起来。从表头开始可以追溯共享主体的整个共享生命周期。共享链路的数目取决于共享主体的数目，所有的共享链路构成了共享链路模型。当同一数字痕迹对应多个共享主体时，共享链路就会出现公共节点。多个公共节点连接也称为公共共享链路段，此时的共享链路模型，从多条链表转变为有向无环图。本文为了解决追溯维度单一问题，针对于应用背景，考虑用户与资源两个维度的追溯，对同一数字痕迹创建两个追溯主体。因此本文中的共享链路模型为图，图中每个节点的最大出度和入度为 2。

共享链路的实现基于数字痕迹的数据结构和 Hyperledger Fabric 框架内部存储，通过向相应共享链路追加数字痕迹完成交接，被追加的数字痕迹作为世界状态，当前主体的拥有权由世界状态决定。本文背景下的共享链路模型如图 2 所示。

如图 2 所示，由于每一个用户与设备都具有唯一标识，因此共享主体也具有唯一性，同一共享主体不可被并行使用，共享主体会随着用户使用共享资源这一事件发生交接，由于本文中仅考虑基于用户与资源维度的追溯，因此在用户使用共享资源时就会发生两次不同维度的共享主体交接，分别是图中横向所示的以用户拥有权作为共享主体的用户维度的交接与图中纵向所示的以资源拥有权为共享主体的资源维度的交接。交接会产生两个维度的相应数字痕迹作为记录节点，即图中的白色圆点，共享主体的数字痕迹会在共享生命周期内沿着时间轴不断的产生，并且自动被链接到对应共享链路的末尾，直到对应共享主体的共享周期结束，即产生如图中所示的黑色圆点的数字痕迹之前，共享主体会像是在沿着一条看

不见的路被一直传递下去，这条路就是一条共享链路。而每一条共享链路的最后一个圆点便组成了对应维度的世界状态。

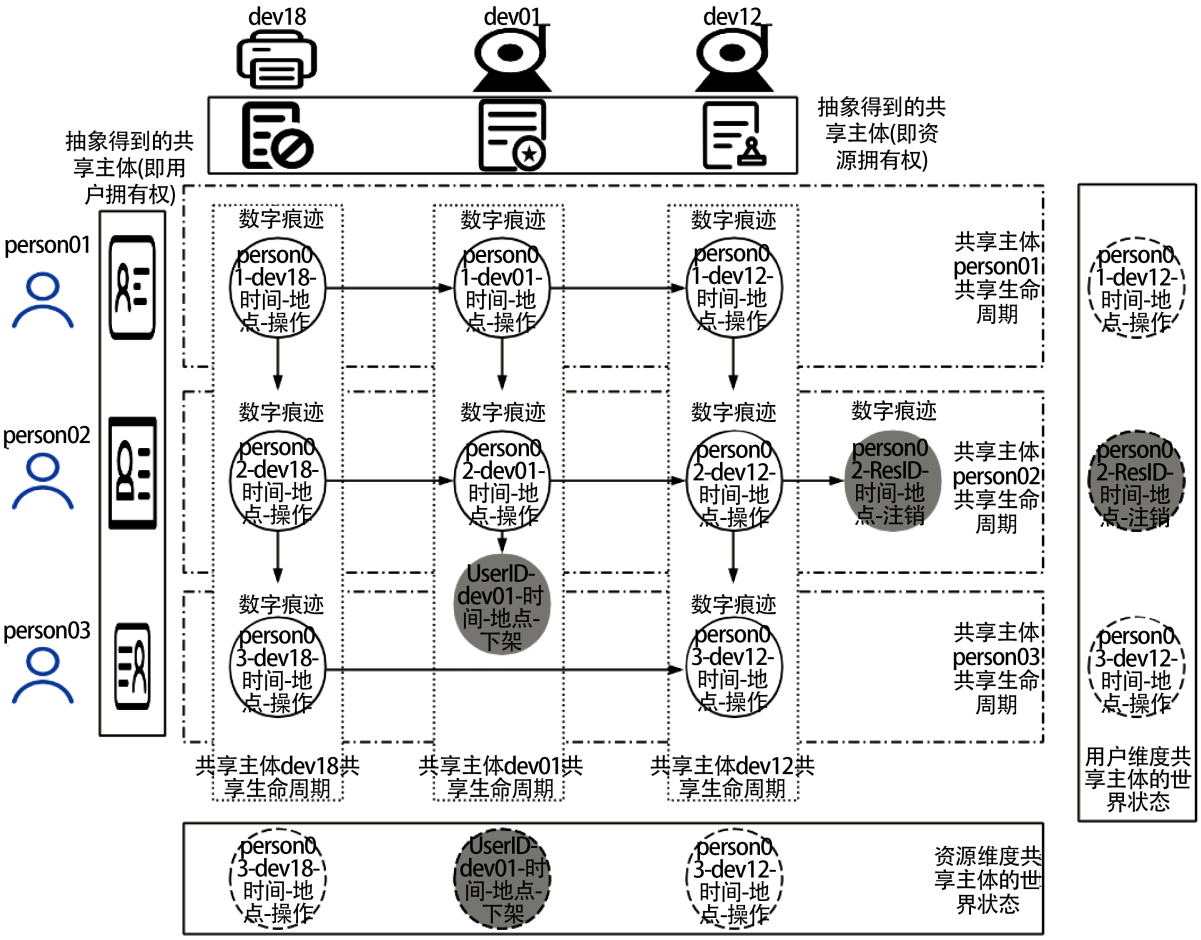


Figure 2. Shared link model
图 2. 共享链路模型

3.2. 追溯优化设计

在提出了共享链路模型的基础上，需要对区块内存储的数字痕迹进行快速追溯查询。传统 Hyperledger Fabric 对于追溯的实现是通过迭代器的方式迭代出所有的记录，然后以一次查询交易响应的方式返回所有数据。这种方式的缺点在于：随着时间的变长，一条共享链路的数据量会越来越巨大，一次追溯响应数据量也会变大，造成延时导致追溯效率降低。

为了解决这一问题，在 Hyperledger Fabric 之上设计了一个索引表。索引表可以根据不同的需求设计，通过唯一标识符将共享链路拆分成多个数据段。例如：索引表组成可以为：共享主体名称、共享主体 ID、操作数量(用于阈值划分)、共享链路段唯一标识符(用于映射关系)，将对于一条共享链路的追溯请求变成多次，并发进行以减少每次查询的数据传输量以及等待时间，提高查询效率。拆分逻辑依赖于索引表中的数量自增，当数量到给定阈值后，生成新的索引，序号自增。根据索引表是表的这种特性，将表以关系型数据库的形式存储，从而使得对于表中字段的查询支持不完全查询，提升了追溯功能。索引表设计如图 3 所示，其中 ptr 与 timestamp 组成一个数据结构，在一个共享链路段中的数量等于操作数量。

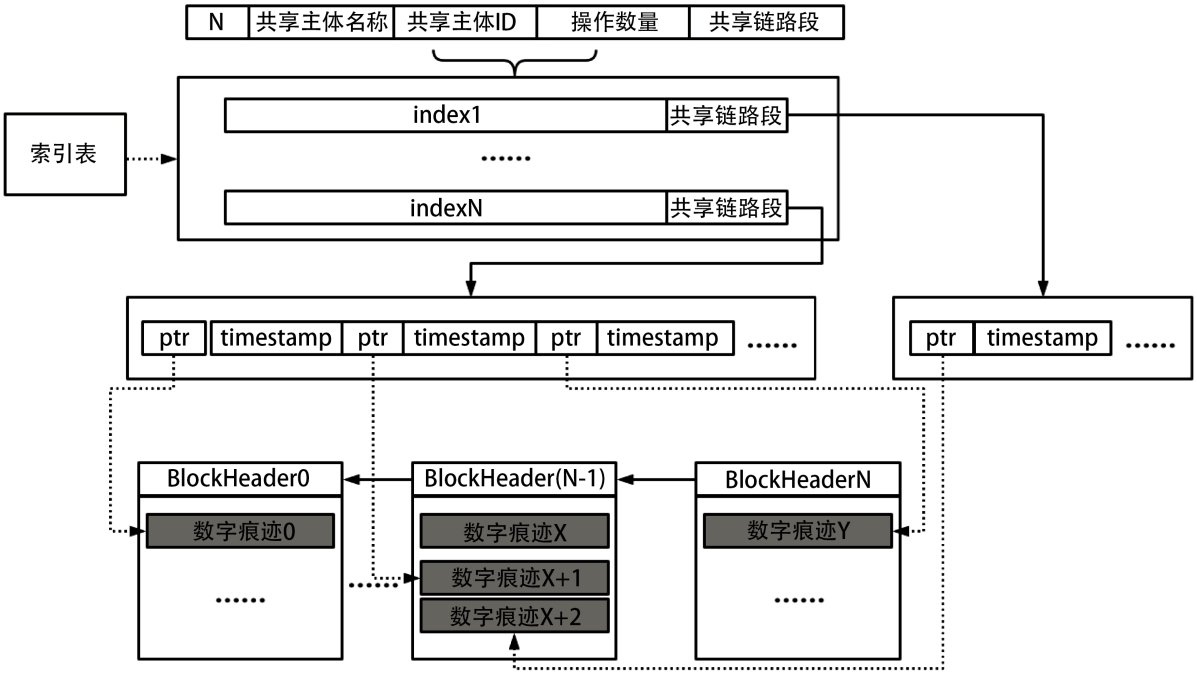


Figure 3. Retrospective optimization index table design
图 3. 追溯优化索引表设计

3.3. 智能合约设计

智能合约是 Hyperledger Fabric 中定义自动执行逻辑的可执行代码，是一种触发式脚本，只需定义触发条件，触发后执行的逻辑，即可自动化执行，无法通过外部进行干预，因此区块链只能运行已规定的智能合约中的执行逻辑，智能合约是操作区块链的唯一途径。本文设计的智能合约实现对于共享链路的创建、追加、追溯功能，数据一旦上链存储在区块链中将不可篡改但可追溯。就针对于本文中的共享链路模型而言，算法流程总结如图 4 所示。

算法 1 共享链路的创建与追加的伪代码流程如下：

(1) 输入用户操作的共享记录
(2) 生成数字痕迹(time,ip,owner,ownerid,object,objectid,operation)
(3) existing, err:=ctx.GetStub().GetState(UserPart)
// 通过 context.GetStub()获取数据库 grpc Client
// 通过 grpc Client 调用 GetState(共享链路段)检查用户维度共享链路段是否存在
(4) If existing == nil
(5) Create UserPart // 当前数字痕迹作为用户维度共享链路的起始数字痕迹
(6) else // existing 中返回最新用户维度共享链路段的起始数字痕迹
(7) Update UserPart // 向用户维度最新共享链路段末尾追加
(8) existing, err=ctx.GetStub().GetState(ResourcePart) // 检查用户维度共享链路段是否存在
(9) If existing == nil
(10) Create ResourcePart // 当前数字痕迹作为资源维度共享链路的起始数字痕迹
(11) else
(12) Update ResourcePart // 向用户维度最新共享链路段末尾追加

算法 2 共享链路段追溯的伪代码流程如下：

(1) 输入要追溯的共享链路段

(2) existing, err:=ctx.GetStub().GetState(Part)

(3) If existing == nil

(4) return “ ” // 返回空字符串

(5) else

(6) HistoryIterator,err := ctx.GetStub().GetHistoryForKey(Part)

(7) for HistoryIterator.HasNext()

(8) HistoryIterator.Next() // Part 向后遍历

(9) return Json 格式的追溯结果

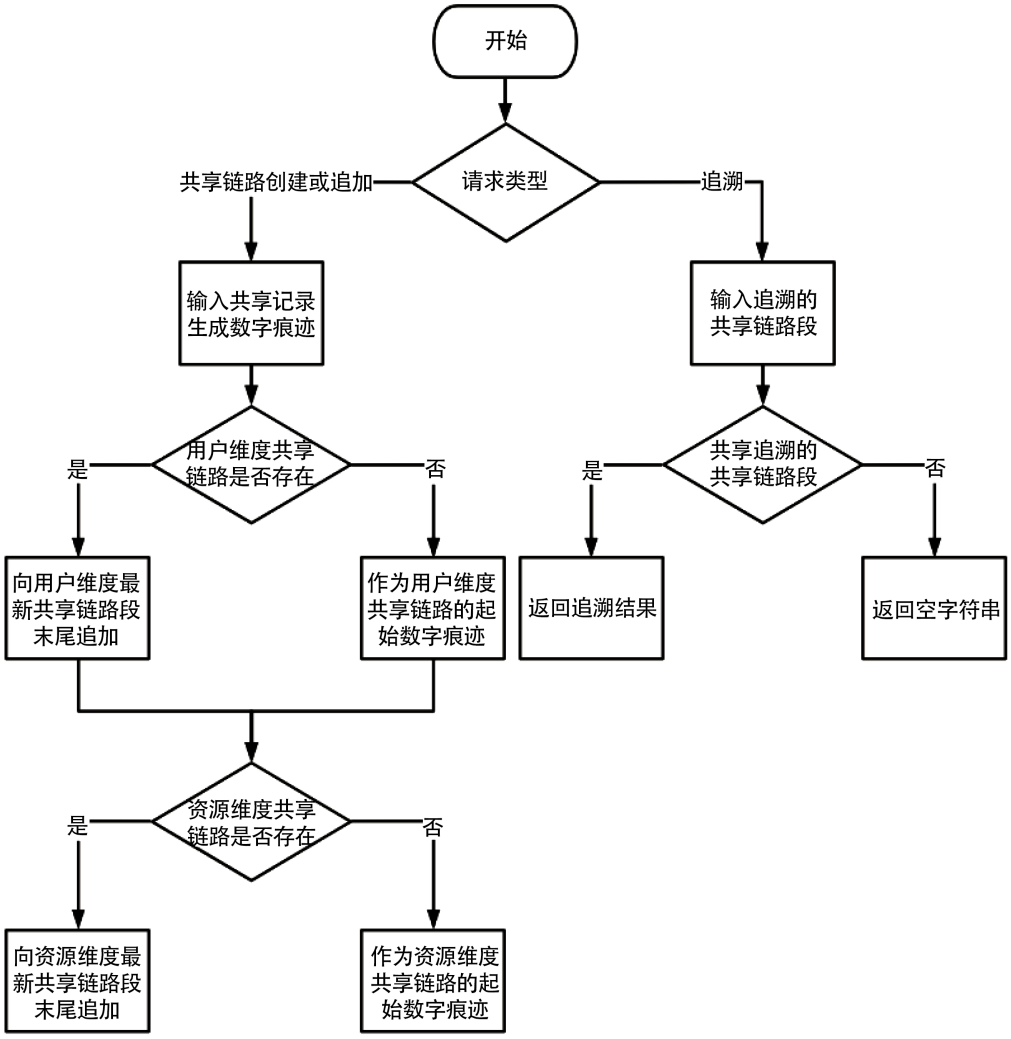


Figure 4. Smart contract algorithm flow
图 4. 智能合约算法流程

通过智能合约，系统将自动判断接收到的请求是共享链路的创建、共享链路的追加还是共享链路段的追溯，根据请求自动执行。如果自动判断为共享链路的创建或追加将进行背书等上链操作，一旦数字

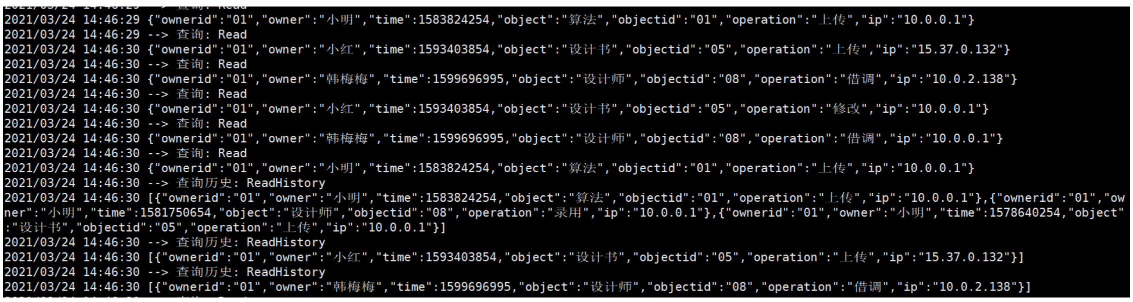
痕迹上链将成为区块中的内容，不可被篡改。智能合约中关于数字痕迹的合法性将通过智能合约中的代码读取世界状态与当前操作生成数字痕迹进行匹配从而自动判断，并且对于不同维度的共享链路的存在也将通过索引表与世界状态的历史版本进行自动查询判断。如果自动判断为共享链路段的追溯，追溯请求不属于共享用户操作的范畴内，因此只需要通过索引表与世界状态的历史版本进行对共享链路存在的判断，不必要进行上链，因而直接从 peer 节点返回结果数据集，有助于提高追溯效率。

4. 系统实现

系统测试的开发环境是部署在 2 台 8 核 3 G 内存的虚拟主机中，系统为 CentOS7，Hyperledger Fabric 版本选用 fabric-sample，此外还配置了 docker，docker-compose 等依赖工具，所搭建的节点有 1 个 Orderer 节点，2 个 Peer 节点，3 个 CA 节点，1 个 Mysql 服务节点。

区块链测试，通过执行编写好的脚本程序，程序中自动触发生成资源共享平台上的相关操作事件，通过登录的 3 个用户在不同的 IP 地址、不同的时间使用不同的资源轮流依次的执行，通过 API、SDK 向搭建的区块链网络系统发送交易请求，测试共享链路的创建、追加、追溯的可用性，执行后以及查询出的请求结果，以 Json 形式展现在终端窗口。如图 5 所示：

(a)

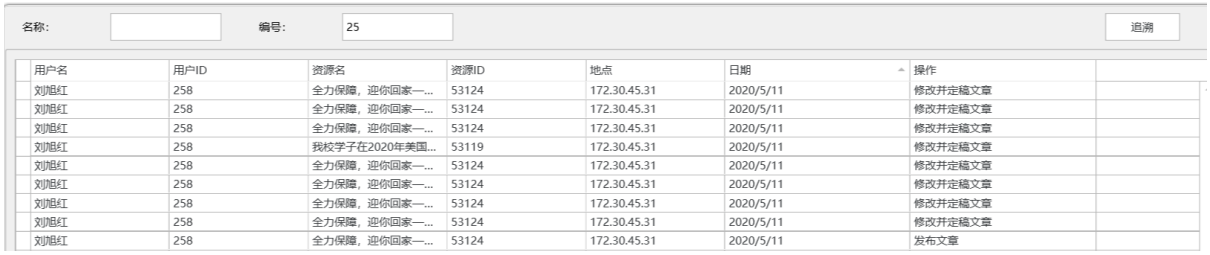


(b)

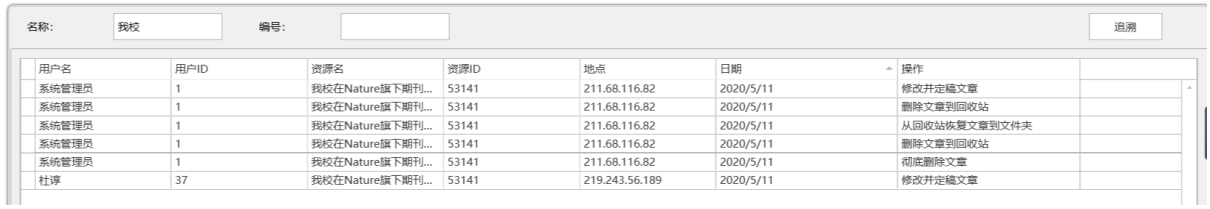
Figure 5. (a) Creation and addition of shared links; (b) Traceability result output of shared link
图 5. (a) 智能合约算法流程；(b) 共享链路的追溯结果输出

结合资源共享平台的测试，通过用户账号身份打开资源共享平台客户端，在相应的模块提供的输入框中随机输入想要查找的字段信息，测试过程中分别对各个字段不同维度进行了不完全查询，系统可以稳定的输出按照共享链路查询的结果并将结果展示，结果如图 6 所示：

(a)



(b)



(c)

(d)

Figure 6. (a) The result of incomplete traceability of the user name; (b) User ID incomplete traceability result; (c) The result of incomplete retrospective resource name; (d) The result of incomplete resource ID traceability

图 6. (a) 用户名称不完全追溯结果；(b) 用户 ID 不完全追溯结果；(c) 资源名称不完全追溯结果；(d) 资源 ID 不完全追溯结果

追溯时间测试，搭建三个相同的测试环境，一者为不进行数据切分的原始系统，一者为数据切分阈值为 50 的系统，还有一者为数据切分阈值为 100 的系统，通过编写好的执行脚本，自动触发生成资源共享平台的相关操作事件，产生足够的测试数据，分别在自动执行相同的数据量之后，通过在发送请求前开启计数函数，在接收到请求之后关闭计时函数的方式，测试通过对比不同数据规模下的不同数据切分阈值与不进行数据切分的同一共享链路的平均追溯时间，验证索引表对于追溯的优化效果。测试数据切分阈值根据平台页面展示最大数量设定为 50、100。在不同系统中通过自动执行脚本设置，每当同一共享链路数据量每增加一万，进行 50 次测试，测试结果经过格拉布斯准测剔除偏差较大的异常值，准则中 $a = 0.05$，$n = 35$，最后求得平均追溯时间。结果如图 7 所示。

测试结果表明：Hyperledger Fabric 内部的历史查询与返回方法是造成上图追溯速度出现明显差异的原因。Hyperledger Fabric 内部的历史查询方法中使用迭代器对数据进行历史追溯，并且迭代后的数据会被全部收集后一次性地发送给追溯端。而引入索引之后可以并发开启多个查询请求，每次查询只需查询到定量的数据就会返回，将追溯端的等待时间缩短。随着集团企业间的共享的时间变长，数据量变大，加入索引的系统在追溯效率上的优势将越发明显，并且索引的添加过程并不复杂，因此本文对于系统追溯效率的优化完全可适用于生产实践。此外根据不同的切分阈值所展示的结果，还可以推测出必然存在一个确切的数据切分阈值，可以达到追溯时间最短。

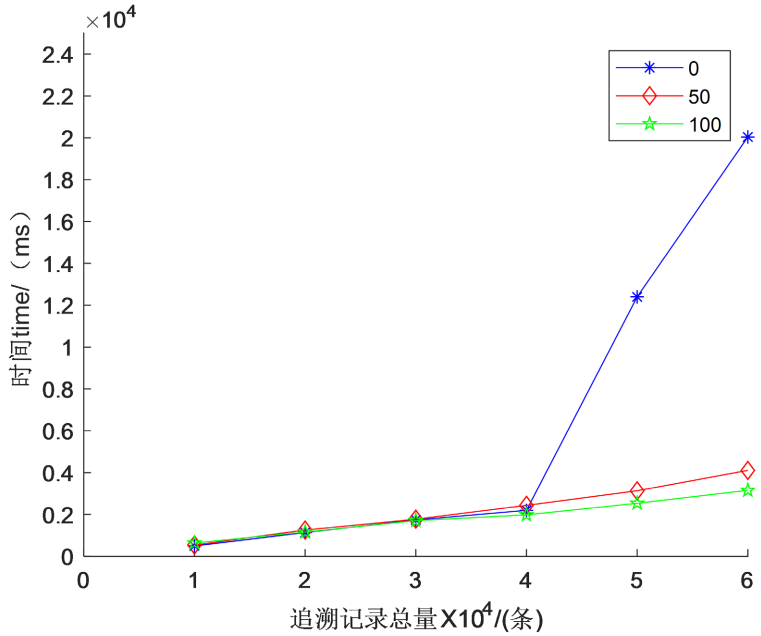


Figure 7. Traceability test results
图 7. 追溯时间测试结果

5. 结束语

区块链的安全、保密、防篡改为集团企业对共享资源追溯中的共享记录可信任问题提供了新的解决思路。本文首先针对于集团企业对共享资源追溯对保密、防篡改、可信任的需求与现有追溯模型不匹配，且追溯维度单一的问题，提出了一种基于区块链的集团企业网络化协同研发设计资源共享追溯设计。其次对于架构设计的内部流程进行了说明，再根据共享记录规定数字痕迹，设计数据结构，提出共享生命周期、共享链路以及共享链路模型，形成多维度追溯模型。然后结合 Hyperledger Fabric 实现，通过智能合约将数字痕迹的上链、实现共享链路的创建、共享链路的追加以及共享链路的追溯。最后，分析 Hyperledger Fabric 内部对于追溯的实现方式，设计更高效的追溯方法，从追溯维度、速度、追溯输入约束等多方面提升追溯效率，并结合测试加以验证。通过对于同一共享链路不同数据切分阈值的对比，表明本文引入索引表在追溯功能上的提升。

6. 展望

基于区块链的网络协同设计资源共享追溯方法，目前已经实现了追溯维度与效率的提升，但是仍有很多问题亟待解决，需要通过实际的使用过程与开发中积累的经验，进一步完善，在目前的研究中发现其中包含以下几点：

1) 系统部署起来比较麻烦，步骤繁琐，而且对于系统的后期运维、维护方面目前还没有制定完善的策略，后续将系统通过 K8S 云计算技术进行部署以及维护，增强系统的可维护性。

2) 本文中的系统对于任何用户都开放追溯查询权限，在权限的分发上可以结合现有的权限管理方式研究在用户权限中做相应的限制，防止用户权限的滥用。

3) 本文提出的区块链系统可以保障数据的一致性安全持久化存储，但是当资源使用出现问题后，对于出现问题的分析仍需要依靠专业技术人员判断。未来可以结合深度学习、人工智能、专家分析等技术，实现问题数据自动发现，使得系统更加智能、更加方便、更加自动化。

基金项目

国家重点研发计划“分布式研发设计资源集成管理与共享关键技术”(2018YFB1701802)。

参考文献

[1] 庄存波, 刘检华, 熊辉. 分布式自主协同制造——一种智能车间运行新模式[J]. 计算机集成制造系统, 2019, 25(8): 1865-1874.

[2] 周新杰, 明新国, 陈志华, 张先燏, 潘杨. 基于模型、数据、知识的设计与制造协同框架[J]. 计算机集成制造系统, 2019, 25(12): 3116-3126.

[3] Verma, D. and Sinha, K.K. (2002) Toward a Theory of Project Interdependencies in High Tech R&D Environments. *Journal of Operations Management*, **20**, 451-468. https://doi.org/10.1016/S0272-6963(02)00024-4

[4] Beyeler, N., *et al*. (2019) Improving Resource Mobilisation for Global Health R&D: A Role for Coordination Platforms? *BMJ Global Health*, **4**, e001209. https://doi.org/10.1136/bmjgh-2018-001209

[5] 于戈, 聂铁铮, 李晓华, 张岩峰, 申德荣, 鲍玉斌. 区块链系统中的分布式数据管理技术——挑战与展望[J]. 计算机学报, 2021, 44(1): 28-54.

[6] Nakamoto, S. (2008) Bitcoin: A Peer-to-Peer Electronic Cash System. https://bitcoin.org/bitcoin.pdf

[7] 张乐君, 刘智栋, 谢国, 薛霄. 基于集成信用度评估智能合约的安全数据共享模型[J]. 自动化学报, 2021, 47(3): 594-608.

[8] Hao, Z., Mao, D., Zhang, B., *et al*. (2020) A Novel Visual Analysis Method of Food Safety Risk Traceability Based on Blockchain. *International Journal of Environmental Research and Public Health*, **17**, 2300. https://doi.org/10.3390/ijerph17072300

[9] 杨信廷, 王明亭, 徐大明, 罗娜, 孙传恒. 基于区块链的农产品追溯系统信息存储模型与查询方法[J]. 农业工程学报, 2019, 35(22): 323-330.

[10] Hong, W., Cai, Y., Yu, Z., *et al*. (2018) An Agri-Product Traceability System Based on IoT and Blockchain Technology. 1*st IEEE International Conference on Hot Information-Centric Networking* (*HotICN*), Shenzhen, 15-17 August 2018, 254-255.

[11] 刘敖迪, 杜学绘, 王娜, 李少卓. 区块链技术及其在信息安全领域的研究进展[J]. 软件学报, 2018, 29(7): 2092-2115.

[12] 唐丹, 庄新田. 制造商融资决策: 银行、商业信用或区块链供应链金融[J]. 东北大学学报(自然科学版), 2021, 42(8): 1202-1209.

[13] 禹忠, 郭畅, 谢永斌, 薛栋. 基于区块链的医药防伪溯源系统研究[J]. 计算机工程与应用, 2020, 56(3): 35-41.

[14] Ding, X. and Yang, J. (2019) An Access Control Model and Its Application in Blockchain. *International Conference on Communications, Information System and Computer Engineering*, Haikou, 5-7 July 2019, 163-167. https://doi.org/10.1109/CISCE.2019.00044

[15] Amofa, S., Sifah, E.B., Agyekum, O., *et al*. (2018) A Blockchain-Based Architecture Framework for Secure Sharing of Personal Health Data. *IEEE* 20*th International Conference on e-Health Networking, Applications and Services* (*Healthcom*), Vitkovice, 17-20 September 2018, 1-6. https://doi.org/10.1109/HealthCom.2018.8531160

[16] 邹秀清, 罗得寸, 林平, 等. 基于区块链的河长制水质信息存证系统[J]. 应用科学学报, 2020, 38(1): 65-80.

[17] Wang, Y., Zhang, A., Zhang, P., *et al*. (2019) Cloud-Assisted EHR Sharing with Security and Privacy Preservation via Consortium Blockchain. *IEEE Access*, **7**, 136704-136719. https://doi.org/10.1109/ACCESS.2019.2943153

[18] Dwork, C. and Naor, M. (1993) Pricing via Processing or Combating Junk Mail. In: *International Cryptology Conference on Advances in Cryptology*, Springer-Verlag, Berlin, 139-147.

[19] King, S. and Nadal, S. (2012) PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake.

[20] Miguel, C., *et al*. (2002) Practical Byzantine Fault Tolerance and Proactive Recovery. *ACM Transactions on Computer Systems*, **20**, 398-461. https://doi.org/10.1145/571637.571640

[21] Suri, N., Amir, Y., Tantillo, T., *et al*. (2016) On Choosing Server- or Client-Side Solutions for BFT. *ACM Computing Surveys*, **48**, 1-30.

[22] Larimer, B.D. (2013) Transactions as Proof-of-Stake!

[23] Leslie, L. (1998) The Part-Time Parliament. *ACM Transactions on Computer Systems*, **16**, 133-169. https://doi.org/10.1145/279227.279229

[24] Ongaro, D. and Ousterhout, J. (2014) In Search of an Understandable Consensus Algorithm. USENIX Association.

[25] Buterin, V. (2014) A Next-Generation Smart Contract and Decentralized Application Platform.

[26] Kuzlu, M., Pipattanasomporn, M., Gurses, L., *et al*. (2019) Performance Analysis of a Hyperledger Fabric Blockchain Framework: Throughput, Latency and Scalability. *IEEE International Conference on Blockchain*, Atlanta, 14-17 July 2019, 536-540. https://doi.org/10.1109/Blockchain.2019.00003

[27] 仵冀颖, 杜聪, 马志远, 等. 应用于食品追溯体系的区块链架构设计[J]. 计算机应用与软件, 2019, 36(12): 46-50.

[28] Thakur, M., *et al*. (2020) A Framework for Traceability of Hides for Improved Supply Chain Coordination. *Computers and Electronics in Agriculture*, **174**, Article ID: 105478.

[29] Lei, H., Ullah, I. and Kim, D.H. (2020) A Secure Fish Farm Platform Based on Blockchain for Agriculture Data Integrity. *Computers and Electronics in Agriculture*, **170**, Article ID: 105251. https://doi.org/10.1016/j.compag.2020.105251