http://bhxb.buaa.edu.cn jbuaa@buaa.edu.cn
DOI: 10.13700/j.bh.1001-5965.2022.0814

一种基于多链协同的联盟链改进模型

赵鲜鲜[1,2]，谭海波[1,2]，赵赫[1,*]，周桐[3]，程昊天[1,2]，李金泽[1,2]

(1. 中国科学院合肥物质科学研究院，合肥 230031；2. 中国科学技术大学，合肥 230026；
3. 安徽中科晶格技术有限公司，合肥 230088)

摘　　要：监管友好、性能优异的联盟链是我国政务和商用区块链的首选。然而联盟链通常因为有效节点少、节点独立性低和发展能力弱的原因，降低系统的可靠性。因此提出了同盟链：一种通过协同多个联盟链来提升系统可靠性的模型。同盟链通过联盟链间相互层叠存储链上数据降低联盟链历史数据被篡改的可能性，通过协同存储区块链加密数据保障账本数据恢复能力；使用聚合签名、数据完整性验证等方法解决数据传输完整性、来源真实性问题；设计双向数据存在性验证方法检验传输数据的连续性，确保协同数据相互背书的真实性；通过与现有方案进行通信、存储和计算消耗上的对比来验证本方案的有效性。

关 键 词：区块链；联盟链；可靠性；去中心化；数据完整性；数据真实性
中图分类号：TP315
文献标志码：A　　**文章编号**：1001-5965(2024)10-3283-14

中央政治局第十八次集体学习以来，区块链技术已上升为国家科技战略。我国区块链产业发展迅速，为多个行业的新发展增信赋能。2021年我国启动了“区块链创新应用试点行动”，在国务院、央行、最高法等18个重要部门启动区块链创新试点工作。同年5月，工信部联合中央网信办发布了《关于加快推动区块链技术应用和产业发展的指导意见》，指出到2025年我国区块链产业综合实力达到世界先进水平，产业初具规模，区块链应用渗透经济社会多个领域，形成场景化示范应用。

区块链按节点准入机制分为公有链、联盟链和私有链[1]。联盟链节点因严格的准入机制具有较高的信任度，故大多数联盟链采用性能较高的传统BFT[2]类分布式共识算法加快达成共识。因此，监管友好、性能优异的联盟链成为区块链落地商用的首选[3]。我国以联盟链为主的区块链产业规模至2021年已突破65亿元[4]。

系统可靠性更加关注数据在系统中的长期存在性[5]，区块链系统的可靠性主要体现在历史数据长期存在和难以篡改两方面。联盟链应用的主要参与方多为权威机构、第三方平台、企业等，仍然存在一定的中心化特征，相较于公有链，联盟链通常在合谋等一些攻击上更易实现[6]，因而区块链的历史数据能在一定时间内被篡改，影响其可靠性。因此，承载了大量重要行业应用的联盟链亟须获得更强的可靠性保障。本文将现有联盟链可靠性低的原因总结如下：

1）有效节点数少。有效节点参与维护联盟链，参与联盟链的有效节点数越少，合谋等攻击越有可能发生[6]。联盟链参与节点数较少的原因是区块链使用者受提供解决方案企业的规模、建链成本以及使用机构的组织结构制约。

2）节点独立性低。通常来说，维护联盟链的节点行为相互独立程度越高，区块链的去中心化程度

收稿日期：2022-09-29；**录用日期**：2022-11-25；**网络出版时间**：2023-03-09 16:07
网络出版地址：link.cnki.net/urlid/11.2625.V.20230309.1406.007
基金项目：国家重点研发计划 (2021YFB2700800)
*通信作者. E-mail：zhaoh@hfcas.ac.cn

*引用格式：赵鲜鲜，谭海波，赵赫，等. 一种基于多链协同的联盟链改进模型 [J]. 北京航空航天大学学报，2024，50（10）：3283-3296.
ZHAO X X，TAN H B，ZHAO H，et al. A consortium chain improvement model based on multi-chain collaboration[J]. Journal of Beijing University of Aeronautics and Astronautics，2024，50（10）：3283-3296（in Chinese）.*

越强，篡改难度越大。节点独立性受区块链本身承载业务的特性所限，多数维护区块链的节点基本隶属同一行业，甚至是同一机构[4]，且一般无法接受其他机构或与行业无关的主体。

3）可持续发展能力弱。区块链系统的可迁移性和恢复能力越强，历史数据长期存在性越好。可持续能力主要体现在技术选型等原因产生的数据需要进行数据迁移[7]和系统数据丢失、崩溃或停止运行情况下的容灾备份。现有大部分联盟链一旦发生数据损毁或停止运行等事件后就无法恢复数据以及证明数据存在。

针对以上原因，本文提出同盟链模型，以期在不干扰区块链原有事务和不显著增加区块链的维护负担下，降低链上历史数据的篡改概率和设计账本恢复机制，以提升区块链的可靠性。通过不同联盟链间持续存储对方每个阶段发送的区块链数据，形成区块链数据的相互层叠存储，使同盟链模型中的每个节点都能间接参与所有联盟链数据的维护；通过对加密区块链数据的协同存储，提升链上数据存储的去中心化程度，在有相关证明的情况下能够对损毁或缺失的数据进行反向恢复或佐证链上数据的真实性。

1 相关工作

1.1 现有联盟链可靠性提升方法

现有提高联盟链可靠性的方法主要为单链改造和多链协同两种。

单链改造分为直接扩展区块链节点数量和改进共识节点可靠性。直接扩展节点数量是最简单的可靠性提升方法，理论上能够降低系统被攻击的风险，但该方法增加的节点并非一定为有效节点或独立性高的节点，一定程度上可能具有中心化特征。改进共识节点可靠性通过提高共识节点的服务可靠性或随机性来提升系统抵御风险的能力[8-9]，但该方法在节点数较少或大部分共识节点信任度不高的情况下不能提高系统可靠性，且难以在已运行的联盟链上进行改造。

多链协同指借助其他区块链来提高原有链可靠性，比较典型的是锚定模式[10-11]和跨链协同[12-15]存储模式。锚定模式是指将原有区块链的数据定期存储至公有链或主链中，通过公有链或主链所有节点存储该段区块链数据进行背书，2013年PoE[10](proof of existence)提出将数据定期锚定至比特币区块链中，通过返回的时间戳等数据反向验证其在原有区块链中的存在性。跨链协同模式指原有区块链数据流通在多个其他区块链中，通过其他链该流转数据的相关交易证明原有链上该数据的存在性。Optimistic rollup[13]是以太坊首主导的扩展性技术，在数据流转的基础上保证了数据可用性。其主要思想：节点缴纳一定保证金成为聚合者，聚合者将一段时间内原有链的交易数据压缩并聚合，再通过智能合约将其存储在另一条链上，并进入较长时间的等待期。若等待期内没有节点发出欺诈证明则交易会被确认，聚合者将得到奖励，反之聚合者将受到惩罚。

1.2 多链协同机制面临的问题

单链改造中除直接提升节点数量方案外，大概率还需要暂停区块链的运行，这对于正在使用区块链处理业务的机构来说是难以接受的，且暂停期间数据容易被篡改，而直接增加节点数量可能会带来中心化的风险和较高的成本（特别是在实际操作节点用户较少的联盟链中），故本文采用多链协同方式来提高区块链的可靠性，而采用多链协同机制则需要对数据来源真实性、数据存在性、完整性进行保障[16]。

数据来源真实性，主要对链间数据来源进行认证，现阶段大多采用数字证书、数字签名等数据认证方案[17]。本文默认每条联盟链由一个真实且唯一的数字身份进行初始化并通信。联盟链的准入机制使得第三方通常无法查看链上数据，数据来源的真实性判断就转变为如何证明节点向外传输的数据是真实的区块链数据。因为联盟链能够追溯到每一个节点的身份，且大多数共识节点是诚实的，因此，我们可以采用多数节点投票签名和多个不同共识节点确认消息的方式解决上述问题，同时运用聚合签名压缩多个节点签名以减少通信数据量的大小。

数据存在性，本文主要针对区块链交易，只有数据发送给其他区块链后验证是否被真实存储，附加到区块链账本中的数据才具有比较高的防篡改能力。区块链的安全性由所有节点进行维护，交易是将数据添加到账本后的承载实体。本文通过对比交易数据并使用Merkle树[18]验证该交易在其他区块链中的存在性。

数据完整性，一方面体现在传输数据所指向区块链的连续性，防止发送方伪造区块链；另一方面体现在传输的加密数据在保存方中的完整性，防止保存方篡改、删除甚至重新外包加密数据。对于发送方，本文通过密码学累加器保证数据链的连续性。密码累加器通过具有准交换属性和单向性的哈希函数开发出一种可用于为时间戳和成员资格测试的单向累加器加密协议，于1994年首次被Benaloh[19]提出，分为对称累加器和非对称累加器，其能够高效地证明元素是否存在于集合中。以每

次传输的数据为累加器元素构造累加器, 若根据之前的累加器根和本次数据能够构造和发送方相同的新累加器值, 即隐式地确认了之前传输数据的存在性, 进而保证多个节点确认后发送方区块链的完整性; 对于保存方, 本文采用数据可拥有证明(provable data possession, PDP)防止加密数据被篡改, 并通过使用不同密钥加密数据防止数据外包。PDP 是一种用于远程完整性的方案, 基于该方案可以检测外包数据中数据的损坏。Ateniese[20] 在 2007 年定义了通用的 PDP 协议架构, 通常由数据发送方随机挑选存储的元数据发起挑战, 数据存储方根据挑战访问小部分文件块生成证据证明保存数据的完整性。

由于本文的重点不在于密码学工具的改进, 简单起见, 所用工具均采用较简单的实现方式, 其他诸如改进或具有相似作用的密码学工具都可以替代本文所述工具, 以提升安全性或效率。

2　同盟链模型

2.1　系统架构

本文的系统架构如图 1 所示, 主要由联盟链、信使节点和普通节点 3 种角色, 以及成员通信配置(alliance intercommunication configuration, AIC)、通信服务、同盟通信协议 (alliance communication protocol, ACP) 组成。相互背书的数据为协同数据, 区块链上确认一定区块数量后由该联盟链的信使节点调用通信服务按照协议 ACP 发送。接受方联盟链的信使节点通过通信服务监听并解析协同数据, 将符合条件的数据构造交易上链, 并将需要保存的加密数据存储到相关介质中, 接受发送方的完整性挑战并给出挑战应答。该存储介质可以是云存储, 也可以是硬盘等。为了保证信使节点的服务质量, 信使节点由联盟链中的共识节点产生。

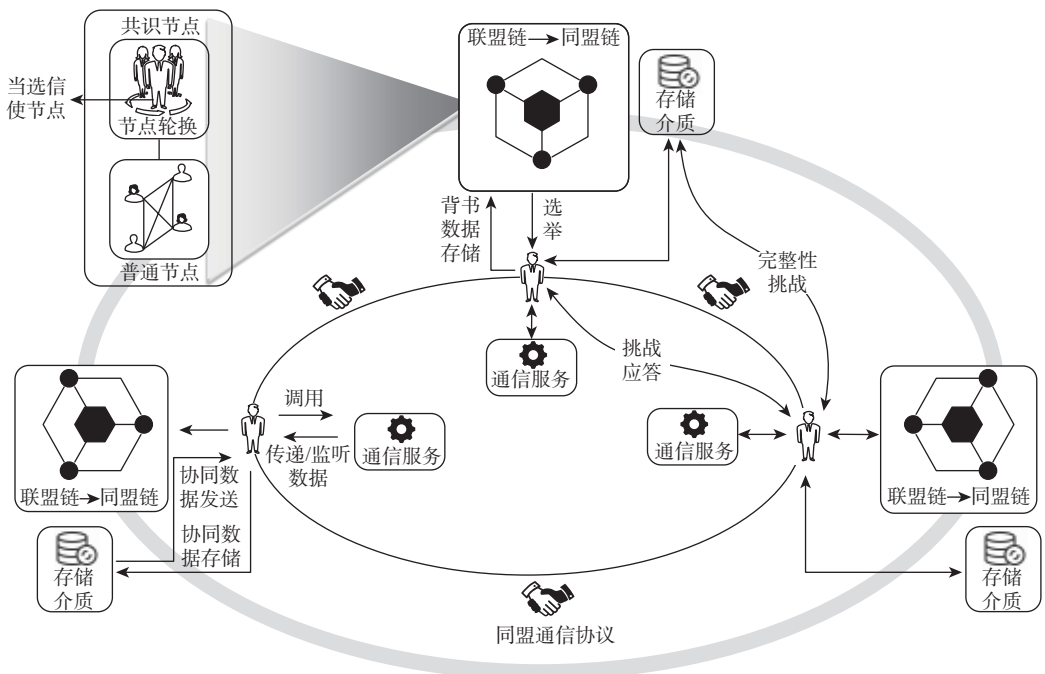

图 1　同盟链系统框架

Fig. 1　System framework of alliance chain

在同盟链模型中, 多个联盟链在自己的账本中不断附加其他结盟区块链协同消息的标识信息, 从而形成区块链数据间的层叠, 若此时篡改一个已协同区块, 难度将增大。如图 2 所示, 联盟链 A 中第一个轮次 Epoch1 的协同数据 A_1, 其唯一标识满足一定要求后在 Epoch2 中被附加在联盟链 B 和 C 中的某个区块里, 随着联盟链之间不断相互存储不同轮次的唯一标识, 联盟链之间的数据层叠存储, 一个已协同的数据还需要篡改其他存储该数据标识的区块链中对应的数据, 联盟链之间形成同盟关系。

2.2　系统组件

同盟链: 需要提升可靠性的联盟链, 相互协同存储后组成同盟链, 符合 BFT 类协议的基本假设。联盟链中包含普通节点、共识节点和信使节点。根据数据流向联盟链可以分为数据发送方和数据协同方。

信使节点: 信使节点是同盟链模型中数据的收发方, 通过通信服务接收验证其他联盟链传输的消息、发起对传输数据的签名、聚合所有消息签名并通过通信服务发送数据至其他联盟链。

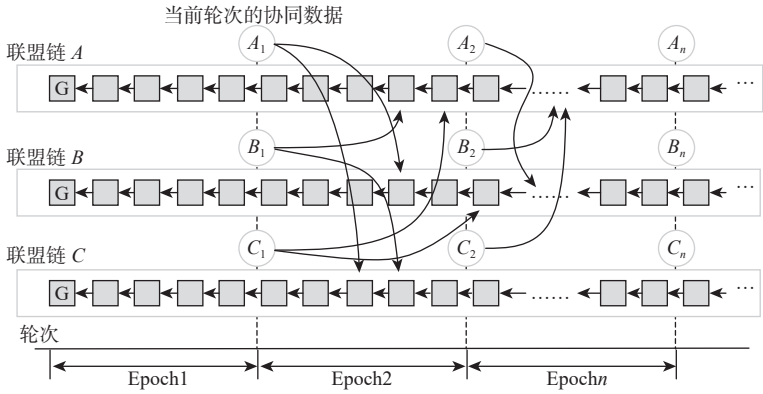

图 2 链间数据存储层叠示意

Fig. 2 Schematic of data storage overlapping between blockchains

普通节点：普通节点负责对同盟链中的事务进行验证和签名。

协同数据：协同数据包括链上背书数据 EndorseData 和链下存储数据 ArchiveData。EndorseData 包括当前轮次发送区块的区块头和必要的验证数据，能够唯一标识该数据的信息通过验证后存储在数据协同方的帐本中，即多个联盟链数据存储形成层叠关系。ArchiveData 包括数据发送方加密的区块链数据和其他需要存储的数据如运行日志等，其不仅可以恢复数据，也能帮助数据发送方恢复到当时的运行状态。ArchiveData 存储在数据协同方的存储介质中，只有当选的信使节点才能向其中追加数据，其他信使节点仅被允许查看，因此该数据不是必需的传输数据。本文使用开源库 Zlib[21] 对 ArchiveData 进行压缩来减少数据大小，后文所述 ArchiveData 均为已被压缩的数据。

通信服务：通信服务和链上节点绑定，具体功能包括监听、路由与数据转换。监听功能分为链上数据监听和协同数据监听，链上数据监听指该服务能够获取链上的状态和数据，协同数据监听指能够获取其他联盟链发送的消息；路由功能根据 AIC 中的路由信息找到传输的对象；数据转换功能将链上监听的数据按照 ACP 的格式进行包装进并请求信使节点对包装后数据的签名。信使节点收到数据后，根据 ACP 对数据展开解析；使用双向数据存在性验证方法（verifiable bidirectional method, VBM）检验区块链的连续性和历史协同数据交易的存在性；校验数据是否符合 AIC 中的各项规则，通过后数据协同方对该数据进行存储。普通节点不具备监听和路由的功能。简便起见，本文后面仅描述节点操作，不再对节点和服务间的交互细节进行阐述。

2.3 成员通信配置 AIC

联盟链间通信时，需要一些配置信息帮助路由找到其他信使节点，并检测联盟链、节点以及传输信息是否合格，关键配置如表 1 所示。

表 1 AIC 关键配置项

Table 1 Key configuration items of AIC

配置项	作用
Ecert	现任信使节点当选证明
AEnvInfo	其他联盟链信使节点证书列表和路由信息
AConfigList	其他联盟链配置信息
EDataQuene	收发双方待确认、已确认数据次数映射列表
ECerList	记录每个成员近期传输证明映射列表
BlackNode	记录禁止通信的信使节点名单
BlackChain	记录作恶联盟链唯一标识列表
Miniconfig	所有联盟链确认加入同盟链模型的最低配置

表中，AConfigList 记录其他联盟链的配置，该配置为成员加入时发送的节点总数、链类型、公钥列表等，可以检验联盟链数据合法性；EDataQuene 能防止其他联盟链的单个信使节点作恶，存储当前轮次发送区块数量、最新区块的高度和对应的确认次数，前一次数据确认后，列表对应字段加 1；ECerList 能够防止由于异常导致的累加器根值不同无法检验或丢弃有用数据的问题；BlackNode 记录作恶节点地址和公钥；BlackChain 记录作恶区块链的 ChainID；Miniconfig 能够防止新成员加入导致的同盟分裂。

2.4 同盟通信协议 ACP

联盟链间的通信依赖于同盟通信协议 ACP，是为了实现联盟链间交换信息而设计的规范，其详细结构如图 3 所示。其中，图 3 左侧为数据传输的详细格式，发送链和接收链有唯一标识，统称为 ChainID，from 和 to 表示数据收发方；消息类型 MType 包括同盟成员加入 MemAdd、协同传输 MemTran、同盟配置更新 MemUpdate、信使节点更新、同盟成员退出 MemQuit 和数据使用 DataUse；Content 是根据 MType 构造的消息内容；Extra 是自

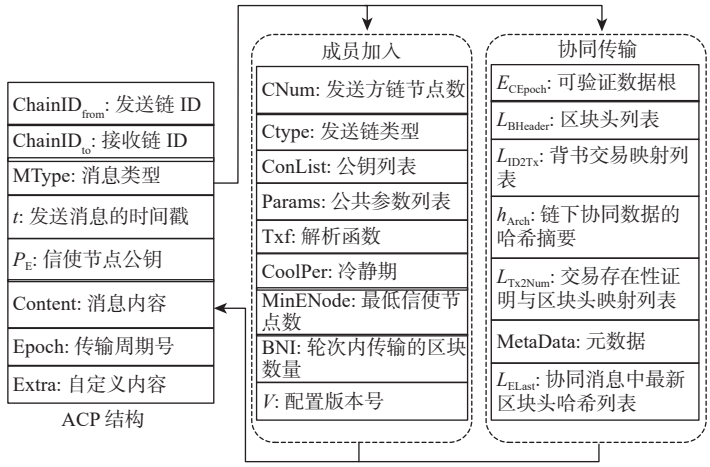

图 3　ACP 结构

Fig. 3　Structure of ACP

定义内容, 用于存放成员间相互协商的其他参数, 以及信使节点当选证明和警告信息。图 3 右侧为 MemAdd 和 MemTran 消息的构造。其中, L_{ID2Tx} 包括<ChainID, T_{x_i}>, T_{x_i}是序号为 i 的交易, 该交易需要对应区块链的信使节点加密; L_{ELast} 包括<ChainID, h_{BLast}, EC>, h_{BLast} 为一个轮次中最新的区块头哈希, EC 为当前链的链上背书证明; L_{Tx2Num}=<T_{x_i}, MerkleProof, N_B>, Merkle-Proof 为交易 T_{x_i} 的证明路径, N_B 为区块号; 成员退出时只存放最后发送消息的可验证数据和返回的数据索引; 同盟配置更新仅构造更新的参数; 信使节点轮换用新信使节点的当选证明 Ecert 和其他消息类型一起发送。

2.5　双向数据存在性验证方法 VBM

联盟链之间不仅需要数据发送方证明传输数据为真实的区块链数据, 而且需要证明数据协同方之前传输的数据经过验证后已经存储在数据发送方的区块链中。VBM 基于累加器构造, 累加器成员为 EndorseData 的唯一标识, 累加器根为背书数据证明 (endorse certification, EC)。

所有联盟链维护一个 EC, 不同链可能会由于网络延迟等原因异步接收消息从而使下一步发送协同消息中的 EC 值不同, 但由于 AIC 中存储了 ECerList, 因此能够借助近期历史数据对不同链进行异步验证。

数据发送方第i个发送轮次发送的区块头列表如图 4 所示, T_{x_3}为轮次 i 中数据协同方被确认的历史协同数据所构造的交易, 序号为 3。EC 构造算法 ConEC() 将发送区块列表哈希值 h_{Blist}、列表最新区块头哈希 h_{BLast}、列表最新区块高度 H_{BLast}、链下存储数据摘要 h_{Arch} 和 Epoch 构造成最新的累加元素添加进累加器中。数据协同方对 EC 的合法性验证算法 VerifyEC() 分为两步, 第一步验证协同方已达确认阈值的协同数据是否在发送方区块链数据中, 首先验证T_{x_3}的内容是否为已达阈值的确认数据, 通

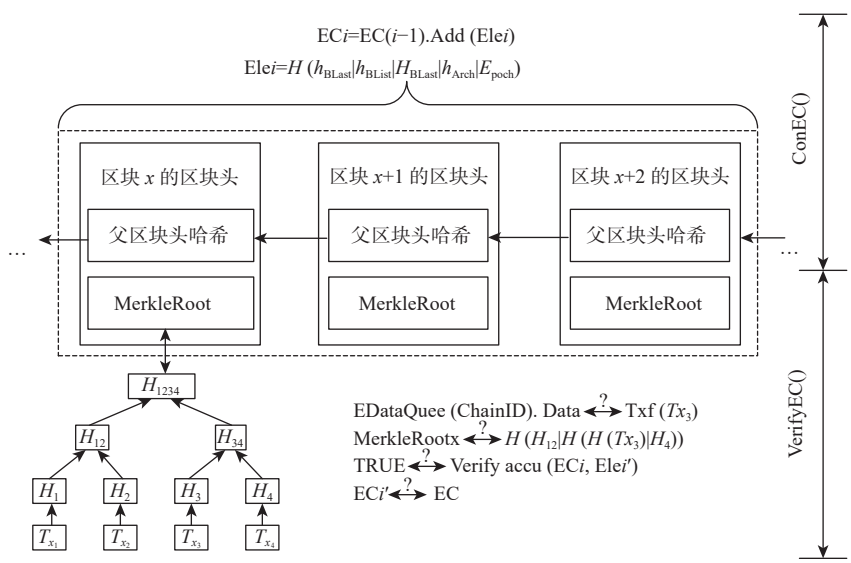

图 4　EC 构造和验证过程

Fig. 4　Process of EC construction and verification

过T_{x_3}的路径证明生成MerkleRootx, 对比是否与区块x中的根值MerkleRoot相等, 即判断其是否在区块链中; 第二步判断传输数据是否连续, 数据协同方根据ConEC()重新计算得到累加元素Ele$i$', 并根据前一次的证明EC(i−1)构造ECi', 与数据发送方发送的ECi进行对比。由于所有联盟链共用一个累加器, 故需要ECerList保存每个联盟链近几次的EC, 否则协同方只能证明该段数据的存在性, 而不能证明单个联盟链数据的连续性。

3 同盟链通信过程

本文所设计方法可以支持多个联盟链, 为了更加简单直观地描述同盟链的通信过程, 在此以联盟链A和B的行为阐述主要流程, 并将A作为行为发起方。

3.1 初始化阶段

由于联盟链不对外公开, 初始化时难以保障参数的正确性, 本文假设ChainID、信使节点公钥与IP、ACP协议已通过可信方式被其他联盟链所知。

1) 参数初始化

选定单个联盟链的安全参数γ, 基数为p的循环群G_1和G_2, 双线性映射$e:G_1\times G_2\to G_T$, 选择两个抗碰撞的安全散列函数H_0、$H_1:\{0,1\}^*\to G_1$, 私钥k和公钥P作为联盟链的聚合公私钥对, 根据每个联盟链的ChainID构造ArchiveData加密的公私钥对$\{k_{\mathrm{ID}},P_{\mathrm{ID}}\}$, 向所有联盟链公开$P$和$P_{\mathrm{ID}}$, 联盟链内部公开私钥$k$和$k_{\mathrm{ID}}$, 为各节点保存获取的密钥。系统参数为

$$\mathrm{Params}=\{\gamma,G_1,G_2,H_0,H_1,p,e,P,P_{\mathrm{ID}}\}$$

2) 信使节点选取

由于不同的区块链共识节点选取方式的差异, 难以使用统一的选举方式选出信使节点, 本文使用原区块链内已被认可的共识算法在共识节点集合中选举信使节点, 对此不再赘述。

A选出任期i内的信使节点$E_A^{i_0}$和j个候选节点$\mathrm{NodeListE}=\{E_A^{i_0},\cdots,E_A^{i_j}\}$, 当信使节点$E_A^{i_0}$作恶或者宕机时, 候选节点会广播当选信息并从已确认的位置开始执行。为了使链中原有的共识过程尽可能不受其他联盟链的影响, 当选的信使节点在任期内将不进行链内事务的共识。

E_A为当选节点, 当选后其在A中广播当选信息Ecert=$\{T_{\mathrm{S}},\mathrm{SigPE},P_{\mathrm{E}}\}$, $\{k_{\mathrm{E}},P_{\mathrm{E}}\}$为节点公私钥, T_{S}是信使节点竞选时公布的时间戳, σ_i表示其他节点对当前信使节点T_{S}和P_{E}的签名, Sig表示聚合签名, SigPE是E_A对Sig的签名。

3.2 构造阶段

E_A遵循ACP协议并根据传输消息的类型构造不同的消息M。

当构造MemAdd时, E_A首先向其他联盟成员请求同盟最低配置Miniconfig和其他联盟链的配置信息, 验证后向所在链节点广播并存入AIC相应字段中。E_A根据协议构造链初始配置, 广播该配置并收集区块链中其他节点对该配置的签名。MemAdd未被确认前不能进行协同数据的发送, 只能接受其他联盟链的信息。当被确认后从创世区块开始每个轮次都发送BNI数量的区块。

当构造MemTran时, E_A根据传输的区块头列表L_{BHeader}调用函数ConEC()计算EC, 检查当次传输的区块中是否有其他联盟链之前发送的协同数据, 若有, 则构造L_{ID2Tx}和L_{Tx2Num}。若发送ArchiveData还需要使用B对应的密钥对原始数据加密, 则会生成同态标签并发起历史协同数据完整性的挑战。同态标签生成时, 信使节点首先会查找ChainID对应的公私钥对$\{k_{\mathrm{ID}},P_{\mathrm{ID}}\}$。将ArchiveData进行分块处理后得到$F=\{m_1,\cdots,m_n\}$, 选取分块的唯一文件标识$v$和辅助变量$u$, 生成认证元$\theta_i=(H(m_i)\cdot u^{m_i})^{k_{\mathrm{ID}}}$, 集合$\Phi=\{\theta_1,\cdots,\theta_n\}$, 对认证元集合构造Merkle认证哈希树[18]M_{Tree}, M_{Tree}存放入字段Metadata中。在标签生成后, E_A可以在任意时间发起对该数据的挑战。E_A从之前的F中随机选出c块, 并从每个块索引中选取一个随机数v_i, 生成挑战$C_{\mathrm{ID}}=\{i,v_i\}$。

应答方需要在相应时间内进行回应。E_B接收到该消息时检索ArchiveData, 并生成应答。E_B根据挑战方发送的挑战数据, 计算证明$\{\theta,\mu\}$、块索引i在M_{Tree}中的路径证明Ω_i和$H(m_i)$, 一起发送给挑战发起的信使节点, 公式如下:

$$\theta=\prod_{i=1}^{c}\theta_i^{v_i} \tag{1}$$

$$\mu=\sum_{i=1}^{c}v_i m_i \tag{2}$$

当构造MemUpdate时, E_A根据所在联盟链或Miniconfig进行参数调整, 在新配置被确认前, 仍需要采用旧配置进行通信。

当构造MemQuit时, 若联盟链正常退出, E_A则将所在区块链最新的EC放入Content中, 等待其他联盟链返回A中已被确认数据的索引信息和对该EC的签名作为归档信息, 等待Coolper时间后可退出。

警告消息将发送错误消息的节点公钥、节点IP、错误消息描述、错误消息摘要、错误消息的聚合签名放入字段Extra中。

3.3　同步阶段

同步流程如图 5 所示, 流程详细叙述如下:

1) 在数据发送方中, 若 E_A 为第一次进行协同数据传输, 则需要将 Ecert 放入 Extra 字段中, 再根据传输消息类型 MType 构造消息 M。

2) E_A 将消息 M 在 A 中发起共识。

3) A 中其他节点对消息 M 验证并签名。

4) 签名的节点将签名 σ 发送给 E_A。

5) 当收集到多数签名后, E_A 将签名进行聚合生成 Sig。签名聚合的具体流程如下:

a. 联盟链内节点验证 E_A 对消息的签名是否真实, 真实则对 M 进行签名, 签名为 $\sigma = kH_0(M)$。

b. 信使节点收到签名后验证签名是否合法, 合法则接受该签名, 并记录签名的公钥至公钥列表 $\text{ConList} = \{P_1, \cdots, P_n\}$。当收集到超过半数的签名后开始进行聚合。

c. 计算 n 个用户对消息 M 的聚合签名 $\text{Sig} = \sum_{i=1}^{n} \sigma_i$。

6) E_A 构造 ACP 并签名后发送给其他联盟链。

7) B 中的信使节点 E_B 解析数据并检查消息的合法性、是否有警告消息, 不合法则构造警告消息, 若有警告消息则查看消息类型并采取相应措施。若为 MemTran, 则可以调用 VBM 方法检验历史协同数据的存在性和确认对发送方之前数据的确认, 并对完整性挑战作出应答, 该应答可以实时传输给 E_A。若消息类型为 DataUse, 则验证后查找相应的数据并构造证明传输给 E_A。

8) 信使节点 E_B 将验证后的数据 M 签名后广播给所在区块链中的其他节点。

9) 普通节点将 M 存放在 EDataQuene 中, 当信使节点发起上链交易后将相应数据从 EDataQuene 中删去, 上链数据构造交易为 $T_x = \{\text{ChainID}, N_B, h_{\text{BLast}}, h_{\text{Arch}}, \text{Epoch}, \text{BNI}\}$, 被确认的数据将会接受完整性挑战。

10) E_A 对 E_B 返回的挑战应答数据进行验证。

11) E_B 将构造的数据传输给 E_A。

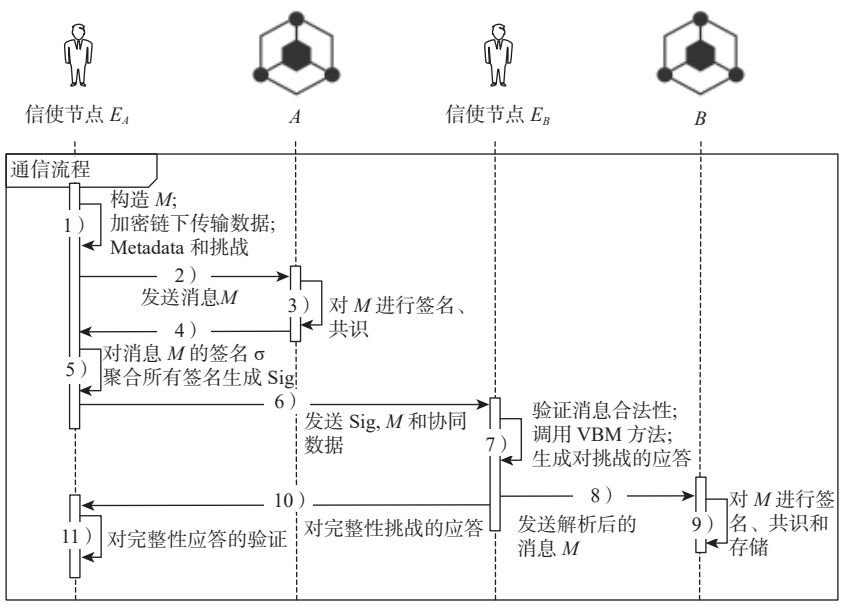

图 5　同步流程

Fig. 5　Synchronization process

3.4　验证阶段

为防止数据发送方中某信使节点伪造数据, 传输数据需经过多个信使节点确认才可保障正确性, 本文在联盟链结盟初期设置了信使节点的最低数量 MinENode。信使节点在接收到数据时需要先对消息进行验证, 包括 t、ChainID、信使节点的 Ecert 以及消息的摘要, 通过验证等式 $e(\text{Sig}, p) = e\left(H(M), \sum_{i=1}^{n} P_i\right)$ 是否相等判断聚合签名是否有效, 然后根据不同的消息类型进行不同的验证。若验证不通过, 将构造警告消息。

当验证 MemAdd 时, E_B 对消息 M 校验流程如图 6 所示。

1) E_B 检查签名消息的节点是否为新的信使节点, 若是, 执行步骤 2; 若否, 执行步骤 4。

2) E_B 检查 ChainID 下是否有归档信息, 即 EC 的存在性证明, 若有, 则证明该联盟链之前参与过同盟链的协同过程, 执行步骤 3; 若没有归档数据, 则执行步骤 4。

3) 查看消息 M 中归档信息对应的相应索引,

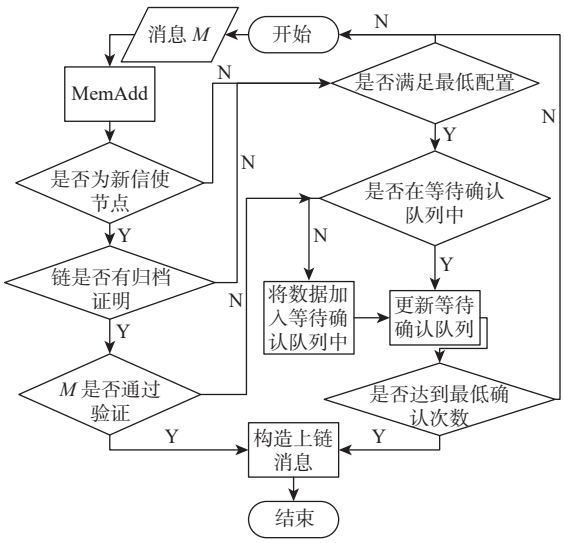

图 6 MemAdd 类型消息验证流程

Fig. 6 Validation flowchart for MemAdd type message

若找到，则执行步骤 5，若找不到则说明该归档信息有误，执行步骤 4。

4）E_B 查看 AIC 中关于联盟的最低配置，若消息 M 满足，则执行步骤 5，若不满足则丢弃 M，构造警告消息，同时继续监听新的消息。

5）E_B 查看消息 M 是否已经在等待确认的数据队列 EDataQuene 中，若无，则将消息 M 加入 EDataQuene 中并将确认次数置为 1，等待满足最低确认次数的阈值 (最低信使节点数)，若 EDataQuene 中已存在，则增加 M 的确认次数，执行步骤 6。

6）若未达到最低确认次数的阈值，等待新的信使节点传递对消息 M 的确认，重复执行步骤 1～5；若已达到最低消息阈值，则执行步骤 7。

7）E_B 对该消息 M 构造上链消息。

当验证 MemTran 类型数据时，E_B 根据 VBM 验证发送的数据是否合法、真实且连续。

当验证 MemUpdate 时，E_B 检查配置项中是否符合满足 MiniConfig，不满足，则丢舍该配置，满足，则将该配置放在 EDataQuene 中，等待满足最低确认次数。在确认最新配置之前，使用最新已被确认的配置进行通信。

当验证 MemQuit 类型数据时，E_B 收到退出消息后，对 A 保存在链上的配置信息和状态信息交易哈希做归档，并将链上有关的交易哈希索引等信息构造进 Content 中，并等待链内共识完整后传输给 E_A。E_A 在接收到其他成员传输的信息后，存储相应数据，等待 Coolper 后退出。正常退出的联盟链在退出后可以根据索引信息在相应区块链中查找历史协同数据，根据归档证明能够快速加入同盟链并证明历史协同数据的存在性。

除此之外，还需要对链下协同数据的完整性挑战应答进行验证，即验证交易路径并计算根节点的哈希值，验证通过则判断等式 3) 是否相同：

$$e(\theta,p)\stackrel{?}{=}e\left(\prod_{i=1}^{c}H(m_i)^{v_i}u^{\mu},P_{\mathrm{ID}}\right) \tag{3}$$

若流程中有不正确的数据，包括假冒信使节点、错误传输消息、错误的完整性证明、严重超时的数据确认和证明以及故意的数据不确认等，信使节点除了针对可能丢失的传输消息进行重传外，严重超时和确认的信使节点将会被放入黑名单中，超过一定次数后便不再与该节点进行通信，而假冒信使节点会被直接禁止通信。异常信息会被构造入警告信息，重传信息会被构造进 ACP 的 Content 里，并短时间内进行传输，不占用正常协同消息的时间。警告消息由 Coolper 时间来发送情况说明撤销处罚。

3.5 数据使用阶段

该阶段在 A 需要账本数据恢复或需要其他联盟链提供链上数据背书证明时进行，分为数据索取和返回两个阶段。

数据索取：E_A 构造 DataUse 类型数据 M_1。E_A 将需要请求的 A 链的数据编号发送给其他联盟链，即区块号区间<N_{Bs}、N_{Be}>和链下数据同步序号区间<N_{Es}, N_{Ee}>、N_{Bs} 和 N_{Be} 表示开始和结束的区块号、N_{Es} 和 N_{Ee} 表示开始和结束的轮次号。

数据返回：E_B 验证 M_1 后按照其中的编号将相关数据构造成 M_2 发送给 E_A。E_B 查找包含 N_{Be} 的链上背书交易，根据交易中的 BNI、N_B 进行查找包含 N_{Bs} 的背书交易 T_x，将<N_{Es}、N_{Ee}>期间所有的交易数据内容生成背书交易列表 ETxList。若还需要 ArchiveData，则根据<N_{Es}、N_{Ee}>查找相应发送轮次内的链下加密数据 ArchiveData，生成链下协同数据列表 ArcList。

E_A 验证 M_2 的合法性后，便开始进行数据的验证和比对。将 A 链的相关数据和 ETxList 进行对比，包括计算数据的完整性、验证数据合法性，若有多个联盟链则均需要进行对比，验证成功后根据 Epoch 将链下数据进行排列，最后通过对应的私钥 k_{ID} 对数据进行解密，从而完成账本数据的恢复。若无 ArchiveData，直接使用证明即可。

4 安全性分析

4.1 合谋攻击

当大部分节点进行作恶时，能够很快篡改区块链数据。总节点数 N_{node}，完整存储数据节点数为 N_{copy}，当网络中作恶节点数超过 N_{copy} 的一半时，攻击节点可以在有限时间内共同篡改数据。成功攻

击的概率 P_{Fault} 如式(4)所示:

$$P_{\text{Fault}}=\begin{bmatrix}N_1\\N_2\end{bmatrix}\times\begin{bmatrix}N_{\text{nodeFault}}\\N_2\end{bmatrix}\Big/\begin{bmatrix}N_{\text{node}}\\N_2\end{bmatrix}\tag{4}$$

式中, $N_1=N_{\text{node}}-N_{\text{nodeFault}}$, $N_{\text{nodeFault}}$ 表示作恶节点数, $N_2=N_{\text{copy}}/2$。在总结点数 N_{node} 为 30, N_{copy} 为 20 时, 由式(4)得出图 7, 即得出账本被篡改的概率。其中横坐标为作恶节点占总节点数的百分比, 纵坐标为篡改数据的成功概率。由图 7 可以看出, 当作恶节点数在 40% 以下时, 被篡改的概率小于 10%。节点数较少时, 作恶节点数超过 40% 的概率会很大。

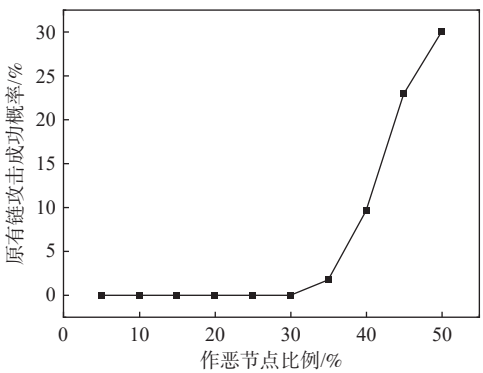

图 7　数据篡改成功概率

Fig. 7　Probability of successful data tampering

采用本文方案时, 该数据在其他联盟链上背书后, 恶意节点篡改数据需要攻破其他联盟链, 设所有联盟链配置相同, 则恶意节点成功更改的概率为

$$P_{\text{Fault}}=\prod_{i=1}^{C\text{num}}\begin{bmatrix}N_1\\N_2\end{bmatrix}\times\begin{bmatrix}N_{\text{nodeFault}}\\N_2\end{bmatrix}\Big/\begin{bmatrix}N_{\text{node}}\\N_2\end{bmatrix}\tag{5}$$

根据该公式作图 8, 得出联盟链数量影响下单个联盟链被成功攻击的概率, 其中横坐标为联盟链节点个数, 纵坐标为攻击成功概率。由此看到, 本文的方案可以有效抵御单个联盟链阶段对历史确认数据的合谋攻击, 增加一个同等配置的联盟链就可以使占比 50% 的攻击节点篡改成功概率下降至

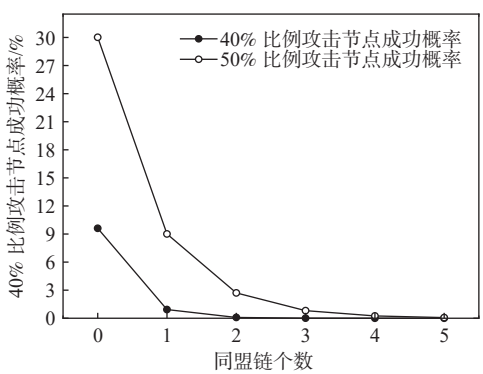

图 8　使用本方案的数据篡改成功概率

Fig. 8　Probability of successful data tampering with our scheme

10% 以下, 随着联盟链数量的增多, 该概率将进一步下降。

4.2　中间人攻击

本方案使用信使节点作为传输数据的媒介, 为了保证双方通信时数据的安全性, 本文针对单个信使节点作恶的情况提出在加入初期联盟链之间协商最低信使节点确认次数来降低作恶的可能。通过在共识节点中选取信使节点来保证服务的可靠性。初始化时,单个节点有概率对区块链的参数进行造假。假设单个节点不诚实的概率为 δ, 那么在最低确认次数为 3 的情况下, 作恶的概率降低为 $\delta^3(0<\delta<1)$。

同时, 为了防止传输期间信使节点进行合谋, 本文要求联盟链中大多数节点对传输的消息进行签名, 并采用聚合签名对所有消息签名进行聚合以减少数据大小。即使恶意节点可以伪装所有其他签名者, 并可以对诚实签名者发起选择消息攻击, 它在任何概率多项式时间内也不可能伪造诚实签名者的有效聚合签名。因此, 信使节点单次发送消息不可伪造, 即单次消息可以代表整个联盟链。在接收到数据时, 聚合签名正确性验证过程如下:

$$e(\text{Sig},p)=e\left(H(M)\sum_{i=1}^{n}k_i,p\right)=e\left(H(M),\sum_{i=1}^{n}P_i\right)\tag{6}$$

ACP 在传输时会附带其他联盟链该轮次发送并已被确认的 EC 值, 该 EC 值可以增强其他联盟链对传输数据正确性的验证。

4.3　链下存储数据完整性欺骗

单次数据损坏或未存储的比例为 f_0, 挑选 c 块, 使用完整性验证方法时有 $1-(1-f_0)^c$ 的概率识别出协同方未完整保存数据。同时, 由于其他联盟链可能会合谋, 导致数据协同方的 ArchiveData 备份数量少于联盟链数量, 可通过使用对应 ChainID 生成的密钥对 ArchiveData 进行加密, 生成不同的数据, 使得所有联盟链存储的数据不一致, 以避免该种情况的发生。

数据协同方若不能提供正确的证据完整性挑战, 则不能通过以下等式:

$$\begin{aligned}e(\theta,p)&=e\left(\prod_{i=1}^{c}(H(m_i)u^{m_i})^{v_ik_{\text{ID}}},P\right)=\\&e\left(\prod_{i=1}^{c}H(m_i)^{v_i}\prod_{i=1}^{c}u^{v_im_i},P\right)=\\&e\left(\prod_{i=1}^{c}H(m_i)^{v_i}u^{\mu},P\right)\end{aligned}\tag{7}$$

4.4 上链背书欺骗

同盟链中联盟链之间由于不一定能够看到其他链的数据,因此在数据协同方接收数据后可能会出现不上链存储数据的情况。本文规定联盟链间需要在一定轮次内传输其他联盟链发送的历史协同数据所在的交易进行验证。通过解析交易中的内容并计算路径 MerkleRoot,该 MerkleRoot 必须存在于传输的区块头列表中的某个区块头中,因为数据传输方不太可能会伪造自己需要背书的区块列表,故通过验证后数据协同方会相信自己之前传输的数据已经在对方的区块链中进行存储。

4.5 背书数据不一致

数据发送方可能会由于节点作恶、轮换等原因造成发送数据不一致。本文采取多信使节点确认的措施,并将每一次传输的数据构造成累加器[22],当传输新数据时,由于累加器元素能够唯一确定单次传输的数据,因此若信使节点传输的新累加器不是基于之前的累加器和当前轮次传输数据构造的,则协同方验证不会通过。本文中,由于所有联盟链都对一个累加器进行更新,采用 RSA 累加器[22]不用担心陷门泄露问题,故本文采用通用累加器来保证传输数据的真实性。

4.6 发送方链分叉导致的数据不一致

当发送方的数据出现分叉后,数据在协同方中存在待确认和已确认两种情况,已确认指协同数据已在协同方中进行上链操作。

待确认数据可以重新发送该分叉所在轮次的相关数据,发送方会对协同方的警告消息做出数据重传的回应,在 Coolper 时间后,未得到最低信使节点数量确认的数据会被协同方删除,此时链上存储的是分叉后的数据。

已确认的数据不能被更改,若发送方链分叉后原先发送的数据不在主链中,则需要发送方根据前一次的协同数据构造 EC,此时的协同方存储了主链数据和分叉数据。

5 实验与评估

本节基于以太坊实现了同盟链的原型系统,并与其他方案在存储、通信和计算开销方面进行对比。

在实验中,对以太坊近一年内的数据进行随机采样,得出平均区块大小为 S_b=83 KB,平均每个区块 124 笔交易。同时设置区块头大小为 0.5 KB,单次发送区块数量为 100,单次协同数据分块为 100,挑战块数量 c 为 20,BLS 公钥长度为 256 bit,聚合签名长度为 256 bit,节点签名长度为 256 bit,哈希摘要为 256 bit,p 为 160 bit,累加器大小为 512 bit,RSA 模长为 1 024 bit,Merkle 树路径长度约为 7 层,此时路径大小为 256 Byte。本文使用基于 BLS 的聚合签名、基于 RSA 的累加器以及其他常规密码学工具。不同场景下可变参数不同,为了减少对实验结果的干扰,本次测量将解析的函数、自定义内容项、信使节点黑名单这些可变长度项设为空或范围内的最小值。在原型系统中选取区块编号为 1 261 到 1 460 的数据进行对比和测量。

5.1 链类型对比

引言提到现有区块链类型根据准入机制分为公有链、私有链和联盟链。随着区块链商用化的普及,现有链暴露出了各自的缺点:公有链对监管不友好,私有链仅适合机构内部使用,联盟链有效节点少、节点独立性低、链可持续发展能力差。本文提出的同盟链模型介于公有链和联盟链之间,每个联盟链作为一个整体成为同盟链模型的维护者。本文通过联盟链之间的相互存储背书使所有参与结盟的区块链节点能间接维护其他联盟链账本,这样可以较好地整合联盟链和公有链的优点,更符合现有商用化落地的需求。不同链类型对比如表 2 所示。

表 2 不同区块链类型对比

Table 2 Comparison of different blockchain types

对比维度	公有链	私有链	联盟链	同盟链
准入机制	无	有	有	有
去中心化程度	高	低	中	较高
交易速度	低	高	高	高
监管难易	难	易	易	易

5.2 方案对比

提高共识节点可靠性不能从本质上提升整个区块链的可靠性,故本文不对其作对比分析。本文主要从账本篡改难度、有效节点数量增幅、节点独立程度、有无账本恢复能力、适用范围和对原有链交易性能的影响程度 6 个方面同其他方案进行对比,结果如表 3 所示。

表 3 不同方案可靠性指标对比

Table 3 Comparison of reliability indicators of different schemes

方案	账本篡改难度	有效节点数量增幅	节点独立程度	账本恢复能力	适用范围	交易体验影响程度
方案1	中	中	低	无	广	中
方案2	高	高	高	无	较广	小
方案3	高	高	高	有	窄	大
本方案	高	高	高	有	广	小

方案 1:直接增加节点数量,能够增大账本篡改难度,若增加的节点之间存在一定行为关联,即不

完全独立, 则有效节点数低于新增节点数。一般而言, 节点数目增多, 区块链的维护难度上升、交易速度下降。该方案在遭受攻击或意外时主要信赖于使用区块链机构自身恢复措施的强度, 本身并不能主动恢复账本数据。

方案 2[11]: 锚定模式, 通过在公有链中发送上链交易存储数据, 借助公有链所有的维护节点共同抵御账本篡改类型的攻击。因该方案将数据存储在公有链中需要手续费, 成本随着时间呈线性增长, 所以有可能存在交易拥堵等情况导致系统可用性显著下降。该方法需要节点加入公有链发起上链交易, 故上链数据真实性依赖于该节点, 该节点是否作恶直接影响原有链的去中心化程度和安全性。该方案只能判断公有链中锚定的数据是否被更改, 无法恢复数据, 后面将不对其进行定量对比分析。

方案 3[13]: 多链协同模式, 利用传统跨链思想进行数据交互。当前跨链技术主要面向不同区块链资产之间的交易兑换, 需要借助可信第三方监听原有链来完成数据交换。Optimistic rollup 考虑了数据可用性, 但需要维护其他链, 且区块链间能够相互访问, 适用范围有限。同时, 该方案等待期较长, 对交易体验影响较大, 若等待期内无验证者提交欺诈证明, 则该数据就会被确认存在安全风险。

本方案: 通过设计较为完备的通信机制和验证措施使参与结盟的区块链存储数据形成层叠存储, 此时若想改动已经协同过的区块链需要所有联盟链的同意, 从而区块链中的各节点能够间接参与到其他区块链数据的维护工作中, 形成区块链间有效节点数的共享。同时通过协同存储进行数据灾备, 最终提升原有链的可靠性。相比方案 1, 本方案能够更好地保证增强节点数的独立性, 不大幅降低原有区块链的交易速度; 相比方案 2, 本方案对监管更友好, 节约成本; 相比方案 3, 本方案不需要强制要求区块链之间的业务关联性和资产透明性, 数据交互时也不需要锁定资产, 对联盟链来说, 没有第三方节点监听和介入原有区块链, 适用范围更广。

5.3　存储消耗

本文为了增强数据的可用性, 采用了链下存储的方式, 但方案 1 与方案 3 的方式都是链上存储, 为了较为全面地衡量不同方案的存储消耗, 将从节点存储开销(上链数据)和区块链存储总量(总共增加的存储量)两个方面进行对比。

1) 节点存储开销

方案 1 仅增加节点, 不增加单节点存储, 故本文仅和方案 3 进行比较。一个轮次中使用不同方案单节点的存储量增幅如表 4 所示, 能够看出, 在每个联盟链有效个数为 6、不同有效节点个数的情况下, 本方案单节点存储开销增幅较小, 优于方案 3, 但劣于方案 1, 相对存储总量和其他成本来说可以忽略不计。

表 4　单节点存储量增幅

Table 4　Increment of single node storage　%

有效节点个数	方案3	本方案
6	0	0
12	53.413 6	0.002 4
18	112.137 2	0.004 8
36	275.985 1	0.120 5
60	495.254 3	0.216 9

2) 区块链存储总量

本方案中, 节点在链下需要存储的数据主要为 AIC 和 ArchiveData。此时计算 3 个联盟链联合情况下的存储, 每条联盟链成员为 6, 最小信使节点数量为 2, 时间戳、成员链 ID 等数字占用 32 bit, ECerList 存储个数为最低信使节点数量加 1, AIC 所占大小为 1.48 KB。N_{Num} 为联盟链数量, 第 i 个联盟链一个轮次内区块数量为 $N_{\mathrm{BNI},i}$, 则压缩后加密数据传输大小总量 S_{c} 约为

$$S_{\mathrm{c}}=\sum_{i=1}^{N_{\mathrm{Num}}}N_{\mathrm{BNI},i}\times S_{\mathrm{b}}\approx 8.92\ \mathrm{MB} \tag{8}$$

在叶子节点数为 100 时, M_{Tree} 大小为 $S_{M_{\mathrm{Tree}}}=\mathrm{Total}(m)\times S_{\mathrm{h}}=(m^0+\cdots+m^{\log_m n})S_{\mathrm{h}}=7.10$ KB, 元数据的存储量为 $S_{\mathrm{mdata}}=n\times S_{\mathrm{h}}=3.13$ KB。

方案 1 中新增节点全量存储区块链数据, 故随节点数量的增加区块链存储总量增多。方案 3 没有增加原有链节点数量, 但所有节点需要存储其他链压缩后的数据, 在增加同等有效节点个数的情况下其存储总量低于方案 1。本方案上链数据最少, 仅为协同数据的唯一标识及其定位数据, 但该协同数据中的 ArchiveData 需要进行链下存储。如图 9 所示, 本方案的存储总量在 3 个方案中最低。

5.4　通信消耗

本节量化本方案联盟链间的通信消耗并和其他方案进行对比。本方案中主要的消耗为不同类型下的 ACP 和 ArchiveData 的传输、辅助信息如对 M 的聚合签名、哈希摘要等。其中, 辅助信息数据通信大小为 I_1=0.09 KB, MemAdd 和 MemUpdate 类型的 ACP 大小为 I_2=0.32 KB。MemTran 类型的 ACP 大小为 I_3=50.34 KB。MemTran 类型的 ArchiveData 数据仅传输加密区块链数据和验证数据,即 I_4=117.42 KB 和 I_5=0.35 KB。DataUse 类型是 MemTran

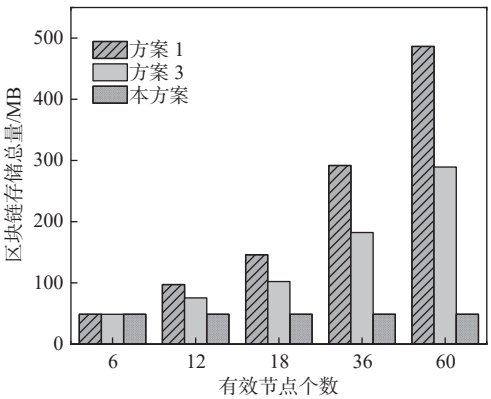

图 9 区块链一个轮次的存储总量

Fig. 9 The total storage capacity of the blockchain in single interval

类型的逆过程，数据返回的过程和 MemTran 通信消耗相同，即 I_3 或 I_3+I_4，数据索取时额外的索引通信消耗为 I_6=0.07 KB。MemQuit 类型消息在 Content 中仅有 EC 消息，但需要发送方和其他联盟链进行交互，此处次数为 3，通信量为 I_7=0.34 KB。

通信消耗和节点数量、共识信息大小、共识信息数量呈正比。一般来说，每个区块内的每笔交易都需要进行全网广播，P2P 网络通信复杂度随着节点数量的增加而上升。方案 1 中增加的节点需要参与每一笔交易的共识，在 100 个区块，每个区块 124 笔交易下共识信息数量显著增多，其额外产生的通信消耗远高于其他方案，故本文仅与方案 3 进行对比。方案 3 不增加原有链节点数量，聚合压缩多笔交易后在其他链上发起交易并进行存储，通信消耗主要表现为原有链和存储方所有节点对交易的共识过程。实验结果如图 10 所示，可以看出，本方案增加的通信消耗在 3 个方案中最少。

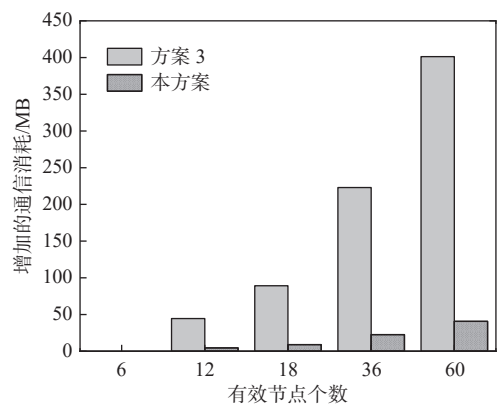

图 10 一个轮次内增加的通信消耗

Fig. 10 Increased communication consumption in single interval

5.5 计算消耗

传输数据的构造为本方案中最耗时的操作，主要分为链下协同数据的加密和数据完整性验证操作中辅助数据的构造。为了保障安全性，本文采用 RSA 算法进行加密，并通过多线程异步加密的方式将加密时间缩短至 327 ms。从单节点和整个区块链两个维度对比不同方案所增加的计算消耗。对单节点来说，方案 1 不增加耗时；方案 3 中，聚合节点压缩数据需耗时 0.2 s 左右，其他节点验证该聚合交易需额外耗时 0.008 s 左右；本方案中，信使节点进行同盟链各项事务需耗时 1.4 s 左右的时间，其他节点签名和验证需花费 0.03 s 左右，信使节点操作和原有链事务异步进行，不停顿原有链。

方案 1 中所有新增节点都要参与交易和区块的共识，在本文所设参数下会产生非常大的计算消耗。方案 3 的计算消耗来源于数据的压缩和共识；本方案的计算消耗主要为传输数据的构造和验证。一个轮次内不同方案给整个区块链带来的计算时间消耗总量如表 5 所示。可以看出，方案 3 的计算消耗总量最少，本方案的计算消耗总量略高于方案 3，但远低于方案 1。

表 5 计算开销额外总耗时

Table 5 Additional total elapsed time spent on computing overhead s

有效节点个数	方案1额外总耗时	方案3额外总耗时	本方案额外总耗时
6	0	0	0
12	117.541 1	0.246 6	1.531 6
18	209.253 1	0.254 4	1.566 0
36	486.230 0	0.277 8	2.835 7
60	855.912 7	0.309 0	3.857 1

6 结 论

联盟链是我国区块链落地的主要形态，但存在有效节点数少、节点独立性低以及可持续发展能力弱的问题，导致联盟链的去中心化程度低、可靠性不足，在运行发展过程中可能出现历史数据被合谋篡改、数据丢失后难以恢复的风险和隐患。对于这些问题，本文贡献如下：

1）提出联盟链之间相互存储其他结盟区块链数据的唯一标识形成数据之间的层叠关系，并协同存储加密区块链数据保证账本的可恢复性，增强系统的可靠性和去中心化程度。

2）设计通信机制使得不同联盟链之间有统一的传输和验证方式，保障不同区块链之间的通信，对方案运行之中会出现的消息类型给出相应构造和验证方案。

3）分析同盟链的安全性问题，证明同盟链能够抵御中间人的攻击，降低区块链对历史数据合谋攻

击的概率, 同时通过完整性验证、密码累加器和延迟确认解决方法中存在的其他问题。

4) 通过分析比对其他方案得出, 同盟链能够提高原有链去中心化程度、有效节点数量、节点间独立性、账本篡改难度, 同时对交易性能影响程度低, 适用范围比较广, 在存储链下协同数据时能够反向恢复数据对应的区块链。

5) 通过实验对比证明本方案在区块链存储总量和通信开销上有优势, 在计算消耗上略有劣势, 但同盟链模型不会阻塞原有链业务, 也不会停顿原有链。

本文所提方法仍有不足之处: ①需要多数节点诚实且可信为前提; ②对信使节点要求较高; ③不同链类型需要开发不同的交易解析工具。同时, 本文仅对同构区块链进行了实验, 所提方法仅对单个同盟链模型进行了分析, 对于多个同盟链模型交叠情况没有过多考虑。下一步将针对以上问题继续开展研究。

参考文献 (References)

[1] ZHENG Z B, XIE S A, DAI H N, et al. An overview of blockchain technology: architecture, consensus, and future trends[C]// 2017 IEEE International Congress on Big Data (BigData Congress). Piscataway: IEEE Press, 2017: 557-564.

[2] CASTRO M, LISKOV B. Practical Byzantine fault tolerance and proactive recovery[J]. ACM Transactions on Computer Systems, 2002, 20(4): 398-461.

[3] ZHU X Y, BADR Y. Identity management systems for the Internet of Things: a survey towards blockchain solutions[J]. Sensors, 2018, 18(12): 4215.

[4] 中国电子信息产业发展研究院. 2021 年中国区块链年度发展白皮书[R/OL]. (2022-06-14)[2022-07-24]. https://dsj.guizhou.gov.cn/xwzx/gnyw/202206/t20220614_74881576.html.

[5] 武腾, 薛磊, 郑东, 等. P2P 持久存储系统可靠性分析与数据维护优化[J]. 信息安全与通信保密, 2009, 7(8): 149-153.
WU T, XUE L, ZHENG D, et al. Reliability analysis and data maintenance optimization of P2P durable storage system[J]. Information Security and Communications Privacy, 2009, 7(8): 149-153 (in Chinese).

[6] GUO H Q, YU X J. A survey on blockchain technology and its security[J]. Blockchain: Research and Applications, 2022, 3(2): 100067.

[7] BANDARA H D, XU X W, WEBER I. Patterns for blockchain data migration[C]// Proceedings of the European Conference on Pattern Languages of Programs 2020. New York: ACM, 2020: 1-19.

[8] LEI K, ZHANG Q C, XU L M, et al. Reputation-based Byzantine fault-tolerance for consortium blockchain[C]// 2018 IEEE 24th International Conference on Parallel and Distributed Systems (ICPADS). Piscataway: IEEE Press, 2018: 604-611.

[9] YAN C Y, ZHANG C, LU Z G, et al. Blockchain abnormal behavior awareness methods: a survey[J]. Cybersecurity, 2022, 5(1): 5.

[10] ARÁOZ M. Proof of existence[R/OL]. [2022-07-24]. http://docs.proofofexistence.com.

[11] 谭海波, 周桐, 赵赫, 等. 基于区块链的档案数据保护与共享方法[J]. 软件学报, 2019, 30(9): 2620-2635.
TAN H B, ZHOU T, ZHAO H, et al. Archival data protection and sharing method based on blockchain[J]. Journal of Software, 2019, 30(9): 2620-2635 (in Chinese).

[12] HOPE-BAILIE A, THOMAS S. Interledger: creating a standard for payments[C]//Proceedings of the 25th International Conference Companion on World Wide Web-WWW '16 Companion. New York: ACM, 2016: 281-282.

[13] 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.
YE S J, WANG X Y, XU C C, et al. BitXHub: side-relay chain based heterogeneous blockchain interoperable platform[J]. Computer Science, 2020, 47(6): 294-302 (in Chinese).

[14] FLOERSCH K. Ethereum smart contracts in L2: optimistic rollup[R/OL]. (2019-08-28)[2022-11-30]. https://medium.com/plasma-group/ethereum-smart-contracts-in-l2-optimistic-rollup-2c1cef2ec537.

[15] BUTERIN V. Chain interoperability[R/OL]. R3 research paper, 2016 (2016-09-09)[2022-07-24]. https://allquantor.at/blockchainbib/pdf/vitalik2016chain.pdf.

[16] VIGIL M, BUCHMANN J, CABARCAS D, et al. Integrity, authenticity, non-repudiation, and proof of existence for long-term archiving: a survey[J]. Computers & Security, 2015, 50: 16-32.

[17] LAKHWANI K, KAUR R, KUMAR P, et al. An extensive survey on data authentication schemes in cloud computing[C]// 2018 4th International Conference on Computing Sciences (ICCS). Piscataway: IEEE Press, 2018: 59-66.

[18] MERKLE R C. A certified digital signature[M]// Advances in Cryptology — CRYPTO' 89 Proceedings. New York: Springer New York, 2007: 218-238.

[19] BENALOH J, DE MARE M. One-way accumulators: a decentralized alternative to digital signatures[M]// Advances in cryptology — EUROCRYPT '93. Berlin: Springer Berlin Heidelberg, 1994: 274-285.

[20] ATENIESE G, DI PIETRO R, MANCINI L V, et al. Scalable and efficient provable data possession[C]// Proceedings of the 4th International Conference on Security and Privacy in Communication Netowrks. New York: ACM, 2008: 1-10.

[21] Zlib home site[EB/OL]. (1995-05-01)[2022-03-28]. http://www.zlib.net.

[22] 祁健. 累加器在区块链中的应用研究[D]. 南京: 南京信息工程大学, 2020: 22-33.
QI J. Research on the application of accumulator in blockchain[D]. Nanjing: Nanjing University of Information Science & Technology, 2020: 22-33 (in Chinese).

A consortium chain improvement model based on multi-chain collaboration

ZHAO Xianxian[1, 2], TAN Haibo[1, 2], ZHAO He[1, *], ZHOU Tong[3], CHENG Haotian[1, 2], LI Jinze[1, 2]

(1. Hefei Institutes of Physical Science, Chinese Academy of Sciences, Hefei 230031, China;
2. University of Science & Technology of China, Hefei 230026, China;
3. Anhui ZhongKeJingGe Technology Co., Ltd., Hefei 230088, China)

Abstract: In recent years, blockchain technology has been developing rapidly, and It plays a critical role in fintech, supply chain, medical health, data security and other fields. Due to friendly supervision and excellent performance, the consortium chain has become China's first choice for government and commercial blockchain. However, it is usually faced with problems such as the small number of effective nodes, low node independence, and weak sustainable development ability, which reduces the system's reliability. We suggest the alliance model, a blockchain approach that coordinates several consortium chains to increase system stability, in light of the aforementioned issues. The model stores data by overlapping each other between chains to reduce the possibility of data tampering and ensure data recovery. In addition, we use the methods of aggregated signature and data integrity verification to solve the problems of transmission data integrity and source authenticity. In order to verify the legitimacy of reciprocal endorsement of collaborative data and to check the continuation of the sent data, we also develop an existence verification method. To confirm its efficacy, this scheme's communication, storage, and computational usage are compared to those of the current techniques.

Keywords: blockchain; consortium chain; reliability; decentralization; data integrity; data authenticity

Received: 2022-09-29; **Accepted**: 2022-11-25; **Published Online**: 2023-03-09 16:07

URL: link.cnki.net/urlid/11.2625.V.20230309.1406.007

Foundation item: National Key Research and Development Program of China (2021YFB2700800)

* **Corresponding author**. E-mail: zhaoh@hfcas.ac.cn