软件学报 ISSN 1000-9825, CODEN RUXUEW
[doi: 10.13328/j.cnki.jos.007699] [CSTR: 32375.14.jos.007699]

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel: +86-10-62562563

跨链事务技术综述*

黄洁华[1], 胡 凯[1], 夏鹏凯[1], 李 洁[1], 李 江[1], 杨鹏武[1], 董林霖[2]

[1](北京航空航天大学 计算机学院, 北京 100191)

[2](云南电网有限责任公司 计量中心, 云南 昆明 650200)

通信作者: 李洁, E-mail: lijie6@buaa.edu.cn

摘 要: 区块链系统依靠共识机制确保节点间数据一致性, 但也因此限定了单一链内数据的可信边界, 导致不同区块链系统形成天然的“数据孤岛”. 然而, 实际应用中区块链间的数据常存在关联性, 独立链上操作的不同步极易破坏这种关联, 威胁链间数据一致性. 跨链技术是打破孤岛、实现区块链互操作、提升系统可扩展性的关键技术. 其中, 跨链事务作为维护跨链系统全局一致性的核心机制, 为构建可扩展的互操作生态提供了基础保障. 现有跨链技术综述大多讨论跨链原子性, 较少全面探讨跨链事务技术. 通过系统调研相关文献: 首先, 厘清跨链系统一致性约束内涵并明确跨链事务定义; 其次, 剖析跨链事务面临的关键技术问题及其研究现状, 并进行多维度比较分析; 最后, 从关键技术角度评述代表性整体解决方案, 并指出若干值得深入探索的未来研究方向.

关键词: 跨链事务; 跨链; 跨链系统一致性约束; 区块链

中图法分类号: TP311

中文引用格式: 黄洁华, 胡凯, 夏鹏凯, 李洁, 李江, 杨鹏武, 董林霖. 跨链事务技术综述. 软件学报. http://www.jos.org.cn/1000-9825/7699.htm

英文引用格式: Huang JH, Hu K, Xia PK, Li J, Li J, Yang PW, Dong LL. Survey on Cross-chain Transaction Technologies. Ruan Jian Xue Bao/Journal of Software (in Chinese). http://www.jos.org.cn/1000-9825/7699.htm

Survey on Cross-chain Transaction Technologies

HUANG Jie-Hua[1], HU Kai[1], XIA Peng-Kai[1], LI Jie[1], LI Jiang[1], YANG Peng-Wu[1], DONG Lin-Lin[2]

[1](School of Computer Science and Engineering, Beihang University, Beijing 100191, China)

[2](Metering Center, Yunnan Power Grid Co. Ltd., Kunming 650200, China)

Abstract: Blockchain systems rely on consensus mechanisms to ensure data consistency among nodes. However, this also confines the trust boundary of data to a single chain, resulting in inherent “data silos” across distinct blockchain systems. In practical applications, data across different blockchains often exhibit correlations. Unsynchronized operations on independent chains can easily disrupt these correlations, threatening inter-chain data consistency. Cross-chain technology is a key solution for breaking down these silos, enabling blockchain interoperability, and enhancing system scalability. Cross-chain transactions, which serve as the core mechanism for maintaining global consistency in cross-chain systems, provide the fundamental underpinning for building a scalable, interoperable ecosystem. Existing surveys on cross-chain technologies predominantly focus on cross-chain atomicity, while providing less comprehensive discussions of cross-chain transaction technologies. Through a systematic review of the relevant literature, this study (1) clarifies the concept of cross-chain system consistency constraints and defines cross-chain transactions; (2) analyzes the key technical challenges faced by cross-chain transactions, reviews the current state of research, and conducts a multidimensional comparative analysis; (3) evaluates representative holistic solutions from the perspective of key technologies and identifies several future research directions worthy of in-depth exploration.

Key words: cross-chain transaction; cross-chain; cross-chain system consistency constraint; blockchain

---

* 基金项目: 国家重点研发计划 (2022YFB2703500); 云南省重大科技专项 (202302AF080006)
收稿时间: 2025-08-25; 修改时间: 2025-11-25, 2026-04-09; 采用时间: 2026-05-15; jos 在线出版时间: 2026-08-19

区块链被看作是构建未来价值互联网的重要支撑技术, 它解决的是一组不信任多方之间实现一致的状态转移问题, 是在不可信多方间实现可信计算的应用范式[1,2]. 随着区块链的发展, 链间资产交换、资产转移和数据共享等应用场景不断涌现[3,4]. 由于不同区块链之间的数据存在内在关联, 链间操作的不同步极易破坏这种关联, 链间数据一致性问题愈发突出. 尽管单个区块链内部的数据一致性可通过共识算法得到保障, 但链间的数据一致性需要额外的机制来维护. 跨链事务技术的目的是保证跨链系统状态满足一致性约束, 为构建高效、稳定的区块链互操作生态奠定了基础.

在跨链事务的研究与实践中, 核心问题聚焦于 3 个方面: 跨链事务执行与数据的可信性、分布式原子性提交、事务隔离. 首先, 跨链事务执行与数据的可信是指保证跨链事务的执行控制和数据与事先定义的跨链操作一致; 其次, 分布式原子性提交是消除跨链操作分步执行风险的关键, 使得跨链事务执行前与执行后的系统状态满足一致性约束; 最后, 合理的事务隔离机制对系统稳定性至关重要, 通过控制事务执行过程中系统状态的可见性和可修改性, 降低并行事务的中止率. 因此, 跨链事务的设计与实现需在可信、原子性与事务隔离之间寻求平衡, 以确保多链协作系统的高效运行. 需要说明的是, 本文中“跨链操作”与“跨链事务”有严格区分, 二者的定义将在后文给出.

目前已有多项综述工作针对跨链研究进行了梳理, 这些工作从不同角度对跨链技术进行了分类和总结, 如表 1, Ren 等人[4]将现有的解决方案分为 5 种: 侧链、公证方案、哈希时间锁定合约 (hashed time-locked contract, HTLC)、中继和区块链无关协议; Robinson[5]根据不同的应用场景, 将跨链协议分为 3 类: 原子交换、跨链消息传递以及区块链状态锁定; Belchior 等人[6]将现有解决方案分为公有连接器 (public connector)、链中链 (blockchain of blockchain) 和混合连接器 (hybrid connector). 然而, 大多数综述对跨链事务的讨论并不全面. 单个区块链系统内对资源的操作是原子的, 但在跨链系统中很难以原子的方式实现跨链操作, 原子性是跨链事务的核心属性之一, 受到广泛的研究. 现有的综述较少深入探讨跨链事务技术的其他关键方面, 例如事务隔离机制、性能优化策略和安全性保障措施[7], 尚未形成系统性的分析框架. 因此, 需对跨链事务技术进行全面而深入的研究, 以填补这一领域的空白, 并为未来的技术发展提供指导.

表 1　综述文章对比

文献	跨链事务				方法					内容			讨论	
	可信性	原子性	隔离性	安全性	协议抽象/建模	性质定义/需求	分类	比较框架	实验	关键技术	整体方案	应用场景分类	机遇	挑战
[1]	√	√	—	—	—	—	√	√	—	√	√	—	√	√
[2]	√	√	—	—	√	√	√	√	—	√	√	—	√	√
[3]	—	—	—	√	—	√	√	√	—	√	—	√	√	√
[4]	√	√	—	√	—	√	√	√	√	√	√	√	√	√
[6]	—	—	—	—	—	√	√	√	—	√	—	√	√	√
[7]	—	√	—	—	√	√	√	√	—	√	√	—	√	√
[8]	—	√	√	—	√	√	√	√	—	√	√	—	√	√
[9]	—	—	—	—	√	√	√	√	—	√	√	√	√	√
[10]	—	—	—	—	—	—	√	—	—	√	—	√	√	√
[11]	—	√	—	—	—	—	√	√	—	√	—	—	√	—
[12]	—	—	—	—	—	—	—	—	—	√	—	—	—	√
[13]	—	—	—	—	—	—	√	√	—	√	—	√	√	√
本文	√	√	√	√	√	√	√	√	√	√	√	√	√	√

为系统地探讨跨链事务技术的核心挑战与发展方向, 本文聚焦于以下 3 个关键研究问题 (research question, RQ).

RQ1 (定义): 如何界定跨链系统一致性约束以及跨链事务的概念?

RQ2 (可信性与事务保障): 如何确保跨链事务的执行可信度 (trustworthiness)? 以及如何保障跨链事务的原子性 (atomicity)、隔离性 (isolation) 与安全性 (safety)?

RQ3 (未来方向): 现有研究的局限性何在? 未来有什么值得深入的研究方向?

为了解答这些问题, 首先筛选出正式文献 61 篇、灰色文献 35 篇, 具体检索策略与筛选流程请查看附录 A. 通过系统的调研, 总结并定义跨链系统一致性约束和跨链事务概念, 然后, 针对跨链事务每个关键技术问题总结当前的技术路径, 并提出分类框架, 提炼不同技术路径基本属性. 接下来, 将分类框架用于分析典型的整体方案, 以便读者更好地理解并评价现有研究工作.

总的来说, 本文主要贡献如下.

1) 给出跨链系统一致性约束的定义, 跨链事务技术的目标是保证跨链系统状态满足一致性约束.

2) 总结当前跨链事务关键技术, 包括跨链可信执行技术与跨链事务并发执行技术.

3) 讨论当前跨链事务技术的安全性, 总结攻击方式、防御手段和典型案例.

4) 指出未来研究方向. 研究有助于完善跨链事务理论体系, 为构建安全可靠的跨链系统提供理论指导.

本文第 1 节介绍区块链中与跨链相关的背景知识, 包括共识证明和状态证明. 第 2 节对跨链事务问题进行分析, 包括跨链操作、跨链系统一致性约束和跨链事务的定义, 方便后续跨链技术的讨论, 以及总结跨链事务关键技术问题. 第 3 节和第 4 节分别讨论跨链事务可信执行关键技术和跨链事务并发执行关键技术. 第 5 节梳理跨链事务的攻击手段、防御策略及典型案例. 第 6 节对代表性的跨链整体解决方案从关键技术问题的角度进行分析和讨论. 最后, 第 7 节指出跨链研究领域尚待解决的问题与挑战.

1 相关背景知识

在开始讨论前, 先介绍区块链的相关背景知识, 方便理解跨链事务相关内容, 包括区块链基本概念以及链相关的验证机制-共识证明和状态证明.

区块链技术是一种按照时间顺序将数据区块以顺序相连的方式组合成链式数据结构, 并以密码学方式保证不可篡改和不可伪造的分布式账本技术[14]. 它解决的是一组不信任多方之间实现一致的状态转移问题, 是在不可信多方间实现可信计算的应用范式.

区块链的全节点都在本地存有完整账本数据, 账本由按序排列的交易组成, 每笔交易是一个状态转移函数, 账本整体即为所有交易依序复合而成的状态转移函数. 假设区块链的初始状态为空, 各个节点上的状态转移函数一样, 状态转移结束后节点存储的区块链状态一致.

作恶节点可能会修改本地的账本, 状态转移函数不一致导致作恶节点的区块链状态与诚实节点不一样. 为了在一组不信任多方之间实现一致的状态转移, 需要共识算法保证不信任多方之间的账本保持一致, 从而达到一致的状态转移. 逐笔交易执行共识算法, 时间成本较大, 将批处理的思想应用到这里, 多个交易按序组合成区块, 每个区块按序组合成账本, 每个区块都通过共识算法保证一致, 从而保证账本一致. 为了方便后面的讨论, 下面给出一些符号的定义.

区块链是以链式结构组织区块的分布式账本 L, 其中,

1) 交易为 Tx.

2) 区块为 $Block$.

3) $L[r]$ 表示第 r 个区块, $L.length \in \mathbb{N}$ 表示链的长度.

4) 交易序列: 第 r 个区块中的交易可表示为数组 $L[r].transactions = \langle Tx_1, Tx_2, \ldots, Tx_n \rangle$.

5) 交易提交: $IN(Tx, L[r]) = \text{true}$ 表示交易 Tx 被写入区块 $L[r]$.

6) 交易有效: $VALID(Tx, L[r]) = \text{true}$ 表示交易 Tx 被写入区块 $L[r]$, 且执行成功.

7) 区块链状态 S 是多副本状态机的全局快照: $S[r]$ 表示链高度为 r 时的状态; 状态变更服从 $S[r] \xrightarrow{Tx} S[r+1]$.

1.1 共识证明

区块链作为分布式账本系统, 每个节点都存储一样的账本, 那么如何证明区块/交易是经过共识有效的? 共识证明就是证明区块链中区块/交易有效的方法[15]. 共识证明与共识机制相关, 下面介绍各个共识机制对应的证明方法.

比特币工作量证明 (proof of work, PoW) 中, 其轻节点[16]仅需获取目标交易所在区块头及对应默克尔树 (Merkle tree)[17]路径, 即可独立验证区块、交易有效性. 区块头中包含的链式哈希结构 (前块哈希值) 构成了交易顺序的时序锁, 配合工作量证明的难度累积效应, 确保交易排序在 6 个区块确认深度后具备统计学不可逆性. 虽然简单支付证明 (simplified payment verification, SPV) 大幅降低客户端资源需求 (存储量减少 99.9%), 但其安全性依赖全网诚实节点算力占比假设.

相比之下, 在权益共识 (proof of stake, PoS) 中, 由于验证者已经在网络中投资了大量资本, 如果他们支持两个互相矛盾的状态转换分支, 或者以无效的状态转换的方式返回了一个块, 则导致自己资本削减, 从而保证大资本节点不会作恶. PoS 的 SPV 节点只需拉取最新块头的签名, 包含超过 2/3 节点质押权重的签名, 则为有效区块. 以以太坊 Casper FFG (the friendly finality gadget)[18]为例, 其采用两阶段投票实现最终确定性: 1) 准备阶段: 验证者对候选区块进行“准备”投票, 需收集超过 2/3 质押权益的 BLS 聚合签名; 2) 提交阶段: 后续对同一区块进行“提交”投票, 触发最终性确认. 验证者双重签名行为会受到惩罚, 这种设计使得共识效率从比特币的“概率确认”提升至“强最终性”.

基于拜占庭容错 (Byzantine fault tolerance, BFT) 的共识算法是许可型联盟链实现强一致性与抵抗恶意节点能力的重要技术路线之一. BFT 允许分布式系统在部分节点发生拜占庭故障时, 仍能对交易顺序达成不可篡改的共识. 在典型 BFT 设定下, 若共识节点规模为 $3f+1$, 系统最多可容忍 f 个拜占庭故障节点. 以 Hyperledger Fabric v3.0[19] 引入的 SmartBFT 排序服务为例: 系统基于成员服务提供者发放的数字证书识别节点身份. 在共识流转中, 节点通过收集并聚合达到法定人数的 commit 消息与门限签名, 构建区块的共识证明. 最终, 该证明需配合通道内的区块验证策略进行校验, 确保区块在追加至账本前已通过合规的共识过程与身份准入.

为突破单一共识机制的性能瓶颈, 学界与产业界正积极探索跨范式融合的创新路径: 1) PoW+PoS 混合链: Decred[20]采用工作量证明出块+权益证明投票的机制; 2) 零知识证明共识: Mina 协议[21]利用 zk-SNARKs 将链历史有效性压缩为简洁证明, 验证者无需存储历史数据, 只需验证这个证明即可确认链的有效性.

Polkadot[22]提供一个通用的共识证明系统, 能够同时维护多个链的共识证明, 其核心由区块生产层与最终性确认层组成: 区块链扩展的盲分配机制 (blind assignment for blockchain extension, BABE) 基于可验证随机函数 (verifiable random function, VRF) 动态分配 Slot Leader 角色实现概率性出块, 同时 GRANDPA 协议 (GHOST-based recursive ancestor deriving prefix agreement) 通过权益加权的拜占庭容错算法在异步网络环境下完成最终性确认, 以“批处理方式”一次性最终确认整条链上的多个区块, 而不是逐块确认.

不同共识的证明方案比较如表 2.

表 2　共识证明方案比较

共识证明方案	证明尺寸	验证耗时	最终性确认延迟
PoW (区块头链+Merkle路径)	大	短	长
DPoS (代表节点签名证书)	小	短	短
BFT (法定人数签名证书)	中等	中等	中等
Mina (递归零知识证明)	小	短	短
Polkadot (GRANDPA 最终性证明)	小	短	短

证明尺寸: 指生成和验证证明所需的数据量. PoW 共识证明通常由连续区块头及交易的 Merkle 路径组成, 其尺寸随确认深度增加; DPoS 和 BFT 共识证明通常由达到法定权重的节点签名构成, 其尺寸取决于验证者数量及签名聚合方式; Mina 使用递归零知识证明, 将链历史有效性压缩为近似恒定大小的证明; Polkadot 的 GRANDPA 最终性证明由达到法定权重的验证者签名及相关区块信息构成.

验证耗时: 指验证一个证明所需的时间. PoW 的验证本身耗时极短, 但其证明生成耗时极长; DPoS (delegated proof of stake) 仅需验证少数代表节点签名, 耗时短; 零知识证明验证耗时短, 但证明生成耗时较长.

最终性确认延迟: 指从交易被提交到被确认为不可篡改的时间. PoW 通常有较长的最终性确认延迟, 因为它

依赖于后续区块的累积来增加安全性. DPoS 和 Polkadot 等则设计有较快的最终性确认机制.

1.2 状态证明

共识证明验证交易记录的全局一致性和不可篡改性. 区块链节点有必要重新实施自创世纪块以来的所有交易, 在全局一致的交易记录基础上, 通过实施所有交易, 每个节点都能产生区块链的一样的状态. 这种机制是使区块链被视为所有参与者都可以信任的防篡改数据库的原因.

但如果为了计算区块链状态需要所有的交易都重新实施一遍, 这样成本太高, 如何在共识证明基础上快速验证区块链证明是重点问题.

如表 3, 有以下状态证明[23]方法: 1) Merkle 包含性证明, 例如比特币使用 Merkle 树承诺区块内交易, 验证者可通过 Merkle 路径验证某笔交易是否包含在相应区块中; 2) Merkle-Patricia trie (MPT)[24], 以太坊使用改进的 MPT 组织区块链数据. 区块头分别包含状态树根、交易树根和收据树根; 其中, 状态树记录账户的余额、nonce、代码哈希和存储树根, 每个合约账户还拥有独立的存储树. 轻节点可依据经过共识认证的区块头, 通过验证 MPT 路径证明来确认账户状态、合约存储、交易或收据的包含性; 3) 多项式承诺, 基于 KZG (Kate-Zaverucha-Goldberg) 承诺构造的 Verkle 树方案[25], 将一个节点的多个子节点值插值编码为低阶多项式, 并以多项式承诺替代 Merkle 树中的哈希承诺. KZG 的单个打开证明具有常数大小, 并支持批量打开, 从而降低多状态查询的证明开销; 4) 零知识证明, Zcash[26]的 Sapling 协议使用 Groth16 算法证明隐私交易满足余额守恒、授权及相关状态转换约束, 而不公开被隐藏的交易信息, 验证者通过验证证明及其公开输入确认交易有效性.

表 3 状态证明方案比较

状态证明方案	证明尺寸	验证耗时	可信假设	适用场景
全量重放执行	大	长	无需特殊假设	数据完整性验证
MPT	中等	中等	哈希函数安全性	区块链、数据完整性验证
KZG 承诺	小	短	需要通用可信设置	区块链状态证明、Rollup数据验证
zk-SNARKs (Groth16)	小	短 (验证短, 证明生成长)	需要电路专用可信设置	区块链、隐私保护

证明尺寸: 指验证所需提交的数据量. 通常, 全量重放执行需包含完整历史交易, 尺寸最大, KZG 承诺与 zk-SNARKs 均可生成常数大小的证明, 尺寸最小.

验证耗时: 指验证一个证明所需的时间. 全量重放执行通常耗时最长, 而 MPT 仅需验证路径哈希, 耗时较短; zk-SNARKs 验证耗时短.

可信假设: 指证明方案所依赖的密码学假设. MPT 仅依赖哈希函数的安全性, 无需额外假设; KZG 承诺需要通用可信设置; Groth16 等 zk-SNARKs 需要电路专用可信设置.

适用场景: 指证明方案适用的具体场景, 如区块链、数据完整性验证等.

2 跨链事务问题分析

现有文献对跨链提出了多种定义, 根据参与区块链间的互动模式, 可以归纳为两类, 1) 调用关系, 指一条区块链以类似远程过程调用 (remote procedure call, RPC) 的方式调用另一条链上的功能; 2) 协作关系, 指一个完整的业务流程需跨越并协调多个区块链上的操作方能完成既定目标. 多位学者用区块链互操作替代跨链. Ren 等人[4]认为“区块链互操作性是指不同区块链 (无论是公有链、私有链还是联盟链) 之间无需修改底层系统即可灵活转移资产、共享数据或调用智能合约的能力”. Wang 等人[7]认为“区块链的互操作性意味着多个区块链 (无论是同构还是异构) 可以安全地相互访问信息, 改变彼此状态, 且不损害原有区块链的去中心化和可信特性”. Belchior 等人[6]认为“区块链的互操作性是通过跨链操作实现的、跨越同构和异构区块链系统组合的能力, 使得一个源区块链能够改变目标区块链的状态 (反之亦然)”, 他们所界定的区块链互操作性概念, 聚焦于区块链系统间相互调用的能力. 而另一类跨链定义侧重于区块链之间的协作关系. Falazi 等人[8]定义跨链智能合约调用 CCSCI (cross-chain smart contract invocation) 是一种分布式交易, 涉及在两个及多个区块链系统上调用智能合约的函数, 体现的是链与链之

间的协作关系.

本文要讨论的跨链是后一种, 跨链技术是将多个区块链操作组合执行的技术.

定义 1. 跨链操作. 跨链操作 CO 是由区块链操作集合 O 和操作依赖关系 E 组成的二元组, 记为 $CO=(O,E)$.

1) $O=\{op_1,op_2,\ldots,op_n\}$ 表示是一个 n 个区块链操作集合, op_i 是区块链操作.

2) $E\subseteq O\times O$ 表示操作间的执行依赖关系, 构成一个有向无环图 (directed acyclic graph, DAG). 若 $\left(op_i,op_j\right)\in E$, 则表示 op_j 的执行依赖于 op_i 的输出或状态确认.

在 CO 中, 根据数据流向, 若 op_j 依赖于 op_i, 则 op_i 所在链称为上游链 (upstream chain), op_j 所在链称为下游链 (downstream chain).

跨链操作的执行路径 P 定义为 DAG 中的一条从起始操作到结束操作的有序序列. 执行路径 P 长度等于跨链开始操作到跨链结束操作执行路径上产生的区块链交易数. 跨链操作执行完整性判定.

1) 如果路径长度为 1, 则跨链操作执行是完整的.

2) 如果路径长度大于 1, 如果路径最后的区块链操作是结束操作且有效, 则跨链操作是完整的.

当路径长度为 1, 说明只有一个区块链操作, 该操作失败说明什么都没发生就结束了, 可以认为该跨链操作是完整的.

跨链操作有 3 个终态, 并根据完整性与所有区块链交易结果映射到 3 种终态, 包括成功、失败和中止.

1) 成功: 完整的跨链操作, 所有子操作都是有效状态 (如资产转移完成).

2) 失败: 完整的跨链操作, 但不是所有子操作都是有效状态 (如余额不足触发回滚).

3) 中止: 不完整的跨链操作因层级失效或超时无法推进 (如某个区块链网络宕机).

为了更直观地理解上述定义, 以典型的跨链资产交换 (swap) 场景为例构建跨链操作模型. 如图 1 所示, 用户 A 在链 1 上给用户 B 转 x 个 token, 用户 B 在链 2 上给用户 A 转 ax 个 token (a 是兑换系数). 该跨链操作可抽象为一个有向无环图 (DAG).

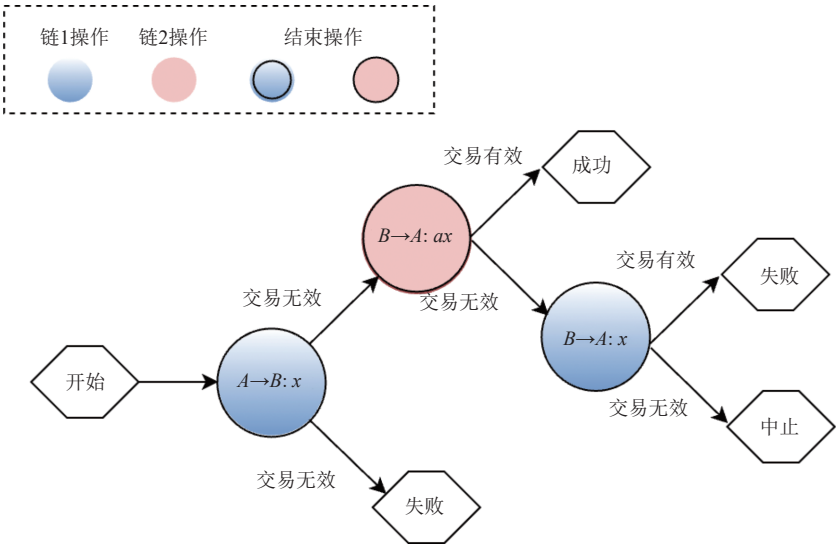

图 1 跨链交换场景的有向无环图 (DAG) 与状态分支

节点 (vertex): 代表区块链上的一个区块链操作.

边 (edge): 代表操作间的因果依赖关系.

分支 (branch): 图中的分支点体现了跨链环境下的非确定性. 与传统分布式数据库不同, 区块链上的交易可能因共识失败、Gas 不足或逻辑校验不通过而无效 (invalid).

图 1 完整展示了从初始状态出发, 由各子交易的有效性驱动系统进入不同终态的所有可能逻辑路径, 是跨链

事务协议设计的基础视图. 链 1 区块链操作是蓝色圆形, 链 2 区块链操作是红色圆形, 圆形内是操作内容, $A \to B$: x 表示用户 A 给用户 B 转 x 个单位的 Token X. 带有黑色边框的圆形是结束操作, 表示如果操作交易成功则跨链操作结束. 最后六边形表示跨链操作的终态 (成功/失败/中止).

基于图 1 的逻辑结构, 图 2 进一步展开了 4 条具体的执行路径.

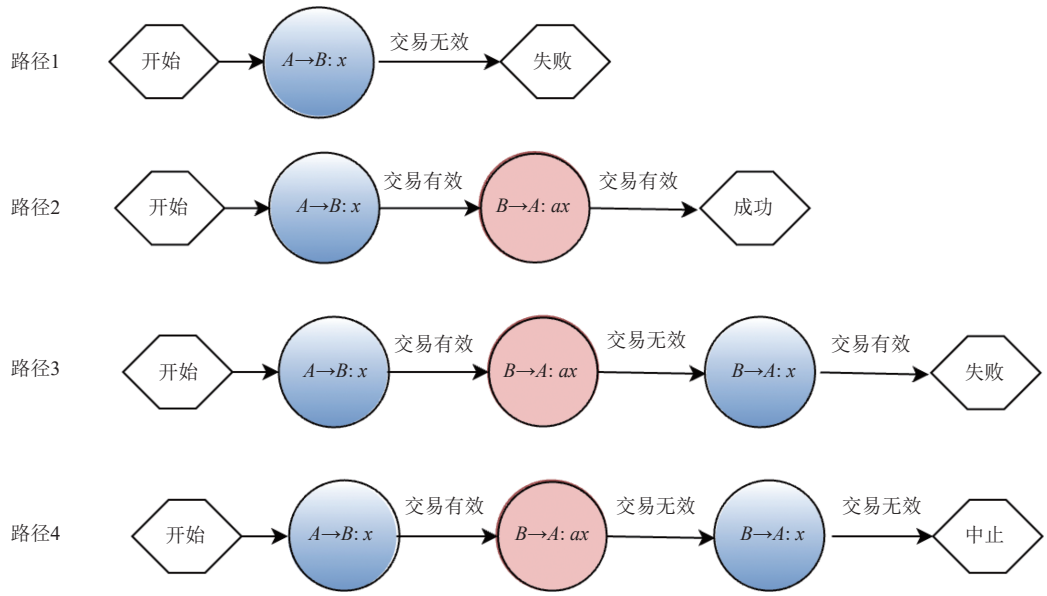

图 2　跨链操作的 4 种典型执行路径及其对一致性约束的满足情况

路径 1: 用户 A 在区块链 1 上给用户 B 转 x 个单位的 Token X, 区块链交易无效, 跨链操作失败. 因为是路径上第 1 个交易, 相当于跨链系统的状态没有被改变, 跨链操作完整但失败.

路径 2: 用户 A 在区块链 1 上给用户 B 转 x 个单位的 Token X, 交易有效; 用户 B 在区块链 2 上给 A 转 ax 个单位的 Token Y, 交易有效, 跨链操作完整成功. 路径 2 所有子交易均有效上链. 系统状态迁移至预期状态, 满足原子性与所有业务约束.

路径 3: 用户 A 在区块链 1 上给用户 B 转 x 个单位的 Token X, 交易有效; 用户 B 在区块链 2 上给 A 转 ax 个单位的 Token Y, 交易无效; 用户 B 在区块链 1 上给 A 转 x 个单位的 Token X, 交易有效; 跨链操作完整但失败. 尽管业务未成功, 路径 3 中链 2 交易失败触发了链 1 的补偿回滚, 系统通过回滚机制回到了初始状态或等价状态, 路径执行后系统状态依然满足所有业务约束.

路径 4: 用户 A 在区块链 1 上给用户 B 转 x 个单位的 Token X, 交易有效; 用户 B 在区块链 2 上给 A 转 ax 个单位的 Token Y, 交易无效; 用户 B 在区块链 1 上给 A 转 x 个单位的 Token X, 交易无效, 跨链操作中止. 这是跨链系统必须极力避免的“不一致状态”, 在此路径中, 链 1 的资产已转移, 但链 2 的交易因网络分区或共识攻击而永久失效, 且未触发补偿, 造成了单边资产损失.

从这个例子可以看出, 交易中止会导致跨链系统不一致, 跨链事务的核心目标是维护跨链系统的一致性状态, 第 3 节 (可信执行) 与第 4 节 (并发控制) 所述的各类技术方案, 本质上都是为了确保执行路径在 DAG 图中能够正确导向一致性终态的机制. 下面先给出跨链系统一致性约束的定义, 然后定义跨链事务, 有助于回答第 1 个研究问题 RQ1.

2.1　跨链系统一致性约束

在跨链系统中, 确保数据的一致性是实现系统正确性和可靠性的关键. 跨链系统一致性约束, 包含链内一致性约束和链间一致性约束. 用 (L, CC) 表示跨链系统, L 表示跨链系统中所包含的所有区块链账本集合. 每个区块链

账本独立运行, 拥有自己的数据结构和共识机制. CC 是跨链一致性约束集合, 定义了跨链操作必须满足的条件和规则, 确保不同区块链之间的交互和数据一致性.

定义 2. 跨链系统状态. S 表示区块链状态, $S[r]$ 表示账本高度为 r 时的状态, 跨链状态由各个区块链状态组成, $CS = \{S_1[r_1], S_2[r_2], \ldots, S_n[r_n]\}$ 表示各个区块链在不同区块高度时的组合状态.

1) 链内一致性约束

链内一致性约束是指单个区块链在交易上链过程中所需满足的条件. 这些约束通常由区块链的共识协议保证, 如工作量证明 (PoW) 或权益证明 (PoS). 由于这些内部约束由区块链本身的机制保证, 因此在跨链中, 默认情况下无需额外处理, 只需确保跨链操作不破坏这些机制即可.

2) 链间一致性约束

用“假设-保证 (assume-guarantee, AG)”方法[27,28] 的方式来描述. 根据 AG 方法, 每个区块链间的协作需求都可用 (ϕ_A, ϕ_G) 表示 ϕ_A 和 ϕ_G 分别表示假设约束 (assumption) 和保证约束 (guarantee), 假设 CS_{pre} 和 CS_{post} 分别是跨链操作执行前和执行后的跨链系统全局状态, 跨链操作执行导致的跨链系统状态变迁满足协作需求 (ϕ_A, ϕ_G) 是指, CS_{pre} 满足约束 ϕ_A, CS_{pre} 和 CS_{post} 满足约束 ϕ_G.

定义 3. 链间一致性约束. 一个链间一致性约束 $c \in CC$ 定义 $c = (\phi_A, \phi_G)$, 其中,

1) $\phi_A(CS_{\text{pre}})$ 是对跨链操作执行前系统状态 CS_{pre} 的谓词判定.

2) $\phi_G(CS_{\text{pre}}, CS_{\text{post}})$ 是对跨链系统状态变迁关系的谓词判定.

跨链操作 CE 满足链间一致性约束 c, 当且仅当跨链操作 CE 执行前系统状态 CS_{pre}, 与执行后系统状态 CS_{post} 满足 $\phi_A(CS_{\text{pre}}) = \text{true}$, $\phi_G(CS_{\text{pre}}, CS_{\text{post}}) = \text{true}$.

跨链操作开始状态 CS_{pre} 是跨链操作开始时各区块链最大高度的组合状态. 跨链操作结束状态 CS_{post} 是跨链操作完成时各区块链最小高度的组合状态. 举个例子, 如图 3, 跨链操作开始时链 1 高度是 2, 链 n 高度是 3, 跨链操作结束时链 1 高度是 3, 链 n 高度是 5. 如果跨链操作满足条件“跨链操作开始状态$\{S_1[2], \ldots, S_n[3]\}$ 满足 ϕ_A, 跨链操作开始状态$\{S_1[2], \ldots, S_n[3]\}$ 和跨链操作结束状态 $\{S_1[3], \ldots, S_n[5]\}$ 满足 ϕ_G”, 则称该跨链操作满足跨链一致性约束.

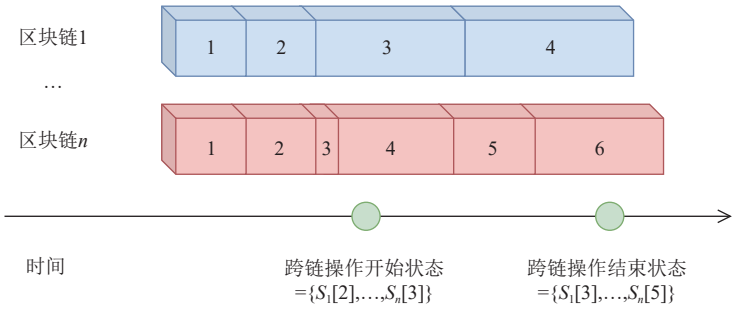

图 3 跨链操作状态变迁示意图

为了更直观地理解, 以跨链交换 (swap) 为例来说明跨链系统的一致性约束, 如后文图 4. 同时满足链内一致性约束和链间一致性约束的系统状态, 称为跨链系统一致性状态.

2.2 跨链事务定义

基于定义 1 跨链操作的基础上, 下面给出跨链事务的定义.

定义 4. 跨链事务. 一个跨链操作被称为跨链事务, 当且仅当其执行前后的系统状态满足跨链一致性约束集合 CC.

继续以跨链交换 (swap) 为例, 跨链交换可能的执行路径有多条, 每条执行路径执行前后的系统状态满足跨链系统一致性约束的跨链操作, 称为跨链事务, 列举如下.

1) 用户 A 在区块链 1 上给用户 B 转 x 个单位的 Token X, 区块链交易无效, 跨链操作失败. 因为是路径上第 1

个交易, 相当于跨链系统的状态没有被改变, 所以保持了跨链一致性约束.

2) 用户 A 在区块链 1 上给用户 B 转 x 个单位的 Token X, 交易有效; 用户 B 在区块链 2 上给 A 转 ax 个单位的 Token Y, 交易有效, 跨链操作成功. 由于满足图 4 的一致性约束, 所以保持了跨链一致性约束.

假设用户 A 在区块链1上拥有 x 个单位Token X, 用户 B 在区块链2上拥有 y 个单位的Token Y, 两人希望进行Token的交换, 兑换比例是 a, 如果用户 A 在区块链1上给用户 B 转账 x 个单位的Token X, 那么用户 B 在区块链2上给用户 A 转账 ax 个单位的Token Y.
A(假设):
用户 A 在区块链1上拥有账户; y >= ax, 用户 B 在区块链2上拥有账户.
G(保证):
满足以下任何一个条件, 都满足一致性约束:
1)区块链1上存在用户 A 给用户 B 转 x 个单位Token X的交易; 区块链2上存在用户 B 给用户 A 转 ax 个单位Token Y的交易.
2)区块链1上不存在用户 A 给用户 B 转 x 个单位Token X的交易; 区块链2上不存在用户 B 给用户 A 转 ax 个单位Token Y的交易.

图 4 跨链交换 (swap) 一致性约束例子

3) 用户 A 在区块链 1 上给用户 B 转 x 个单位的 Token X, 交易有效; 用户 B 在区块链 2 上给用户 A 转 ax 个单位的 Token Y, 交易无效; 用户 B 在区块链 1 上给用户 A 转 x 个单位的 Token X, 交易有效; 跨链操作失败. 由于区块链 1 上用户 A 给用户 B 转的 token 又被转回去了, 区块链 2 上用户 B 给用户 A 的交易失败, 所以满足图 4 的一致性约束, 所以保持了跨链一致性约束.

4) 用户 A 在区块链 1 上给用户 B 转 x 个单位的 Token X, 交易有效; 用户 B 在区块链 2 上给用户 A 转 ax 个单位的 Token Y, 交易无效; 用户 B 在区块链 1 上给用户 A 转 x 个单位的 Token X, 交易无效; 跨链操作中止. 由于只有一方转账成功, 所以不满足图 4 的一致性约束, 所以导致跨链的不一致性.

从这个例子可以看出, 操作中止会导致跨链系统不一致. 更进一步, 执行完整性和原子性的目的都是维护跨链一致性, 原子性比完整性更加严厉, 比如路径 3 的恢复操作满足完整性但违反原子性, 但都达到了维护跨链一致性的目的.

2.3 跨链事务关键技术问题

跨链事务必须被完整执行才能保证跨链系统的一致性. 实现跨链事务需要解决以下 3 个问题.

1) 跨链事务执行的可信性问题: 一个跨链操作包含多个区块链操作, 区块链操作需要被触发执行, 即在不改变区块链本身机制的情况下, 为了执行跨链事务, 必须依赖于链下跨链系统, 按照事先定义的跨链操作逻辑去触发各个区块链操作, 如何保证跨链事务执行的可信性和区块链操作输入的可信性?

2) 并发控制与性能的权衡问题: 事务并行执行有利于提高效率, 但会导致资源冲突, 由于区块链之间是独立的, 所以事务中的区块链操作的提交不是同时的, 资源冲突有可能会导致后提交的区块链操作失败, 导致事务中止, 破坏跨链一致性约束, 如何平衡效率和中止率?

3) 跨链事务安全性与防御机制问题: 跨链事务涉及多方交互, 攻击面广且复杂. 跨链事务面临哪些特有的攻击手段? 有哪些防御手段? 以及它们对系统会产生何种程度的影响?

3 跨链事务可信执行关键技术

事务执行的可信性要保证跨链的执行与协商的跨链操作一致. 区块链操作的执行结果, 可以通过区块链账本确认, 但跨链操作的执行控制需要额外的机制保证可信.

跨链事务执行包含多个区块链操作的执行, 区块链操作执行前如何验证数据的有效性, 如何确保当前跨链事务被正确执行, 是可信执行的关键问题. 根据区块链交易、数据有效性验证所在位置分为两类: 1) 链下验证可信执行机制; 2) 链上验证可信执行机制.

3.1 链下验证可信执行机制

链下验证可信执行机制是指链下负责上游链数据的获取与验证工作, 以及跨链事务执行, 链上负责验证链下工作节点身份和权限, 或者验证链下计算的完整性, 如图 5.

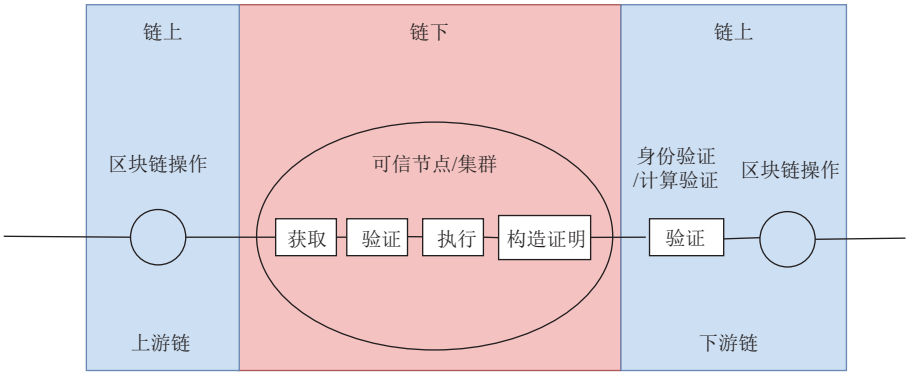

图 5 链下验证可信执行机制

根据链下工作控制逻辑特点, 分为集中式执行与分布式执行, 集中式执行指由单节点执行跨链请求, 如 XCLAIM 提出去中心化的跨链货币交换的方法, 其中 Vault 是负责抵押资产并执行跨链兑换的节点, Vault 需要先在链上做货币抵押, 如果执行过程 Vault 说谎, 则扣除 Vault 的抵押物, 从而保证可信[29]. 也可以通过 TEE 提供可信执行环境, 充当可信第三方, Bentov 等人[30]提出了基于可信计算硬件的 token 交换方案, 在可信计算硬件中验证交易有效性, 利用硬件中的私钥给下游链发送交易. 分布式执行即由多个节点通过共识算法共同验证数据有效性和执行跨链事务, 大部分分布式执行采用区块链作为中继, 称为中继链方案[31]. 同时, 也有采用多签技术保证链下验证工作的, 如 Wormhole 通过 19 个验证节点组成 Guardian 网络, 负责验证跨链消息的有效性并签名, 通过多签技术, 使得下游链能够验证消息的有效性[32]. 集中式执行设计简单且速度快, 只需要一个实体控制状态转移过程, 但面临单点故障的风险. 分布式执行减小单点故障的风险, 由于有多方共同决定, 更加可信, 但同时会增加计算时长, 降低性能.

链上负责验证链下工作节点的身份和权限, 或者验证链下计算的完整性. 存在多种节点身份与权限的验证技术, 包括多签技术、奖惩机制.

1) 基于多签技术的验证机制, 如 Wormhole[32]网络由 19 个守护节点提供多签证明.

2) 基于奖惩制度的验证机制, 验证节点提供验证结果之前会先抵押资产, 设置一个验证器, 如果没有其他验证节点提供可以反驳的证明, 则会给验证节点奖励[33,34], 否则以没收抵押资产为惩罚. 如 Optimism Bedrock[35]的欺诈证明窗口设为 7 日, 挑战成功者可获得被挑战方部分或全部抵押资产作为奖励, Chainlink CCIP[36]的反欺诈网络要求节点同样质押 LINK, 一旦检测到恶意或错误消息, 违规节点质押金将被罚没, 并用于补偿受损用户.

链上除了可以对链下验证节点的身份和权限进行验证, 还可以验证链下计算的完整性, 有不同的计算证明方案.

1) 有的链本身的共识证明是对链验证节点的签名做验证, 所以签名验证工作放到链下, 然后生成签名验签的工作证明, 如 zkBridge[37] 链下计算的内容是验证 Cosmos 验证节点的签名, 生成的证明是签名验证的工作证明. Cao 等人[31]提出一个中继架构 MAP 和 zk 轻客户端, 针对以太坊 PoS 多签共识证明的情况, 通过将签名验证工作放到链下证明从而优化链上轻客户端的计算, 从而减少跨链验证成本.

2) 有的链本身没有验证节点, 则通过跨链的验证节点集群对链做共识证明, 跨链验证节点生成验证结果以及对应的签名, 再对跨链验证节点的签名生成签名验证的工作证明, 如 Sober 等人[38]通过 zk-SNARKs 实现链下聚合. 预言机为用户查询链数据的时候, 分为一个聚合节点和多个验证节点, 聚合节点收集签名, 对签名验证做工作证明.

3) 除了签名验证的计算证明, 也可以在链下做共识证明, 并生成链下验证计算的完整性, 如 Cosmos[39]的 Tendermint 轻客户端可在链下证明后将验证结果发布到以太坊, 从而实现经济高效的轻客户端验证. Polygon zkEVM 等方案[40]则通过在链下生成有效性证明、在链上由验证器合约验证的方式降低目标链验证负担.

当链下只对链下验证节点的身份和权限进行验证, 需要保证节点本身的可信问题, 信誉值机制是一种评估节点历史行为的有效方式, 它进一步决定了节点提交的交易数据是否可靠[41]. 通过对节点行为的持续监测和评估, 信誉管理系统能够提高链下验证机制的安全性. 翟社平等人[42]提出一种基于改进的 PowerTrust 算法的中继链信

誉管理模型. Chen 等人[43]提出了一种基于节点过去交易行为的动态信誉管理方案, 将节点行为作为评估节点信誉的基础, 以支持对恶意行为的惩罚, 并使系统能够及时拦截.

3.2 链上验证可信执行机制

链上验证可信执行机制是利用分布式账本的可信, 通过区块链为事务执行提供可信计算和其他链上交易、数据的可信验证, 如图 6. 基于分布式账本的信任机制依赖于区块链的共识证明与状态证明技术, 如 ETH Relay[44]的主要思想是通过减少在链上执行状态更新的成本, 从而降低智能合约的交易成本. 它在链上存储区块头信息, 从而实现链上验证.

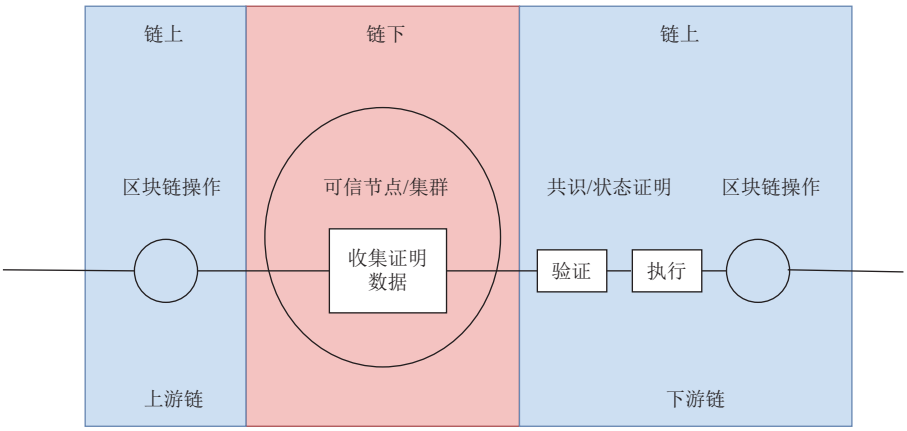

图 6 链上验证可信执行机制

链上验证机制充分利用了区块链本身的信任根 (共识安全性与不可篡改性), 通过在下游链上直接存储和验证上游链的共识证明 (如区块头) 和状态证明 (如 Merkle proof), 为跨链操作提供最高级别的安全保证. 然而, 这种机制存在显著的扩展性瓶颈: 1) 存储开销巨大: 持续存储大量上游链的区块头信息占用宝贵的链上存储资源; 2) 验证成本高昂: 在下游链上验证上游链的共识签名或复杂状态证明的计算成本 (Gas 消耗) 可能非常高昂, 尤其当上游链共识机制复杂 (如 PoS 多签) 或状态证明体积庞大时; 3) 跨链延迟增加: 链上验证流程本身会增加跨链操作的确认时间. 这些因素共同限制了链上验证机制在高吞吐量、低成本跨链场景中的应用.

相比之下, 链下验证机制将繁重的数据获取、验证计算和事务执行逻辑移出下游链主流程, 交由链下的可信执行者 (TEE、中继链、预言机网络、多签委员会等) 处理, 仅将最终的、经过压缩或证明的验证结果 (如聚合签名、零知识证明、有效性声明) 提交到链上进行轻量级验证. 这种架构带来了显著优势: 1) 大幅降低链上负载: 避免在下游链存储大量原始数据和执行复杂验证计算; 2) 提升可扩展性与性能: 链下执行可以并行化、优化, 不受链上 Gas 限制和区块时间约束, 显著提高吞吐量并降低延迟; 3) 降低成本: 链上只需进行轻量级验证 (如验证零知识证明、检查多签阈值), Gas 费用大幅降低; 4) 架构灵活性: 可以灵活选择不同的链下信任模型 (基于经济抵押、TEE 硬件安全、多方共识等) 以适应不同安全性和性能需求.

尽管链下验证机制引入了额外的信任层 (依赖于链下执行者的诚实性或共识), 但通过精心设计的经济激励机制 (如抵押惩罚)、密码学证明 (如零知识证明 zk-SNARKs/zk-STARKs、有效性证明) 以及多样化的信任最小化方案 (如去中心化中继网络、多签委员会), 其安全性在实践中已能达到相当高的水平, 并在安全与效率之间取得了更好的平衡. 因此, 考虑到区块链扩展性的核心挑战以及跨链应用对高性能、低成本日益增长的需求, 链下验证可信执行机制凭借其显著的效率优势和不断成熟的信任最小化技术, 正成为跨链基础设施发展的主要趋势.

3.3 跨链事务可信执行方案评价

- 评价指标

为了系统评估现有方案, 本文基于公开论文、项目官方文档、开源实现基准与技术说明, 从信任假设、目标

链验证开销、证明/提交数据量、最终性延迟、通用性与可扩展性、实现复杂度、吞吐量、容错率以及去中心化程度九个维度进行归纳分析, 整理得到表 4. 需要说明的是, 表 4 并非在统一软硬件平台下直接复现实验所得, 而是对不同方案在公开资料中披露的代表性结果进行横向比较. 若官方资料仅披露区间值或相对下降幅度则保留区间, 并在“测试环境、方法与数据来源”列注明测试条件.

表 4 跨链事务可信执行方案对比

方案	信任假设	目标链验证开销 (Gas)	证明大小	延迟	通用性/扩展性	实现复杂度	吞吐量	容错率	去中心化程度	测试环境、方法与数据来源
Wormhole[32]	强信任 (Guardian 委员会)	392 789	~1.5 KB	低	高	低	高	⩽6/19	低	Optimism→Ethereum主网交易样本; 数据来源: 文献[45]
Cosmos[46]	弱信任 (上游链共识)	>500k	~10–20 KB	中	中	高	中	<1/3 拜占庭节点	高	以太坊主网模拟; 测试条件: 验证Tendermint轻客户端头; 数据来源: 文献[37]
Polkadot[47]	弱信任 (中继链共识)	~300k–600k	~16 KB	中	低	高	中	<1/3	中	Polkadot→Ethereum主网; 验证包含随机抽样的验证者签名、位域及MMR证明; 数据来源: 以太坊主网交易[48]
MAP Protocol[31]	密码学信任 (ZKP)	~650k	~200 Bytes	中	高	极高	低-中	密码学假设	中	以太坊主网; 测试条件: 混合轻客户端方案, 单次跨链交易验证; 数据来源: 文献[31]
Polygon[40]	密码学信任 (ZKP)	~200k	~400 Bytes	中	高	极高	低-中	密码学假设	中	官方技术文档; 测试条件: FFLONK验证器单次验证; 数据来源: 文献[49]
XCLAIM[29]	经济理性 (超额抵押)	~50k–100k	~500 Bytes	高	中	中	低	经济假设	中	Ethereum Ropsten测试网; 采用Bitcoin P2PKH交易和BTC-Relay; 数据来源: 文献[29]

为了客观评估不同的可信执行方案, 建议引入以下 9 个关键维度.

(1) 信任假设 (trust assumption): 系统安全性的根基, 包括如下 4 类.

a) 强信任/外部信任: 依赖选定的外部实体 (如多签委员会) 诚实.

b) 弱信任/内生信任: 仅依赖上游链的共识安全, 无额外信任层.

c) 密码学/数学信任: 依赖零知识证明的数学完备性.

d) 经济理性假设: 假设参与者为了利益不会作恶 (如抵押惩罚机制).

(2) 链上验证开销 (on-chain cost): 在下游链 (以 Ethereum 为基准) 验证跨链消息所需的资源开销. 这是决定方案能否大规模商用的关键经济指标.

(3) 证明大小 (proof size): 提交到下游链的数据包大小 (包含签名、Merkle 分支或 ZKP).

(4) 最终性延迟 (finality latency): 从上游链发起事务到下游链确认生效所需的时间. 包含生成证明的时间、挑战期 (乐观验证) 或共识确认时间.

(5) 通用性与扩展性 (universality & extensibility): 方案是否支持异构链 (如 EVM 到非 EVM)? 接入新链是否需要重新开发复杂的轻客户端?

(6) 实现复杂度 (implementation complexity): 链下证明生成系统和链上验证合约的开发难度.

(7) 吞吐量 (throughput): 系统在单位时间内可处理的跨链消息或事务数量. 受上游链共识速率、链下证明生成时延、网络传输带宽等因素共同制约, 是衡量方案能否支撑高频跨链应用的核心性能指标. 由于各方案瓶颈机制差异显著且缺乏统一测试基准, 此处采用高/中/低定性标注.

(8) 容错率 (fault tolerance): 系统在维持安全性与活性的前提下可容忍的恶意或失效节点比例. 基于委员会签名的方案由协议公开的门限参数决定 (如 Wormhole 13/19 门限, 可容忍最多 6 个节点故障); 基于 BFT 共识的方案

依据共识协议安全属性可容忍不超过 1/3 的拜占庭节点; 密码学 ZKP 不依赖节点诚实假设, 安全性由密码学难题假设保障.

(9) 去中心化程度 (decentralization degree): 评估方案对信任根的分散程度, 参考验证节点数量规模、节点准入机制 (许可/无许可) 和信任来源 (委员会/链共识/数学证明), 分为高/中/低这 3 档. 高度去中心化意味着系统抵御单点攻击与审查的能力更强, 但可能以牺牲吞吐量或增加协调成本为代价.

● 方案对比

(1) 信任假设方面: Wormhole 采用强信任假设, 其安全性完全依赖于外部 Guardian 委员会的诚实度, 但效率高; Cosmos 与 Polkadot 采用弱信任假设, 通过轻客户端直接验证上游链共识签名, 将安全性锚定于上游链本身; XCLAIM 引入经济理性假设, 依赖 Vault 的抵押来遏制恶意行为; 而 MAP Protocol 与 Polygon 则进一步将信任锚点升级为密码学信任, 依靠零知识证明的知识安全属性, 在不引入第三方实体的前提下, 通过数学手段保障状态无法被伪造.

(2) 链上验证开销方面: 以以太坊主网为例, Wormhole 需要在链上验证至少 13 个 Guardian 的签名, 按单次 VAA (verified action approval) 消息验证的公开统计口径, 平均 Gas 消耗约为 392 789 Gas. 相比之下, Cosmos IBC 若在以太坊上运行, 需验证 Tendermint 轻客户端头 (包含数百个签名), Gas 消耗可能超过 500k Gas. Polkadot 采用 BEEFY 协议来适配以太坊 EVM, 通过引入 ECDSA 签名和 MMR 数据结构降低验证难度. 然而, 根据公开技术文档, 在以太坊上验证包含 MMR 根和验证者位域的证明仍需消耗约 300k–600k Gas. MAP Protocol 采用混合轻客户端方案, 在以太坊主网对单次跨链交易进行验证时, 链上 Gas 消耗约 650k. Polygon 采用 FFLONK (fast-Fourier-transform based Plonk) 做最后验证, 交易 Gas 开销~200k. XCLAIM 在以太坊上验证此类证明 (主要涉及 SHA-256 哈希计算) 的 Gas 开销较低, 约为 50k–100k Gas.

(3) 证明大小方面: Cosmos IBC 依赖轻客户端验证上游链的区块头. 由于早期 Tendermint 共识要求收集超过 2/3 验证者的签名, 且不支持聚合签名 (如 BLS), 其证明大小随验证者数量线性增长. 以 Cosmos (约 180 个验证者) 为例, 单个区块头的证明数据约为 10–20 KB. 这不仅消耗了大量的链上存储, 也导致验证成本居高不下. 相比之下, Wormhole 通过简单的 13/19 门限签名将大小控制在 ~1.5 KB, 而 ZKP 方案则通过生成常数大小的证明 (~200 Bytes) 实现了极致的压缩. MAP Protocol 生成常数大小的零知识证明, 其提交到目标链的证明大小仅为 ~200 Bytes. Polygon 基于 FFLONK 的压缩证明大小~400 Bytes. XCLAIM 采用基于 SPV 的轻客户端模式, 其跨链证明主要由比特币区块头和 Merkle 包含路径组成, 证明大小约为 500 Bytes.

(4) 最终性延迟方面: 最终性延迟由物理共识确认时间和验证生成时间共同决定. Wormhole 凭借公证人签名机制, 在观测到事件后可几乎实时转发, 延迟最低; Cosmos、Polkadot 及 XCLAIM 必须等待上游链达到概率最终性或绝对最终性 (如 Bitcoin 的 6 个区块), 存在不可避免的物理延迟 (中等); ZKP 方案在此基础上还需叠加链下电路生成证明 (proving time) 的计算耗时, 尽管硬件加速正在优化这一过程, 但目前仍使得整体延迟略高于单纯的签名转发方案.

(5) 通用性与扩展性方面: 通用性反映了方案跨越异构链的能力. Wormhole 与 ZKP 轻客户端 (MAP/Polygon) 具有高通用性, 前者仅需部署标准化的签名验证合约, 后者通过标准化的 ZKP 验证器适配不同链, 无需重构链上逻辑即可接入新链; XCLAIM 通常绑定特定的资产类型和 SPV 验证逻辑, 通用性中等; Cosmos 通过标准化的 IBC 协议具备一定跨链通用性, 但接入非 Cosmos 生态的异构链仍需较大开发代价, 通用性中等; Polkadot 的 XCMP 高度依赖其中继链架构, 与外部异构公链的互操作需要引入额外适配层, 通用性与扩展性最为受限.

(6) 实现复杂度方面: 实现复杂度呈现出从协议层向计算层转移的特征. Wormhole 的工程逻辑最为简洁, 核心仅涉及链下 P2P 网络与链上多签验证; XCLAIM 需维护抵押率预言机与挑战逻辑, 复杂度中等; Cosmos 与 Polkadot 涉及深度的协议层握手与状态机同步, 工程量大; 而 ZKP 方案 虽然链上逻辑简洁, 但链下需设计针对特定共识算法的算术电路并维护高性能 Prover 集群, 涉及高深的密码学工程与算力优化, 实现复杂度极高, 是当前技术大规模落地的主要门槛.

(7) 吞吐量方面: 各方案的吞吐量瓶颈位置不同. Wormhole 依托 Guardian 网络并行签名与消息转发, 是现有

方案中吞吐量最高的. Cosmos IBC 与 Polkadot 的吞吐量上限由上游链 (或中继链) 共识速率决定, 本质受限于链层本身. XCLAIM 受比特币出块速率与确认周期约束, 吞吐量相对较低. MAP Protocol 与 Polygon 等 ZKP 方案的瓶颈在于链下 Prover 集群的证明生成速率: 当前 Prover 通常需要数十秒至数分钟完成一次电路证明, 尽管并行 Prover 与专用硬件 (FPGA/GPU 加速) 持续优化, 实际吞吐量仍处于低-中水平.

(8) 容错率方面: Wormhole 采用 13/19 门限签名, 可容忍最多 6 个 Guardian 节点作恶或失效 (容错率约 32%). Cosmos IBC 与 Polkadot 继承各自上游链/中继链的 BFT 共识安全性, 依据 BFT 协议安全属性可容忍不超过 1/3 的拜占庭验证节点. MAP Protocol 与 Polygon 将信任锚点转移至密码学假设, 系统安全性不依赖任何节点的诚实性, 只要底层密码学假设成立则无法被伪造, 理论容错能力最强. XCLAIM 的安全性依赖经济理性假设——Vault 维持超额抵押且理性不作恶, 当资产价值发生剧烈波动时存在抵押不足风险.

(9) 去中心化程度方面: Cosmos 的验证直接复用上游链的公开验证者集合, 节点准入完全依靠经济质押, 去中心化程度高. Polkadot 依托中继链共识, 中继链验证者集合同样以质押选举产生, 但节点规模相对有限且生态内准入特性更强, 去中心化程度中等. XCLAIM 的 Vault 为无许可角色, 任何人均可参与, 但用户需自行甄别可信 Vault, 去中心化程度中等. MAP Protocol 与 Polygon 目前的 Prover 节点存在一定的许可/半许可属性, 处于逐步去中心化过程中, 综合评定为中等. Wormhole 的 Guardian 委员会仅有 19 个许可成员, 去中心化程度最低, 是其最主要的单点信任风险.

● 技术选型建议

基于上述 9 个维度的对比, 不同应用场景对跨链可信执行方案存在差异化的选型需求, 综合建议如下.

(1) 高频跨链消息传递场景: 推荐 Wormhole. 其 Guardian 委员会并行转发机制提供较高吞吐量低延迟, 链上验证成本约 392k Gas. 代价是接受许可委员会的信任假设.

(2) 高价值资产跨链与安全优先场景: 推荐 MAP Protocol 或 Polygon 等 ZKP 方案. 密码学信任假设无需依赖任何节点诚实, 提供最强安全保障; 常数级证明能够显著压缩链上存储与验证负担. 主要约束是当前 ZK Prover 集群的吞吐量瓶颈, 适用于安全性要求极高、对吞吐量要求相对宽松的场景.

(3) 强去中心化与生态系内跨链场景: 推荐 Cosmos IBC 或 Polkadot XCMP. 两者将安全性锚定于上游链/中继链共识本身, 无需额外信任层; 其中 Cosmos IBC 去中心化程度高, Polkadot 去中心化程度中等. 需注意异构链接入成本较高, 建议已与 Cosmos/Polkadot 生态深度集成的项目优先考虑.

综上, 跨链可信执行方案的选型应依据具体场景在安全性 (信任假设、容错率)、性能 (吞吐量、延迟)、成本 (Gas 消耗、抵押率) 和去中心化程度之间做出权衡. ZKP 方案代表了长期演进方向, 委员会签名方案 (如 Wormhole) 则提供了当前阶段较优的性能-成本平衡.

4 跨链事务并发执行关键技术

事务并发执行技术保证事务执行的原子性和隔离性, 解决和减少事务并发执行的冲突问题, 同时不能给跨链系统带来不必要的负担. 跨链事务并发执行技术包括两个部分, 并发控制协议与资源访问控制方法. 并发控制协议用于调度并发事务的操作执行顺序, 使得并发事务的执行过程不会破坏跨链系统的一致性. 资源访问控制方法为并发控制协议提供隔离能力, 同时为应用场景提供隐私性与安全性.

4.1 并发控制协议

跨链事务系统一般采用分层模式, 将系统纵向分为事务层和区块链层. 如图 7 所示, 事务层负责统一调度并发事务的执行顺序, 区块链层通过合约或者账本协议实现资源访问控制.

两阶段提交机制 (two phase commit, 2PC)[50]是常用的”自上而下”并发控制协议, 保持区块链之间跨链事务执行状态一致的一种分布式算法. 它的第 1 阶段先预执行, 判断事务是否能完整执行, 缓存状态, 如果判断可以完整执行, 则进入第 2 阶段事务提交, 持久化状态.

1) 投票阶段, 其中协调器进程向所有事务的参与进程 (指定参与者、群组或工作者) 发出投票请求, 如果事务

参与者的本地部分执行已正确结束, 则投票“是”, 否则投票“否”.

2) 提交阶段, 在此阶段, 协调者根据参与者的投票决定是提交 (仅当所有参与者都投了“是”票时) 还是中止事务 (否则), 并将结果通知所有参与者. 然后, 参与者使用其本地事务资源 (也称为可恢复资源; 例如数据库数据) 以及事务其他输出中的各自部分 (如果适用) 执行所需的操作 (提交或中止).

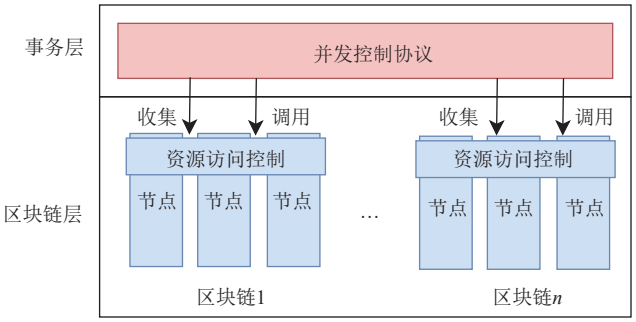

图 7 跨链事务分层模式

两阶段提交协议最大的缺点就是它是一种阻塞协议. 如果协调者永久失败, 一些参与者将永远无法解决他们的事务: 在参与者发送同意消息作为对协调者发出的提交请求消息的响应后, 它将被阻塞, 直到收到提交或回滚. 在提交阶段, 如果协调器和队列成员都发生故障, 两阶段提交协议无法可靠地恢复. 如果只有协调器发生故障, 而没有队列成员收到提交消息, 则可以安全地推断未发生任何提交. 但是, 如果协调器和队列成员都发生故障, 则发生故障的队列成员可能是第一个收到通知的成员, 并且实际上已经提交了. 即使选择了新的协调器, 它也无法自信地继续操作, 直到收到所有队列成员的同意, 因此必须阻塞直到所有队列成员响应.

如 Falazi 等人[51]提出的 TCCSCI (transactional cross-chain smart contract invocation) 方法, 引入区块链两阶段提交 (two-phase commit for blockchains, 2PC4BC) 协议, 保证全局原子性与有序性, 并由合约实现资源管理, 每个参与跨链的业务合约都调用资源管理合约 (resource manager smart contract, RMSC), 利用 RMSC 来确保事务隔离性. 长安链的跨链协议 TCIP (trust chain interoperability protocol)[52]也是通过两阶段提交机制保证事务的原子性. 还有的事务并发控制策略是基于“补偿机制”保证原子性的, 通过显式给出事务处理失败时的补偿操作[53], 保证原子性.

4.2 资源访问控制方法

链上资源访问控制方法可以分为 4 个方面: 锁定范围、状态切割、控制实现方式以及状态终止与合并.

● 锁定范围

锁定是数据的一种经过共识的状态, 所以锁定机制必然是要实现在共识机制里, 根据锁定状态的感知范围分为系统锁定和合约锁定. 系统锁定是指区块链状态的锁定是全局可见的, 合约锁定是指状态的锁定状态是合约范围可见的.

锁定状态的可见范围决定锁定范围. 系统锁定要求链共识机制本身有锁定机制, 或者链共识机制本身具有可扩展性, 如比特币的 HTLC[54], 利用比特币共识机制的脚本功能来实现的, 这样达到系统锁定的要求.

合约锁定则将资源的支配权固定在合约中, 合约根据协议将支配权进行再分配, 例如 HTLC, 哈希锁依靠预像抵抗属性.

● 状态切割

状态切割是缩小锁定范围的方法, 将需要锁定的状态切割出来, 使其不影响未锁定部分的并发执行. 最常见也是最方便的状态切割是针对账户余额场景的切割, 将用户一部分余额转移到另一个地址做状态锁定, 比如 XCLAIM[29]中的 Vault 仅冻结交易所需额度.

● 控制实现方式

锁定状态的控制实现方式, 最常用的是“智能合约+承诺”的方式, 合约实现锁定状态的控制逻辑, 包括更新与

终止, 同时会将承诺写入合约, 满足承诺条件的即可将状态解锁.

承诺有多种实现方式, 包括 HTLC、ZK、多签和经济质押.

(1) HTLC

HTLC 在比特币上是一个脚本, 比特币支付者 (简称买方) 给这个脚本支付比特币以及设定 image(image = hash(preimage)), 其中 preimage 是买方的私有数据), 哈希时间锁定合约将获取这些比特币一段时间 (块数) 的支配权, 支配逻辑已经写好在脚本中, 允许比特币接收方 (简称卖方) 通过提供 preimage 来获取比特币, 超过一段时间卖方没有提取则买方能够取回这个比特币.

HTLC 的优点是支持异步锁定同步解锁, 支持多链的原子交换 (atomic cross-chain swaps), 从而支持没有中介机构的资产交换[55,56], 也被用于确保跨支付网络的支付安全[57].

HTLC 的潜在问题如下.

a) 该方案成功的前提是存在买卖双方交换商品和 preimage 的安全通道, 否则就会出现两种不好的情况, 买方接收商品后没有提供 preimage 给卖方, 或者卖方接收 preimage 后没有提供商品.

b) 根据接收方是否有限定分为限定接收方和不限定接收方两种, 如果不限定接收方会存在安全隐患, 首先, 接收方提取比特币需要附带明文 preimage, 如果 preimage 在上链前被恶意节点窃听则有可能会被窃取, 其次, 发送方知道 preimage, 会不会私下用其他账户提取.

c) 在推广的时候, 区块链节点之间对于“一段时间”的共识是很难的, 比特币由于块上链速度慢, 可以使用块数作为“一段时间”的等价物. 但其他区块链继续用块数是否合适, 如果不合适应该用什么作为时间.

d) HTLC 对于激励操纵攻击是脆弱的. Winzer 等人[56]表明, B 可以贿赂矿工使用指定的智能合约忽略 A 的事务直到超时结束. 同样地, Harris 等人[58]显示, B 可以通过向系统中添加自己的事务来延迟 A 的事务确认. 这两种方法都允许 B 获得 HTLC 代币而剥夺 A 的权利, 即使 A 发布了预像, MAD-HTLC 针对这个问题提出了新的激励机制来防止矿工的贿赂[59], HCNCT 针对这个问题, 将门限与 HTLC 的结合, 解决 HTLC 的恶意超时交易攻击以及公证人中心化问题[60].

e) HTLC 也容易受到外界干扰, 如果因为节点宕机或网络故障等原因在超时的时候没有办法提供秘密 s, 则卖方将损失这笔钱. 针对这个问题, Zakhary 等人[61]提出了 AC3WN 协议, 通过开放区块链网络来监视 AC2T (atomic cross-chain transaction) 交易是否提交或中止.

(2) 零知识证明

ZKP (zero knowledge proof) 使得 HTLC 的潜在问题 *a* 在面向信息交换的场景中得到了解决. 面向信息交换的场景, 首先, 买方能提供所需信息的自动验证程序 *P*, 自动验证程序被写入智能合约, 相当于把货币控制权交给了合约, 合约能通过自动验证程序 *P* 判断卖方信息真伪, 当卖方提供的信息满足买方的验证程序则把货币转给卖方, 买方也得到了所需信息. 卖方提供的内容如下.

① 卖方随机选择加密密钥 *k*, 对卖方信息进行加密得到 *Ex*, *Ex* 满足自动验证程序 *P*.

② 卖方提供加密密钥 *k* 的哈希值 *h*.

具体流程如图 8, 合约能够通过自动验证程序 *P* 验证卖方是否持有买方所需信息, 并且能够当卖方提供秘密的时候, 合约根据哈希值判断是否将货币转给卖方. 如 ZKCP[62]通过 ZKP 完成不揭露信息的前提下完成安全交易的.

(3) 多签

Zie 等人[63]提出了仅支持多签名 (multisig) 交易的 atomic cross-chain swaps 协议的扩展. A 将钱转入 A 链上合约, B 在 B 链的提取支票上签名发给 A, A 对提取支票签名表示 A 到 B 链上做钱的提取, B 在 B 链上得到 A 已签名的提取支票, 然后到 A 上合约取钱.

(4) 经济质押

XCLAIM 是一种基于原子跨链交换协议的信任最小化方案, 包含 3 个核心参与角色: 请求方 (Requester)、保管库 (Vault) 和赎回方 (Redeemer). 协议执行流程如下.

a) 抵押阶段: 保管库在源区块链 (记为 Blockchain I) 的互操作性智能合约 (iSC) 中锁定足额抵押资产.

b) 资产锁定与证明提交: 请求方在目标区块链 (记为 Blockchain B) 上完成标的资产 (Token b) 的锁定操作, 并向跨链中继服务 (ChainRelay) 提交该锁定交易的可验证密码学证明.

c) 跨链验证: ChainRelay 对请求方提交的证明进行有效性验证, 确认 Token b 在 Blockchain B 上的锁定状态.

d) 验证通过与资产铸造: 验证通过后, 协议在 Blockchain I 上触发铸币流程, 向请求方发行与锁定资产价值锚定的映射通证 Token i(b). 该通证的价值由保管库在 iSC 中锁定的抵押资产足额支撑.

e) 原子化交换操作: 获得 Token i(b) 后, 请求方与赎回方可在 Blockchain I 上通过原子交换或原子转移完成资产交换.

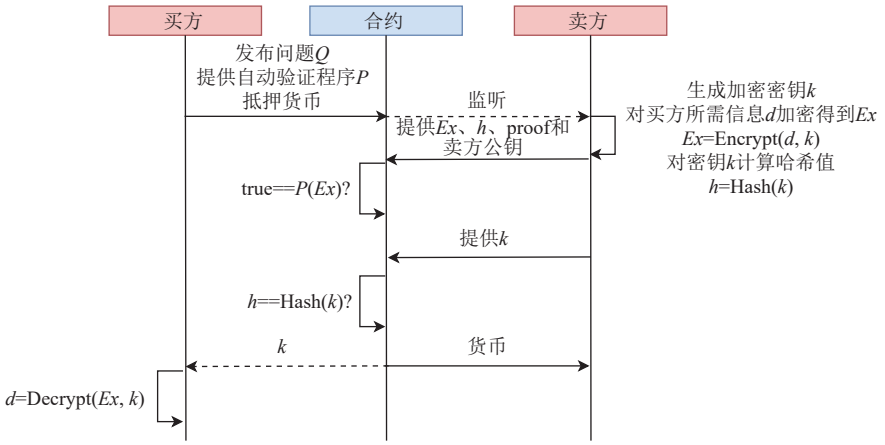

图 8 ZKP 资源控制流程

● 状态终止与合并

状态终止与状态合并构成资源访问控制的闭环生命周期管理, 其核心是通过预定义规则实现锁定状态的释放与回写, 并保障状态一致性.

(1) 状态终止机制

状态终止由承诺达成条件或超时回滚条件触发, 具体实现方式如下.

a) 条件满足型终止

当承诺条件 (如哈希原像提交、零知识证明验证、多签阈值达成) 被满足时, 智能合约自动执行状态解锁. 例如: HTLC 中接收方提交 preimage 后立即释放锁定资产; ZKP 承诺锁在链上验证零知识证明成功后终止锁定状态.

b) 超时回滚型终止

预设时间窗口内未满足条件时, 触发状态回滚 (如资金返还发起方). 异构区块链采用不同时间度量 (块高度 vs. 真实时间), 需通过中继器或预言机实现时间共识, 也可以采用超额抵押机制 (如 XCLAIM 要求>100% 抵押) 抵御恶意延迟交易攻击.

(2) 状态合并流程

终止后的状态需安全合并回主状态池, 其过程包含: 当被切割状态未受其他操作影响时, 直接更新主状态树 (如账户余额合并); 若合并时目标状态已被修改 (如双花攻击), 可采用多签委员会 (如 Aragon Court[64]) 或质押投票机制 (如 Cosmos Governance) 进行人工仲裁, 典型流程: 冲突状态冻结并提交仲裁提案, 质押代币的治理者投票裁决, 按多数决结果强制执行状态合并.

4.3 跨链事务并发执行方案评价

● 评价指标

为了系统评估现有方案, 本文从并发控制时间开销、并发控制成本、状态锁定范围、事务隔离级别、资产抵

押率、实现难度、吞吐量、容错率以及去中心化程度这 9 个维度进行深度解析, 整理得到表 5.

表 5 跨链事务并发执行方案对比

文献	并发控制时间开销	并发控制成本	状态锁定范围	事务隔离级别	资本抵押率	实现难度	吞吐量	容错率	去中心化程度	测试环境、方法与数据来源
HTLC[54]	高 (受限于超时)	低	对象级	读已提交	100%	低	低	N/A	高	Bitcoin主网; 测试条件: 标准HTLC脚本执行, 超时窗口通常数小时至24 h; 数据来源: 文献[54,57]
TCCSCI[51]	高 (两轮阻塞)	高	合约级	可串行化	100%	中	中	弱	中	模拟多链环境 (Ethereum+Fabric); 测试条件: 2–4链跨链事务场景, 涵盖Prepare与Commit两阶段; 数据来源: 文献[51]
XCLAIM[29]	中 (SPV验证)	中	对象级	快照隔离	大于100%	中	低	经济假设	中	Bitcoin-Ethereum测试网; 测试条件: Bitcoin SPV Merkle路径验证, 典型抵押率150%; 数据来源: 文献[29]
ZKCP[62]	中 (证明生成)	中	对象级	可重复读	100%	高	低-中	密码学假设	高	Bitcoin主网实测; 测试条件: ZK证明生成约60 s (2016年环境); 数据来源: 文献[62]

(1) 并发控制时间开销 (time overhead): 指跨链事务从发起到最终确认所需的总时长. 主要受跨链通信轮次、锁定时间窗口以及证明生成/验证时间的影响.

(2) 并发控制成本 (control cost): 指为了维护并发安全性所付出的资源代价. 包含链上成本 (智能合约执行计算资源消耗、状态存储费) 和链下成本 (生成零知识证明或 SPV 证明的计算资源消耗).

(3) 状态锁定范围 (state locking scope): 指并发控制机制在执行事务时对共享资源的占用边界, 直接决定了系统的资源争用程度, 取决于状态数据存储的数据结构, 允许并发事务同时操作同一资源的不同部分, 从而避免全局锁竞争. 有系统级、合约级和对象级别.

(4) 事务隔离级别 (isolation level): 为了系统地评估不同跨链事务并发执行方案的一致性保障能力, 本文引入经典数据库理论中的事务隔离级别概念[65]. 隔离级别为用户在系统性能 (吞吐量、延迟) 与并发隔离性 (数据一致性) 之间提供了权衡标准[66]. 常见的隔离级别由弱到强依次为: 读未提交 (read uncommitted)、读已提交 (read committed) 、可重复读 (repeatable read)、快照隔离 (snapshot isolation) 和可串行化 (serializable).

大多数数据库系统通过避免某些特定的非可串行化异常来定义隔离级别[67–69]. 在跨链环境下, 关键的 5 类并发异常定义如下.

a) 脏写异常 (dirty write): 指一个事务修改一个数据, 在这个事务提交或者回滚之前存在另一个并行的事务也修改同一个数据, 如果第 1 个事务回滚, 这会使得第 2 个事务的修改也被同时回滚.

b) 脏读异常 (dirty read): 指一个事务读取到另一个未提交事务写入的结果. 在跨链场景中, 这对应于读取了未达到最终确认的区块数据.

c) 丢失更新 (lost update): 指两个并行的事务同时读取一个数据, 并且基于读取的数据进行更新, 使得先执行事务产生的更新未能得以保留的并发问题.

d) 读偏异常 (read skew): 指在一个事务读取数据库的过程中, 存在另外一个并行的事务同时修改数据库, 导致第 1 个事务的读取结果既包含第 2 个事务修改之前的数据, 也包含第 2 个事务修改之后的数据.

e) 写偏异常 (write skew): 指事务并行读取一个满足一致性约束的数据集, 并且事务各自修改该数据集中不相交的数据项, 进而破坏数据集一致性约束的并发问题.

一般而言, 隔离级别缺乏统一的工程实现标准, 本文基于上述异常避免情况, 将跨链事务的隔离级别定义如表 6 所示.

表 6 跨链隔离级别定义

隔离级别	脏写异常	脏读异常	丢失更新	读偏异常	写偏异常
读未提交	禁止	允许	允许	允许	允许
读已提交	禁止	禁止	允许	允许	允许
可重复读	禁止	禁止	允许	禁止	允许
快照隔离	禁止	禁止	禁止	禁止	允许
可串行化	禁止	禁止	禁止	禁止	禁止

a) 读未提交是最低的隔离级别. 一个事务可以读取另一个事务尚未提交的修改数据. 不提供任何隔离性保障, 性能最高, 但数据一致性最差.

b) 读已提交这是大多数数据库 (如 PostgreSQL、Oracle) 的默认隔离级别. 它保证一个事务只能读取到已经提交的数据. 不允许脏读异常, 但无法保证在同一个事务中多次读取同一数据时结果一致 (因为其他事务可能在两次读取之间提交了修改).

c) 可重复读该级别保证在同一个事务内, 多次读取同一行数据的结果是一致的. 即使其他事务在两次读取之间修改并提交了该行数据, 当前事务也看不到这些修改.

d) 快照隔离, 事务的所有读取操作都基于事务开始时刻的一个一致性快照.

e) 可串行化这是最高的隔离级别. 它要求事务的并发执行结果必须与某种顺序的串行执行 (即一个接一个执行) 结果完全相同.

(5) 资产抵押率 (collateral ratio): 指为了保障跨链原子性, 协议要求参与节点锁定的资产价值与交易价值的比例, 反映了资本效率 (capital efficiency).

(6) 实现难度 (implementation difficulty): 指工程落地的复杂度, 包括智能合约逻辑的复杂性、链下节点的维护难度以及密码学电路 (如 ZK Circuits) 的开发门槛.

(7) 吞吐量 (throughput): 系统在单位时间内能够完成的跨链并发事务数量. 受并发控制协议的交互轮次、锁定持有时间、链下证明生成速率等因素制约. 由于各方案瓶颈机制差异显著, 此处采用高/中/低定性标注并括注瓶颈来源.

(8) 容错率 (fault tolerance): 并发控制机制在面对节点故障、网络分区或恶意行为时维持系统活性与安全性的能力. 对于基于锁定的协议 (如 2PC), 需特别关注协调器单点故障风险; 对于基于密码学承诺的协议, 关注密码学假设的健壮性.

(9) 去中心化程度 (decentralization degree): 并发事务执行过程中对中心化实体 (如协调者、中继器、信任第三方) 的依赖程度, 分为高/中/低这 3 档. 高度去中心化方案避免单点信任, 但可能以引入复杂的分布式协调为代价.

● 方案对比

(1) 并发控制时间开销方面: 时间开销主要取决于协议的交互模式与锁定机制. HTLC 和 TCCSCI 的时间开销被评为高, 但原因不同, HTLC 为了应对链活性故障, 必须设置较长的安全超时窗口 (通常为数小时), 导致异常情况下的资金周转极慢; TCCSCI 作为同步阻塞协议, 需要跨链进行两轮通信 (prepare+commit), 网络延迟被放大且容易造成队头阻塞. 相比之下, XCLAIM 和 ZKCP 的开销为中, 因为它们主要依赖单向的证明提交, 无需长时间的交互式等待, 延迟主要取决于上游链确认数和证明生成时间.

(2) 并发控制成本方面: 成本评估综合了链上资源消耗和链下计算. HTLC 仅涉及基础的哈希原像校验和时间锁脚本, 链上资源消耗最低, 且无复杂链下计算, 成本低. TCCSCI 需要在两条链上部署复杂的资源管理合约 (RMSC) 来维护事务状态, 多次跨链调用导致链上资源消耗成本高. XCLAIM 需在目标链验证 SPV Merkle 路径, 资源消耗随树深度增加, 成本中等. ZKCP 虽然链上验证成本极低, 但链下生成零知识证明需要消耗大量计算资源, 综合成本评为中.

(3) 状态锁定范围方面: 状态锁定范围界定了并发控制机制在维持一致性时对共享资源的占用边界, 直接关联系统的资源争用程度与吞吐量上限. XCLAIM 采用精细的对象级锁定, 引入“保管库”机制将资产池化, 允许跨链事

务仅锁定 Vault 资产池中与当前请求对应的特定份额, 而剩余的未锁定资产仍可同时服务于其他并发请求, 这种机制显著降低了全局锁竞争并提升了并行处理能力. HTLC 与 ZKCP 实施明确的对象级锁定, 它们针对每一次跨链交换生成独立的哈希锁或零知识证明, 虽然实现了交易间的隔离, 但对特定的资金对象实施全额互斥访问. TCCSCI 则需在资源管理合约中实施合约级锁定, 对参与事务的整体资源状态进行独占保护, 这在高频交易场景下极易引发资源瓶颈与排队阻塞.

(4) 事务隔离级别方面: 引入文中的数据库异常定义, 各方案的一致性保障能力呈现阶梯分布. TCCSCI 通过在预提交阶段严格锁定所有涉及资源, 完全消除了并发异常, 达到了最高的可串行化级别; XCLAIM 利用状态切割实现了类似多版本并发控制 (multi-version concurrency control, MVCC) 的效果, 有效避免了丢失更新, 符合快照隔离特征; ZKCP 通过零知识证明确保了交易内容的隐私性与原子性, 避免了不可重复读, 达到可重复读级别; 而 HTLC 虽然保证了资金安全, 但由于缺乏全局协调, 交易执行期间易受外部状态 (如矿工行为或网络延迟) 干扰, 面临“抢跑”风险, 隔离级别主要体现为读已提交级别.

(5) 资产抵押率方面: 资产抵押率反映了方案的资本效率与安全模型. XCLAIM 为了在去信任环境中防止保管库作恶 (如双花或拒绝赎回), 必须要求 Vault 抵押价值高于用户资产的保证金, 以覆盖汇率波动风险, 属于超额抵押 (>100%); 而 HTLC、ZKCP 以及多签方案本质上是原子交换或即时支付, 遵循“一手交钱一手交货”原则, 要求交易双方进行 1:1 的等额资产锁定, 虽然资本利用率相对较高, 但无法利用杠杆效应.

(6) 实现难度方面: HTLC 基于区块链最基础的哈希和时间锁脚本, 几乎所有公链都原生支持, 难度低. TCCSCI 和 XCLAIM 需要开发复杂的跨链中继、资源管理合约及挑战机制, 涉及分布式系统状态同步, 难度中. ZKCP 的难度被评为高, 因为开发者不仅需要理解共识协议, 还需设计专门的算术电路来实现逻辑证明, 并维护高性能的 Prover 基础设施.

(7) 吞吐量方面: HTLC 的吞吐量受超时窗口长度严重制约, 超时通常设为数小时, 期间锁定的资金无法用于其他交易, 单通道有效吞吐量低. TCCSCI 作为 2PC 协议需要两轮阻塞式通信, 在跨越多条高延迟区块链时, 每轮通信耗时可达数十秒, 但协议本身不限制并发事务总数, 吞吐量处于中等水平. XCLAIM 的吞吐量上限受比特币底层出块速率约束 (~7 TPS). ZKCP 受 ZK 证明生成时延制约, 在 2016 年基准环境中单次证明生成约 60 s[62], 在硬件加速优化后可达低-中水平吞吐量.

(8) 容错率方面: HTLC 和 ZKCP 均为双方直接博弈协议, 不依赖中心化节点, 不存在节点失效的委员会容错问题, 但双方均需保持活性 (在超时前响应); 若一方宕机, 另一方将面临资金临时冻结. TCCSCI 依赖协调器的持续可用性, 若协调器永久失效, 所有参与者将陷入阻塞状态——这是其容错能力最弱的设计缺陷; 通过将协调器逻辑部署在高可用链上合约 (如长安链 TCIP[52]) 中可部分缓解此风险. XCLAIM 的容错性依赖于 Vault 的持续活性和足额抵押, 当 Vault 宕机或抵押不足时, 触发挑战机制进行仲裁.

(9) 去中心化程度方面: HTLC 和 ZKCP 完全去中心化, 无需任何第三方节点, 通过链上脚本或密码学证明实现无信任原子交换, 去中心化程度最高. XCLAIM 通过无许可 Vault 角色实现了中等程度的去中心化, 但用户仍需信任所选 Vault 的持续活性与诚实性, 且 ChainRelay 中继节点存在一定中心化风险. TCCSCI 在现有实现中依赖指定协调器节点驱动 2PC 流程, 去中心化程度相对较低.

● 技术选型建议

基于上述 9 个维度的对比, 不同应用场景对跨链并发执行方案存在差异化的选型需求, 综合建议如下.

(1) 点对点原子资产交换场景: 推荐 HTLC. 其纯链上脚本实现无需任何第三方节点介入, 去中心化程度最高, 实现难度低, 100% 等额抵押保障双方资产安全. 主要约束是超时窗口导致的低吞吐量与资金占用问题, 适合对延迟不敏感、追求极简信任模型的点对点资产互换场景.

(2) 隐私信息与资产联动交换场景: 推荐 ZKCP. 零知识证明承诺保障了交易内容的隐私性与原子性, 完全去中心化且容错性依赖密码学假设而非节点活性. 主要约束是证明生成延迟 (当前硬件加速后已显著改善), 适合需要隐私保护的数字商品或数据资产跨链交换场景.

(3) 企业级/联盟链强一致性跨链事务场景: 推荐 TCCSCI. 两阶段提交机制达到可串行化最高隔离级别, 能够支撑多链复杂业务逻辑的原子性与全局一致性. 需注意协调器单点故障风险, 建议将协调逻辑部署于高可用链上合约 (如长安链 TCIP) 以提升容错能力. 适合对数据一致性要求极高、可接受较高延迟与成本的企业跨链互操作场景.

综上, 跨链并发执行方案的选型应依据具体场景在一致性 (隔离级别、容错率)、性能 (吞吐量、时间开销)、资本效率 (抵押率) 和去中心化程度之间做出权衡. 当前方案在功能覆盖上各有侧重, 尚无单一方案能同时满足高吞吐、强一致、低抵押与完全去中心化的全部需求, 实践中可结合场景优先级选择或组合使用上述方案.

5 跨链事务安全挑战与案例

跨链系统涉及多条异构区块链、中继网络及链上合约的复杂交互, 其攻击面远超单链系统. 一旦跨链事务的一致性被破坏, 往往会导致巨额资产损失. 基于上述讨论, 本节从跨链事务的可信执行与并发执行两个关键维度出发, 如表 7, 系统梳理特有的攻击手段、防御策略及典型案例.

表 7 跨链事务安全挑战与案例

安全层面	威胁类型	防御策略	典型案例
可信执行	链下节点/集群私钥窃取与合谋[70–73]	安全多方计算、可信执行环境硬件隔离、节点轮换机制	Ronin Bridge[70,74]事件、Harmony Horizon Bridge事件[75]
	轻客户端漏洞[76]	乐观验证	BNB Chain Bridge 事件[77]、Multichain 事件[78]
并发执行	恶意耗时[61,79–85]	抵押资产	Lightning Network 漏洞[86]

5.1 攻击手段与防御策略

● 针对跨链事务可信执行的攻击

可信执行机制的目标是确保“下游链接收到的消息真实反映上游链的状态”, 针对此环节的攻击主要旨在伪造跨链消息或绕过验证逻辑, 目前有以下攻击手段和防御策略.

(1) 链下节点/集群私钥窃取与合谋

链下验证机制依赖于链下节点/集群获取、验证、执行和构造证明, 攻击者无需攻破区块链共识, 链上验证机制也依赖于链下节点/集群收集上游链执行状态. 所以攻击者只需控制超过阈值的链下节点私钥, 即可生成合法的签名, 在下游链上伪造任意跨链. 可采取的防御策略如下.

a) 安全多方计算: 利用分布式密钥生成技术, 私钥从未在任何单一节点上完整出现过. 跨链消息的签名需要 t-of-n 个节点协作计算产生分片签名, 再聚合为签名. 攻击者必须同时攻破 t 个独立环境的服务器才能伪造签名, 极大提高了攻击成本.

b) 可信执行环境硬件隔离: 将链下验证工作与签名逻辑运行在 CPU 的安全模块中. 可信执行环境保证了即使操作系统被攻破, 攻击者也无法读取内存中的私钥或篡改运行中的验证代码, 提供了硬件级别的隔离保护.

c) 节点轮换机制: 不依赖固定的验证者集合, 而是按照预定的时间周期或特定事件, 从更大的候选池中随机选择一组新的节点来负责下一阶段的跨链消息验证与签名.

(2) 轻客户端漏洞

针对链上验证机制, 攻击者主要通过伪造证明或利用代码逻辑漏洞来绕过验证. 一种方式是在上游链发生 51% 攻击或区块回滚时, 向下游链提交“幽灵区块头”或已被回滚的交易证明. 另一种更为直接的手段是利用轻客户端合约中密码学验证逻辑的实现缺陷, 构造虚假的 Merkle proof (默克尔证明), 在没有真实业务发生的情况下欺骗合约验证通过. 可引入“乐观验证”机制, 假设提交的跨链消息有效, 但保留一个挑战窗口期. 在此期间, 任何挑战者若检测到上游链状态与提交的证明不符, 可提交欺诈证明并罚没链下验证节点的质押金. 通过 1-of-N 的信任模型 (只要有一个诚实节点即可保障安全) 替代了多签的 M-of-N 模型.

● 针对跨链事务并发执行的攻击

并发执行机制的目标是确保事务的原子性和隔离性. 针对此环节的攻击主要利用时间差、信息不对称或竞争条件来获利. 目前主要是恶意耗时攻击, 针对 HTLC 和原子交换, 攻击者在锁定资产后, 故意不进行后续操作或在最后一刻才揭示秘密. 如果资产价格向有利于攻击者的方向波动, 则完成交易; 否则让交易超时回滚. 攻击者无风险套利, 而诚实用户的资金被长期锁定, 破坏了资产流动性和公平性. 可要求双方都抵押资产, 如果一方在揭示秘密阶段不作为, 根据不作为时长支付另一方时间成本补偿.

5.2 典型案例分析

案例 1: Ronin Bridge 事件 (私钥窃取)

事件经过: 2022 年 3 月, 攻击者控制了 Ronin 链 9 个验证节点中的 5 个 (包括 Sky Mavis 自运营的 4 个节点和 1 个 Axie DAO 节点), 随后使用被盗私钥伪造提款凭证, 仅通过两次交易就成功提取 173 600 ETH 和 2 550 万 USDC, 后因用户无法提款才被发现.

损失: 该事件共造成约 6.24 亿美元损失, 是当时 DeFi 史上金额最大的跨链桥攻击案, 被盗资产主要为 ETH 和 USDC. 由于资金未能追回, 严重打击了 Ronin 生态信誉.

漏洞根因: 根本原因在于私钥窃取.

启示: 验证节点私钥应物理分散管理, 避免单点失陷引发连锁反应; 名义上的多签去中心化不应掩盖实质的中心化控制; 链下验证可信执行机制其安全性比纯链上验证更难保障, 必须从架构、流程、监控这 3 层建立纵深防御体系.

案例 2: BNB Chain Bridge 事件 (轻客户端漏洞)

事件经过: 2022 年 10 月, 黑客利用 BNB Chain 跨链桥 BSC Token Hub 的 IAVL 树验证漏洞, 通过伪造 Merkle 证明分两次提取 200 万枚 BNB. 币安发现后紧急联系全部验证者暂停 BSC 网络, 阻止部分资金流动.

损失: 事件初始损失达 5.66 亿美元, 但由于币安快速响应并冻结约 4.6 亿美元资金, 实际净损失约 1 亿美元.

漏洞根因: 漏洞根源在于 IAVL 树验证漏洞, 证明验证流程未完整校验到根哈希的完整路径.

启示: 跨链桥需建立实时监控和自动熔断机制; 链上验证可信执行机制的链上合约需引入多方安全审计.

案例 3: Lightning Network 漏洞 (恶意耗时)

事件经过: 闪电网络的通道阻塞攻击是针对 HTLC 并发执行的典型恶意耗时攻击, 由 Harris 等人在文献 [58] 中进行了系统分析.

损失: 通道阻塞攻击无需大规模资本投入即可瘫痪目标节点的转发能力, 严重影响网络流动性与公平性, 至今仍是闪电网络未完全解决的开放问题.

漏洞根因: 具体来说, 攻击者能拦截交易双方签名的智能合约, 通过延迟或阻止 HTLC 的适当生成使交易无效化. 更深层的根因在于闪电网络的开放连接特性, 节点可自由加入网络并声称提供零费用路由, 但协议缺乏对中继节点行为有效约束机制, 导致恶意节点能够“免费搭车”后拒绝服务.

启示: 跨链事务的资源访问控制方法单一依赖经济激励 (如路由费用) 不足以防范恶意行为, 需结合声誉系统、通道监控和自动惩罚机制. 闪电网络应实施更严格的入网验证和节点行为约束, 避免匿名节点随意承诺服务后失信.

6 跨链事务整体解决方案

对代表性的跨链整体解决方案从关键技术问题的角度进行分析和讨论. 基于前文提出的分析框架, 本节选取了 Polkadot、Cosmos、WeCross 等代表性跨链项目进行深入剖析. 将紧扣 RQ2 (可信性与事务保障), 从可信执行机制和并发控制策略, 阐述各方案如何具体实现技术权衡与落地.

6.1 Polkadot

如图 9, Polkadot 网络包含 4 种节点: 1) 验证节点: 抵押 token, 负责对平行链的块进行验证并将块作为候选者

上中继链, 从平行链采集节点获取块, 由提名者选择与支持验证者; 2) 提名节点: 支持并选择验证节点候选人的利益相关者, 用于监管验证节点, 图中虚线连接表示提名节点对验证节点的监督与质押关系; 3) 采集节点: 收集并将平行链数据提交给中继链 (箭头向上指向中继链), 通过观察平行链共识过程, 将最新块签名发送给验证节点. 如果平行链里有数据想要发送给其他平行链, 采集节点会从里面获取出来; 4) 纠错节点: 检查平行链的正确操作, 类似于赏金猎人.

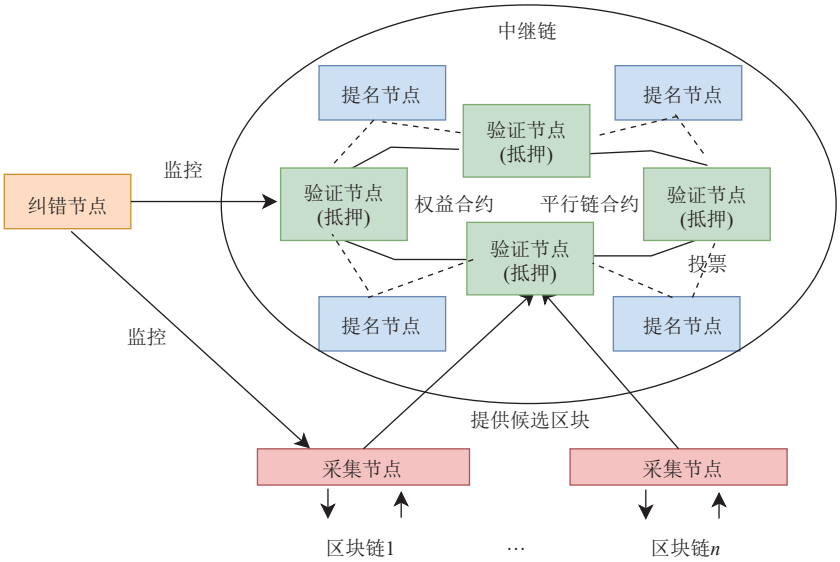

图 9 Polkadot 逻辑架构图

为了支持异构链跨链可信执行, Polkadot 提出采用 BABE+GRANDPA 的共识机制的中继链, 是典型的链下验证可信执行机制, 中继链提供链下通用的区块链共识证明方法, 业务链对中继链验证者身份做验证.

1) BABE: 以概率方式快速出块, 利用 VRF 实现“盲分配”, 保证权益公平且防止预测攻击.

2) GRANDPA: 是 Polkadot 的最终性工具 (finality gadget), 其主要职责是确定性地选择规范链, 即决定哪些区块的更改是最终的. 它的工作原理是让一组验证器投票决定哪个区块应该被认为是最终的. 只有当大多数验证器都同意一个区块是最终的时, 该区块才会被认为是不可变的.

BABE 产块必须始终在 GRANDPA 已确定的链上构建. 当最终确定的头部之后出现分叉时, BABE 通过在具有最多主要区块的链上构建来提供概率确定性.

这是一种兼具概率最终性和可证明最终性优点的方法. 1) 区块的生成与确认分离: BABE 负责区块生成, GRANDPA 负责区块的最终确认. 这种分离使得区块生成可以快速进行, 而确认过程则在单独的进程中完成, 避免了两者相互干扰, 提高了整体效率. 2) 批量确认能力: GRANDPA 能够一次性确认多个区块, 而不是像传统拜占庭容错算法那样每次只确认一个区块. 这大大减少了确认时间, 提高了网络的处理速度. 其次, 安全性好, 1) 权益分配公平性: BABE 基于权益分配区块生产权, 同时引入了随机性, 避免了权益集中化. 这种机制使得更多的节点有机会参与区块生产, 增强了网络的去中心化程度. 2) 支持大规模验证人网络: Polkadot 的提名权益证明 (nominated proof of stake, NPoS) 机制允许构建庞大的验证人网络, 最多可达 1 000 个验证人. 这进一步增强了网络的去中心化和安全性.

Hyperbridge[87–89]是 Polkadot 的跨链互操作协处理器, 保证安全跨链通信: 共识验证、共识容错、状态证明和状态转换. 通过利用 BEEFY (bridge efficiency enabling finality yielder) 提供的经济高效的共识证明, Hyperbridge 确认了网络保护的所有平行链状态转换的合法性. 互操作状态机协议 (interoperable state machine protocol, ISMP), 促进连接链之间的跨链消息传递.

6.2 WeCross

WeCross[90,91]是一款区块链跨链协作平台, 由微众银行区块链团队自主研发并完全开源. WeCross 设计一套通用的区块链数据协议, 抽象提炼主流区块链共通的核心数据结构与资源定义, 使多种区块链平台可以用统一的数据协议交互, 极大程度减小区块链平台之间的交互难度. 同时设计主流区块链平台通用的网络交互协议及统一的交互模式, 通过简便适配, 即可实现异构区块链平台的连通. 为了满足原子事务, 支持两阶段提交协议和哈希时间锁定合约.

针对跨链数据的可信性, WeCross 采用了链上验证可信执行机制的变体. 它不仅传输交易数据, 还传输交易回执的默克尔证明. 接收方利用这些证明进行 SPV 验证, 从而确认交易在源链上的真实性. 这种机制属于前文表 4 中的“弱信任”模型, 依赖于源链的默克尔树根.

为了保障跨链一致性约束, WeCross 事务管理支持前文提到的两阶段提交协议和哈希时间锁定合约, 为了避免第三方依赖, WeCross 事务管理支持用户在多个业务区块链之外可选地搭建一个专门用于协调事务的区块链, 称为治理链. 各个机构中参与事务的跨链路由通过配置区块链适配器连接治理链, 在处理两阶段事务的过程中, 事务的状态都记录在治理链上, 这样恶意的协调者就无法轻易地篡改两阶段事务的状态.

6.3 Cosmos

Cosmos 对跨链部分的设计与 Polkadot 的某些理念相似, 通过一个中间链对跨链信息进行中转, 从而创造一个异构的跨链平台. 在 Cosmos 中提出了两个新的概念 Hub 和 Zone. 如图 10, Hub 是用于处理跨链交互的中继链, Zone 是 Cosmos 中的平行链. 为了支持平行链之间的跨链互操作, Cosmos 提出了一种跨链交互协议 IBC[46], 并利用 tendermint 共识算法的即时确定性实现多个异构链之间的价值和数据传输.

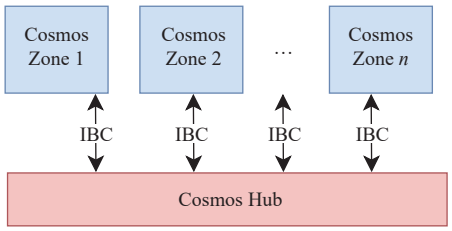

图 10 Cosmos 逻辑架构图

IBC (inter blockchain communication): 是一种端到端的、面向连接的、有状态的协议, 适用于未知和动态拓扑排列的异构区块链之间的可靠、有序和认证通信. IBC 可用于构建广泛的跨链应用程序, 包括令牌传输、原子交换、多链智能合约, 以及各种类型的数据和代码分片.

Hubs: 中心枢纽链, 是专门将区域链连接在一起而设计的区块链, 是跨链消息的信任中心. 中心枢纽链负责追踪记录各个区域链的状态. CosmosHub 是目前 Cosmos 实现的一个官方枢纽链.

Zones: 区域链, 是参与到 Cosmos 网络中的应用链, 可以是互相异构的. 区域链通过跨链协议 IBC 与中心枢纽链进行跨链操作, 每个区域链需要把产生的新区块汇报给中心枢纽链. 当一个区域链与中心枢纽链创建 IBC 连接时, 中心枢纽链可以自动访问连接到它的所有区域链. 因此, 每个区域链只需要与有限个中心枢纽链建立有限的连接. 当一个区域链从中心枢纽接收令牌时, 它只需要信任此令牌的来源区域链和中心枢纽链.

Cosmos 和 Polkadot 之间的区别主要在于网络中子链的主权. Cosmos 中子链在维护事务共识时是自治的, 而 Polkadot 要求子链通过中继链形成全局共识以实现安全共享.

传统的 Tendermint 轻客户端在以太坊等异构链上验证签名的 Gas 开销巨大. 为解决这一挑战, Cosmos 采用基于 ZKP 的链上验证的可信执行机制, 与以太坊之间形成安全快速通道. 以太坊轻客户端作为 CosmWasm 合约被部署在 Cosmos Hub 上, Tendermint 的轻客户端使用 Succinct 的 SP1 zkVM 运行在 Cosmos Hub 上, 并将证明发布到以太坊, 从而实现经济高效的轻客户端验证. 这一改进也验证了前文中提出的“利用 ZK 技术降低链上验证成

本”的技术趋势.

6.4 TCIP

TCIP (trust chain interoperability protocol) 是长安链的跨链协议, 如图 11 是 TCIP 的逻辑架构图. 1) 业务链是指用户正在使用的链, 这个链需要接收来自其他业务链的跨链请求或者需要向其他业务链发起跨链请求; 2) 中继链 (relay-chain) 又名中继器, 是一个第三方链, 通过跨链消息传递协议, 连接区块链网络中的其他链; 3) 中继网关顾名思义, 它与中继链建立了链接, 通过与中继链中的跨链管理合约进行交互, 对跨链操作的各个阶段进行调度, 并提供交易验证功能; 4) 跨链网关: 跨链网关是指与业务链交互的网关, 它通过监听业务链的合约事件, 来判断该事件是否为跨链事件, 若满足触发条件则发起跨链操作; 5) 跨链触发器: 跨链触发器是一种跨链网关如何处理监听到业务链中的合约事件, 如果不满足跨链触发器的判断条件, 那么会将此条合约事件丢弃, 如果满足了跨链触发器的判断条件, 那么跨链网关会根据跨链触发器的配置, 构建跨链操作.

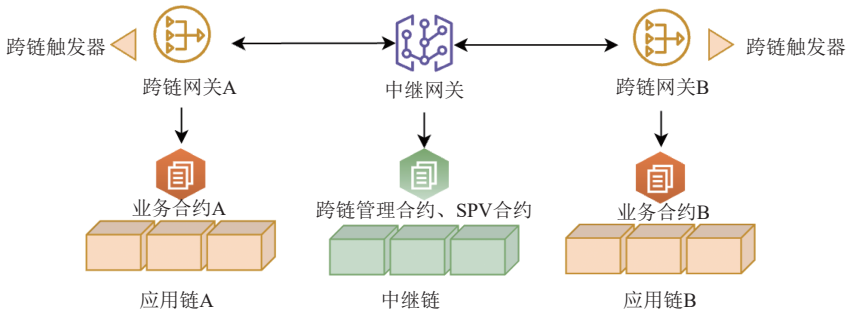

图 11 TCIP 逻辑架构图

TCIP 采用链上验证可信执行机制, 在中继链上部署了 SPV 合约. 跨链网关提交业务链的交易证明到中继链, 中继链通过运行轻客户端逻辑直接验证交易的合法性. TCIP 将跨链事务拆分为 Try (预执行/锁定)、Confirm (提交) 和 Cancel (回滚) 这 3 个阶段. 中继链充当事务协调者, 仅当所有参与链在 Try 阶段均锁定资源成功时, 才触发全局 Confirm, TCIP 通过两阶段提交协议保证事务的原子性.

6.5 MAP Protocol

Cao 等人[31]提出中继链跨链架构 MAP Protocol, 针对以太坊 PoS 多签共识证明的情况, 利用 ZKP 技术将签名验证工作移至链下, 降低链上轻客户端的计算开销, 进而减少跨链验证成本.

图 12 展示了 MAP Protocol 如何利用零知识证明优化跨链验证流程. 传统的跨链验证通常需要在下游链上运行重型轻客户端, 而如图 12 中间部分所示, MAP 引入了“中继链轻客户端”与“链下证明器”的双层结构. 复杂的签名验证计算被从链上 (蓝色方框) 剥离, 转移到了链下 (红色方框) 进行. 链下证明器生成计算完整性的 ZK 证明后, 再提交回链上. 这种“链下计算、链上验证”的数据流向, 直观地诠释了其降低跨链验证 Gas 开销的核心技术路径.

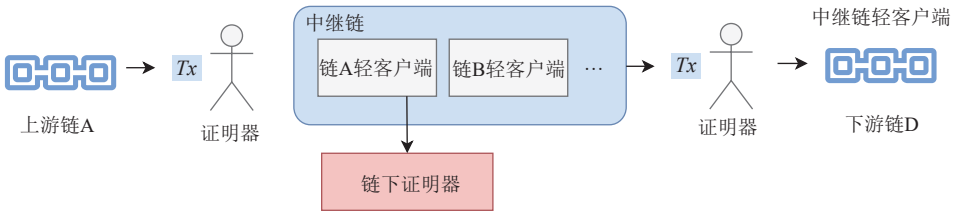

图 12 MAP Protocol 逻辑架构图

6.6 Polygon

Polygon[40]旨在在以太坊上构建一个由 ZK 驱动的、可弹性扩展且统一的 Layer 2 生态系统, 用户可以在其中

创建、编程和交换价值.

AggLayer 是 Polygon 的互操作协议, 跟跨链相关的各个状态以状态树的形式构建一个全局视图. 首先, 连接到 AggLayer 的每条链都会维护一棵本地退出树 (local exit tree), 用于追踪该链上的所有跨链操作; 其次, AggLayer 可以利用每条链的本地退出树的根, 构建一个涵盖统一桥上所有链上所有跨链操作的全局视图; 这被称为“全局退出树 (global exit tree)”.

如图 13, AggLayer 采用链下验证可信执行机制, 在链下验证区块链操作有效性, 生成验证计算的证明, 链上对这个验证工作进行验证. 首先, AggLayer 验证当前全局视图到下一个全局视图的转换正确, 通过 ZKP 证明系统为验证计算生成计算证明. 然后, 下游链能够通过该证明验证计算完整性, 保证跨链执行的可信性.

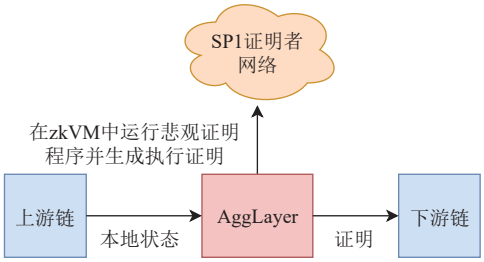

图 13 AggLayer 逻辑架构图

6.7 AntChain Bridge

AntChain Bridge[92]是蚂蚁链推出的通用异构链跨链协议, 旨在构建区块链之间的开源可信通信网络. 在可信执行机制上, 它采用基于委员会的链下验证可信执行机制, 通过引入委员会证明转化组件 (committee proof transform component, Committee PTC) 来提供去中心化的跨链验证与背书能力.

在跨链交互流程中, Committee PTC 的工作机制如下: 每个委员会成员持有独立的私钥, 负责对源链的跨链数据进行验证. 验证通过后, 成员使用私钥对数据进行数字签名并返回背书结果. 跨链客户端负责收集这些签名, 将其聚合成第三方证明 (third party proof, TpProof). 该 TpProof 最终被提交至目标链, 用于验证跨链消息的合法性.

从架构上看, Committee PTC 由以下两类核心角色组成.

验证节点 (Node): 作为服务端响应客户端请求, 负责执行具体的验证与背书逻辑. Node 通过加载异构链数据验证服务 (hetero-chain data verification service, HCDVS) 适配不同区块链的验证规则. 其核心任务包括验证共识状态 (通常是对区块头的抽象) 以及跨链消息证明 (如账本数据的 Merkle proof). 验证通过后, Node 对数据进行签名并返回. 为了保证可发现性, Node 的服务地址会在区块链域名服务 (blockchain domain name service, BCDNS) 中公开注册.

管理者 (Supervisor): 负责管理 Committee PTC 的身份生命周期. Supervisor 持有 PTC 跨链证书对应的管理私钥, 负责维护和更新 BCDNS 上的 PTC 信任根. 信任根中包含了当前活跃 Committee Nodes 的公钥列表及网络连接信息, 确保 AntChain Bridge 网络中的其他参与者 (如目标链网关) 能够安全、便捷地获取信任根, 从而验证由委员会出具的 TpProof 的有效性.

7 挑战与总结

7.1 挑 战

跨链技术已经取得了显著的进步, 但是它仍然是一个充满挑战的研究领域. 本文在此列出若干亟待解决的挑战性问题.

1) 分布式高效跨链并发执行机制

若将单个区块链视为一个数据分片, 跨链技术则相当于在构建一个由多个分片组成的分布式数据库. 然而, 跨

链系统与传统分片分布式数据库[93–96] (通常通过中心化的事务管理层保障一致性) 存在本质差异. 传统分片分布式是自上而下设计的统一系统, 用户通过单一入口提交操作, 事务协调层利用全局性并发控制协议 (如两阶段锁) 有效管理冲突. 与之相对, 跨链系统是自下而上的组合体, 各区块链 (分片) 具有先验的独立存在性与完全自治的写入权限. 这种架构缺乏统一的全局事务入口与协调中心, 每个分片均可独立处理外部写入请求, 天然排斥依赖全局锁的传统方案. 因此, 亟需研究适用于自下而上架构、高效、分布式的跨链事务并发控制方法.

2) 跨链数据语义互操作与统一标准

不同区块链在核心数据结构、智能合约语言、状态机模型以及加密算法等方面存在显著异构性. 这种差异为跨链交互中数据传递 (如资产、事件或状态证明) 的语义理解和一致性保障带来了严峻挑战. 亟需定义并推广数据表示、转换与验证的跨链统一标准, 以实现异构系统间的无歧义数据互操作.

3) 非交互式跨链多用户授权机制

跨链事务通常由跨多个区块链的子操作构成, 且每个链上操作均需经过所属区块链的身份验证 (如用户签名授权). 当一个跨链操作需多个独立用户的授权 (如多方参与的资产交换) 时, 要求所有用户实时在线签名的方案, 用户体验差且实用性低; 而采用预授权委托账户的方式则引入了额外的安全风险. 因此, 亟须设计高效、安全的非交互式授权机制, 使得多个用户的授权承诺在下游链操作的整个生命周期 (从发起至最终执行确认) 内仅需一次签名即可完成绑定.

4) 高性能低成本的跨链可信执行验证技术

基于链上存储的简化支付验证 (SPV) 等技术虽具基础可信性, 但依赖存储大量外部链信息 (如区块头、Merkle 路径) 并需在下游链上重复执行复杂的密码学验证算法, 其链上成本 (Gas/资源消耗) 高昂, 已成为跨链应用规模化的重要瓶颈. 一种极具前景的研究方向是探索链上与链下计算资源紧耦合的可验证计算范式: 将计算密集型的验证任务高效卸载至链下执行, 并设计轻量级的链上验证机制, 确保在极简的链上开销下, 对链下计算结果的可信性与正确性进行高效验证.

5) 跨链与人工智能的深度融合

随着大语言模型 (large language model, LLM) 与生成式 AI 的突破, LLM 能够降低跨链交互门槛, 以及 AI 模型能够在链下监测环节弥补传统形式化验证的不足. 一方面, 通过引入 AI Agent 作为跨链求解器, 用户仅需表达自然语言意图, 智能体即可利用 LLM 强大的推理能力在链下自动规划最优跨链路径、估算 Gas 成本并生成底层 CallData, 从而大幅降低跨链交互门槛, 实现从“手动操作”到“意图直达”的跃迁. 另一方面, 人工智能技术为跨链安全构建了主动防御体系, 利用深度学习与图神经网络对跨链内存池及交易图谱进行实时监控, 不仅能精准识别异常签名聚合或闪电贷攻击前兆, 还能利用针对 Solidity 优化的垂直大模型辅助代码生成与动态审计, 弥补传统形式化验证的不足, 在交易上链前实现毫秒级的风险拦截与逻辑漏洞挖掘.

7.2 总 结

跨链技术是解决当前区块链数据孤岛现状的关键技术, 通过构建链间数据与计算能力的组合范式, 突破单链的协作边界. 跨链事务是维护跨链系统一致性的核心跨链技术, 为构建可扩展的区块链互操作生态提供基础性前提. 本文首先定义跨链系统一致性和跨链事务, 以及总结通用的跨链事务工作流模型. 随后, 本文分别综述、讨论各个关键技术问题的研究现状, 包括跨链事务执行的可信保证及并发执行技术, 分析了跨链事务安全性, 并进一步分析当前具有代表性的跨链事务整体解决方案. 最后, 本文指出了几个值得探索的研究方向. 总体而言, 跨链事务技术研究还在起步阶段, 保证跨链系统一致性尚有许多理论与技术问题亟待解决, 实现可以实际应用的、安全高效的跨链事务则更是充满挑战.

References

[1] Duan TT, Zhang HW, Li B, Song ZX, Li ZC, Zhang J, Sun Y. Survey on blockchain interoperability. Ruan Jian Xue Bao/Journal of Software, 2024, 35(2): 800–827. http://www.jos.org.cn/1000-9825/6950.htm [doi: 10.13328/j.cnki.jos.006950]

[2] Zamyatin A, Al-Bassam M, Zindros D, Kokoris-Kogias E, Moreno-Sanchez P, Kiayias A, Knottenbelt WJ. SoK: Communication across distributed ledgers. In: Proc. of the 25th Int'l Conf. on Financial Cryptography and Data Security. Berlin, Heidelberg: Springer, 2021. 3–36. [doi: 10.1007/978-3-662-64331-0_1]

[3] Augusto A, Belchior R, Correia M, Vasconcelos A, Zhang LY, Hardjono T. SoK: Security and privacy of blockchain interoperability. In: Proc. of the 2024 IEEE Symp. on Security and Privacy (SP). San Francisco: IEEE, 2024. 3840–3865. [doi: 10.1109/SP54263.2024.00255]

[4] Ren KP, Ho NM, Loghin D, Nguyen TT, Ooi BC, Ta QT, Zhu FD. Interoperability in blockchain: A survey. IEEE Trans. on Knowledge and Data Engineering, 2023, 35(12): 12750–12769. [doi: 10.1109/TKDE.2023.3275220]

[5] Robinson P. Survey of crosschain communications protocols. Computer Networks, 2021, 200: 108488. [doi: 10.1016/j.comnet.2021.108488]

[6] Belchior R, Vasconcelos A, Guerreiro S, Correia M. A survey on blockchain interoperability: Past, present, and future trends. ACM Computing Surveys (CSUR), 2022, 54(8): 168. [doi: 10.1145/3471140]

[7] Wang G, Wang Q, Chen SP. Exploring blockchains interoperability: A systematic survey. ACM Computing Surveys, 2023, 55(13s): 290. [doi: 10.1145/3582882]

[8] Falazi G, Breitenbücher U, Leymann F, Schulte S. Cross-chain smart contract invocations: A systematic multi-vocal literature review. ACM Computing Surveys, 2024, 56(6): 142. [doi: 10.1145/3638045]

[9] Lohachab A, Garg S, Kang B, Amin MB, Lee J, Chen SP, Xu XW. Towards interconnected blockchains: A comprehensive review of the role of interoperability among disparate blockchains. ACM Computing Surveys (CSUR), 2022, 54(7): 135. [doi: 10.1145/3460287]

[10] Monika, Bhatia R. Interoperability solutions for blockchain. In: Proc. of the 2020 Int'l Conf. on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE). Bengaluru: IEEE, 2020. 381–385. [doi: 10.1109/ICSTCEE49637.2020.9277054]

[11] Siris VA, Nikander P, Voulgaris S, Fotiou N, Lagutin D, Polyzos GC. Interledger approaches. IEEE Access, 2019, 7: 89948–89966. [doi: 10.1109/ACCESS.2019.2926880]

[12] Schulte S, Sigwart M, Frauenthaler P, Borkowski M. Towards blockchain interoperability. In: Proc. of the 2019 Int'l Conf. on Business Process Management. Vienna: Springer, 2019. 3–10. [doi: 10.1007/978-3-030-30429-4_1]

[13] Qasse IA, Abu Talib M, Nasir Q. Inter blockchain communication: A survey. In: Proc. of the 6th ArabWIC Annual Int'l Conf. Research Track. Rabat: ACM, 2019. 2. [doi: 10.1145/3333165.3333167]

[14] China Academy of Information and Communications Technology. White Paper on Blockchain Technology and Application Development in China. Beijing: China Academy of Information and Communications Technology, 2016.

[15] Polytope Labs. Consensus proofs. 2023. https://research.polytope.technology/consensus-proofs

[16] Nakamoto S. Bitcoin: A peer-to-peer electronic cash system. 2008. http://nil.csail.mit.edu/6.824/2016/papers/bitcoin.pdf

[17] Wikipedia. Merkle tree. 2024. https://en.wikipedia.org/wiki/Merkle_tree

[18] Buterin V, Hernandez D, Kamphefner T, Pham K, Qiao Z, Ryan D, Sin J, Wang Y, Zhang YX. Combining GHOST and casper. arXiv: 2003.03052, 2020.

[19] GitHub. hyperledger/fabric. 2025. https://github.com/hyperledger/fabric

[20] GitHub. decred. 2024. https://github.com/decred

[21] Bonneau J, Meckler I, Rao V, Shapiro E. Mina: Decentralized cryptocurrency at scale. 2020. https://eprint.iacr.org/2020/352

[22] Burdges J, Cevallos A, Czaban P, Habermeier R, Hosseini S, Lama F, Kilinç Alper H, Luo XM, Shirazi F, Stewart A, Wood G. Overview of Polkadot and its design considerations. arXiv:2005.13456, 2020.

[23] Polytope Labs. State (machine) proofs. 2023. https://research.polytope.technology/state-machine-proofs

[24] Wood G. Ethereum: A secure decentralised generalised transaction ledger shanghai version efc5f9a – 2025-02-04. 2025. https://ethereum.github.io/yellowpaper/paper.pdf

[25] Polytope Labs. Polynomial commitments. 2023. https://research.polytope.technology/polynomial-commitments

[26] Sasson EB, Chiesa A, Garman C, Green M, Miers I, Tromer E, Virza M. Zerocash: Decentralized anonymous payments from bitcoin. In: Proc. of the 2014 IEEE Symp. on Security and Privacy. Berkeley: IEEE, 2014. 459–474. [doi: 10.1109/SP.2014.36]

[27] Benveniste A, Caillaud B, Ferrari A, Mangeruca L, Passerone R, Sofronis C. Multiple viewpoint contract-based specification and design. In: Proc. of the 6th Int'l Symp. on Formal Methods for Components and Objects. Amsterdam: Springer, 2007. 200–225. [doi: 10.1007/978-3-540-92188-2_9]

[28] Benveniste A, Caillaud B, Nickovic D, Passerone R, Raclet JB, Reinkemeier P, Sangiovanni-Vincentelli A, Damm W, Henzinger TA,

Larsen KG. Contracts for system design. Foundations and Trends® in Electronic Design Automation, 2018, 12(2–3): 124–400. [doi: 10.1561/1000000053]

[29] Zamyatin A, Harz D, Lind J, Panayiotou P, Gervais A, Knottenbelt W. XCLAIM: Trustless, interoperable, cryptocurrency-backed assets. In: Proc. of the 2019 IEEE Symp. on Security and Privacy (SP). San Francisco: IEEE, 2019. 193–210. [doi: 10.1109/SP.2019.00085]

[30] Bentov I, Ji Y, Zhang F, Breidenbach L, Daian P, Juels A. Tesseract: Real-time cryptocurrency exchange using trusted hardware. In: Proc. of the 2019 ACM SIGSAC Conf. on Computer and Communications Security. London: ACM, 2019. 1521–1538. [doi: 10.1145/3319535.3363221]

[31] Cao YF, Cao JN, Bai DB, Wen L, Liu Y, Li RD. Map the blockchain world: A trustless and scalable blockchain interoperability protocol for cross-chain applications. In: Proc. of the 2025 ACM on Web Conf. Sydney: ACM, 2025. 717–726. [doi: 10.1145/3696410.3714867]

[32] Wormhole. Financial infrastructure for the modern economy. 2025. https://wormhole.com/

[33] Nissl M, Sallinger E, Schulte S, Borkowski M. Towards cross-blockchain smart contracts. In: Proc. of the 2021 IEEE Int'l Conf. on Decentralized Applications and Infrastructures (DAPPS). IEEE, 2021. 85–94. [doi: 10.1109/DAPPS52256.2021.00015]

[34] Hyperlane | open interoperability for any blockchain. 2025. https://hyperlane.xyz/

[35] GitHub. ethereum-optimism/optimistic-specs. 2025. https://github.com/ethereum-optimism/optimistic-specs

[36] Chainlink. Cross-chain interoperability protocol (CCIP). 2025. https://chain.link/cross-chain

[37] Xie TC, Zhang JH, Cheng ZR, Zhang F, Zhang YP, Jia YZ, Boneh D, Song D. zkBridge: Trustless cross-chain bridges made practical. In: Proc. of the 2022 ACM SIGSAC Conf. on Computer and Communications Security. Los Angeles: ACM, 2022. 3003–3017. [doi: 10.1145/3548606.3560652]

[38] Sober M, Scaffino G, Schulte S. Cross-blockchain communication using Oracles with an off-chain aggregation mechanism based on zk-SNARKs. Distributed Ledger Technologies: Research and Practice, 2024, 3(4): 27. [doi: 10.1145/3678187]

[39] Cosmos Network. IBC Eureka technical walkthrough. 2025. https://cosmos.network/blog/ibc-eureka-technical-walkthrough

[40] Polygon. 2025. https://polygon.technology/

[41] Deng QY, Zuo QH, Li ZT, Liu HL, Xie Y. Blockchain-based reputation privacy preserving for quality-aware worker recruitment scheme in MCS. IEEE/ACM Trans. on Networking, 2024, 32(6): 5188–5203. [doi: 10.1109/TNET.2024.3453056]

[42] Zhai SP, Wei J, Yang R, Zhang WX. Reputation management model of relay chain based on PowerTrust algorithm. Application Research of Computers, 2025, 42(10): 2939–2946. [doi: 10.19734/j.issn.1001-3695.2025.03.0069]

[43] Chen K, Lee LF, Chiu W, Su CH, Yeh KH, Chao HC. A trusted reputation management scheme for cross-chain transactions. Sensors, 2023, 23(13): 6033. [doi: 10.3390/s23136033]

[44] Frauenthaler P, Sigwart M, Spanring C, Sober M, Schulte S. ETH relay: A cost-efficient relay for Ethereum-based blockchains. In: Proc. of the 2020 IEEE Int'l Conf. on Blockchain (Blockchain). Rhodes: IEEE, 2020. 204–213. [doi: 10.1109/Blockchain50366.2020.00032]

[45] zkGasm. 2025. https://research.union.build/zkGasm-II-12e79e5af45d8038b075f64da60f4b64

[46] Wei QY, Zhao XF, Zhu XY, Zhang WH. Formal analysis of IBC protocol. In: Proc. of the 31st IEEE Int'l Conf. on Network Protocols (ICNP). Reykjavik: IEEE, 2023. 1–11. [doi: 10.1109/ICNP59255.2023.10355573]

[47] Bhatt BN, Shirazi F, Stewart A. Trustless bridges via random sampling light clients. In: Proc. of the 7th Conf. on Advances in Financial Technologies (AFT 2025). Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2025. 31: 1–31:24.

[48] Etherscan. Ethereum mainnet transactions of the snowbridge BEEFY light client. 2025. https://etherscan.io/.

[49] Polygon. Polygon CDK: Private blockchains with public liquidity. 2025. https://docs.polygon.technology/zkEVM/

[50] Bernstein PA, Hadzilacos V, Goodman N. Concurrency Control and Recovery in Database Systems. Reading: Addison-Wesley, 1987.

[51] Falazi G, Breitenbücher U, Leymann F, Schulte S, Yussupov V. Transactional cross-chain smart contract invocations. Distributed Ledger Technologies: Research and Practice, 2025, 4(2): 17. [doi: 10.1145/3616023]

[52] ChainBridge-TCIP relay cross-chain solution. 2023 (in Chinese). https://docs.chainmaker.org.cn/tech/TCIP%E4%B8%AD%E7%BB%A7%E8%B7%A8%E9%93%BE%E6%96%B9%E6%A1%88.html

[53] Duan TT, Guo Y, Li B, Zhang HW, Song ZX, Li ZC, Zhang J, Sun Y. PieBridge: An on-demand scalable cross-chain architecture. Journal of Computer Research and Development, 2023, 60(11): 2520–2533. [doi: 10.7544/issn1000-1239.202230284]

[54] GitHub. bitcoin/bips. 2024. https://github.com/bitcoin/bips/blob/master/bip-0199.mediawiki

[55] Interledger. Interledger protocol V4. 2024. https://interledger.org/developers/rfcs/interledger-protocol/

[56] Winzer F, Herd B, Faust S. Temporary censorship attacks in the presence of rational miners. In: Proc. of the 2019 IEEE European Symp. on Security and Privacy Workshops (EuroS&PW). Stockholm: IEEE, 2019. 357–366. [doi: 10.1109/EuroSPW.2019.00046]

[57] Poon J, Dryja T. The bitcoin lightning network: Scalable off-chain instant payments. 2016. https://lightning.network/lightning-network-paper.pdf

[58] Harris J, Zohar A. Flood & loot: A systemic attack on the lightning network. In: Proc. of the 2nd ACM Conf. on Advances in Financial Technologies. New York: ACM, 2020. 202–213. [doi: 10.1145/3419614.3423248]

[59] Tsabary I, Yechieli M, Manuskin A, Eyal I. MAD-HTLC: Because HTLC is crazy-cheap to attack. In: Proc. of the 2021 IEEE Symp. on Security and Privacy (SP). San Francisco: IEEE, 2021. 1230–1248. [doi: 10.1109/SP40001.2021.00080]

[60] Ren YJ, Lv ZY, Xiong NN, Wang J. HCNCT: A cross-chain interaction scheme for the blockchain-based metaverse. ACM Trans. on Multimedia Computing, Communications, and Applications, 2024, 20(7): 188. [doi: 10.1145/3594542]

[61] Zakhary V, Agrawal D, El Abbadi A. Atomic commitment across blockchains. Proc. of the VLDB Endowment, 2020, 13(9): 1319–1331. [doi: 10.14778/3397230.3397231]

[62] Bitcoin Core. The first successful zero-knowledge contingent payment. 2016. https://bitcoincore.org/en/2016/02/26/zero-knowledge-contingent-payments-announcement/

[63] Zie JY, Deneuville JC, Briffaut J, Nguyen B. Extending atomic cross-chain swaps. In: Proc. of the 2019 Int'l Workshop on Data Privacy Management. Luxembourg: Springer, 2019. 219–229. [doi: 10.1007/978-3-030-31500-9_14]

[64] Aragon. Build something worth owning. 2025. https://www.aragon.org/

[65] Gray JN, Lorie RA, Putzolu GR, Traiger IL. Granularity of locks and degrees of consistency in a shared data base. In: Stonebraker M, Hellerstein JM, eds. Readings in Database Systems. San Francisco: Morgan Kaufmann Publishers Inc., 1998. 175–193.

[66] Berenson H, Bernstein P, Gray J, Melton J, O'Neil E, O'Neil P. A critique of ANSI SQL isolation levels. ACM SIGMOD Record, 1995, 24(2): 1–10. [doi: 10.1145/568271.223785]

[67] Adya A. Weak consistency: A generalized theory and optimistic implementations for distributed transactions. Cambridge: Massachusetts Institute of Technology, 1999.

[68] Ports DRK, Grittner K. Serializable snapshot isolation in PostgreSQL. Proc. of the VLDB Endowment, 2012, 5(12): 1850–1861. [doi: 10.14778/2367502.2367523]

[69] Yang ZK, Yang CH, Han FS, Zhuang MQ, Yang B, Yang ZF, Cheng XJ, Zhao YZ, Shi WH, Xi HF, Huang Y, Liu B, Pan Y, Yin BX, Chen JQ, Xu QQ. OceanBase: A 707 million tpmC distributed relational database system. Proc. of the VLDB Endowment, 2022, 15(12): 3385–3397. [doi: 10.14778/3554821.3554830]

[70] Lee SS, Murashkin A, Derka M, Gorzny J. SoK: Not quite water under the bridge: Review of cross-chain bridge hacks. In: Proc. of the 2023 IEEE Int'l Conf. on Blockchain and Cryptocurrency (ICBC). Dubai: IEEE, 2023. 1–14. [doi: 10.1109/ICBC56567.2023.10174993]

[71] GitHub. ArbitrumFoundation/governance. 2023. https://github.com/ArbitrumFoundation/governance/blob/main/audits/trail_of_bits_governance_report_1_6_2023.pdf

[72] Belchior R, Somogyvari P, Pfannschmidt J, Vasconcelos A, Correia M. Hephaestus: Modeling, analysis, and performance evaluation of cross-chain transactions. IEEE Trans. on Reliability, 2024, 73(2): 1132–1146. [doi: 10.1109/TR.2023.3336246]

[73] Zhang JS, Gao JB, Li Y, Chen ZM, Guan Z, Chen Z. Xscope: Hunting for cross-chain bridge attacks. In: Proc. of the 37th IEEE/ACM Int'l Conf. on Automated Software Engineering. Rochester: ACM, 2022. 171. [doi: 10.1145/3551349.3559520]

[74] Ronin. Ronin bridge. https://bridge.roninchain.com/

[75] Elliptic. The harmony horizon bridge hack. 2025. https://www.elliptic.co/resources/harmony-horizon-bridge-hack

[76] zkrouter. 2025. https://drive.google.com/file/d/1ibuHChcYcYCN6JelRAQPnM4rkaB9EgAM

[77] samczsun. 2025. https://x.com/samczsun/status/1578167198203289600

[78] Multichain contract vulnerability post mortem | by multichain (previously anyswap) | medium. 2025. https://medium.com/multichainorg/multichain-contract-vulnerability-post-mortem-d37bfab237c8

[79] Herlihy M. Atomic cross-chain swaps. In: Proc. of the 2018 ACM Symp. on Principles of Distributed Computing. Egham: ACM, 2018. 245–254. [doi: 10.1145/3212734.3212736]

[80] Shlomovits O, Leiba O. JugglingSwap: Scriptless atomic cross-chain swaps. arXiv:2007.14423, 2020.

[81] Wang G, Nixon M. InterTrust: Towards an efficient blockchain interoperability architecture with trusted services. In: Proc. of the 2021 IEEE Int'l Conf. on Blockchain (Blockchain). Melbourne: IEEE, 2021. 150–159. [doi: 10.1109/Blockchain53845.2021.00029]

[82] Xue YJ, Herlihy M. Hedging against sore loser attacks in cross-chain transactions. In: Proc. of the 2021 ACM Symp. on Principles of Distributed Computing. ACM, 2021. 155–164. [doi: 10.1145/3465084.3467904]

[83] Mazumdar S. Towards faster settlement in HTLC-based cross-chain atomic swaps. In: Proc. of the 4th IEEE Int'l Conf. on Trust, Privacy

and Security in Intelligent Systems, and Applications (TPS-ISA). Atlanta: IEEE, 2022. 295–304. [doi: 10.1109/TPS-ISA56441.2022.00043]

[84] Manevich Y, Akavia A. Cross chain atomic swaps in the absence of time via attribute verifiable timed commitments. In: Proc. of the 7th IEEE European Symp. on Security and Privacy (EuroS&P). Genoa: IEEE, 2022. 606–625. [doi: 10.1109/EuroSP53844.2022.00044]

[85] Eizinger T, Hoenisch P, del Pino LS. Open problems in cross-chain protocols. arXiv:2101.12412, 2021.

[86] Lightning. 2020. https://diyhpl.us/~bryan/irc/bitcoin/bitcoin-dev/linuxfoundation-pipermail/lightning-dev/2020-October/002857.txt

[87] Web3 Foundation. Hyperbridge. 2024. https://wiki.polkadot.network/docs/learn-hyperbridge

[88] GitHub. polytope-labs/hyperbridge. 2025. https://github.com/polytope-labs/hyperbridge

[89] Hyperbridge. Interoperability done right. 2023. https://hyperbridge.network/

[90] WeCross technology document. 2025 (in Chinese). https://wecross.readthedocs.io/zh-cn/latest/index.html

[91] Blockchain Team of WeBank. WeCross: A technical whitepaper on blockchain cross-chain collaboration platform. 2020 (in Chinese). https://docs.huihoo.com/blockchain/%E5%BE%AE%E4%BC%97%E9%93%B6%E8%A1%8C%E5%8C%BA%E5%9D%97%E9%93%BEWeCross%E8%B7%A8%E9%93%BE%E6%8A%80%E6%9C%AF%E7%99%BD%E7%9A%AE%E4%B9%A6.pdf

[92] GitHub. AntChainBridge. 2025. https://github.com/AntChainOpenLabs/AntChainBridge/tree/main

[93] Corbett JC, Dean J, Epstein M, *et al*. Spanner: Google's globally distributed database. ACM Trans. on Computer Systems (TOCS), 2013, 31(3): 8. [doi: 10.1145/2491245]

[94] Chandra TD, Griesemer R, Redstone J. Paxos made live: An engineering perspective. In: Proc. of the 26th Annual ACM Symp. on Principles of Distributed Computing. Portland: ACM, 2007. 398–407. [doi: 10.1145/1281100.1281103]

[95] Taft R, Sharif I, Matei A, VanBenschoten N, Lewis J, Grieger T, Niemi K, Woods A, Birzin A, Poss R, Bardea P, Ranade A, Darnell B, Gruneir B, Jaffray J, Zhang L, Mattis P. CockroachDB: The resilient geo-distributed SQL database. In: Proc. of the 2020 ACM SIGMOD Int'l Conf. on Management of Data. Portland: ACM, 2020. 1493–1509. [doi: 10.1145/3318464.3386134]

[96] Huang DX, Liu Q, Cui Q, Fang ZH, Ma XY, Xu F, Shen L, Tang L, Zhou YX, Huang ML, Wei W, Liu C, Zhang J, Li JJ, Wu XL, Song LY, Sun RX, Yu SP, Zhao L, Cameron N, Pei LQ, Tang X. TiDB: A Raft-based HTAP database. Proc. of the VLDB Endowment, 2020, 13(12): 3072–3084. [doi: 10.14778/3415478.3415535]

[97] Haddaway NR, Page MJ, Pritchard CC, McGuinness LA. PRISMA2020: An R package and Shiny APP for producing PRISMA 2020-compliant flow diagrams, with interactivity for optimised digital transparency and Open Synthesis. Campbell Systematic Reviews, 2022, 18(2): e1230. [doi: 10.1002/cl2.1230]

附中文参考文献

[1] 段田田, 张瀚文, 李博, 宋兆雄, 李忠诚, 张珺, 孙毅. 区块链互操作技术综述. 软件学报, 2024, 35(2): 800–827. http://www.jos.org.cn/1000-9825/6950.htm [doi: 10.13328/j.cnki.jos.006950]

[14] 中国信息通信研究院. 中国区块链技术和应用发展白皮书. 北京: 中国信息通信研究院, 2016.

[42] 翟社平, 魏杰, 杨锐, 张卫星. 基于 PowerTrust 算法的中继链信誉管理模型. 计算机应用研究, 2025, 42(10): 2939–2946. [doi: 10.19734/j.issn.1001-3695.2025.03.0069]

[52] ChainBridge-TCIP 中继跨链方案. 2023. https://docs.chainmaker.org.cn/tech/TCIP%E4%B8%AD%E7%BB%A7%E8%B7%A8%E9%93%BE%E6%96%B9%E6%A1%88.html

[53] 段田田, 郭仪, 李博, 张瀚文, 宋兆雄, 李忠诚, 张珺, 孙毅. PieBridge: 一种按需可扩展的跨链架构. 计算机研究与发展, 2023, 60(11): 2520–2533. [doi: 10.7544/issn1000-1239.202230284]

[90] WeCross 技术文档. 2025. https://wecross.readthedocs.io/zh-cn/latest/index.html

[91] 微众银行区块链团队. WeCross 技术白皮书: 区块链跨链协作平台. 2020. https://docs.huihoo.com/blockchain/%E5%BE%AE%E4%BC%97%E9%93%B6%E8%A1%8C%E5%8C%BA%E5%9D%97%E9%93%BEWeCross%E8%B7%A8%E9%93%BE%E6%8A%80%E6%9C%AF%E7%99%BD%E7%9A%AE%E4%B9%A6.pdf

附录 A

本节将详细介绍系统调查方法, 图 A1 展示了筛选流程图 PRISMA, 文献除了已发表的正式文献 (例如期刊和

会议论文) 外, 还包括灰色文献 (例如博客文章、视频和白皮书). 选择了 6 个声誉良好的正式文献来源: 1) ACM Digital Library, 2) IEEE Xplore, 3) SpringerLink, 4) ScienceDirect, 5) Google Scholar 和 6) 知网. 也选取 Google Search 作为灰色文献的来源. 检索日期是 2025-08-17, 筛选条件: 1) 与跨链相关的, 搜索关键词为 (“inter?blockchain” | “cross?blockchain” | “multi?blockchain” | “inter?chain” | “cross?chain” | “multi?chain” | “cross?ledger” | “multi?ledger” | “inter?ledger” | “blockchain?interoperability” | “ledger?interoperability” | “chain?interoperability”), 2) 日期是 5 年内的. 具体筛选流程见图 A1.

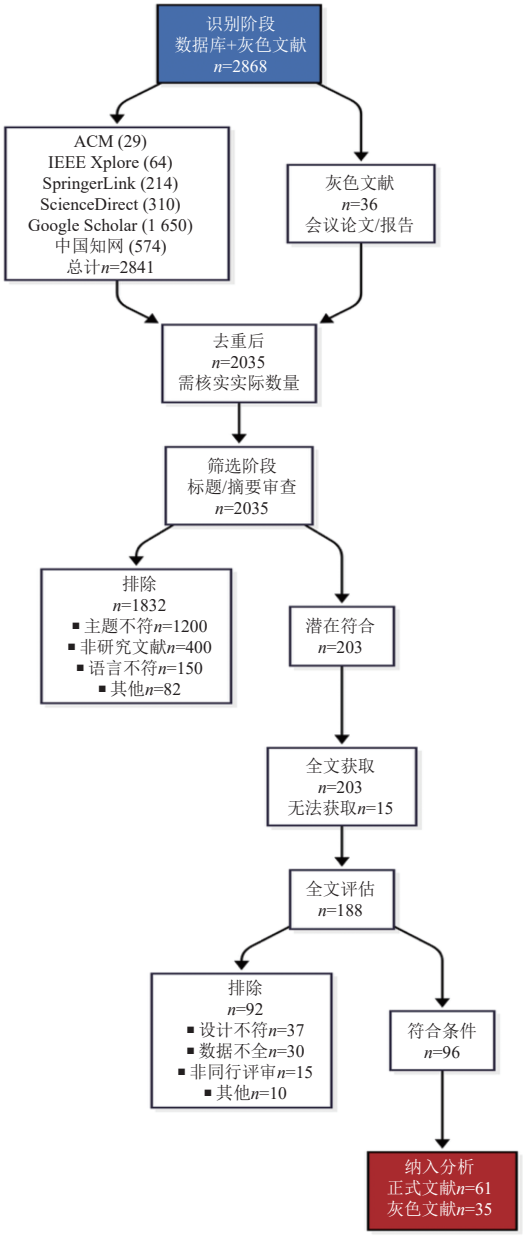

图 A1　筛选流程图 PRISMA[97]

作者简介

黄洁华, 博士生, CCF 学生会员, 主要研究领域为区块链, 智能合约, 跨链.
胡凯, 博士, 教授, 博士生导师, 主要研究领域为区块链, 分布式系统, 形式化方法.
夏鹏凯, 博士生, 主要研究领域为区块链, 隐私保护.
李洁, 博士生, 主要研究领域为形式化方法, 区块链, 软件工程.
李江, 博士生, 主要研究领域为神经网络解释的形式化方法.
杨鹏武, 硕士生, 主要研究领域为区块链, 数据跨链, 隐私保护.
董林霖, 硕士生, 助理工程师, 主要研究领域为区块链, 数据跨链.