软件学报 ISSN 1000-9825, CODEN RUXUEW
Journal of Software,2020,31(8):2557−2573 [doi: 10.13328/j.cnki.jos.006020]

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

基于双层协同的联盟区块链隐私数据保护方法*

蔡 亮[1], 端 豪[1], 鄢 萌[2], 夏 鑫[3]

[1](浙江大学 计算机科学与技术学院,浙江 杭州 310007)

[2](重庆大学 大数据与软件学院,重庆 401331)

[3](Faculty of Information Technology, Monash University, Melbourne, VIC 3800, Australia)

通讯作者: 鄢萌, E-mail: mengy@cqu.edu.cn

摘 要: 为了解决联盟区块链平台中的隐私保护问题,提出了一种基于双层协同的隐私数据保护方法,包括: (1) 链间隐私保护:通过将不同业务的数据进行分流处理、分区存储,实现了不同业务之间的隐私机密性保护; (2) 链内隐私保护:通过在交易体中嵌入字段来指定链内隐私数据的参与方,并由接收交易的区块链节点作为中转节点进行链内隐私数据的同步,中转节点同时负责将隐私数据替换成其哈希值后,构造公开交易进行正常公开交易的上链,待公开交易上链成功后,由隐私参与方节点各自进行隐私账本的更新.为了验证该方法的有效性,分别对链间隐私方法吞吐量以及链内隐私保护方法的延迟性进行了测试与对比,结果表明,通过结合粗粒度的链间隐私保护与细粒度的链内隐私保护,在满足了隐私需求的同时,也保证了可观的性能,为区块链平台的隐私性与安全性做出了贡献.

关键词: 区块链;分区共识;隐私保护

中图法分类号: TP311

中文引用格式: 蔡亮,端豪,鄢萌,夏鑫.基于双层协同的联盟区块链隐私数据保护方法.软件学报,2020,31(8):2557−2573. http://www. jos.org.cn/1000-9825/6020.htm

英文引用格式: Cai L, Duan H, Yan M, Xia X. Private data protection scheme for consortium blockchain based on two-layer cooperation. Ruan Jian Xue Bao/Journal of Software, 2020,31(8):2557−2573 (in Chinese). http://www.jos.org.cn/1000-9825/6020.htm

Private Data Protection Scheme for Consortium Blockchain Based on Two-layer Cooperation

CAI Liang[1], Duan Hao[1], YAN Meng[2], XIA Xin[3]

[1](College of Computer Science and Technology, Zhejiang University, Hangzhou310007, China)

[2](School of Big Data and Software Engineering, Chongqing University, Chongqing 401331, China)

[3](Faculty of Information Technology, Monash University, Melbourne, VIC 3800, Australia)

Abstract: This study proposes a two-layer collaborative approach for privacy protection in consortium blockchain. The proposed approach consists of two layers. 1. Inter-chain privacy protection: such protection approach aims to protect the privacy between different businesses. This approach is realized by separating and storing the data of different businesses. 2. Intra-chain privacy protection: such protection approach is realized by embedding collection field into transaction body to specify the participants of privacy protection. Then, this approach regards the receiving blockchain node as a relay node to synchronize private data. At the same time, the relay node is also responsible to construct a public transaction by replacing the private payload with its hash after which the public transaction will be synchronized to all participants through consensus. Finally, private participant nodes update their private ledgers to achieve intra-chain privacy. To verify the validity of the proposed approach, some experiments are designed to measure the throughput of inter-chain privacy protection and the delay of intra-chain privacy protection. The experimental results show that the privacy protection approach which is combined with coarse-grained inter-chain protection and fine-grained intra-chain protection ensures the considerable performance and

* 收稿时间: 2019-10-28; 修改时间: 2019-12-30, 2020-02-07; 采用时间: 2020-02-20; jos 在线出版时间: 2020-05-26

satisfies the privacy requirements at the same time. Thus, the proposed approach has made a potential contribution to the privacy and security of blockchain platform.

Key words: blockchain; namespace; privacy protection

2008 年 11 月,化名“中本聪”的作者发表了一篇名为《比特币:一种点对点的电子现金系统》[1]的比特币白皮书,详细地论述了如何创建一套去中心化的电子现金系统.2009 年 1 月,中本聪创建了比特币世界的第 1 个区块——创世区块,标志着比特币交易系统的诞生.而区块链技术作为比特币的底层技术,由于其去中心化、公开透明性、防篡改等特性,渐渐受到人们的重视.直到 2013 年 12 月,试图挖掘区块链技术更深层次意义的 Vitalik Buterin 发布了以太坊白皮书《下一代智能合约和去中心化应用平台》[2],标志着区块链 2.0 时代的到来.以太坊开创性地将智能合约[3]和区块链技术结合起来[4],将区块链技术从简单的转账交易场景迁移到了灵活多变的合约交易场景.为了解决企业级区块链的需求,2015 年 12 月,由 Linux 基金会牵头,联合 30 家初始企业成员,共同宣布了超级账本(hyperledger)联合项目的成立.超级账本项目致力为透明、公开、去中心化的企业级分布式账本技术提供开源参考实现,并推动区块链和分布式账本相关协议、规范和标准的发展.Hyperledger Fabric 项目[5]是首个面向企业的开放区块链技术的重要探索,其具备严格的身份识别和权限控制,并设计了可插拔的高效共识算法和数据存储设计,支持多种编程语言的智能合约,同时实现了联盟链上的隐私保护,预示着区块链 3.0 的到来.

经过 10 年左右的演进,区块链技术本身已经得到了长足的发展[6−11],但是风光背后却存在着不少的问题.区块链的安全性与隐私性越来越多地受到黑客攻击的挑战[12−19],这不仅仅与用户本身对数字资产的管理不当有关,还与区块链底层协议设计的不完善有关.

1 相关工作

1.1 比特币中的隐私保护

比特币网络有如下 3 个重要的特点.

(1) 公开可访问性:所有交易记录全网公开,这是为了防止交易双花的必然结果.

(2) 交易之间有串联关系:每一笔交易都包含若干的输入与输出,而每一个输入又是之前某笔交易的输出或者是挖矿奖励.通常来说,一笔交易的输入都是由一个余额较大的 UTXO 或者若干个余额较小的 UTXO 组成,而一笔交易的输出通常包含两个:一个是发向收款方的金额,一个是返回给付款方的余额(change).

(3) 公钥地址的重用:收款方与付款方身份通过公钥地址来标识.事实上,对于收款方来说,最好的方式就是为每一笔接收的交易都生成一个新的公钥地址.

鉴于上述 3 大特征,Reid 等人[20]通过分析 2009 年 1 月 3 日到 2011 年 7 月 12 日之间的交易历史,总结出了比特币网络中的两大网络结构模型——交易网络模型与用户网络模型,并指出了比特币潜在的匿名性威胁.

为了解决上述的匿名泄漏[21]问题,许多提升比特币匿名性的提案被提出[22,23],许多类比特币的新币种也应运而生[24],其中较为著名的有:

(1) 比特币隐私提案 CoinJoin[25,26]:由于比特币交易由多个输入以及多个输出构成,因此可以将多笔交易合并以构造成一笔大的交易(混币).具体实现过程中,可以通过一个交易池将同时发生的交易汇聚到一起构造成一笔大额交易,这样对于外部观察人员来说,交易里的个别资金走向与发送方、接收方的地址是没有明显的关联性的.

(2) 比特币隐私提案 TumbleBit[27,28]:与 CoinJoin 不同,在 TumbleBit 中,用户可以通过 tumble 混合器进行交易的发送,同时,任何人,包括 tumble 本身,都不可能获知交易的发起方与接收方的任何信息,避免了交易双方信息被泄漏的风险.

(3) 门罗币 Monero[29−31]:Monero 基于 CryptoNote[32]协议,并在区块链模糊化方面有显著的算法差异,其通

过使用环签名机制实现了交易发送方的匿名性,通过隐形地址技术实现了交易接收方的匿名性,并通过环匿名交易实现了交易本身的匿名性.但是过多的加密操作的结果就是门罗币的交易体比其他币种要大得多,且其本身的兼容性也无法得到保障.

上述提出诸多匿名方案都旨在保护转账交易类型的匿名性,但是匿名并不等于隐私,尤其是在联盟链场景下,节点通常与机构挂钩,而机构的身份信息往往又与其权限挂钩,因此在联盟链中,身份信息反而不能够进行匿名化(尤其是在能源与健康医疗领域[33,34]).如何在参与方不匿名的前提下保护数据的隐私,是本文工作的重点与难点所在.

1.2 以太坊中的隐私保护

为了实现合约级别的隐私保护,以太坊客户端 Parity 实现了一套隐私合约的特性,允许用户通过 Parity 客户端在以太坊区块链上存储、修改与查询加密数据.每一个隐私合约对应产生一个公开合约,隐私合约以加密方式存储在对应的公开合约中,而公开合约将被部署到所有区块链节点中;同时,对合约的访问者进行严格的权限控制,只有合约部署时指定的访问才能获取到解密公开合约中的隐私合约的密钥,由此来实现合约的隐私性.目前,隐私保护在 Parity 客户端中还处于开发阶段,因此还存在一些限制,包括:

(1) 由于目前暂未实现隐私状态的状态缓存,因此当前每个区块每个合约只能调用一次隐私交易请求.

(2) 当前实现版本中,必须要收集齐所有验证者节点的签名才能验证通过隐私交易,暂不支持验证者集合的可配置.

1.3 Fabric中的隐私保护

Hyperledger Fabric 是企业级开源联盟区块链的代表,一直注重于企业级的权限控制体系的实现,这其中就包括企业间的隐私保护.Fabric1.0 自其 1.2 版本[35,36]开始支持隐私交易(sideDB),相应的,隐私数据应该保证对非隐私参与方的背书节点、排序节点以及记账节点都是不可见的.Fabric 中,隐私交易保护的流程主要分为两个阶段:背书阶段与提交阶段.在背书阶段,客户端构造隐私交易广播给授权的背书节点,授权的背书节点首先将隐私交易缓存至本地临时数据库,随后通过 gossip 协议进行隐私交易数据的传播,将其同步至其他的授权节点(包括背书节点以及记账节点)中,最后授权的背书节点将隐私数据 key-value 对的哈希值返回给客户端.在提交阶段,客户端将交易提交到排序服务中(此时,该交易中仅包含隐私数据的哈希值)进行正常的上链流程,包含该笔公开交易的区块最终同步到了所有节点上,在记账节点进行区块提交的时候,授权节点会根据授权策略判断自身是否有访问隐私数据的权限:如果有的话,则首先检查本地的临时数据库中是否存在对应的隐私交易;如果不存在的话,需要先向其他授权参与方拉取隐私数据,随后,授权节点根据公开交易中的哈希值来校验隐私数据的完整性.最后,当节点进行账本提交的时候,将隐私交易数据从临时数据库中移除并持久化到隐私账本中.

上述方法摒弃了通过加解密方式来保护隐私数据的方案,并在其原先成熟的权限控制体系下实现了一套相对较为完备的隐私保护方案,但是依旧存在如下两点问题:一是隐私交易数据需要通过 gossip 协议进行传播,很容易导致授权节点在达到最后的执行环节时缺失隐私交易,需要通过额外的拉取流程获取隐私交易数据;二是隐私状态的读写一致性并未得到保证,即 Fabric 并未保证不同节点存储的隐私状态是否一致.

1.4 本文工作

为了解决当前区块链领域内隐私保护技术方法的诸多不足,本文首次提出了一套完善的、以数据隔离为主要思想的、适用于联盟链环境的双层协同隐私保护方法,该方法分两个层次进行隐私数据的保护.

(1) 链间隐私保护:实现了业务级别、粗粒度的隐私保护.

(2) 链内隐私保护:实现了交易级别、合约级别、细粒度的隐私保护.

2 链间隐私保护方法

为了解决上述联盟链内的隐私保护效率低下、权限控制复杂的问题,本文提出了以数据隔离为主要思想的链间隐私保护方法.该方法通过对节点的不同业务流进行分区隔离处理,实现了业务级别的隐私保护,省去了耗

时的数据加解密流程,并通过分区管理器 NSM(namespace manager)实现了严密的分区权限管理.为了与不同的参与方构建多条互不交互的业务流系统,传统的联盟链解决方法是为不同的业务分别构建不同的区块链网络,搭建各自的区块链应用.而分区隔离方法的提出,大大提升了多业务流并行处理的速度,大大降低了多业务流场景的部署难度,提升了区块链平台的可扩展性.

2.1 架构设计

下图1给出了链间隐私保护的多节点多分区集群架构图,本文将一个业务流网络称为一个分区(namespace,简称 NS),每个分区都由一组可变的分区参与方节点(namespace participants,简称 NSP)共同维护,节点每参与一个新的业务,都需要开辟一个新的分区,所有节点在初次启动时默认参与全局分区(global NS),以便进行全节点的管理以及区块链功能的扩展.如图 1 所示(未标识出 global 分区)为多分区的节点集群架构图,其中,节点 1~节点 4 参与到了分区 1 中,而节点 3~节点 6 参与到了分区 2 中.相应的,节点 1~节点 4 之间共享账本 1 的数据,而节点 3~节点 6 共享账本 2 的数据.需要注意的是,节点 3 和节点 4 虽然都参与到了两个业务分区中且都需要维护两份账本数据,但是这两份账本数据是隔离存储的,即不同分区之间的数据是互不交互的.

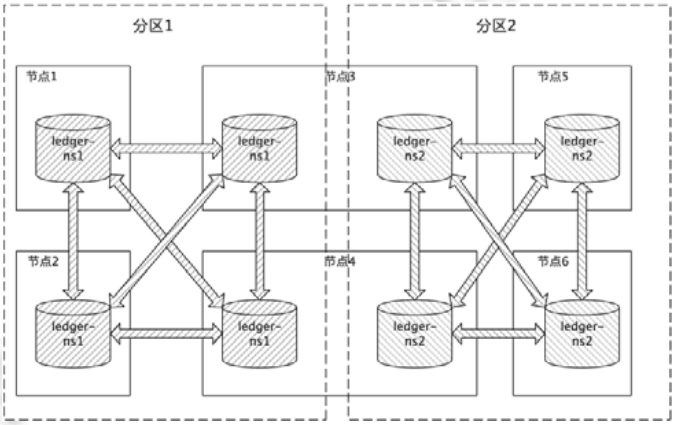

Fig.1 Architecture of multi-node multi-partition cluster

图 1 多节点多分区集群架构

2.2 分区管理器NSM

分区管理器 NSM 是链间隐私保护的核心,保证了不同分区间数据的分流执行与隔离存储,实现了单节点多业务的并行处理.图 2 展示了单节点内部的模块架构图.

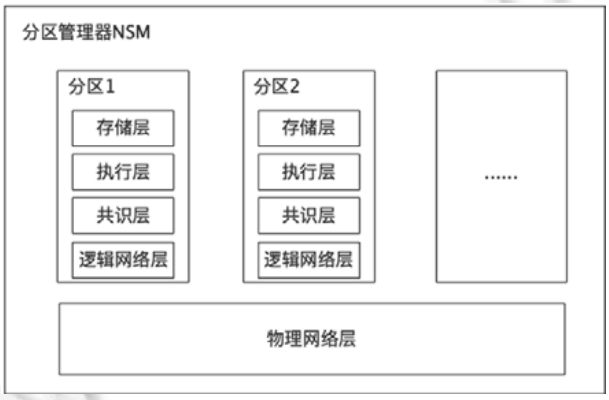

Fig.2 Architecture of single-node multi-partition module

图 2 单节点多分区模块架构

一个区块链节点内部通常由网络层、共识层、执行层与存储层构成,在引入 NSM 之前,由于单节点只需要处理一个业务分区的交易请求,因此各层之间相互耦合度较高也不影响交易的分发与处理;而在引入 NSM 之后,为了实现交易的分流处理,需要将各个模块进行拆分解耦,将主要模块由节点级别转化为分区级别.一个分区可以理解为一个虚拟的区块链网络,因此一个分区由上述 4 个模块构成,所有分区可以单独的处理各自分区内部的交易请求,有各自的交易定序机制和执行引擎,并统一的由 NSM 进行交易请求的分发.同时,为了提高底层网络的复用,不同分区之间共用同一套物理网络.

由于所有分区的物理网络层是共用的,因此网络接口层需要承担起交易分流的作用,客户端在向节点发送交易请求时,需要附带上该交易所处的分区的唯一 ID(NS_ID),接口层通过解析请求中的 NS_ID 将交易转发给 NSM,由 NSM 进行交易的分发.

2.3 分区生命周期管理

有些业务分区存在周期较长,而有些业务分区的存在周期较短,因此,一个完善的分区生命周期管理是链间隐私保护必备的需求,一个分区的生命周期包括:分区的注册、启动、停止与注销.其中,分区的注册与注销仅能由节点的管理员通过调用外部接口触发,而节点的启动与停止则可以由两种方式触发:一是节点管理员通过调用外部接口触发,二是通过节点内部状态进行自动触发.例如,节点 LICENSE 过期时自动停止分区服务,而在替换新的有效 LICENSE 之后可自行启动相应分区.

2.4 系统复杂度及性能分析

链间隐私保护的主要思想在于对用户的请求进行分区处理.由于所有分区的物理网络层是共用的,因此网络接口层需要承担起交易分流的作用,客户端在向节点发送交易请求时,需要附带上该交易所处的分区的唯一 ID(NS_ID),接口层通过解析请求中的 NS_ID 将交易转发给 NSM,由 NSM 进行交易的分发.图 3 展示了多分区交易请求的处理流程,客户端向节点发送了 3 笔交易,分别是分区 1 内的请求 3(点线表示)、分区 2 内的请求 2(虚线表示)、分区 3 内的请求 1(实线表示).节点在接收到交易请求后,需要在 RPC 接口层进行一次交易解析,读取出交易所属的 NS_ID,并将交易 ID 连同交易本身一起转发到 NSM 分区管理器中;随后,NSM 根据给定的 NS_ID 将交易请求分发到相应分区的处理模块中,实现交易的分流处理.

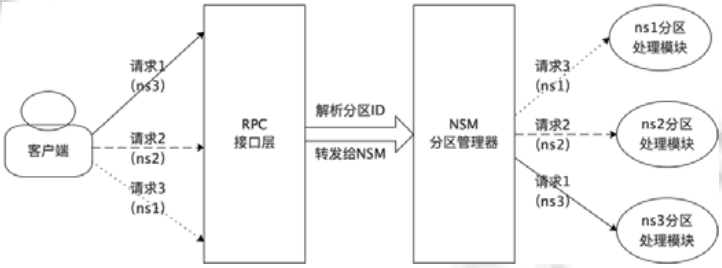

Fig.3 Shunt processing flow of multi-partition transaction

图 3 多分区交易分流处理流程

从图 3 可以看出,分区共识的交易处理流程相较于传统的交易处理流程多了两个步骤:一是 RPC 接口层解析分区 ID,二是分区管理器 NSM 根据分区 ID 进行交易的分流.在 NSM 将交易分流之后,分区内部的处理逻辑与原有的解决方案一致,不做更改.由于这两步操作都是可以忽略时间差的,因此整个分区共识的交易处理流程的时间复杂度与传统的交易处理流程几乎一致;同时,由于在分区共识方案的重构过程中,系统内部的各个模块需要进行解耦,从而使得模块间调用逻辑变得更加清晰.因此,重构之后的分区共识方案相比于传统的解决方案在处理逻辑上有了一定的优化,系统性能也会有一定的提升.

3 链内隐私保护方法

链间隐私保护通常适用于联盟链内各参与方均有多条复杂业务流的场景,但并不是所有的隐私数据请求

都需要通过新建分区的方式来完成:一方面,简单的隐私需求如多方隐私存证,仅需要一条或者若干条隐私交易请求就能完成了,为每一个诸如此类的数据量较小、交易频次也较小的隐私需求单独新建一个分区将是一个较为耗费资源的方式;另一方面,一个业务分区内部往往也存在着许多的隐私需求.

为了解决上述问题,本节提出了链内隐私保护方法.该方法实现了分区内部交易级别、合约级别的隐私保护,通过在交易体内部指定隐私交易的参与方信息(collection),用户可以选择分区参与方的任意合法子集作为本次隐私存证或者隐私合约的参与方,并通过客户端构造双签名交易向区块链平台发起隐私交易.接收隐私交易的区块链节点作为中转节点,将隐私数据同步至所有隐私参与方之后,构造公开交易并进行分区内部的全网共识,最终将公开交易同步至分区参与方节点后,隐私参与方节点单独进行隐私账本的更新.该方法通过中转节点来保证隐私数据的同步,防止隐私数据的泄漏,实现了较为灵活的交易级别隐私保护.

3.1 架构设计

链内隐私保护的架构设计如图 4 所示,所有区块链节点需要在每一个分区下维护两份账本信息,分别是公开数据账本与隐私数据账本,其中,

- 公开数据账本记录的是所有的公开交易信息,包括普通的公开交易(普通转账或者公开合约请求等)以及由中转节点构造的公开交易(作为隐私交易的存证信息存储在公开账本中).公开区块以块链式结构相连,所有节点的公开账本一致且同步进行账本的更新.
- 而隐私数据账本记录的则是各节点自己参与的所有隐私交易信息.由于不同节点在特定分区下参与的隐私交易不尽相同,因此隐私账本也不尽相同,隐私区块之间也无须以块链式结构相连以保证数据的一致性,每一次隐私账本的更新只在隐私参与方节点内部进行同步更新.

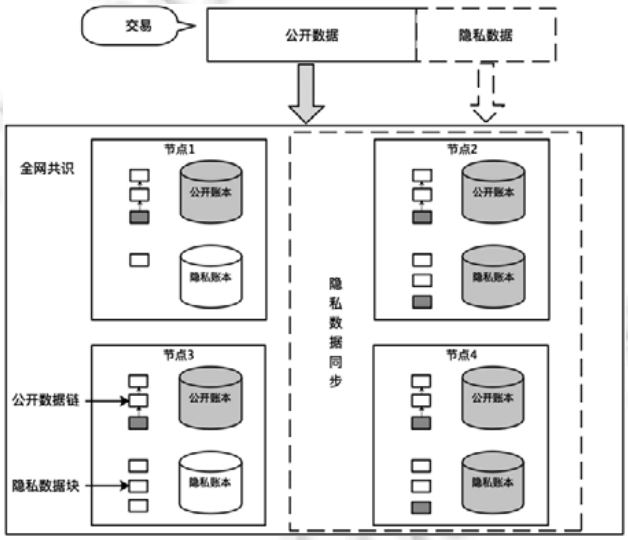

Fig.4 Architecture of intra-chain privacy protection

图 4 链内隐私保护架构

链内隐私保护的整体工作流程如图 5 所示.客户端构造包含双签名的隐私交易,并在交易中指定隐私参与方列表信息,将构造完成的隐私交易发向任一可连接的参与方节点;接收隐私交易的节点作为中转节点,首先检查隐私交易的合法性与自身权限的合法性,确认合法后,将隐私交易交给隐私交易管理器,由隐私交易管理器负责缓存至本地的隐私交易缓存区,并将隐私交易请求同步存储到其他所有的隐私参与方;其他隐私参与方节点在接收到隐私请求时,检查自身权限的合法性,确认合法后,将隐私交易交由隐私交易管理器,缓存至本地的隐私交易缓存区,随后向中转节点返回确认消息;中转节点在接收到所有隐私参与方节点的确认消息后,将需要保护的隐私数据替换成其哈希值后,构造公开交易,进行正常的公开交易上链流程;待到公开交易通过正常的共识

流程同步至所有分区参与方之后,所有分区参与方节点同步更新各自的公开账本,随后,各节点检查自身是否有对应的隐私交易,如果存在,则各自从隐私交易管理器中取出隐私交易、执行隐私交易,最后进行隐私账本的更新.

其中,上述的交易缓存区需要采用两级存储策略,每一笔进入到隐私交易管理器的隐私交易首先会保存到内存 cache 中,随后还需要持久化到数据库中,保证隐私数据存储的安全性与读取的高效性.具体来说,

- 第 1 层的缓存 cache 使用定长 map 实现,可以保证在最终提交隐私交易时快速的根据交易哈希值读取出隐私交易迁移到隐私账本中,一旦隐私交易 cache 数量超出规定的缓存大小限制,则后续进入的隐私交易只做持久化而不做 cache 缓存,因为先进入到隐私交易管理器缓存的交易提交顺序必定先于后到的隐私交易.但是如果缓存中存在过多的失效 cache(可能是因为同步存储失败导致的失效缓存的堆积),则有可能影响后续正常的隐私交易的读取延迟.为此,需要设计一套隐私交易失效清理的策略,即对于超过一定时长(如 1 天)或者超过一定区块数(如 1 000 个区块)迟迟无法提交的隐私交易缓存,隐私交易管理器定期地将他们从缓存中删除.
- 第 2 层持久化策略的存在主要考虑到了以下两点:一是隐私交易管理器缓存 cache 不能设置的过大,否则容易遭受 DDOS 攻击导致内存崩溃,对于超过缓存大小的隐私交易,可以只做持久化操作,这样在后续提交读取时,也会相应地多进行一次数据库的读操作;二是防止节点在意外宕机情况下丢失隐私数据,虽然宕机重启的节点可以通过隐私数据异常恢复策略重新获取到丢失的隐私数据,但是在应对多节点同时宕机的场景时依旧存在丢失的风险,因此所有节点都需要有自身的隐私数据持久化策略,以尽量保证隐私数据不会丢失.

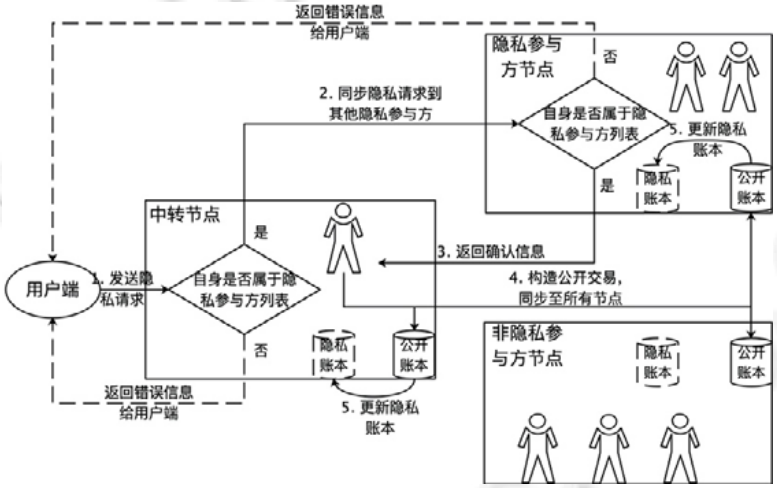

Fig.5 Flow of intra-chain privacy protection

图 5 链内隐私保护流程

3.2 隐私参与方列表Collection

为了实现交易级别细粒度的隐私保护,隐私交易必须支持灵活可变的隐私参与方列表 Collection 信息,因此可以将 Collection 字段嵌入到 Extra 字段中,作为隐私交易的额外信息存在,最终纳入隐私交易签名的计算中.需要注意的是,对于隐私存证场景,每一次的隐私存证都可能存在不同的参与方列表,即每一笔隐私存证交易请求都必须指定隐私参与方信息;但是对于隐私合约场景而言,每一个隐私合约在合约部署时就确定了隐私合约的参与方列表,因此在后续的合约调用、升级等操作时,客户端无须再次指定该隐私合约的参与方信息,即每一笔隐私合约部署交易请求都必须指定隐私合约参与方列表,后续所有关于该合约的调用交易请求都无须指定隐私合约参与方列表,区块链节点会根据给定的合约地址查询到隐私合约的参与方列表.

3.3 隐私数据字段的保护

无论是对于隐私交易存证场景还是对于隐私合约场景,其所需要保护的隐私数据都记录在交易的 Payload 字段中,因此,交易级别隐私的主要目的就在于对 Payload 字段的保护.本文选取 SHA3 加密哈希散列函数计算 Payload 原数据的哈希值,并用哈希值替换 Payload 原数据构造公开交易,以避免隐私原数据的泄漏.同时,为了对隐私相关的数据段进行整合,Payload 字段也被嵌入到 Extra 字段中.隐私交易中,原先 Payload 字段的部分始终置空.

3.4 中转节点

本文将与客户端直连的区块链节点定义为中转节点,该节点同时也是第 1 个接收到隐私交易的区块链节点,由其负责隐私交易的同步.中转节点的具体工作流程为:中转节点在接收到一笔来自客户端的隐私交易请求后,首先检查隐私交易的签名信息是否合法,并拒绝签名非法的交易;对于签名合法的交易,中转节点需要检查自身是否属于交易指定的隐私参与方列表,中转节点只接受自身参与的隐私交易;对于权限检查通过的隐私交易,中转节点将其交由隐私交易管理器 PTM 进行隐私数据的同步;在确认隐私数据同步完成之后,中转节点通知客户端同步成功的结果,并由客户端构造一比新的公开交易,发送到当前分区的共识网络中进行正常公开交易的上链流程.公开交易的存在,一方面作为隐私交易发生的证明记录在公开账本中,另一方面也是隐私数据同步过程中的索引信息,如果隐私数据同步在规定时间内失败,则中转节点也需要返回给客户端相应的失败信息,由客户端决定是否进行隐私交易请求的重传.

3.5 隐私交易管理器

隐私交易管理器(private transaction manager,简称 PTM)是链内隐私保护的核心所在,是保证隐私数据读写一致性的关键.如图 6 所示,PTM 作为分区级别的组件,也会被纳入 NSM 的管理中.

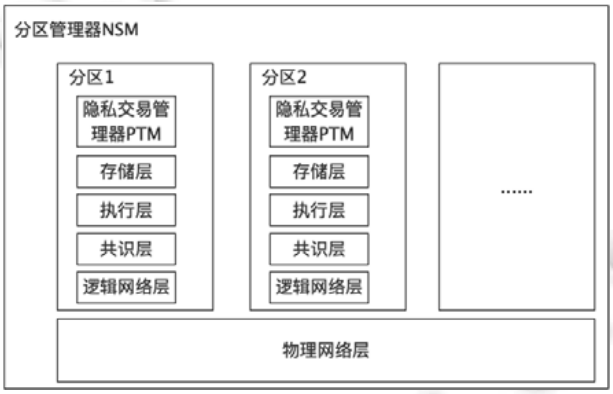

Fig.6 Architecture of single-node

图 6 单节点架构

3.6 隐私账本的设计

区块链账本记录了自创世区块以来的所有状态变迁的历史,具体的包括:

(1) 区块信息:即一系列链接起来的区块,后序区块通过在区块头中记录前序区块的区块哈希值来形成一条块链式结构,保证区块链的完整性与不可篡改性.

(2) 状态信息:即当前所有账户的状态数据集,也称为世界状态,世界状态是所有历史交易的最终表现,区块链节点之间通过当前状态数据的哈希值来进行全网状态的对比,确保所有节点状态的一致性.

在链间隐私保护设计方法中,每一个分区都有一份独立的分区账本,因此,每个节点会维护多份分区账本,每一份账本记录的是各自分区的区块信息与状态信息.同样的,隐私交易也会引起隐私状态的变迁,每一份隐私合约也有相应的隐私状态需要维护,因此,每一个分区也同样的需要维护一份隐私账本,其中记录的是隐私区块

信息与隐私状态信息.公开账本与隐私账本的更新流程如下.

(1) 节点通过共识模块进行交易的定序打包,并将打包完的区块定序后抛到公开交易执行线程中进行执行,此时区块中仅包含公开交易.

(2) 节点按序执行区块中的所有交易,在执行每一笔公开交易的时候,缓存当前对于公开世界状态的更改,同时检查是否有对应的隐私交易,如果有的话,再确认是否本节点是隐私参与方,如果是隐私参与方的话,则缓存该笔隐私交易.

(3) 节点执行完区块中所有的公开交易,将更新后的公开账本世界状态连同当前区块的信息持久化到数据库中.

(4) 在完成公开账本的更新之后,节点将前述缓存的隐私交易抛到隐私交易执行线程中进行执行,节点按序执行完所有的隐私交易后,进行隐私世界状态与隐私区块的持久化.需要注意的是,隐私区块仅包含两个信息,即与对应公开区块一致的区块号和所有的隐私交易.

从上述流程可以看出,隐私交易的执行与公开交易的执行是分处在不同的线程中完成的,这是为了尽量避免隐私交易对于公开区块执行效率的影响.如果将隐私交易与公开交易放在单个线程中执行,则不同节点执行相同的公开区块的速度将会不一致的.即参与隐私交易过多的节点会花费较多的时间在执行隐私交易上,最终导致不同节点间公开账本的更新速率不一致,需要通过频繁的区块快速同步来达到节点间公开状态的一致.

此外,隐私区块通过区块号来达成与其对应的公开区块的关联性,这一点设计是为了能够保证节点执行到最新的隐私区块.由于所有区块链节点的公开世界状态是一致的,公开区块的链式记录也是一致的,因此节点可以通过共识模块来确定当前全网最新的公开区块的信息,如果落后,就可以通过区块快速同步算法进行公开区块的索取与恢复.但是不同节点的隐私账本的内容不尽相同,因此节点无法通过其他节点的状态来判断自身是否执行到了最新的隐私状态.通过保留隐私区块号与公开区块号的关联性,可以让节点获知自身的隐私区块是否跟上了最新的公开区块,间接地确认自身是否执行到了最新的隐私状态.

图 7 展示了引入链内隐私保护之后的节点账本分布图,其中每一个节点需要为每一个分区创建两份账本信息:记录公开状态变迁的公开账本与记录隐私状态变迁的隐私账本.

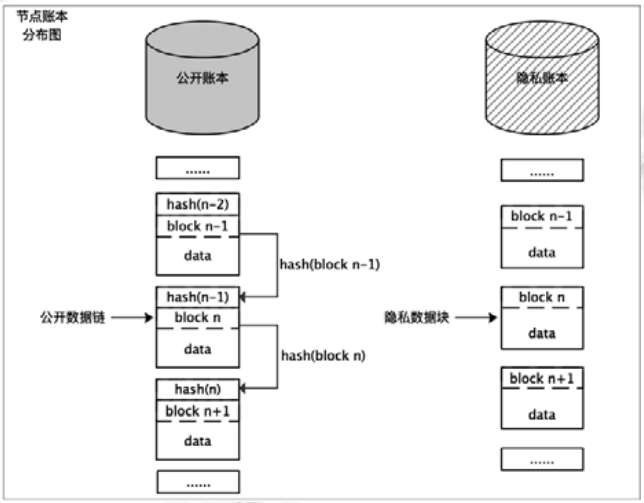

Fig.7 Ledger-distribution of intra-chain privacy protection

图 7 链内隐私保护账本分布

区别于公开账本中区块之间的链式结构,隐私账本中的区块之间无须形成链式结构,每一个隐私区块中只需要存储区块号、交易列表即可.原因在于,不同节点所参与的隐私交易不尽相同,因此不同节点之间的隐私世界状态也就不尽相同,那么维护一致的隐私区块哈希也就失去了意义.因此,隐私区块之间仅通过区块号辨别隐

私交易发生的历史顺序,不会再记录前序区块的哈希以形成块链式结构.

3.7 系统复杂度及性能分析

链内隐私保护的主要思想在于将隐私交易与公开交易分离处理,从而达到仅有隐私参与方能够获取到隐私交易的目的.图 8 展示了隐私交易及其对应的公开交易的生命周期图.

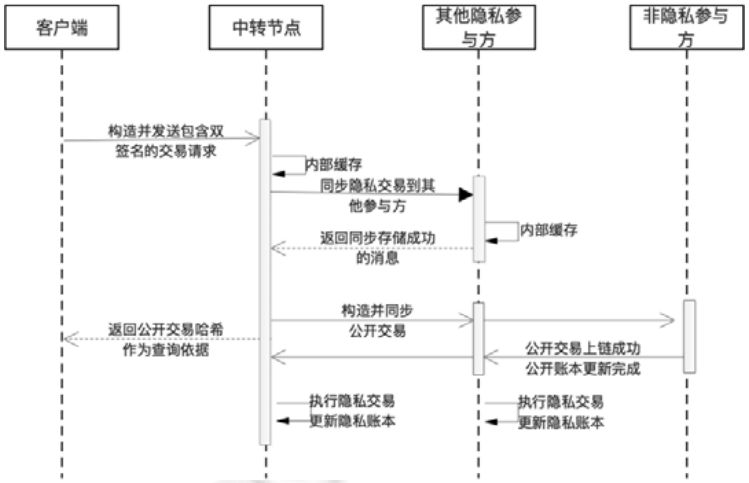

Fig.8 Life cycle of private transaction

图 8 隐私交易生命周期

从图 8 可以看出,一笔隐私交易的生命周期包含了:客户端构造隐私交易;中转节点接收隐私交易并进行本地的缓存;中转节点同步到其他隐私参与方,等到其他隐私参与方进行本地缓存之后回复同步成功的消息;中转节点构造公开交易进行上链,所有隐私参与方等待公开交易上链之后进行隐私交易的存储.隐私交易对应的公开交易与普通的公开交易上链流程一致,所需的时间开销 $T_{pub\text{-}tx}$ 包括:

(1) 公开交易验签时间 T_s;

(2) 公开交易共识时间 T_c;

(3) 公开交易执行时间 T_e;

(4) 公开交易存储时间 T_p.

上述 4 项时间开销是所有公开交易都需要花费的,其中,T_s 表示验证交易签名所花费的时间,该时间与交易的复杂度成正比;T_c 表示交易共识所花费的时间,该时间与系统采用的共识算法以及网络环境相关;T_e 表示交易执行的时间,该时间与系统所采用的虚拟机以及机器性能相关;T_p 表示交易存储时间,该时间与系统所采用的数据库类型以及机器存储介质相关.公开交易上链的总时间为上述所有时间的总和:

$$T_{pub\text{-}tx}=T_s+T_c+T_e+T_p \tag{1}$$

隐私交易上链所需要的时间开销 $T_{priv\text{-}tx}$ 包括:

(1) 隐私交易的验签时间 T_s;

(2) 中转节点本地持久化缓存时间 T_{p1};

(3) 中转节点同步到其他隐私参与方的网络时间开销 T_{n1};

(4) 其他隐私参与方本地持久化缓存时间 T_{p2};

(5) 其他隐私参与方回复同步存储成功的网络时间开销 T_{n2};

(6) 隐私交易对应的公开交易上链时间 $T_{pub\text{-}tx}$;

(7) 隐私交易存储时间 T_{p3}.

其中,隐私交易的验签时间与公开交易的验签时间一样,与交易的复杂度成正比;中转节点与其他隐私参与

方节点都需要进行隐私交易的本地持久化缓存,该时间与公开交易的存储时间一样(即 $T_p=T_{p1}=T_{p2}=T_{p3}$),与系统所采用的数据库类型以及机器存储介质相关;中转节点将隐私交易同步到其他隐私参与方节点的网络开销 T_{n1} 以及相应的回复消息的网络时间开销 T_{n2} 均与网络环境相关.公开交易上链的总时间为上述所有时间的总和.

$$T_{priv\text{-}tx}=T_{pub\text{-}tx}+T_s+3T_p+T_{n1}+T_{n2} \tag{2}$$

相比于公开交易的上链时间,隐私交易的上链时间主要受制于隐私交易的验签、隐私交易的持久化以及隐私数据的网络同步时间.由于隐私交易的同步过程是并行地向所有参与方节点发送隐私交易数据并同步地等待存储成功的响应消息,因此这一部分的时间主要受制于延迟最大的参与方节点.链内隐私保护主要面向于交易量较小、但是参与方灵活多变的隐私需求,因此链内隐私保护对于时效性的要求比较高,更多注重的性能指标是隐私交易的延迟而非吞吐量,具体的延迟指标包括隐私交易请求的延迟以及隐私交易查询的延迟.

4 实验结果与分析

本节分别对链间隐私保护方法以及链内隐私保护方法的实现进行测试与分析,其中,链间隐私保护方法的测试主要包括正确性测试、分区后多业务并行的性能与多平台部署方法性能的对比;链内隐私保护方法的测试主要包括正确性测试、隐私交易与公开交易的延迟对比.本测试全部都基于知名区块链平台 Hyperchain 进行,Hyperchain 平台作为国内知名的联盟链区块链平台,致力于实现国产、自主、可控的区块链平台,为开发者提供了非常方便的接口,因此作为本文的测试对象,保证了数据的真实性.

4.1 实验环境

本实验采取 6 台部署了 Hyperchain 节点的机器作为服务器模拟 6 家参与多条业务的机构,以及一台测试机作为客户端向区块链节点发送模拟交易.相应的,作为对比,每台服务器上也准备了加入隐私保护特性改造的 Hyperchain 节点.为了简单起见,不对客户端与区块链节点进行网络分区场景的模拟,测试机可以同时连接上所有的节点.所有机器的配置一致,见表 1.

Table 1 Configuration of test machine

表 1 测试机器配置表

配置	具体信息
操作系统	CentOS Linux release 7.3.1611 (Core)
CPU	Intel(R) Core(TM) i7-6700 CPU @ 3.40GHz
网卡	Intel Corporation Ethernet Connection (2) I219-LM (rev 31)
内存	16G Samsung PC4-2400T-UA2
硬盘	512G SSD Samsung SSD, 1T HDD ST1000DM003-1SB1 CC4

4.2 链间隐私保护测试

链间隐私保护的关键在于,不同分区之间的数据需要进行分流处理、隔离存储.因此,所有分区内的交易数据只能由该分区的参与方维护,也只对该分区的参与方可见.分区共识的特性尤其适用在同一个区块链节点需要参与到多个业务流的场景,该场景的传统解决方法是为不同的业务分别部署一套单独的区块链平台,而分区共识可以让同一个节点同时参与多个分区业务,满足多业务并行的需求.

为了尽量模拟多分区场景,本节测试利用 6 台节点服务器构造了 3 个分区,具体的分区参与状态如图 1 所示:所有节点都参与到了 global 分区(为了方便起见,未标出 global 分区及其网络连接状态),节点 1~节点 4 同时参与到了 ns1 分区,节点 3~节点 6 参与到了 ns2 分区.

4.2.1 正确性测试

本节主要设计链间隐私保护的正确性测试,分区内部的数据由分区的参与方进行维护,任意一个恶意节点对数据的破坏都无法造成对整个分区数据的破坏.由于所有区块链数据都是通过加密方式存储的,很难进行手动地更改,因此,本文仅通过删除相应分区的数据库来模拟恶意节点对数据的破坏.具体的测试步骤如下:在所有节点之间网络连接状态正常、客户端与所有节点连接正常的情况下,客户端分别向不同节点发送不同分区的

模拟交易,随后,模拟非法分区的交易请求;删除某合法节点的分区数据库,立即向其进行该分区数据的查询,并等待该节点数据同步完成之后,再向该节点进行分区数据的查询.

针对链间隐私保护正确性测试,本文设计了两组测试场景,并对每一种场景进行了 100 次的功能性测试.两组测试场景如下.

- 测试场景 1:向节点 1~节点 6 发送 ns1 分区的交易请求,向节点 1~节点 6 发送回执查询请求.
  预期结果:所有发向 1~节点 4 的请求发送成功,回执查询成功;所有发向节点 5、节点 6 的请求与回执查询都失败.
- 测试场景 2:向节点 1~节点 4 发送 ns1 分区的数据,并删除 1 号节点的 ns1 分区数据库,立即向其查询请求回执;一段时间之后,再向其查询请求回执.
  预期结果:所有请求发送成功,第 1 次查询失败;一段时间后,第 2 次查询成功.

表 2 展示了测试结果.

Table 2 Test result of correctness of inter-chain privacy protection

表 2 链间隐私保护正确性测试结果

场景编号	测试组数	符合预期组数(传统解决方案)	符合预期组数(链间隐私方法)
1	100	100	100
2	100	100	100

从上述测试结果可以看出,链间隐私保护方案符合了测试预期,与传统的解决方案一样可以限制非法的分区交易请求.例如向节点 1~节点 4 节点发送 ns1 分区的请求与回执查询都能成功,但是向节点 5、节点 6 发送 ns1 分区的请求均失败,回执查询相应的也均失败.同时,恶意破坏其中某个节点的分区数据库是无法破坏整个分区数据的完整性的.例如,实验 3 中,向节点 1~节点 4 发送一定量的交易数据后,直接删除 1 号的分区数据库,此时 1 号节点处于数据的丢失恢复阶段.因此,直接向 1 号节点查询交易的回执时会报查询失败的错误;但是过一段时间后,1 号通过区块的快速同步算法进行了丢失数据的恢复,此后再向 1 号节点查询时都能够正常地查询到丢失的数据.可见,链间隐私保护方法保证了数据的完整性与正确性.

4.2.2 性能测试

首先,本文选择 4 台服务器部署区块链节点,每一个节点分别部署 1~10 套区块链平台,模拟 4 家机构同时参与 1~10 个业务的场景,并由一台测试机器模拟向区块链节点发送 1~10 个业务的交易请求.测试一共分 10 组进行,每组测试中,客户端的压力请求都均摊到各个分区中.例如,模拟 1 个业务场景,即每个节点部署一套区块链平台时,则向节点发送 6 000TPS(transaction per second)的压力请求;而模拟 2 个业务场景,即每一个节点部署两套区块链平台时,则分别向每一个业务平台发送 3 000TPS 的压力请求,以此类推…

随后,本文选择同样的 4 台服务器部署增加了分区共识特性的区块链节点,每一个节点分别参与 1~10 个分区,模拟 4 家机构同时参与 1~10 个业务的场景,并由同样的测试机器模拟向区块链节点发送 1~10 个分区的交易请求.测试一共分 10 组进行,每组测试中,客户端的压力请求都均摊到各个分区中.例如,模拟 1 个业务场景,即每个节点启动一个分区时,则向该分区发送 6 000TPS 的压力请求;而模拟 2 个业务场景,即每一个节点启动两个分区时,则分别向每一个分区发送 3 000TPS 的压力请求,以此类推….

图 9 从区块链节点的角度,展示了单节点处理总性能值对比.

从上图可以看出,单节点在分区共识方法下的处理总性能比多平台部署方法提升了 10%~15%左右,而且在使用了分区共识的功能之后,2 个分区明显比 1 个分区的性能有了显著的提高.主要原因在于,多平台部署方法中,每多一个业务,就需要多部署一套区块链平台,为此需要单独启用若干网络端口资源;而分区共识方法中,所有分区之间的底层网络资源是共享的,因此不论单节点参与到多少个分区,只需要启动一份网络端口资源,大大降低了网络资源的消耗.总的来说,分区共识方案相比于传统的多平台部署方案而言,在保证了正确性与完整性的同时,还带来了可观的性能提升.

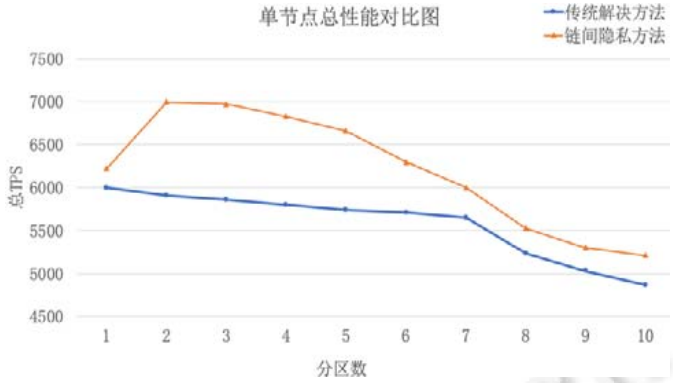

Fig.9 Comparison of single-node performance

图 9 单节点性能对比

4.3 链内隐私保护测试

4.3.1 正确性测试

本节主要设计链内隐私保护的正确性测试,隐私交易的数据应该仅由隐私参与方节点之间共享,非隐私参与方无法获取到隐私交易的读取权限.同样的,由于所有区块链数据都是通过加密方式存储的,很难进行手动地更改,因此本文仅通过删除相应分区的数据库来模拟恶意节点对数据的破坏.具体的测试步骤如下:在所有节点之间网络连接状态正常、客户端与所有节点连接正常的情况下,首先模拟恶意节点发送非法的隐私交易请求;随后模拟恶意节点发送非法的隐私交易查询请求;最后向节点发送正常的隐私交易,并删除该节点的分区数据库,查询隐私交易回执,一段时间后,再次查询隐私交易的回执.

针对链内隐私保护正确性的测试,本文设计了 3 组测试场景,并对每一种场景进行了 100 次的功能性测试.为方便起见,节点 n 的哈希值用 hn 的方式表示.3 组测试场景如下.

- 测试场景 1:向 *node*1 发送隐私交易请求、*Collection*=[*h*2,*h*3].
  预期结果:隐私交易请求失败.
- 测试场景 2:向 *node*1 发送隐私交易请求、*Collection*=[*h*1,*h*2,*h*3];分别向 *node*4,*node*5,*node*6 查询隐私交易回执.
  预期结果:隐私交易请求成功,隐私交易回执查询失败.
- 测试场景 3:向 *node*1 发送隐私交易请求、*Collection*=[*h*1,*h*2];删除 1 号的分区账本数据库,立即向 1 号查询隐私交易回执;一段时间后,再次查询.
  预期结果:隐私交易请求成功,第 1 次查询失败,第 2 次查询成功.

表 3 展示了测试结果.

Table 3 Test result of correctness of intra-chain privacy protection

表 3 链内隐私保护正确性测试结果

场景编号	测试组数	符合预期组数(传统解决方案)	符合预期组数(链内隐私方法)
1	100	0	100
2	100	0	100
3	100	0	100

从上述测试结果可以看出,传统的区块链平台并不能提供到交易级别细粒度的隐私保护,而链内隐私方法则可以解决联盟区块链中的交易隐私问题,非法的隐私交易请求与隐私交易回执查询请求都受到了严格的限制.同时,恶意破坏节点的隐私账本并不会破坏整个的系统隐私账本的完整性.如实验 3 所示:虽然 1 号节点的账本被恶意破坏了,但是经过一段时间的隐私交易异常恢复之后,1 号节点的隐私账本依旧可以恢复丢失的隐私数据,保证了隐私数据的完整性.

4.3.2 性能测试

由于链内隐私保护方法主要面向小数据量、小群体之间的隐私保护需求,因此该方法注重的是隐私交易的延迟而非吞吐量.本文选择在 6 节点情况下,分别设置 1~6 个节点作为隐私参与方,测试不同参与方数量下隐私交易请求与隐私交易回执查询的延时.为了防止网络波动对本测试的影响,本测试选择在夜间无网络负载的情况下进行,且所有节点均处在同一机房同一内网.同时,为了作为对比,也将相应的公开交易请求与公开交易回执查询的延迟画在了图中.图 10 展示了隐私请求延迟对比图.

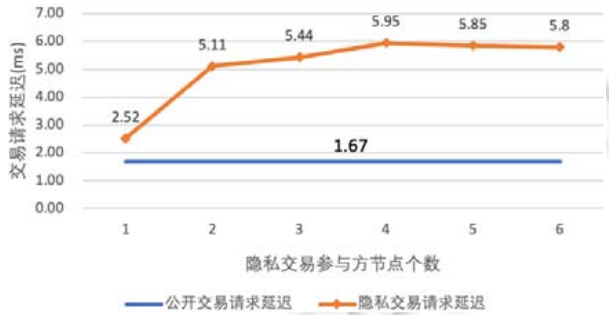

Fig.10 Comparison of transaction's request latency

图 10 交易请求延迟对比

从图 10 可以看出,相比于正常公开交易的请求延迟(1.67ms),所有隐私交易的请求延迟都略有上升.其中,当参与方只有 1 个时(该节点必然为中转节点),由于中转节点已经无需将隐私交易同步至其他的参与方节点,因此该场景只比公开交易场景多了一次隐私交易验签的时间加上隐私交易本地持久化的时间,延迟上升约 0.85ms;当参与方有 2 个时,中转节点除了自身需要进行隐私交易验签加本地持久化之外,还需要将隐私交易同步至其余的 1 个隐私参与方节点,等待该隐私参与方节点进行验签、本地持久化以及返回确认消息的时间,因此延迟上升了约 2.59ms;后续,随着参与方的增多,隐私同步的总时间也基本上稳定在了 5ms~6ms 之间,总体的延迟时间在用户可接受的范围内.

图 11 展示了隐私交易回执查询延迟对比图.

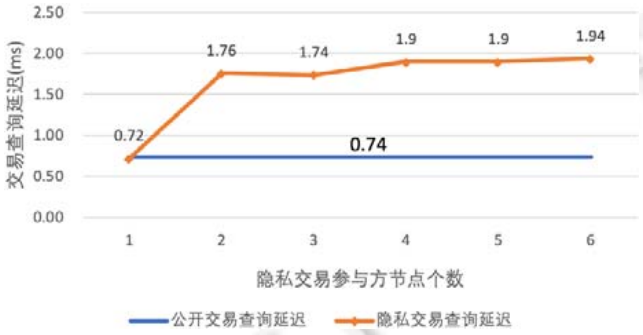

Fig.11 Comparison of transaction's query latency

图 11 交易查询延迟对比

从上图可以看出,相比于正常公开交易的查询延迟(0.74ms),隐私交易的查询延迟略有上升.特别地,当参与方只有 1 个时(该节点即为接收节点),由于接收节点已经无需向其他参与方查询回执了,因此该场景下只需要接收节点查询本地数据库的结果即可,相比于公开交易查询时一样的要从本地数据库查询交易回执,该场景反而比公开交易查询快了 0.02ms;当参与方有 2 个时,接收节点除了自身需要进行本地数据库的查询外,还需要向其他 1 个参与方节点请求回执信息,等待该隐私参与方节点进行本地查询加返回查询结果,因此延迟上升了约

1.04ms;后续,随着参与方的增多,隐私查询的总时间也基本上稳定在 1.7ms~2ms 之间,总体的延迟时间在用户可接受的范围内.

5 结 论

随着区块链技术的火热发展,越来越多的区块链应用开始落地实施,随之而来的问题也逐渐暴露出来.本文主要着手研究了联盟区块链中的隐私保护问题,具体工作内容包括:

(1) 设计并实现了链间隐私保护方法.

通过对不同业务流的数据进行分流处理、分区存储实现了业务流之间的数据隔离保护,所有分区的信息通过统一的分区管理器 NSM 进行管理,所有节点在启动完之后,需要首先加入到全局分区,以便进行后期节点的管理.一个分区的生命周期包括注册、启动、停止、注销.需要注意的是,为了保证分区参与方变动时共识机制的完备性,分区注册之前需要进行分区名与分区配置项的线下协商,分区启动时需要进行协议版本的线上协商,分区注销之前需要进行删除节点操作.

(2) 设计并实现了链内隐私保护方法,具体包括隐私交易与隐私合约的保护.

通过在交易体中嵌入 Collection 字段,用户可以指定分区参与方的任意子集作为一笔隐私交易的参与方,实现了灵活的交易级别隐私保护.为了简化隐私交易同步存储的流程,本文将第 1 个接收隐私请求的区块链节点记为中转节点,并由其负责进行隐私数据的同步存储;同时,本文设计了独特的双签名交易,使得中转节点可以直接构造合法的公开交易进行上链操作,而无需由客户端进行二次的交易请求.为了实现隐私交易与隐私合约数据的隔离存储,节点需要为每一个分区维护两份账本数据:一份是正常的公开交易及公开状态的数据,一份是隐私交易及隐私状态的数据.公开账本与隐私账本之间隔离存储,保证了两者之间的互不影响.

最后,本文分别对链间隐私方法吞吐量以及链内隐私保护方法的延迟性进行了测试与对比,结果表明,通过结合粗粒度的链间隐私保护与细粒度的链内隐私保护,在满足了隐私需求的同时也保证了可观的性能,为区块链平台的隐私性与安全性做出了贡献.

References:

[1] Nakamoto S. Bitcoin: A peer-to-peer electronic cash system. 2008. 1–9. https://bitcoin.org/bitcoin.pdf

[2] Buterin V. A next-generation smart contract and decentralized application platform. 2014. 1–36. https://cryptorating.eu/whitepapers/Ethereum/Ethereum_white_paper.pdf

[3] Szabo N. The Idea of Smart Contracts. Nick Szabo's Papers and Concise Tutorials, 1997. 1–2. https://nakamotoinstitute.org/the-idea-of-smart-contracts

[4] Wood G. Ethereum: A Secure Decentralised Generalised Transaction Ledger. Ethereum Project Yellow Paper. 2014. 1–34.

[5] Cachin C. Architecture of the hyperledger blockchain fabric. In: Proc. of the Workshop on Distributed Cryptocurrencies and Consensus Ledgers. 2016. 1–4.

[6] Zheng Z, Xie S, Dai H, Chen X, Wang H. Blockchain challenges and opportunities: A survey. Int'l Journal of Web and Grid Services, 2018,14(4):352–375.

[7] Zheng Z, Xie S, Dai H, Chen X, Wang H. An overview of blockchain technology: Architecture, consensus, and future trends. In: Proc. of the IEEE Int'l Congress on Big Data (BigData Congress). 2017. 557–564.

[8] Vujičić D, Jagodić D, Ranđić S. Blockchain technology, bitcoin, and Ethereum: A brief overview. In: Proc. of the 17th Int'l Symp. on Infoteh-Jahorina (infoteh). IEEE, 2018. 1–6.

[9] Khan MA, Salah K. IoT security: Review, blockchain solutions, and open challenges. Future Generation Computer Systems, 2018, 82:395–411.

[10] Alphand O, Amoretti M, Claeys T, Dall'Asta S, Duda A, Ferrari G, Rousseau F, Tourancheau B, Veltri L, Zanichelli F. IoTChain: A blockchain security architecture for the Internet of Things. In: Proc. of the IEEE Wireless Communications and Networking Conf. (WCNC). 2018. 1–6.

[11] Yaga D, Mell P, Roby N, Scarfone K. Blockchain technology overview. 2019. 1–46. https://doi.org/10.6028/NIST.IR.8202

[12] Eyal I, Sirer E G. Majority is not enough: Bitcoin mining is vulnerable. In: Proc. of the Int'l Conf. on Financial Cryptography and Data Security. Berlin, Heidelberg: Springer-Verlag, 2014. 436−454.

[13] Marcus Y, Heilman E, Goldberg S. Low-resource eclipse attacks on Ethereum's peer-to-peer network. IACR Cryptology ePrint Archive, 2018.

[14] Kalra S, Goel S, Dhawan M, Sharma S. Zeus: Analyzing safety of smart contracts. In: Proc. of the 25th Annual Network and Distributed System Security Symp. (NDSS). 2018. 1−12.

[15] Dagher GG, Mohler J, Milojkovic M, Marella PB. Ancile: Privacy-preserving framework for access control and interoperability of electronic health records using blockchain technology. Sustainable Cities and Society, 2018,39:283−297.

[16] Praitheeshan P, Pan L, Yu J, Liu J, Doss R. Security analysis methods on Ethereum smart contract vulnerabilities: A survey. arXiv preprint arXiv:1908.08605, 2019.

[17] Zhang R, Xue R, Liu L. Security and privacy on blockchain. ACM Computing Surveys (CSUR), 2019,52(3):1−34. https://doi.org/10.1145/3316481

[18] Feng Q, He D, Zeadally S, Khan MK, Kumar N. A survey on privacy protection in blockchain system. Journal of Network and Computer Applications, 2019,126:45−58.

[19] Dwivedi AD, Srivastava G, Dhar S, Singh R. A decentralized privacy-preserving healthcare blockchain for IoT. Sensors, 2019, 19(2):326. https://doi.org/10.3390/s19020326

[20] Reid F, Harrigan M. An analysis of anonymity in the Bitcoin system. In: Proc. of the Security and Privacy in Social Networks. New York: Springer, 2013. 197−223.

[21] Bonneau J, Miller A, Clark J, Narayanan A, Kroll JA, Felten EW. Sok: Research perspectives and challenges for Bitcoin and cryptocurrencies. In: Proc. of the IEEE Symp. on Security and Privacy. IEEE, 2015. 104−121.

[22] Bonneau J, Narayanan A, Miller A, Clark J, Kroll J A, Felten EW. Mixcoin: Anonymity for Bitcoin with accountable mixes. In: Proc. of the Int'l Conf. on Financial Cryptography and Data Security. Berlin, Heidelberg: Springer, 2014. 486−504.

[23] Ruffing T, Moreno-Sanchez P. ValueShuffle: Mixing confidential transactions for comprehensive transaction privacy in Bitcoin. In: Brenner M, *et al*., eds. Proc. of the Financial Cryptography and Data Security (FC 2017). LNCS 10323. Cham: Springer-Verlag, 2017. 133−154.

[24] Miers I, Garman C, Green M, Rubin AD. Zerocoin: Anonymous distributed e-cash from Bitcoin. In: Proc. of the IEEE Symp. on Security and Privacy. 2013. 397−411.

[25] Maxwell G. CoinJoin: Bitcoin privacy for the real world. In: Proc. of the Post on Bitcoin Forum. 2013. https://bitcointalk.org/index.php?topic=279249.0

[26] Maurer FK, Neudecker T, Florian M. Anonymous CoinJoin transactions with arbitrary values. In: Proc. of the 2017 IEEE Trustcom/BigDataSE/ICESS. 2017. 522−529.

[27] Heilman E, Alshenibr L, Baldimtsi F, Scafuro A, Goldberg S. TumbleBit: An untrusted Bitcoin-compatible anonymous payment hub. In: Proc. of the Network and Distributed System Security Symp. 2017. 1−37.

[28] Heilman E, Baldimtsi F, Goldberg S. Blindly signed contracts: Anonymous on-blockchain and off-blockchain Bitcoin transactions. In: Proc. of the Int'l Conf. on Financial Cryptography and Data Security. Berlin, Heidelberg: Springer-Verlag, 2016. 43−60.

[29] Noether S. Ring signature confidential transactions for Monero. IACR Cryptology ePrint Archive, 2015,1098:1−34.

[30] Möser M, Soska K, Heilman E, Lee K, Heffan H, Srivastava S, Hogan K, Hennessey J, Miller A, Narayanan A, Christin N. An empirical analysis of traceability in the Monero blockchain. Proc. on Privacy Enhancing Technologies, 2018,(3):143−163.

[31] Borggren N, Yao L. Correlations of multi-input Monero transactions. arXiv preprint arXiv:2001.04827, 2020.

[32] Van Saberhagen N. CryptoNote v 2.0. 2013. 1−20. https://cryptonote.org/whitepaper.pdf

[33] Gai K, Wu Y, Zhu L, Qiu M, Shen M. Privacy-preserving energy trading using consortium blockchain in smart grid. IEEE Trans. on Industrial Informatics, 2019,15(6):3548−3558.

[34] Zhang A, Lin X. Towards secure and privacy-preserving data sharing in e-health systems via consortium blockchain. Journal of Medical Systems, 2018,42(8):Article No.140.

[35] Androulaki E, Barger A, Bortnikov V, *et al*. Hyperledger fabric: A distributed operating system for permissioned blockchains. In: Proc. of the 13th EuroSys Conf. 2018. 1−15.

[36] Sousa J, Bessani A, Vukolic M. A byzantine fault-tolerant ordering service for the hyperledger fabric blockchain platform. In: Proc. of the IEEE 48th Annual IEEE/IFIP Int'l Conf. on Dependable Systems and Networks (DSN). 2018. 51−58.

蔡亮(1976－),男,博士,副教授,CCF 高级会员,主要研究领域为计算机应用.

端豪(1994－),男,硕士,主要研究领域为区块链技术与应用.

鄢萌(1989－),男,博士,研究员,博士生导师,CCF 专业会员,主要研究领域为智能软件工程,软件仓库挖掘,软件维护与演化.

夏鑫(1986－),男,博士,讲师,博士生导师,CCF 专业会员,主要研究领域为软件仓库挖掘,经验软件工程.