计算机科学
COMPUTER SCIENCE

基于可跟踪环签名的拜占庭容错共识算法

涂俊, 贾东立, 王津

引用本文

涂俊, 贾东立, 王津.基于可跟踪环签名的拜占庭容错共识算法[J]. 计算机科学, 2023, 50(6A): 220300100-7.

TU Jun, JIA Dongli, WANG Jin. Byzantine Fault Tolerant Consensus Algorithm Based on Traceable Ring Signature [J]. Computer Science, 2023, 50(6A): 220300100-7.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.220300100

基于可跟踪环签名的拜占庭容错共识算法

涂 俊 贾东立 王 津
河北工程大学信息与电气工程学院 河北 邯郸 056038
(tj18832045990@163.com)

摘 要 针对联盟链的实用拜占庭容错(PBFT)共识算法在共识过程中节点间的隐私保护弱、网络结构静态、选取主节点不可靠和通信开销大的问题,提出一种基于可跟踪环签名的拜占庭容错共识算法(tracePBFT)。首先,随机将节点分为主域节点和副域节点并且赋予其不同的权重,选择权重高的主域节点为主节点;然后在准备阶段引入可追踪环签名对节点进行隐私保护,并且节点可以通过权重选择可靠节点,在确认阶段验证签名和跟踪拜占庭节点;最后适当惩罚拜占庭节点。这样选择的主节点更加可靠,减少因主节点出错而更换视图导致的通信开销。实验结果表明,相比传统的 PBFT 算法,tracePBFT 算法在通信复杂度、安全性、吞吐量等方面均有一定的提高。

关键词:联盟链;实用拜占庭容错共识算法;可跟踪环签名;主节点
中图法分类号 TP301.6

Byzantine Fault Tolerant Consensus Algorithm Based on Traceable Ring Signature

TU Jun,JIA Dongli and WANG Jin
School of Information and Electrical Engineering,Hebei University of Engineering,Handan,Hebei 056038,China

Abstract The practical Byzantine fault tolerance(PBFT) consensus algorithm of alliance chain has the problems of weak privacy protection between nodes,static network structure,unreliable selection of master node and high communication overhead. A Byzantine fault-tolerant consensus algorithm(tracePBFT) based on traceable ring signature is proposed. Firstly,the nodes are randomly divided into primary domain nodes and secondary domain nodes,and different weights are given,and the primary domain node with high weight is selected as the primary node. Then,the ring signature is introduced in the preparation stage to protect the privacy of the node,and the node can select the reliable node through the weight,verify the signature and track the Byzantine node in the confirmation stage,and finally appropriately punish the Byzantine node. In this way,the selected master node is more reliable and reduce the communication overhead caused by changing the view due to the error of the master node. Experiments show that the tracePBFT algorithm is better than the traditional PBFT algorithm in communication complexity,security,throughput and so on.

Keywords Alliance chain,Practical Byzantine fault tolerant consensus algorithm,Traceable ring signature,Master node

1 引言

2008 年《比特币:一种点对点式的电子现金系统》一文的发表标志着比特币诞生[1],比特币的提出被用于解决“双花”问题,并以区块链作底层技术[2-3]。由于比特币和其他数字加密货币迅速发展,区块链技术作为保证比特币安全可靠的底层技术得到研究学者的广泛关注。区块链本质上是一种独特的去中心化的分布式账本数据库。它由密码学、智能合约、共识算法、时间戳等技术集成,具备有防篡改、透明可溯源和去中心的特性。共识算法[4]作为区块链系统的底层核心技术,可保证系统中的各节点对特定时间内打包的提案交易顺序达成一致,即实现分布式系统完成提案的最终一致性的作用。

事实上,共识算法很早以前就出现了,最常见的共识算法有 PoW 算法、PoS 算法、Raft 算法和 BFT 算法等[5]。在解决拜占庭问题上,BFT 算法像 PoS 那样存在垄断机会,也不像 PoW 那样需要耗费巨大算力。但为了达到共识结果,BFT 算法需要进行三阶段通信[6],其中后两阶段中每个节点需要向系统全节点传递消息。BFT 算法后两阶段的性质,导致在共识过程中,随着节点数增加,通信复杂度呈指数级增加,不符合现实发展需求。且网络的扩展性不足[7],系统无法感知节点的变化,节点退出和加入系统需要重启应用。以上缺点的存在,使得 BFT 算法没有受到很大的关注。直至区块链技术的出现,BFT 算法才焕发新的活力,同时为后来的实用拜占庭容错(Practical Byzantine Fault Tolerance,PBFT)算法的出现提供了指导的方向[8]。在共识过程中,随着节点数增加,PBFT 算法的通信复杂度呈多项式增加,相比 BFT 算法提高了共识效率。同时 PBFT 算法也继承 BFT 算法的一些缺点,例如随着共识节点数量增加,共识效率逐渐降低,以及网络结构是静态类型,仍然不能满足我们实际的应用要求。而且共识算法不能只注重共识效率,对用户的隐私保护也是研究者

基金项目:河北省高等学校科学技术研究项目(ZD2015087)
This work was supported by the Science Technology Research Project of the Higher Eduction in Hebei Province(ZD2015087).
通信作者:贾东立(jwdsli@163.com)

不能忽视的问题。本文由此提出了基于可跟踪环签名的拜占庭容错算法(tracePBFT)。经过实验验证和分析,tracePBFT算法利用可跟踪环签名的特性,对恶意的节点进行惩罚,减少了通信开销,提高了共识效率,同时也保障了共识过程中节点间的隐私。

当前已有诸多研究人员从各个方面对PBFT算法提出一些修改方案。Hao等[9]在原PBFT算法的基础上增加了节点信息登记机制及恶意节点管控机制,以提高系统的安全性,但是节点安全性问题没有得到解决。Han等[10]提出了一种改进的算法,降低了算法的复杂度,并且允许共识节点加入和退出系统,解决了可扩展性问题,但是主节点的选择是随机的。Tang等[11]通过引入节点的可靠性评分,选择可靠性评分最高的节点作为主节点和选择可靠性评分较高的部分节点做副节点进行共识的过程。虽然它解决了PBFT算法通信复杂度高、主节点选取简单、对节点拥有管控机制等一些弊端。但在共识过程中,可以故意使一个恶意节点的可靠性评分最高作为主节点,在共识过程中此恶意节点可以联合其他恶意节点,评选此恶意节点为可靠性最高的节点,伪造一条新的链条,储存虚假信息。因此在共识过程中使节点匿名是非常重要的,这样评选出的节点更加可靠[12]。Wang等[13]提出一种高效监督拜占庭容错算法,主要是针对效率问题,高效监督拜占庭容错算法将节点随机划分为多个节点簇,设置信誉值,通过信誉值从节点簇中选举共识节点来监督节点,尽可能提升共识节点的高效性及可靠性;监督节点对共识节点进行监控,避免了在Global Stabilization Time(GST)开始之前共识节点可能遭遇的系统不协调问题,进一步保证算法的安全性;但是该算法没有考虑网络状态复杂的情况下存在节点宕机或者其他状况。虽然该算法解决了一部分安全问题,但是没有实用的情况。Fang等[14]提出了基于环签名的PBFT算法,保护了共识过程中节点间的隐私,并解决了动态加入或退出网络的问题,但是通信复杂度高这一问题依然存在。Zhang[15]提出基于关联环签名的PBFT算法,通过引入了节点权重,权重高的节点拥有参与共识过程的权力,解决了通信复杂度高的问题。引入了关联环签名不仅解决了共识过程中节点间隐私保护问题,也解决了节点"双花"问题,但出现恶意节点时没有对该节点作出适当的惩罚,导致该恶意节点可以继续参加下一轮共识过程。

本文针对PBFT算法普遍存在的网络结构类型静态、主节点选取的不可靠性、通信复杂度高、节点间的隐私性保护弱等问题,提出一种基于可跟踪环签名的拜占庭容错共识算法(TracePBFT)。针对PBFT算法的特点,将节点分为主域节点和副域节点。主域节点权重较高,因此可被选择为主节点,参与共识过程;副域节点相当于备份节点,可以通过投票选择可靠的主节点,可以动态退出,新的节点加入可以作为备份节点。在共识过程中的三个阶段,在准备阶段设置参与共识的节点进行数字签名处理,通过投票,为下一轮选择主节点做准备,确认阶段确认节点投出的权重是否有效,并惩罚恶意节点投出权重。

本文的主要贡献如下:

(1)从节点的基础配置及共识过程两个方面着手分析,开始分配主副域节点的权重,以权重高的节点作为主节点,主持本轮共识,同时选择权重较高的节点参与共识全部过程,解决了PBFT算法选主节点不安全、通信复杂度高以及扩展性不足的问题。

(2)在共识过程中增加数字签名算法以保护节点间隐私,在准备阶段对节点进行签名,并且节点间匿名投出权重给信任的节点,为下一轮共识选择主节点做准备;在确认阶段对数字签名进行验证,判断投出的权重是否有效,以及是否出现"双花"的现象,并作出适当的惩罚。

(3)通过对tracePBFT算法进行实验,结果表明,在安全和可信的前提下,合理地选取主节点以减少触发视图协议,其共识效率(时延和吞吐量)优于PBFT算法。

2 相关背景介绍

2.1 实用拜占庭容错共识算法

PBFT算法最早是由Castro等提出的[16],是对BFT算法的优化,因此它拥有BFT算法的优点,且通信的复杂度只有多项式级别。PBFT算法可以在已知网络系统中拜占庭节点数不超过三分之一情况下,确保最初提案在最终决策的统一性和正确性,这是实用拜占庭容错算法的首次实现。PBFT算法在保证网络通信系统安全可靠的前提下,可以允许网络中存在 $f=\frac{N-1}{3}$ 的恶意节点(其中 N 表示网络中的总节点数量)。

PBFT算法共识过程中的节点由一个主节点(Primary)以及多个备份节点(Backup)组成,主节点的任务是主持这次共识过程。首先客户端将提案消息发送给主节点,主节点将收到的消息进行打包和编号处理,然后将处理的消息广播给所有的备份节点,收到消息的备份节点会对消息进行验证以及进行有关操作将最终决策返回给客户端。PBFT算法通信协议的核心部分是三阶段,它保证共识结果最终的正确性。该三阶段协议又被称为一致性协议,是以投票为基础的共识协议,其中包括预准(Pre-Prapare)、准备(Prepare)和确认(Commoit)3个阶段。PBFT流程图如图1所示,图中节点3是拜占庭节点。

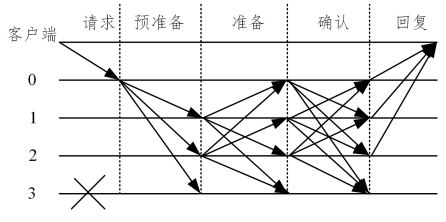

图1 PBFT流程图

Fig. 1 PBFT flowchart

2.2 环签名

环签名于2001年由Shamir等提出[17],环签名只需要签名者本身的密钥和环中其他成员公钥的集合,就可以签署验证通过的环签名。签名验证过程只需要环中任意成员利用公钥集合就可以对签名消息进行验证,但验证者只知道环中某个合法成员代表环签署了信息,不能判别公钥集合中实际签名者的具体身份。基础的环签名由下面3个算法组成。

生成Gen:首先输入系统安全参数 δ,该算法会为用户 i 生成一对密钥 (pk_i, sk_i),密钥需要用户自己保护好,其中密钥

pk_i是公钥，需要发送给环中的每一个成员。

签名 Sign：签名算法是利用自己产生的密钥（pk_i，sk_i），以及收到环中所有人发送的公钥集合 $\boldsymbol{L}=\{pk_1,pk_2,\cdots,pk_n\}$对信息 M 进行签名，最后产生对应的环签名 σ。

验证 Verify：γ 验证算法输入签名 σ、消息 M 和环成员的公钥集合 $\boldsymbol{L}$，γ 输出 Valid 或者 Invalid。

环签名具有以下性质：

无条件匿名：如果攻击者可以获取所有环成员的私钥，攻击者是环外成员可能真正判断签名者的概率不超过$\frac{1}{n+1}$，攻击者是环内成员可能真正判断签名者的概率不超过$\frac{1}{n}$，其中 n 表示环成员的数量[18-19]。

不可伪造性：在攻击者没有环中任何成员私钥的条件下，攻击者可以伪造一个信息 M 的签名，但它是不可能通过验证者验证。

良好的特性：签名者可以根据自己的需要指定匿名范围，并可以对任何消息签名，自发形成一个环状，实现一个去中心化但拥有群签名的特性。

3　基于可跟踪环签名的共识算法

3.1　共识节点的域分布

首先将参与共识过程的所有共识节点分为两组 $\boldsymbol{X}=\{x_1,x_2,\cdots,x_n\}$和 $\boldsymbol{Y}=\{y_1,y_2,\cdots,y_n\}$。然后赋予每组节点在共识过程中投票拥有的权重，规定 $\boldsymbol{X}$ 组节点投票时拥有的权重 $Q_x=2$，$\boldsymbol{Y}$ 组节点投票时拥有的权重 $Q_y=1$。这里设定权重为 2 的节点是主域节点，权重为 1 的节点是副域节点。可以看出 $\boldsymbol{X}$ 组节点拥有较高的权重，本文设置其安全性比 $\boldsymbol{Y}$ 组高，且相比 $\boldsymbol{Y}$ 组有恶意节点的概率更小。如果有新的节点加入，该节点自动加入 $\boldsymbol{Y}$ 组。

由于给节点设置不同的投票权重，PBFT 算法的准则发生了一些变化。传统的 PBFT 算法都默认节点的投票权重为 1，在正常情况下，如果节点总数为 N，通信过程中系统允许最大的恶意节点数为 $f=\frac{N-1}{3}$。然而在本文的共识环境下，定义所有节点的投票权重的总和是 M（见式(1)），允许恶意节点的权重是 F（见式(2)）。

$$M=\sum_{i=1}^{n}Q_x x_i+\sum_{i=1}^{n}Q_y y_i \tag{1}$$

$$F=\frac{M-1}{3} \tag{2}$$

对于这里设定的新环境，在逻辑上可以认为 $\boldsymbol{X}$ 组的节点由两个 $\boldsymbol{Y}$ 组节点组成，如果拥有一个大的恶意节点，它可以联系其他恶意节点一起做恶。在 PBFT 算法中，它至少需要控制 $f+1$ 个节点，但是这里的节点权重不同，如果 $f+1$ 个节点都来自 $\boldsymbol{Y}$ 组，显然它控制的权重不能破坏共识系统，而且本文还设置了想控制 $\boldsymbol{X}$ 组节点所消耗的成本 Vx，控制 $\boldsymbol{Y}$ 组节点所消耗的成本是 Vy。控制权重越大的节点需要消耗的成本越高，且满足控制 $\boldsymbol{X}$ 组节点消耗的成本满足 $Vx\geqslant 2Vy$，这样就能提高共识系统的安全性。在传统的 PBFT 算法中，共识节点的权重是一样的，大的恶意节点可以选择安全性低的节点从而破坏系统网络。这里节点权重的划分消除了传统系统的缺点，使共识系统的安全性得以提高。同时将它们分组可以大大减少对可跟踪环签名算法的加解密时间，降低共识算法的时延。

3.2　主节点的选择

因为主节点出错需要切换视图，而确保视图一致性需要增加大量的通信开销，从而降低共识效率。传统 PBFT 算法是在整个共识系统随机选取主节点 $Primary=v\bmod|R|$，主节点出错概率大。本文提出的算法最初选择的主节点是在主域节点中随机选取的，选择好主节点后确定视图 v 共识。由于最后的主域节点都是由其他节点匿名选择的可信任节点，因此安全性较高。在主域节点中选择高权重节点作为主节点，出错的概率较小，视图切换频率将会降低，在减少通信的开销的同时提高了共识效率。

3.3　共识算法流程

算法运行流程图如图 2 所示，和传统的 PBFT 算法基本相似。

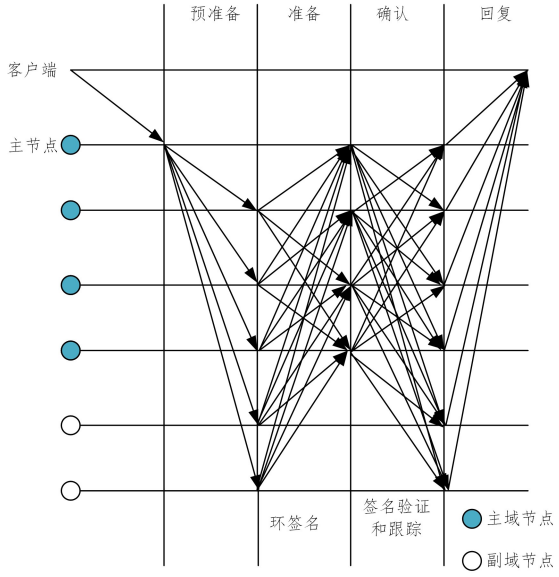

图 2　tracePBFT 算法流程图

Fig. 2　tracePBFT algorithm flowchart

Pre-Prapare 阶段：

primary 节点收到来自客户端的请求 m 后，它会将请求 m 打包排序以及编号 s。然后主节点将需要共识的提案请求在网络系统中进行广播。广播 RRE-PREPARE 消息：〈〈v，s，d，m〉，Pre-prepare〉，其中 v 表示视图的编号，s 是请求编号，d 是 m 的摘要。

Prepare 阶段：

当其他所有节点收到 Pre-prepare 消息后，会对 Pre-prepare 的消息是否达到共识要求（即要求验证摘要 d，判断 v 视图与当前视图是否一致，s 是否在设置的水平线范围内）进行验证，验证都通过之后，节点开始对准备消息投票以及进行可跟踪环签名处理。

这里假设在 $\boldsymbol{X}$ 组节点中进行选择，选择一个大的素数 q，$\boldsymbol{G}$ 是阶为 q 的循环乘法群，选择一个生成元 $g\in Z_q^*$。定义 3 个哈希函数 $H:\{0,1\}^*\rightarrow\boldsymbol{G}$，$H':\{0,1\}^*\rightarrow\boldsymbol{G}$ 和 $H'':\{0,1\}^*\rightarrow\mathbb{Z}_q$。其中一个节点 i 的密钥产生遵循下面方法：节点 i 随机选择一个私钥 $x_i\in Z_q^*$，计算节点公钥是 $P_i=g^{x_i}\bmod q$，因此 i 的公钥是 $pk_i=\{g,P_i,\boldsymbol{G}\}$以及私钥$=ski=\{pk_i,x_i\}$，将结构

组合成公钥群集合 $PKG_g=\{pk_i \mid i=1,2,\cdots,N\}$，节点可公开的系统参数 $\boldsymbol{P}=\{\boldsymbol{G},g,q,H,H',H'',PKG_g\}$。

现在节点 i 待加密的消息为 m，L 争议设定任意字符串是 $\{0,1\}^*$ 作为标记，节点 i 使用自己的私钥 sk_i 进行下面的签名投票操作：

(1)计算 $h=H(L)$ 和 $\alpha_i=h^{x_i}$。

(2)设 $U_0=H'(L,m)$ 和计算 $U_1=\left(\frac{\alpha_i}{U_0}\right)^{\frac{1}{i}}$。

(3)对于 $j=i+1,\cdots,n,1,\cdots,i-1$，计算 $\alpha_j=U_0U_1^j\in\boldsymbol{G}$，注意这里的每个 $(j,\log_h(\alpha_j))$ 被定义为由 $(0,\log_h(U_0))$ 和 (i,x_i) 组成的线上，这里的是 $x_i=\log_h(a_i)$。

(4)在 $(\boldsymbol{L},m)$ 中，产生签名 $(\boldsymbol{k}_N,\boldsymbol{d}_N)$，这里是基于零知识证明中衍生出的关系知识[20-21]：

$$\mathscr{L}\triangleq\{(L,h,\alpha_N)\mid \exists i'\in N \text{ 以致} \log_g(p_{i'})=\log_h(\alpha_{i'})\}$$

这里的 $\alpha_N=(\alpha_1,\alpha_2,\cdots,\alpha_n)$，因此：

1)任意选取 $\varphi_i\leftarrow\mathbb{Z}_q$，然后设 $r_i=g^{\phi_i}$，$t_i=h^{\phi_i}\in\boldsymbol{G}$。

2)任意选取 $k_j,d_j\leftarrow\mathbb{Z}_q$，计算设 $r_j=g^{d_j}P_i^{k_j}$，$t_j=h^{k_j}\alpha_j^{d_j}\in\boldsymbol{G}$ 这里 $j\neq i$。

3)设 $\boldsymbol{d}=H''(L,U_0,U_1,r_N,t_N)$，而这里的 $\boldsymbol{r}_N=(r_1,r_2,\cdots,r_n)$ 和 $\boldsymbol{t}_N=(t_1,t_2,\cdots,t_n)$。

4)设 $d_i=d-\sum_{j\neq i}d_j \pmod q$ 和计算 $k_i=\phi_i-d_ix_i \pmod q$。返回 $(\boldsymbol{k}_N,\boldsymbol{d}_N)$，这里的 $\boldsymbol{k}_N=(k_1,k_2,\cdots,k_n)$ 和 $\boldsymbol{d}_N=(d_1,d_2,\cdots,d_n)$ 证明 $\mathscr{L}$。

5)输出 $\boldsymbol{\alpha}_W(m)=(U_1,\boldsymbol{k}_N,\boldsymbol{d}_N)$ 是在 (L,m) 签名。

环签名完成之后，节点会向主域中所有节点广播 PREPARE 消息：$\langle\langle v,s,d,i,W\rangle,\boldsymbol{\alpha}_W(m),Kw,\text{PREPARE}\rangle$。$i$ 表示节点的身份，W 表示节点来自哪个分组，比如节点是 X 组，则 $W=X$，Kw 代表 W 组节点的公钥组成的环 PKG_W，$\boldsymbol{\alpha}_W(m)$，表示节点对 (L,m) 的环签名。

Commit 阶段：

当主域节点收到 PREPARE 消息后，还是需要判断 v,d 和 s 是否符合共识要求。然后判别该信息来自哪组，只可以确定哪个组形成的环签名，但是不知道组中节点的身份，这样就能够保护节点身份的安全。之后主域节点会将 PREPARE 消息的环签名进行解密操作，判别签署消息的节点是否来自 Kw 环内的节点，再进行可跟踪验证，保证同一个人不能进行两次投票，如果存在，将扣除该节点的权重为零。具体的操作步骤如下：

(1)判断 Kw 内部的节点是否来自同一个组，如果不在同一个组表示 Invlid，认为消息是无效的。否则继续下一步。

(2)对环签名消息进行验证，输入 L(其设定与前文一样)和公钥 PKG_W，首先检测 $g,U_1\in\boldsymbol{G},k_i,d_i\in\mathbb{Z}_q$ 和 $p_i\in\boldsymbol{G}(i\in N)$。$N$ 表示环成员数，计算 $h=H(L)$ 和 $U_0=H'(L,m)$。

(3)计算 $\alpha_i=U_0U_1^i\in\boldsymbol{G}(i\in N)$。

(4)计算 $r_i=g^{d_i}P_i^{k_i}$ 和 $t_i=h^{d_i}\alpha_i^{k_i}(i\in N)$。

(5)$H''(L,m,U_0,U_1,\boldsymbol{r}_N,\boldsymbol{t}_N)\equiv\sum_{i\in N}k_i \pmod q$ 是否成立，而这里的 $\boldsymbol{r}_N=(r_1,r_2,\cdots,r_n)$，$\boldsymbol{t}_N=(t_1,t_2,\cdots,t_n)$。

(6)如果上面的检测都成立，验证通过，否则投票无效并进行如下跟踪操作。

检测同一个节点是否两次投票，即设第一次投票的签名消息和第二次投票的签名消息一样，这里的 $\boldsymbol{\alpha}_G(m)=(U_1,\boldsymbol{k}_N,\boldsymbol{d}_N)$ 和 $\alpha_W'(m)=(U_1',\boldsymbol{k}_N',\boldsymbol{d}_N')$，输入与前面争议一样的 L，输入 PKG_g 进行检测。

1)计算 $h=H(L)$ 和 $U_0=H'(L,m)$。

2)计算 $\alpha_i=U_0U_1^i\in\boldsymbol{G}(i\in N)$。

3)用同样的方式求出 $\alpha_i'=U_0'U_1'^i\in G(i\in N)$。

4)然后判断 $\alpha_i=\alpha_{i'}$ 是否成立($i\in N$)，如果成立就能知道对应节点的公钥 P_i，这样可以对该节点进行惩罚，扣除该节点的权重为 0。

经过以上步骤的检验，就可以检测出是否有拜占庭节点，并且对其作出惩罚后，主域节点会计算收到的合法信息加权权量和，然后比较加权权量和是否大于 $2F+1$，若大于 $2F+1$，说明已经达到了拜占庭算法共识的门限条件。主域节点就可以对系统所有节点广播 COMMIT 消息，在广播时也会和上面一样对确认消息进行一次环签名，这里的主要节点是 X 组，节点签名步骤和前面的 Prepare 阶段一样，主域节点发送 COMMIT 消息。环签名后 COMMIT 消息为：$\langle\langle v,s,i,d\rangle,\boldsymbol{\alpha}_X(m),K_X,\text{COMMIT}\rangle$，其中 K_X 表示主域节点组成的公钥环 PKG_X。

Reply 阶段：当节点收到 COMMIT 消息之后，还是需要判断 v,d 和 s 是否符合共识要求，对消息的环签名进行验证操作，判断消息的合法性和签名是否由 X 组节点完成。如果验证通过，且节点收到来自主域节点发送合法消息大于 $2f_X+1$(f_X 表示容忍最大容错节点数，即 $f_X=\frac{(X-1)}{3}$，X 表示主域节点的节点数)，说明系统提案一致，所有节点会将提案信息写在区块链，最后回复客户端交易成功。

3.4 视图转换

虽然这里的算法选择的主节点可靠性高，但是还是需要考虑主节点出现宕机，或者在 commit 阶段一些节点收到 $2f_X+1$ 个其他节点发来确认消息，还有一些节点没有收到 $2f_X+1$ 个确认消息等导致分布式节点状态不一致性问题(f_X 表示主域节点可以容忍的最大拜占庭节点数)。节点间状态不同，出现这样的状况会触发视图切换，然后对数据不一致性进行修补，这一修补过程在分布式节点中进行。

当请求超时，主域节点进入视图 $v+1$，$v+1$ 是新主节点的视图，广播视图更改消息 $\langle\text{view-change},\langle v+1,s,C,P,i\rangle$，$s$ 表示最新稳定检查点编号，C 是稳定检查点证明，P 是这次副本节点未完成请求的 Pre-prepare 消息和 Prepare 消息的集合。

当主域中节点收到切换视图的信息，主域节点会自动形成一个环进行环签名处理，计算出主域节点中的消息加权重和，当消息的加权和大于 $2F+1$，此时确定了新的主节点，主节点可以向主域其他节点和副域节点发送 new-change 消息进行分布式数据一致性处理。

4 实验验证

Hyperledger Fabric 是一种区块链开发平台，支持模块化可插拔[22]，这里算法的实现用 Golang 语言。然后利用 Caliper(Fabric 中用于测试共识算法性能的框架)对 tracePBFT 算法进行性能测试，使用 docker 容器让每个节点隔离开来，

这样每个节点都有一个单独的空间，相互没有干扰，但是每个容器都包含着共识算法的整个流程，实验验证环境如表 1 所列。

表 1　环境配置情况

Table 1　Environment configuration

Project environment	Parameter configuration
CPU	Intel i5-5200U
Operating system	Ubuntu 20.04
Container system	docker 20.10.10
Fabric version	1.4.1
Caliper 版本	0.3.0

4.1　吞吐量测试

吞吐量指单位时间内系统能够打包交易数量，它可以衡量系统承受负载或者处理事务的能力。在测试共识算法性能方面，大部分都是用每秒交易数（Transaction Per Second，TPS）来表示系统吞吐量，即：

$$TPS=Transaction_{\Delta t}\ \Delta t \tag{3}$$

其中：Δt 表示交易提案开始到区块上链之间的时间。$Transaction_{\Delta t}$ 表示 Δt 的时间里完成的交易总数。最开始吞吐量随着每个区块的容量的增大逐渐增大，区块的容量增大会导致共识时间增加、系统的负载增大，当区块的容量增加到一定大小吞吐量反而会减少。为了便于测试，采用控制变量，设置每个区块容量中包含的交易数都是 500 个。实验测试 4 个节点、7 个节点、10 个节点、13 个节点和 16 个节点，并且每个节点是在一个单独的 docker 容器，投票阶段相互没有干扰，设置了只有 4 个主域节点的 5 组对照实验。经过多次测试取平均值，得出统计结果。

两种算法在相同条件下的吞吐量对比如图 3 所示。

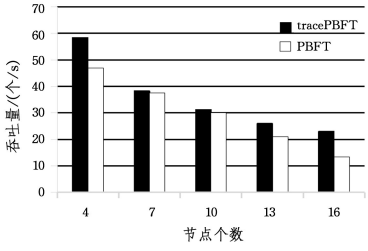

图 3　两种算法的吞吐量对比

Fig. 3　Throughput comparison of two algorithms

可以看出，随着节点数的增加两种算法的吞吐量都有不同程度的减少，但是改进的 tracePBFT 算法的吞吐量减少越来越缓慢，而且比原 PBFT 算法高出许多。在只有 4 个主域节点和测试 4 个节点时，tracePBFT 算法的吞吐量比 PBFT 算法高出许多，原因是它选取的主节点更加可靠，导致切换视图次数减少，同节点时吞吐量有所增加，在节点数量越多时增加越明显。主域节点只有 4 个，多节点选取主节点可靠度更高，切换视图的频率更低。

4.2　时延测试

共识时延（DelayTime）表示开始提案到区块被写入链所需要的时间，是衡量区块链性能的一个指标，共识的时延越低，系统处理事务越快，这样符合现实发展的需求。共识时延越小，区块链的安全性就越高。这里测试的共识时延为区块完成一次共识过程的时间，即：

$$DelayTime=Tconfirm-Ttransmit \tag{4}$$

其中，$Tconfirm$ 为区块已经确定上链的时间，$Ttransmit$ 为请求提案开始时间，这里的数据是取多次测试的平均值，得出的统计结果如图 4 所示。

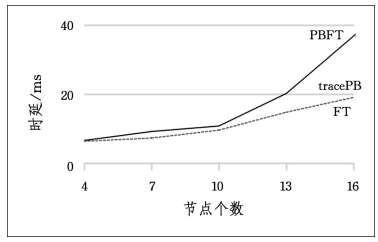

图 4　时延比较

Fig. 4　Time delay comparison

从图 4 可以看出 tracePBFT 的时延比 PBFT 低，但在 4 个节点、7 个节点和 10 个节点时差距不明显，主要是因为节点数太少了，选择的主节点的可靠性不高。随着节点数越来越多，选取的主节点可靠性越来越高，而 PBFT 的时延随着节点的增加而快速增加，原因是节点越多，一次共识通信复杂性增加了。而 tracePBFT 随着节点的增加时延增加缓慢，除了主节点选取可靠，减少切换视图，其中准备阶段和确认阶段是主域节点之间的通信，大大减少通信开销，从而提高共识效率。

4.3　可跟踪签名算法仿真分析

可跟踪环签名加解密，在这一部分中，我们对环签名算法的性能进行了仿真测试，判断可跟踪环签名加解密操作对 tracePBFT 算法的影响，然后采用相应的解决方案。我们使用长度为 1024 的公钥和私钥，并计算不同数目的环签名成员进行 10 次加密解密的时间，测试结果如图 5 所示。从图中可以得知，加密和解密所用的时间基本拟合，可以从本质上看出，解密是加密的逆运算过程。加密解密的时间在共识算法节点比较少时，对共识算法时间开销的影响较小。对不同数量的节点进行加解密是一个线性函数，随着节点数量的增加，时间也在增加，因此我们可以对节点进行分组，使共识算法过程中可跟踪环签名加密和解密的时间减少。本文提出的可跟踪环签名算法使用分组加密与解密，其中 Y 组不需要进行解密，X 组需要加解密操作，而主域节点较少，这样可以大大减少本文共识算法的时间开销。

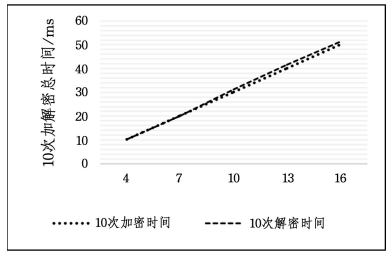

图 5　可跟踪环签名算法加解密的时间

Fig. 5　Encryption and decryption time of traceable ring signature algorithm

4.4　算法安全性

（1）首先给出可跟踪环签名的安全性，下面是证明方案。

这里的可跟踪环签名方案，在保证其密钥的安全性的同时，也可以证明这里的方案是标志性链接环签名（前面的标志

L),这里的证明会用以下的引理。我们认为敌手 A 攻击我的签名方案,A 给出一个 1^k 和在随机预言模型机下的 H' 和 H'' 以及较多的时间 $s_{H'}$ 和 $s_{H''}$。这里的 A 没有必要多项式时间是有界。如果允许敌手 A 输出有效对(L,m,α)去伪造签名,需要达到下面的要求。

1)要求$\{i\in N\mid \log_h(\alpha_i)=\log_g(P_i)\}<1$ 的概率最多是 $\frac{s_{H''}}{s}$。

2)要求$\{i\in N\mid \log_h(\alpha_i)=\log_g(P_i)\}>1$ 的概率最多是 $\frac{s_{H'}}{s}$。

这里敌手 A 的概率的选择由 H'、H''和抛硬币决定。

证明:事件 1 $\{i\in N\mid \log_h(\alpha_i)=\log_g(P_i)\}<1$:证($L$,$m$,)=1 表示 $r_i=g^{d_i}P_i^{k_i}\in G$ 和 $t_i=h^{d_i}\alpha_i^{k_i}\in G(i\in N)$,这就意味着 $\log_g(r_i)=k_i+d_i*\log_g(P_i)$ 和 $\log_h(t_i)=k_i+d_i*\log_h(\alpha_i)(i\in N)$。注意,如果 $\log_g(P_i)\neq\log_h(\alpha_i)$,$d_i$ 是目标,则事件 1 对于所有的 $d_i(i\in N)$ 是唯一的目标。由于 H'' 是随机预言模型,每个都给出(L,m,U_0,U_1,r_i,t_i),敌手 A 在最多可以 $s_{H''}$ 数量询问随机预言模型 H'' 的情况下,得出 $H''(L,m,U_0,U_1,r_N,t_N)\equiv\sum_{i\in N}k_i(\bmod q)$ 的概率最多是 s^{-1},而得出事件 1 的概率最多是 $\frac{s_{H''}}{s}$。

事件 2 $\{i\in N\mid \log_h(\alpha_i)=\log_g(P_i)\}>1$:由 $\alpha_i=U_0U_1^i\in \boldsymbol{G}(i\in N)$,每个点$(i,\log_h(\alpha_i))(i\in N)$是在 $y=\log_h(U_1)+\log_h(U_0)$线上。事件 2 表示至少有两个点$(i,\log_g(P_i))$是在这线上。这意味着 PKG_N 是固定的,这条线目标,因此 $\log_h(U_1)$和$\log_h(U_0)$是目标。虽然我们也需要 $\log_h(U_0)=\log_h(H'(L(issue,PKG_N),m))$,这里的 $H'(L,m)$是目标,不依靠上面的线,因为 H'是随机预言模型。而敌手 A 在最多 $s_{H'}$ 数量询问随机预言模型 H'的情况下,这里的$\log_h(H'(L,m))=\log_h(U_0)$的概率最多是 s^{-1}给(L,m)。而事件 2 的概率最多是$\frac{s_{H'}}{s}$。

由此可以看出,可跟踪环签名算法在随机预言模型机下证明攻击者无法在合理时间内输出合法签名,保证了节点的安全性,在不知道其他节点身份的情况可以公平地投票,使共识算法过程选出来的主节点更加可靠,减少了更换主节点和视图的次数,合理地减少了更换视图通信时间,降低了时延。

(2)无条件匿名性

和其他签密方案不同的是,该方案保护各节点签名者的匿名性。验证者获得签名信息后可以验证其合法性,从而肯定其是否为来自特定环的某个成员所签署的签名信息,但是无法确定签名者具体是谁。本文提出的算法产生了一个正常的环签名 α,在计算r_j,t_j以及最后的d_j时都需要两个随机需求的值k_j,$d_j\leftarrow\mathbb{Z}_q$,经过计算$r_j=g^{d_j}P_j^{k_j}$,$t_j=h^{k_j}\alpha_j^{d_j}$,$d=H''(L,U_0,U_1,r_N,t_N)$组合而成。在此过程中,签名环中的成员在全部节点上是均匀分布的,因此攻击者推测出签名者真实身份的概率为 $1/n$。即使攻击者位于签名环内,是环签名 L 中的成员之一,但攻击者将自身排除在外后对真实签名者身份的命中概率依然为 $1/(n+1)$。

(3)可跟踪性

如果节点希望同时投票两个不同的节点信息m'和 m,首先需要验证签名是否合法,如果合法,对算法进行检测,利用标签 *issue* 和各节点的公钥PK_i对每个节点 i 进行检测,通过同样的方式产生两个的签名α_i和α_i',$i\in N$,判断两个签名是否相等,若相等,保存对应节点的公钥PK_i,则寻找到了一个节点进行了违规操作,两个消息由一个节点签署,进行“双花”操作,这里可以得到这个节点的公钥PK_i。

结束语 本文利用可跟踪环签名方案对 PBFT 算法进行改进,提出基于可跟踪环签名的拜占庭容错共识算法。在准备阶段利用环签名处理,这样节点可以自由地组成一个环,它可以保证节点间的安全性,利用环签名投票选择可靠的节点。然后根据可跟踪环签名的特性,在确认阶段对签名投票的节点进行认证,发现存在恶意投票和“两次花费”的节点进行惩罚,使选取的主节点更加可靠,降低主节点出错导致的视图切换的频率,同时利用权重选择减少在确认阶段的通信开销。实验结果表明,本文算法解决了节点间安全性、主节点随机选择、通信开销大和节点动态加入和退出的问题,比较适合较大的联盟链网络。

参 考 文 献

[1] NAKAMOTO S. Bitcoin: A peer-to-peer electronic cash system [EB/OL]. https://bitcoin.org/bitcoin.pdf, 2008.

[2] ZHANG A, BAI X Y. Overview of Research and Practice of Blockchain Privacy Protection[J]. Journal of Software, 2020, 31(5): 1406-1434.

[3] HAN X, YUAN Y, WANG F Y. Blockchain Security Issues: Research Status and Prospect[J]. Acta Automatica Sinica, 2019, 45(1): 206-225.

[4] XIA Q, DOU W, GUO K W, et al. Overview of Blockchain Consensus Agreements[J]. Journal of Softwar, 2021, 32(2): 277-299.

[5] ZHENG M, WANG H, LIU H, et al. Survey on Consensus Algorithms of Blckchain[J]. Netinfo Security, 2019, 19(7): 8-24.

[6] LAMPOR T L, SHOSTAK R, PEASE M. The Byzantine generals problem[J]. ACM Transactions on Programming Languages and Systems, 1982, 4(3): 382-401.

[7] WANG W B, HONG D T, HU P Z. A Survey on Consensus Mechanisms and Mining Management in Blockchain Network [J]. IEEE Access, 2019, 7: 22328-22370.

[8] GAN J, QIANG L I, CHEN Z, et al. Improvement of blockchain practical Byzantine fault tolerance consensus algorithm[J]. Journal of Computer Applications, 2019, 26: 45-55.

[9] HAO X, YU L. Dynamic Practical Byzantine Fault Tolerance [C]// 2018 IEEE Conference on CNS. 2018: 1-8.

[10] HAN S C, ZHU X R, ZHANG X X. A secure and efficient decentralized conditional anonymous payment system based on blockchain[J]. Journal of the Internet of Things, 2020, 4(2): 18-25.

[11] TANG H, LIU S, JIU Y H, et al. Improvement of Practical Byzantine Fault Tolerance Algorithm[J/OL]. Computer Engineering and Application. https://kns.cnki.net/kcms/detail/11.2127.TP.20210927.2052.010.html.

[12] FANG Y B, ZHOU C M, LI S, et al. Improvement of Practical Byzantine Fault Tolerance Algorithm in Alliance Chain[J].

Computer Engineering and Applications，2022，58(3)：135-142.

[13] WANG R H，XIN C Y，XU Q Q，et al. Byzantine Fault-tolerant Algorithm with Supervision Mechanism[J]. Computer Engineering and Applications，2021，57(18)：142-148.

[14] FANG Y，DENG J Q，CONG L H，et al. An Improved PBFT Blockchain Consensus Algorithm Based on Ring Signaturg[J]. Computer Engineering，2019，45(11)：32-36.

[15] ZHANG L S. Research on Blockchain Consensus Algorithm Based on Byzantine Fault Tolerance Algorithm[D]. Chengdu：University of Electronic Science and Technology，2020.

[16] CASTRO M，LISKOV B. Proctical Byzantine fault Tolerance [C]// Operating Systems Design and Implementation(OSDI). New Orleans，1999：173-186.

[17] RIVEST R L，SHAMIR A，TAUMAN Y How to leak a secret [C]// International Conference on the Theory and Application of Cryptology and Information Security. 2001：552-565.

[18] LIN C，HE D，HUANG X，et al. DCAP：a secure and efficient decentralized conditional anonymous payment system based on blockchain[J]. IEEE Trans. Inf. Forens. Secur.，2020(15)：2440-2452.

[19] LI X F，MEI Y R，GONG J. A blockchain privacy protection scheme based on ring signature[J]. IEEE Access，2020，8：76765-76772.

[20] MA L M，ZHANG W，LIU X Y. Research and Design of A Secure Wireless Body Area Network Medical Information management system[J]. Netinfo Security，2019(5)：38-46.

[21] YU R W. ZHOU B X，WANG L，et al. Research on Zero Knowledge Location Proof Method Based on Blockchain[J]. Journal of Electronics and Information，2020，42(9)：2142-2149.

[22] KUZLU M，PIPATTANASOMPORN M，GURSESL. Performance Analysis of a Hypereldger Fabric Blockchain Framework：Throughput，Latency and Scalability[C]// Proceedings of the 2019 2nd IEEE International Conference on Blockchain. Piscataway：IEEE，2019：536-540.

TU Jun，born in 1996，postgraduate. His main research interests include blockchain and cryptography.

JIA Dongli，born in 1972，Ph. D，associate professor. His main research interests include intelligent signal processing and so on.