基于 Gossip 协议的拜占庭共识算法

张仕将[1] 柴 晶[1] 陈泽华[1] 贺海武[2]
(太原理工大学信息工程学院 太原 030024)[1] (中国科学院计算机网络信息中心 北京 100190)[2]

摘 要 区块链是一种对等网络的分布式账本系统,具备去中心化、不可篡改、安全可信等特点,因此受到了广泛关注。在区块链系统中,典型的拜占庭错误包括操作错误、网络延迟、系统崩溃、恶意攻击等。现有共识算法不仅对区块链中拜占庭节点的容错能力低,而且对区块链系统的可扩展性差。针对这一问题,文中提出了基于 Gossip 协议的拜占庭共识算法,使系统可以容忍小于一半的节点为拜占庭节点,能够达到 XFT 共识算法的容错能力。同时,因为采用了统一的数据结构,所以系统具有更好的可扩展性,并且有利于正确节点识别区块链系统中的恶意节点。在该算法中,提案节点随着区块链长度的变化而转移,系统中所有节点都处于对等的地位,从而避免了单点故障问题,进而使得系统具有更好的动态负载均衡的性能。

关键词 区块链,拜占庭错误,共识算法,Gossip 协议,可扩展性
中图法分类号 TP302.8 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2018.02.004

Byzantine Consensus Algorithm Based on Gossip Protocol

ZHANG Shi-jiang[1] CHAI Jing[1] CHEN Ze-hua[1] HE Hai-wu[2]
(Department of Information Engineering, Taiyuan University of Technology, Taiyuan 030024, China)[1]
(Computer Network Information Center, Chinese Academy of Sciences, Beijing 100190, China)[2]

Abstract Blockchain is a kind of distributed ledger system with peer-to-peer network, which has drawn widespread attention because of its characteristics such as decentralization, non-tempering, security and credibility. In a blockchain system, some nodes have the Byzantine errors such as operational errors, network latency, system crashes, malicious attacks, and so on. The existing consensus algorithms are less tolerant to the Byzantine nodes in the blockchain, and the scalability of the blockchain system is poor. In order to solve these problems, this paper proposed a Byzantine consensus algorithm based on Gossip protocol, which allows the system to tolerate less than half of the nodes as the Byzantine node and achieve the fault-tolerant performance of XFT consensus algorithm. This paper proved that the algorithm can reach consensus in a distributed system with Byzantine defects from the agreement, correctness and termination. At the same time, the system adopts the uniform data structure, and thus has better scalability and facilitates the right node to identify the Byzantine nodes in the blockchain system. In this algorithm, the proposed node is shifted with the change of the length of blockchain, so that all nodes in the system are in the same position, thus avoiding the single point of failure problem, and making the system have better dynamic load balancing performance.

Keywords Blockchain, Byzantine error, Consensus algorithm, Gossip protocol, Scalability

1 引言

区块链技术是利用加密链式的区块结构来验证与存储数据,利用分布式节点共识算法来生成和更新数据,利用自动化脚本代码来编程和操作数据的一种全新的去中心化基础架构与分布式计算范式[1]。Nakamoto[2]于 2008 年在密码学邮件列表中提出了比特币的概念。比特币是目前为止区块链技术最为成功的应用。由于区块链技术有去中心化、不可篡改、安全可信等特点,因此出现了大量基于区块链技术的应用,如众筹区块链上的智能合约、基于区块链的支付系统、基于区块链的股权交易系统等。在区块链系统中,由于某种原因引起的操作错误、网络延迟、系统崩溃、恶意攻击等都会引起系统错误,这样的错误被称为拜占庭错误[3]。为了避免此错误,区块链系统引入了共识机制,要求每一个节点都有唯一一个公认的全局账本。

2017 年 5 月 12 号爆发的勒索比特币病毒使得比特币再

到稿日期:2017-11-20 返修日期:2018-01-10 本文受中科院计算机网络信息中心百人计划项目(1101002001),国家自然科学基金(61402319,61403273),山西省自然科学基金项目(2014021022-4)资助。
张仕将(1993—),男,硕士,CCF 会员,主要研究方向为区块链技术、机器学习;柴 晶(1983—),男,博士,讲师,主要研究方向为机器学习、数据挖掘;陈泽华(1974—),女,博士,教授,主要研究方向为工业大数据、智能信息处理及应用;贺海武(1977—),男,博士,研究员,CCF 会员,主要研究方向为区块链分布式系统、分布式及并行计算,E-mail:hehaiwu@gmail.com(通信作者)。

次受到广泛关注。比特币的底层采用了区块链的技术架构[4]。比特币采用竞争记账的方式达成共识，竞争结果的判定由工作量证明机制(Proof of Work，PoW)[1]来完成。但是，PoW 共识机制具有很明显的缺点，例如需要巨大的电力成本、共识效率低、资源浪费、有分叉的可能性等。实质上，PoW 共识算法是一种概率性的拜占庭协议[4]。Micali 提出的 Algorand 算法通过密码学的方法决定下一个区块的生产者，从而减小了网络出现分叉的可能性[5]。点点币采用了权益证明(Proof of Stake，PoS)达成共识[6]。PoS 引入了币龄的概念，币龄能反映一个用户在交易时刻拥有的货币数量。PoS 共识机制弥补了 PoW 共识机制共识效率低的缺点，可以在更短的时间内达成共识。但是，该机制由于采用了权益结余，可能会导致某些账户的权利很大。文献[7]中提出了实用拜占庭容错算法(Practical Byzantine Fault Tolerance，PBFT)。在 PBFT 算法中，客户端的请求需要经过请求、序号分配、交互、序号确认、响应 5 个阶段才能完成，且需要 $3f+1$ 个节点才能容忍 f 个拜占庭节点。在 PBFT 中，一旦主节点发生错误，就会触发视图更新和所有节点的视图变化，这在一定程度上影响了系统的性能。Raft 是一种用于管理复制日志的共识算法[8]，其通过选举 leader 节点来完成日志复制、记账等操作，leader 节点崩溃后，会重新选举 leader 节点，这将引发系统震荡，从而降低系统的稳定性。在 IBM 主导的 Hyperledger 中，PBFT 和 Raft 是可选的共识算法。

本文提出了一种在区块链系统中基于 Gossip 协议的拜占庭共识算法(Gossip protocol-based Byzantine Consensus algorithm，GBC)。相对于前人的工作，本文算法具有以下特点：1)可以提高系统的容错性能，即系统可以容忍小于一半的节点为拜占庭节点；2)具有可扩展性，本文算法能够保证在系统规模增大时，节点间仍能高效率地传递信息；3)动态负载均衡，系统中的提案节点随着区块链长度的变化而转移，不会因主节点或者 leader 节点变更而影响系统的性能，具有较好的稳定性。

2　预备知识

2.1　Gossip 协议

类似于人群中的谣言传播，Gossip 协议是一种按照自己的需求自行选择邻近节点并与之交换信息的通信方式[9]。协议的大致思想为：节点 A 的数据得到更新以后，其会将最新的状态发送给它的邻居节点 B。节点 A 如果在一段时间内没有得到更新，则会主动要求邻居节点 B 向其推送数据，从而进行被动更新。经过几轮交换以后，最终能够使所有的节点都获得最新的数据。

在 Gossip 协议中，每个节点均有一个自己的视图，视图中包含其邻近节点的信息。在每轮通信中，每个节点会在其视图中选择若干个目标节点进行通信。通信方式有 3 种[10]：

1)Push-gossip，即节点 A 选择节点 B 为目标节点，并将信息发送给节点 B，节点 B 根据收到的信息进行更新；

2)Pull-gossip，即节点 A 选择节点 B 为目标节点，节点 B 将信息发送给节点 A，节点 A 根据收到的信息进行更新；

3)Push-Pull gossip，即 Pull-gossip 和 Push-gossip 的结合。

文献[11-12]证明了 Gossip 协议的正确性，文献[13]证明了 Gossip 协议可以保证信息被高效率地传播。

2.2　区块链系统模型

区块链是一个分布式账本系统[14]，系统内的节点基于 Gossip 协议进行通信。系统中存在两种角色：提案者和验证者。提案者负责广播提案，系统中的节点轮流扮演提案者的角色。在每个提案的过程中，有且仅有一个提案者；剩余节点为验证者，验证者验证提案的正确性并对其进行计算。提案者和验证者把最后的共识结果都记入自己的账本中。

为了避免信息在传递过程中被篡改，引入数字签名技术，即要求节点在发送信息时加入自己的签名。在系统中，假设节点可以随意地离开、加入系统。节点的编号从 0 到 $N-1$，区块链长度从 1 开始。节点会定期向其视图中的其他节点发送信息，以确保其他节点在线，如果发现其视图中的某一节点不在线，则向网络发送节点变更信息。

验证节点对提案的验证包括以下方面[15]：

1)提案内容是否符合系统规则，如果符合，则判定为合法；

2)提案中的交易是否已经存在于区块链中，如果没有存在，则判定为合法；

3)提案中的交易是否有多重支付，如果没有，则判定为合法；

4)提案中交易的所有合约脚本是否都正确执行，如果都正确执行，则判定为合法；

5)如果以上判定均合法，则提案正确。

3　基于 Gossip 协议的拜占庭共识算法

3.1　数据结构

本文提出的基于 Gossip 协议的拜占庭共识算法中设定了 3 种数据结构：本地数据 LD，全局数据 GD，共识数据 CD。LD 为节点对于输入计算的值，GD 是由 N 个节点的 LD 值组成的一维向量，CD 为最终的一致值。

定义 1　节点计算自身的 GD 向量中超过半数的一致元素，此元素为 CD。

定义 2　设向量 GD_i 和 GD_j 分别为节点 i 和节点 j 的全局数据。若 $GD_i[x]\neq GD_j[x]$ 且 $GD_j[x]\neq\emptyset$，置 $GD_i[x]=GD_j[x]$，则 GD_i 中的第 x 个元素得到更新；若 $GD_j[x]\neq GD_i[x]$ 且 $GD_i[x]\neq\emptyset$，置 $GD_j[x]=GD_i[x]$，则 GD_j 中的第 x 个元素得到更新。

在 2.1 节的每一种通信方式中，通信双方的节点有一个是主动更新数据，称为主动节点；有一个是被动更新数据，称为被动节点。主动节点算法和被动节点算法分别如算法 1 和算法 2 所示。

算法 1　主动节点算法

输入：提案

输出：共识数据(CD)

步骤 1　计算节点自身的 LD，并赋值给 GD 向量中对应的元素。

步骤 2　若通信方式为 Push-gossip，则将自身 GD 向量推送给目标节点；若通信方式为 Pull-gossip，则接收目标节点发送的 GD 向量，更新自身 GD 向量(根据定义 2)。

步骤 3　重复步骤 2，直至 GD 向量中无空值。

步骤 4　计算共识数据(根据定义 1)。

步骤 5 输出共识数据,结束。

算法 2 被动节点算法

输入:提案

输出:共识数据(CD)

步骤 1 计算节点自身的 LD,并赋值给 GD 向量中对应的元素。

步骤 2 节点监听自身是否被选作目标节点。若被选为目标节点,且通信方式为 Pull-gossip,则将自身的 GD 向量推送给对方;若通信方式为 Push-gossip,则接收对方发送的 GD 向量,并更新自身的 GD 向量(根据定义 2)。

步骤 3 重复步骤 2,直至通信结束。

步骤 4 计算共识数据(根据定义 1)。

步骤 5 输出共识数据,结束。

3.2 算法描述

区别于其他共识算法,GBC 算法的提案由参与共识的节点轮流提出。选取提案节点的方法为:$j=(h-i)\%N$,其中 j 为提案节点的编号,h 为当前区块链的长度,N 为参与共识的总节点数,i 表示当前区块链长度下共识过程中的轮数,% 为求余符号。当新的区块加入到区块链时,将 i 重置为 0。提案的内容包括交易、当前区块链长度 h、当前区块链长度下共识过程中的轮数 i 和上一个区块提交的 GD 向量。共识网络根据规则选择提案节点,提案节点从交易池中获取提案所需的交易并广播提案。验证节点收到提案节点广播的提案后,首先对提案节点和提案进行验证,如果提案节点和提案都正确,则计算 LD,否则 LD 为 False。节点之间通过主动节点算法和被动节点算法更新各自的 GD 向量,节点会检查其是否达成共识。若达成共识且共识的结果不是 False,则提案生成区块,加入区块链中;若共识的结果为 False 或者没有达成共识,则重新进行提案。如果提案节点或提案错误,则进入共识的新一轮,重新选择提案节点进行提案。具体算法步骤如算法 3 所示。

算法 3 基于 Gossip 协议的拜占庭共识算法

输入:提案

输出:新的区块链

步骤 1 选择提案节点。

步骤 2 提案节点广播提案。

步骤 3 验证节点验证提案节点和提案,若提案节点或提案错误,则进入第 i+1 轮共识,重复步骤 1。

步骤 4 根据算法 1 和算法 2 得出 CD。

步骤 5 检查是否达成共识,若达成共识且共识结果不是 False,则发布完整区块,否则重复步骤 1。

步骤 6 将区块加入区块链。

步骤 7 输出新的区块链,结束。

4 实例分析

共识网络的系统设置如表 1 所列。共识网络共有 5 个节点(A,B,C,D,E),每个节点的视图中包含 2 个邻居节点,其中节点 A 和节点 C 为拜占庭节点,节点 B、节点 D 和节点 E 为正确节点,节点关系满足 $N\geqslant 2f+1$,N 为总节点数,f 为拜占庭节点数。节点在每轮通信中选择 1 个目标节点进行通信,且采用 Push-Pull gossip 通信方式。设当前区块链的长度为 3,正确节点对提案计算的 LD 为 m,根据 GBC 算法,应由节点 D 广播提案。

表 1 算法实例的系统设置

Table 1 System setting of algorithm instance

Node	View	LD	BFT
A	B,C	any	yes
B	E,D	m	no
C	B,D	any	yes
D	A,C	m	no
E	C,D	m	no

节点间利用 Gossip 协议进行通信,第一轮通信结果如表 2 所列。序号 1 表示节点 A 将视图中的节点 B 选为目标节点进行通信,由于节点 A 为拜占庭节点,因此对接收到的提案计算的 LD 可以为任意值。因为节点在发送信息时加入了自己的签名,所以拜占庭节点不能伪造或者篡改 GD 向量中其他节点的 LD。节点 A 和节点 B 分别将自己的 GD 向量发送给对方,更新 GD 向量,得到一致结果[f,m,−,−,−]。序号 2−5 的分析类似。

表 2 第一轮通信结果

Table 2 Communication results of the first round

Number	Node	Initial GD	Output GD
1	A	[f,−,−,−,−]	[f,m,−,−,−]
	B	[−,m,−,−,−]	
2	B	[f,m,−,−,−]	[f,m,−,−,m]
	E	[−,−,−,−,m]	
3	C	[−,−,s,−,−]	[f,m,s,−,m]
	B	[f,m,−,−,m]	
4	D	[−,−,−,m,−]	[f,m,−,m,−]
	A	[f,m,−,−,−]	
5	E	[f,m,−,−,m]	[f,m,s,−,m]
	C	[f,m,s,−,m]	

在第一轮通信结束后,由于节点的 GD 向量有空值,因此进行第二轮通信,结果如表 3 所列。第二轮的分析与第一轮的分析类似。从表 3 可知,序号 2 表示在操作结束以后,节点 B 和节点 D 可以判定节点 A 和节点 C 为拜占庭节点。序号 3 和序号 4 表示在操作结束以后,节点 D 可以判定节点 C 为拜占庭节点。序号 5 表示在操作结束以后,节点 E 和节点 D 可以判定节点 A 和节点 C 为拜占庭节点。该实例表明,仅经过两轮的通信,所有节点的 GD 向量就没有空值,能够根据 GD 向量得出超过一半的一致元素 m,并且能识别出拜占庭节点。因此,在系统存在小于一半的节点为拜占庭节点的情况下,正确节点能对提案达成共识,达到了拜占庭容错的目的。

表 3 第二轮通信结果

Table 3 Communication results of the second round

Number	Node	Initial GD	Output GD
1	A	[w,m,−,m,−]	[f&w,m,q,m,m]
	C	[f,m,q,−,m]	
2	B	[f,m,s,−,m]	[f,m,s,m,m]
	D	[f,m,−,m,−]	
3	C	[f&w,m,q,m,m]	[f&w,m,s&q,m,m]
	D	[f,m,s,m,m]	
4	D	[f&w,m,s&q,m,m]	[f&w,m,s&q,m,m]
	C	[f&w,m,s&q,m,m]	
5	E	[f,m,s,−,m]	[f&w,m,s&q,m,m]
	D	[f&w,m,s&q,m,m]	

5　算法分析

5.1　正确性分析

若要在一个存在拜占庭缺陷的分布式系统中达成共识，则 GBC 算法需要满足以下 3 个条件[4]。

1）GBC 算法符合一致性条件，即所有正确节点都必须同意同一个值。

证明：（反证法）假设所有正确节点同意的值不一样，即 CD 不同。设系统有 M 个正确节点、f 个拜占庭节点（$M\geqslant f+1$），其中节点 i 和节点 j 为正确节点，节点 i 和节点 j 收到的为同一个正确提案。节点 i 计算 LD，并将其赋值给 $GD[i+1]$。经过若干轮通信以后，节点 i 通过 GD 向量的更新，将其计算的 LD 赋值给其他节点 GD 向量的 $GD[i+1]$。同样，对于节点 j，经过更新以后，将其计算的 LD 赋值给其他节点 GD 向量的 $GD[j+1]$。因为节点 i 和节点 j 为正确节点，所以 $GD[i+1]$ 等于 $GD[j+1]$。在共识网络中有超过一半的节点为正确节点的情况下，每个节点的 GD 向量中也有超过一半的值为正确值，因此节点 i 与节点 j 就 GD 向量达成共识并得到相同的 CD。同理，在遵守 GBC 算法的前提下，对于系统中的 M 个正确节点也能达成共识，并得到相同的 CD。这与假设相矛盾，因此 GBC 算法满足一致性得证。证毕。

2）GBC 算法符合正确性条件，即如果所有正确节点有相同的初始值，那么所有的正确节点所同意的必须是同一个初始值。

证明：①设系统有 M 个正确节点、f 个拜占庭节点（$M\geqslant f+1$），提案者为正确节点。当区块链长度增加时，新的提案者会提出新的提案，所有正确节点接收提案并验证提案者和提案。若提案正确，每个正确节点都会对提案计算自己的 LD，并将其赋值给 GD 中对应的元素。经过 Gossip 协议通信，M 个正确节点对其 GD 向量达成共识并得到相同的 CD，此时 CD 等于正确节点对提案计算的 LD。因此，在所有正确节点收到相同的提案时，所有的正确节点所同意的是同一个提案。

②如果提案者为拜占庭节点，则其提出的提案错误或提出不同的提案。每个正确节点 GD 向量有超过一半的值为 False，因此最终 M 个正确节点对其 GD 向量达成共识并得到一个确定的值，即 False。共识网络会重新选择提案节点进行提案。因此，当系统有 M 个正确节点、f 个拜占庭节点时（$M\geqslant f+1$），GBC 算法满足正确性。证毕。

3）GBC 算法符合可结束性条件，即每个正确节点必须最终确定一个值。

证明：①当系统只有 1 个节点且其为正确节点时，该节点计算的 LD 即为 GD 向量中唯一的值，因此可以直接得出 CD。GBC 算法满足可结束性要求。

②假设系统有 M 个正确节点、f 个拜占庭节点（$M\geqslant f+1$），其中节点 i 和节点 j 为正确节点。当节点 i 和节点 j 收到同一个正确的提案时，节点 i 计算提案的 LD，且其他正确节点 GD 向量中的 $GD[i+1]$ 元素也为节点 i 所计算的 LD；同理，其他正确节点 GD 向量中的 $GD[j+1]$ 的元素也为节点 j 计算的 LD。最终，在每个节点的 GD 向量中有 M 个正确节点计算的 LD，使得 M 个正确节点在各自的 GD 向量中都有 M 个正确的元素。因为 $M\geqslant f+1$，所以每个正确节点的 GD 向量都有超过一半的一致元素，因此最终 M 个正确节点对其 GD 向量达成共识，得到一个确定的值，即 CD。当节点 i 和节点 j 收到的是错误的提案时，正确节点的 LD 为 False，每个正确节点 GD 向量有超过一半的值为 False，因此最终 M 个正确节点对其 GD 向量达成共识，得到一个确定的值，即 False。共识网络会重新选择提案节点进行提案。因此，当系统有 M 个正确节点、f 个拜占庭节点时（$M\geqslant f+1$），GBC 算法满足可结束性。证毕。

推论 1　系统可以容忍小于一半的节点是拜占庭节点。系统中的节点数满足 $N\geqslant 2f+1$，N 为总节点数，f 为拜占庭节点数。

证明：节点之间通过主动节点算法和被动节点算法交换信息。Gossip 协议能够保证信息被高效率地传播，在达到一定轮次的信息交换之后，所有正确节点根据自己的 GD 向量确定最终的 CD。假设系统有 K 个正确节点，正确节点最终得到含有 K 个一致元素的 GD 向量。如果 K 大于节点数的一半，即在 GD 向量中有超过一半的元素为同一元素，则 GD 向量为有效向量，可以得到 CD；否则达不成共识。因此，系统的总节点数 N 决定了系统中可以存在的拜占庭节点数 f。当 N 为偶数时，GBC 算法容许系统中存在的拜占庭节点数为 $\lfloor N/2\rfloor-1$；当 N 为奇数时，GBC 算法容许系统中存在的拜占庭节点数为 $\lfloor N/2\rfloor$。证毕。

综上所述，在一个有拜占庭缺陷存在的分布式系统中，GBC 算法满足达成共识的条件，证明了 GBC 算法的正确性和有效性，并且对推论 1 的证明，说明 GBC 算法能够容忍小于一半的节点为拜占庭节点，其具有较好的容错性能。

5.2　可扩展性分析

节点增加到共识网络时，寻找邻近节点来构造自己的视图，向视图内的节点发起同步区块链请求，并发送节点变更信息。节点离开共识网络时，在自己的视图内发送节点变更信息。每个节点收到节点变更信息时，通过 Gossip 协议通告共识网络，改变 GD 的数据结构。传统拜占庭容错算法的消息复杂度为 $O(N^2)$，对于节点比较多的网络是不可扩展的。由于 Gossip 协议本身具有较好的可扩展性，文献[16]已经证明当系统的规模增大时，适当增加与视图中通信节点的个数仍然能够保证节点间高效率地传递信息。因此，GBC 算法提高了共识网络的可扩展性。

结束语　本文针对区块链系统中的拜占庭问题，提出了一种基于 Gossip 协议的拜占庭共识算法。该算法采用了统一的数据结构，由提案节点提出提案，验证节点计算提案的 LD，节点间通过 Gossip 协议进行通信，各节点通过计算其 GD 向量中的元素达成共识。本文通过实例分析、数学证明、正确性分析说明了算法的正确性、有效性和可行性。该算法

的优势体现在以下 3 个方面:1)提高了系统的容错性能,相比于 PBFT,本文算法可以容忍小于一半的节点为拜占庭节点,可以达到 Hyperledger 中 XFT 共识算法相同的容错能力;2)具有较好的可扩展性,相比于传统的拜占庭共识算法,本文算法能够保证在系统规模增大时,节点间仍能高效率地传递信息;3)系统中的提案节点随着区块链长度的变化而转移,使系统具有均衡动态负载的性能。系统中所有节点都处于对等的地位,从而解决了单点故障问题。本文工作对区块链技术的共识算法在容错性和可扩展性两个方面的研究提供了一定的参考。

参 考 文 献

[1] YUAN Y, WANG F Y. Blockchain: The State of the Art and Future Trends [J]. Acta Automatica Sinica, 2016, 42(4): 481-494. (in Chinese)
袁勇,王飞跃. 区块链技术发展现状与展望[J]. 自动化学报, 2016,42(4):481-494.

[2] NAKAMOTO S. Bitcoin: A peer-to-peer electronic cash system [OL]. https://www.cs.bgu.ac.il/~crp161/wiki.files/Bitcoin-Paper.pdf.

[3] LAMPORT L, SHOSTAK R, PEASE M. The Byzantine Generals Problem [J]. Acm Transactions on Programming Languages & Systems, 2016, 4(3): 382-401.

[4] 邹均. 区块链技术指南[M]. 北京:机械工业出版社, 2016: 109-128.

[5] MICALI S. ALGORAND: The Efficient and Democratic Ledger [OL]. http://pdfs.semanticscholar.org/0dc0/55052cda7179cd74d43e07479565121ef733.pdf.

[6] Larimer D. Transactions as proof-of-stake [OL]. [2017-07-05]. https://bravenewcoin.com/assets/Uploads/TransactionsAsProofOfStake10.pdf.

[7] CASTRO M. Practical byzantine fault tolerance and proactive recovery [J]. Acm Transactions on Computer Systems, 1999, 20(4): 398-461.

[8] ONGARO D, OUSTERHOUT J. In search of an understandable consensus algorithm [C]// Usenix Conference on Usenix Technical Conference. USENIX Association, 2014: 305-320.

[9] LEI C J, LIN Y P, LI J G, et al. Research on Byzantine Fault Tolerance Under Volunteer Cloud Environment [J]. Computer Engineering, 2016, 42(5): 1-7. (in Chinese)
雷长剑,林亚平,李晋国,等. 志愿云环境下的拜占庭容错研究[J]. 计算机工程, 2016, 42(5): 1-7.

[10] LI J. Distributed Gossip algorithms for Quantized Consensus [D]. Harbin: Harbin Institute of Technology, 2013. (in Chinese)
李婧. 基于量化共识的分布式 Gossip 算法研究[D]. 哈尔滨:哈尔滨工业大学, 2013.

[11] ANDRÉ A, DEMERS A, HOPCROFTT J E. Correctness of a gossip based membership protocol [C]// Twenty-Fourth ACM Symposium on Principles of Distributed Computing. ACM, 2005: 292-301.

[12] GUREVICH M, KEIDAR I. Correctness of gossip-based membership under message loss [C]// ACM Symposium on Principles of Distributed Computing. ACM, 2009: 151-160.

[13] GANSESH A J, KERMARREC A M, MASSOULI, et al. Peer-to-Peer Membership Management for Gossip-Based Protocols [J]. IEEE Transactions on Computers, 2003, 52(2): 139-149.

[14] 黄步添,王云霄,王从礼,等. 一种应用于区块链的拜占庭容错共识方法:中国,CN106445711A [P]. 2017-02-22.

[15] 张铮文. 一种用于区块链的拜占庭容错算法[OL]. [2017-07-03]. http://www.onchain.com/paper/66c6773b.pdf.

[16] KERMARREC A M, MASSOULIE L, GANESH A J. Probabilistic reliable dissemination in large-scale systems [J]. IEEE Transactions on Parallel & Distributed Systems, 2003, 14(3): 248-258.

(上接第 19 页)

[5] 王帅宇,李晨. 一种基于区块链技术的大数据确权方法及系统:中国,CN106815728A[P]. 2017-06-09.

[6] PETITCOLAS F A P, ANDERSON R J, KUHN M G. Information Hiding-a Survey[J]. Proceedings of the IEEE, 1999, 87(7): 1062-1078.

[7] NAKAMOTO S. Bitcoin: A Peer-to-Peer Electronic Cash system [OL]. https://bitcoin.org/bitcoin.pdf.

[8] BONEH D, LYNN B, SHACHAM H. Short Signatures from the Weil pairing[C]// International Conference on the Theory and Application of Cryptology and Information Security. Springer Berlin Heidelberg, 2001: 514-532.

[9] WANG C, CHOW S S M, WANG Q, et al. Privacy-Preserving Public Auditing for Secure Cloud Storage[J]. IEEE Transactions on Computers, 2013, 62(2): 362-375.

[10] DU W, JIA J, MANGAL M, et al. Uncheatable Grid Computing [C]// International Conference on Distributed Computing Systems, 2004. IEEE, 2004: 4-11.

[11] ZHANG F G. From Bilinear Pairings to Multilinear Maps[J]. Journal of Cryptologic Research, 2016, 3(3): 211-228. (in Chinese)
张方国. 从双线性对到多线性映射[J]. 密码学报, 2016, 3(3): 211-228.

[12] ATENIESE G, BURNS R, CURTMOLA R, et al. Provable Data Possession at Untrusted Stores[C]// ACM Conference on Computer and Communications Security. ACM, 2007: 598-609.

[13] WANG B, LI B, LI H. Oruta: Privacy-Preserving Public Auditing for Shared Data in the Cloud[C]// IEEE Fifth International Conference on Cloud Computing. IEEE Computer Society, 2012: 295-302.

[14] fabric[OL]. (2017-10-20). https://github.com/hyperledger/fabric.