高技术通讯 2018 年 第28 卷 第9 – 10 期:771 ~ 783

doi:10.3772/j.issn.1002-0470.2018.09 – 10.002

CHB-Consensus:一种基于一致性哈希算法的区块链共识机制研究①

于 雷②[*,**] 赵晓芳[*] 金 岩[*] 胡 斌[*,**]

([*]中国科学院计算技术研究所 北京 100190)

([**]中国科学院大学 北京 100049)

摘 要 区块链中的共识协议使得网络中相互不信任的节点对全网的交易状态达成一致的确认。当前的共识协议在“去中心化、安全性、能耗”3 个方面存在矛盾,无法同步优化。针对该问题,本文基于一致性哈希算法,设计实现了全新的区块链共识协议,称为“CHB-Consensus”。该共识协议在“诚实”节点创建新区块时不耗费多余的“算力”资源,但是“恶意”节点进行新块创建攻击或“双花”攻击时需要耗费海量算力。“CHB-Consensus”形成的区块链网络与比特币系统基于同样的安全性假设,因此,“CHB-Consensus”节省了海量算力的同时没有牺牲去中心化和安全性的优势。本文分析了“CHB-Consensus”共识协议可能存在的攻击过程,并给出了严格但可调整的验证策略。“CHB-Consensus”共识过程引入了 CA,CA 数字证书只作为共识过程的准入凭证,CA 对区块链网络及区块链数据结构不具有任何特殊的管理控制权限,但有交易隐私泄露的风险,这取决于 CA 的安全性和可信性。本文最后分析了“CHB-Consensus”的健壮性及对应不同网络环境的优化策略。

关键词 区块链,共识协议,一致性哈希,低能耗,去中心化

0 引 言

近年,区块链(Block chain)技术成为当前的学术研究和应用研究的热点之一。区块链技术的核心价值包括去中心化、分布式共识、非对称秘钥的签名和加密、时间戳,在节点无需互相信任的分布式系统中实现基于去中心化信用的点对点交易、协调与协作,从而为解决中心化机构普遍存在的高成本、信用垄断、可靠性依赖等问题提供了方案。具体的区块链的定义可描述如下:区块链是一种按照时间顺序将数据区块用类似链表的方式组成的数据结构,并以分布式共识和密码学方式保证区块链的数据全局一致、不可篡改和不可伪造的分布式去中心化账本,能够安全存储简单的、有先后关系的、能在系统内进行验证的数据[1]。区块链的出现解决了数字货币的 2 大问题,即双重支付问题和拜占庭将军问题[2-7]。区块链技术在金融、保险、支付、公证等领域有广阔的应用前景。

1 相关研究

1.1 区块链技术

2008 年之后出现的比特币[8]技术,在近年来被广泛关注,比特币被认为是加密数字货币实用系统的开创。数字货币的底层技术平台是区块链技术,区块链的核心协议可以概括为以下几个技术术语的组合:P2P 网络、基于非对称秘钥机制的签名验证、全网共同遵守的当前时间段交易信息共识、基于单

① 国家自然科学基金(61202413)资助项目。

② 男,1981 年生,博士生,工程师;研究方向:大数据管理,区块链;联系人,E-mail: yulei@ncic.ac.cn

(收稿日期:2018-03-06)

向哈希(Hash)摘要算法的交易历史链式数据结构,这在 Nakamoto[8] 的论文中进行了详细的描述。可以将区块链技术的本质视为分布式数据库,该数据库保存历史交易数据,这个数据库被所有节点通过分布式一致协议共享[9]。

一般而言,目前的区块链类型可以分为两类:记录交易单链接类型的区块链,以下简称交易类型的区块链;记录账户变化信息的区块链,简称账户类型的区块链。

本文只利用交易类型的区块链协议,账户类型的区块链不在本文进行讨论,本文以下内容都是基于交易类型的区块链协议进行的描述及讨论。

当前,交易类型区块链技术并没有形成行业标准,基本的区块链的数据结构如图 1 所示。

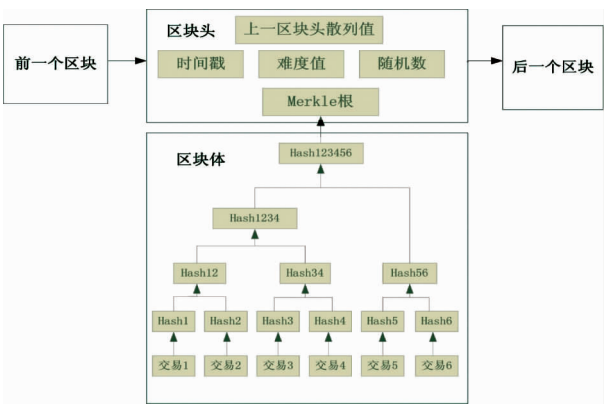

图 1 区块链的基本数据结构

区块链协议中的链式结构、交易信息的 Merkle 树和共识机制,保证了历史交易数据极难被篡改,其中的交易数据为本段时间内的交易单信息,其中的交易单的逻辑结构如图 2 所示。

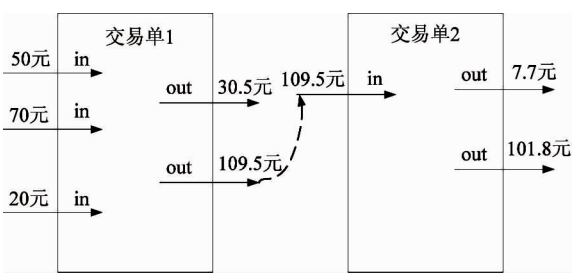

图 2 交易型区块链交易单的逻辑链接

以比特币为例,从创世区块开始,区块链历史账本数据中,包含了数字资产的首尾相接转账交易单构成的交易链条,上一个交易单的输出(out)成为当前交易单的输入(in),当前交易单的 out,又可以作为下一个交易单的 in,每个交易单的具体数据结构如下:

```
{
  Hash:[代表本交易单的哈希值],
  in:  //收入来源项内容
  {
    prev _ out:  //前置交易信息
    {
      Hash:[代表前置交易的哈希值],
      n:[代表输出项 out 的索引值]
    },
    scriptSig: < sig > < pubKey >   //拥有者的签名和公钥
  },
  out:  //支出去向项内容
  {
      value:[付款金额],
      address:[代表收款方地址的哈希值],
      scriptPubKey:脚本内容
          验证使用者的公钥地址与当前指向的公钥地址哈希值匹配(基于非对称秘钥机制);
          验证使用者的数字签名与使用者的公钥匹配(基于非对称秘钥机制);
  },
}
```

首尾相接的交易单数据(包含交易发送方的数字签名)经过脚本内容的验证为合法后,被不同时间戳的区块进行记录,构成首尾相接的区块链的数据主体。区块链网络的节点通过共识过程,竞争交易单的记账权,避免“双花”问题,避免历史交易数据被轻易篡改。

基于去中心化的点对点交易需求以及系统可靠性方面的考虑,区块链技术普遍基于 P2P 网络,网络中的每个节点以扁平式拓扑结构相互连通和交互,不存在任何中心化的特殊节点、不存在层级结构,每个节点均会承担网络路由、验证交易单、验证区块数据、传播区块数据、发现新节点等功能。区块链的网络结构研究正在受到广泛关注,针对 P2P 网络的改进以实现区块链网络的线性扩展正成为研究热点,文献[10]提出了共识网络的递进式分区方

法,用于解决网络,将全网统一的共识过程递进式地划分为多个子网共识,并解决了跨网交易和节点跨网移动的问题,使得区块链网络具有了线性扩展能力。

1.2　一致性哈希算法

Karger[9]等在 1997 年提出了一致性哈希算法,目标是在动态变化的分布式系统上,实现负载(或副本分布)的均衡性、动态适应性和高效性。一致性哈希算法是在哈希算法的基础上提出的,初衷是解决分布式系统的"热点"问题。一致性哈希算法将整个哈希值空间映射成一个虚拟的圆环,整个哈希空间的取值范围为 0 到($2^{32}-1$),哈希空间按顺时针方向组织。0 点和 2^{32} 点是重合的。其原理如图 3 所示。

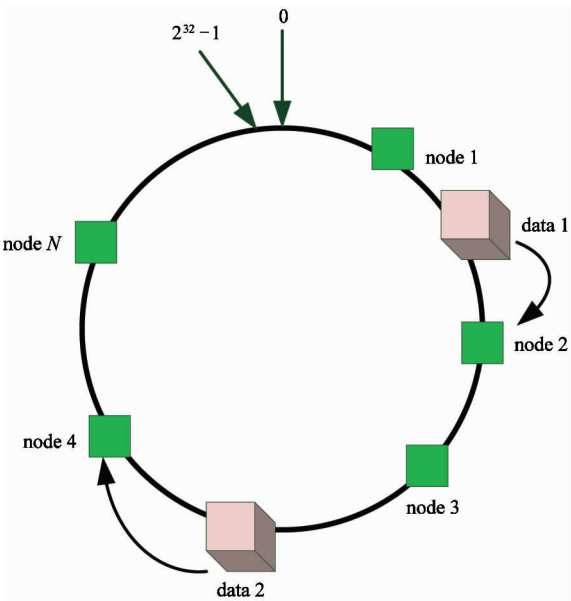

图 3　一致性哈希原理

一致性哈希算法解决的经典问题是负载均衡问题,目标是将大量负载"data"均衡地分配到节点"node"构成的服务器集群中。首先将所有"node"计算哈希值,将每个"node"映射到哈希环上的确定位置,然后计算某个"data"或某个"负载"的哈希值 $H(d)$,将 $H(d)$ 的值映射到这个哈希环上,从 $H(d)$ 的值开始顺时针找到的第一个"node",该"node"即为该"data"或"负载"的处理"node"。该方法适用于分布式系统环境下"node"自由进出的情况,当"node"自由进出时,受到影响的只是该"node"的邻接"node"。容易发现,由于"node"哈希值的不确定性和"node"数量的不确定性,"data"或"负载"很难完全在所有"node"间均衡。对此,提出了改进的一致性哈希算法,基于每个真实"node"的处理能力,为每个真实"node"生成对应数量的虚拟"node",将虚拟"node"均衡映射到哈希环上,"data"或"负载"在哈希环上的映射对应于某个虚拟"node",最终处理该"data"或"负载"的是虚拟"node"对应的真实"node"。

本文的主要贡献如下。

(1) 提出了基于一致性哈希的区块链共识机制,使得区块链网络创建新区块时不需要浪费算力,恶意节点的攻击行为需要浪费海量算力。

(2) 分析本共识机制的安全性、可靠性、隐私和负载相关问题,提出了针对不同网络环境的改进的策略。

本文第 2 节对当前区块链共识机制研究进行总结。第 3 节对本文设计实现的一致性哈希共识机制进行描述,包括新区块的创建过程、区块的验证方法、初始链的形成、节点的加入与退出过程;详细讨论可能存在的可能攻击。第 4 节对本文实现的共识机制进行公平性、安全性和隐私性的讨论。第 5 节描述了本文共识机制的概要内容并总结全文。

2　区块链共识研究

区块链的一致性共识机制的目标是实现全网参与节点对"交易"合法性和交易顺序的一致确认。并且经过全网确认的历史"交易"将不能被恶意节点篡改(或很难篡改)。区块链的共识机制一直是区块链技术的研究热点。

工作量证明(Proof of Work,POW)共识机制的核心思想源于防止垃圾邮件的研究[11],之后 Back[12]提出了 Hashcash,首次提出了基于哈希函数的工作量证明方法。由 Nakamoto[8]设计的 POW 共识可有效解决非信任的匿名节点自由进出而可能出现的女巫攻击(Sybil attack)[13],是目前应用最广的区块链共识机制之一,包括比特币系统在内的很多区块链系统采用 POW 或改进的 POW。POW 机制

要求每个节点基于自身算力解决求解复杂但验证容易的 SHA256 计算难题,即寻找一个合适的随机数 *Nonce*,使得区块头部元数据与 *Nonce* 组合构成的输入计算两次 SHA256 哈希值 H, 使得 H 小于区块头中难度目标的设定值,即:

$$H(nVersion, hashPreBlock, hashMerkleRoot, nTimes, nBits, Nonce) < \left(\frac{MaxTarget}{Diff}\right) \quad (1)$$

其中,两次 SHA256 哈希计算的输入参数来自当前待建区块的区块头元数据,由于两次 SHA256 哈希计算的不可逆,节点不得不付出足够的算力进行 *Nonce* 的搜索,以便在尽可能短的时间内完成式(1)的计算(即比特币挖矿)。POW 共识要求,“诚实”节点创建新区块(挖矿)需要耗费足够的算力,“恶意”节点的“双花”攻击需要耗费全网 51% 的算力才能获得足够高的攻击成功概率。POW 的优点是算法简单,容错达到 50%;其缺点是诚实节点的竞争“挖矿”浪费了海量的资源。据可靠数据,比特币的“挖矿”算力早已超过了全球 500 强计算机算力的总和。为保证比特币系统稳定,难度值被周期(每 2 016个块)调整,以保证区块平均间隔 10 min,某个交易的可靠确认,通常需要链接 6 个区块,这造成了交易的确认时间达到 60 min。并且,POW 机制导致的大矿池已经威胁到了比特币系统的去中心化。

针对 POW 的缺点,2011 年 Quantum Mechanic 在 Bitcointalk 论坛首次提出 Proof of Stake(POS)共识机制,提出以节点当前权益作为该节点挖矿难度的凭证,点点币(peercoin)实现了区块生成难度与节点所占股权成反比的权益证明协议[14],由币龄和币值构成权益证明的两个因子,其核心算法为

$$Hash(T,c) \leqslant d \times T.time \times T.value \quad (2)$$

其中, T 表示节点尚未使用的某个交易单, c 表示节点当前状态, $T.time$ 表示交易单 T 的拥有时间,即币龄, $T.value$ 表示交易单 T 可使用的币值。权益证明机制的随机搜索空间是有限的,每秒才能进行一次哈希尝试。从式(2)可以看出,区块的生成难度与节点拥有的某个交易单币龄成反比,与币值成反比,并且,不再采用暴力搜索的方式。由此可知,POS 共识在一定程度上解决了 POW 共识的能耗问题,并且可以缩短区块的间隔。但是 POS 共识在网络同步性较差时,每轮区块创建周期会产生多个区块,极易分叉。恶意节点可以控制网络通信,有权生成区块的恶意节点可以向不同的网络分区发送不同的区块,从而形成“双花”攻击。POS 共识在区块链形成初期无法保证公平性,因为少数拥有足够币龄和币值的节点更易产生区块。

Delegated Proof of Stake(DPOS)[15]共识在 POS 共识的基础上,将产生区块的权利进行专业化,首先以各个节点的权益作为投票权值选出一个“委员会”,由“委员会”成员轮流产生新区块,“委员会”成员必须保证 90% 的在线状态,DPOS 共识将区块的创建权利控制在少数的权威节点,可以提高共识的效率,达到秒级确认。缺点也是明显的,当“委员会”成员成为恶意节点,产生“双花”区块时,其他节点将无能为力;DPOS 也非完全的去中心化。

Practical Byzantine Fault Tolerance(PBFT)[16]共识是基于消息传递模式的一致性共识机制,在恶意节点数 F 小于 $\frac{(N+1)}{3}$ 时可以使得系统就某个值达成共识(N 表示节点总数)。所有节点经过 3 个阶段的网络消息传递,使得诚实节点对某个发起值达成一致。区块链应用 PBFT 时,每次共识周期都是主节点产生区块,避免了区块链分叉,避免了海量算力的浪费,缩短了区块间隔和交易确认周期。但是,PBFT 共识过程无法应对女巫攻击(Sybil attack),恶意节点可以产生多个节点,从而使得整个网络的恶意节点数量超过 $\frac{(N+1)}{3}$, 破坏一致性和安全性。由于每次的区块周期都由主节点产生,因此,PBFT 共识不适合网络节点规模过大的情况而更适用于联盟链。

3 基于一致性哈希算法的共识机制

本文方法的安全性建立在以下假设的基础之上。

假设 1 非对称秘钥算法是公开的,在已知公钥的情况下,通过算法求逆破解私钥或随机尝试破解私钥都是不可行的,伪造数字签名是不可行的。

假设 2 原文通过公开的哈希摘要算法生成原文

摘要,通过原文哈希摘要,推算原文是不可行的。

假设 3 区块链网络内诚实节点的数量超过 50%,如果非诚实节点数量小于 50% 时,整个区块链系统将会是无价值、无意义的,这对非诚实节点也是不利的。

假设 4 最长的区块链是最安全且正确的。

假设 5 一致性哈希算法中哈希环有足够的空间容纳足够多的“节点”,并且保证任何节点在哈希环上的映射不会重叠,且一致性哈希的计算过程求逆是不可行的。

本文方法的安全性建立在以上 5 个假设之上,其中,前 4 个假设也是比特币安全性保证的基石,第 5 个假设是一致性哈希算法的基本要素。因此,本文方法的安全性是没有苛刻的前提条件的,建立在基于密码学安全性的基本保障之上,本文提到的所有相关名称的算法都是建立在以上 4 个假设基础之上。

在 5 个安全性假设的基础上,本文不对 CA(certificate authority)进行任何安全性假设。本文引入 CA,不会由此带来任何额外的区块链数据安全性问题。通过算法及协议保证了 CA 管理机构对区块链不具有任何特殊的控制或操作权限。但是 CA 可能引入隐私泄露问题。

3.1　数据结构和网络结构

3.1.1　数据结构

沿用交易类型区块链的基本结构(类似比特币交易单的数据结构和区块结构)包含区块头(见表 1)和区块体(见表 2)。

表 1　区块头

名称	描述
Pre-Hash	上一区块哈希
No. X	当前区块编号
Timestamp	区块时间戳
Transaction merkle root	交易的 Merkle 树根
Cert merkle root	CA 数字证书的 Merkle 树根
BaseCoinSig	代币奖励的数字签名

表 2　区块体

名称	描述
Transaction-num	包含的交易数量
Transactions	依交易时间戳排序的交易单信息
Cert-num	CA 数字证书个数
Cert-serialNums	包含的所有 CA 数字证书的序列号
CoinBase-Trans	“代币”奖励交易单

区块链的数据结构如图 4 所示。

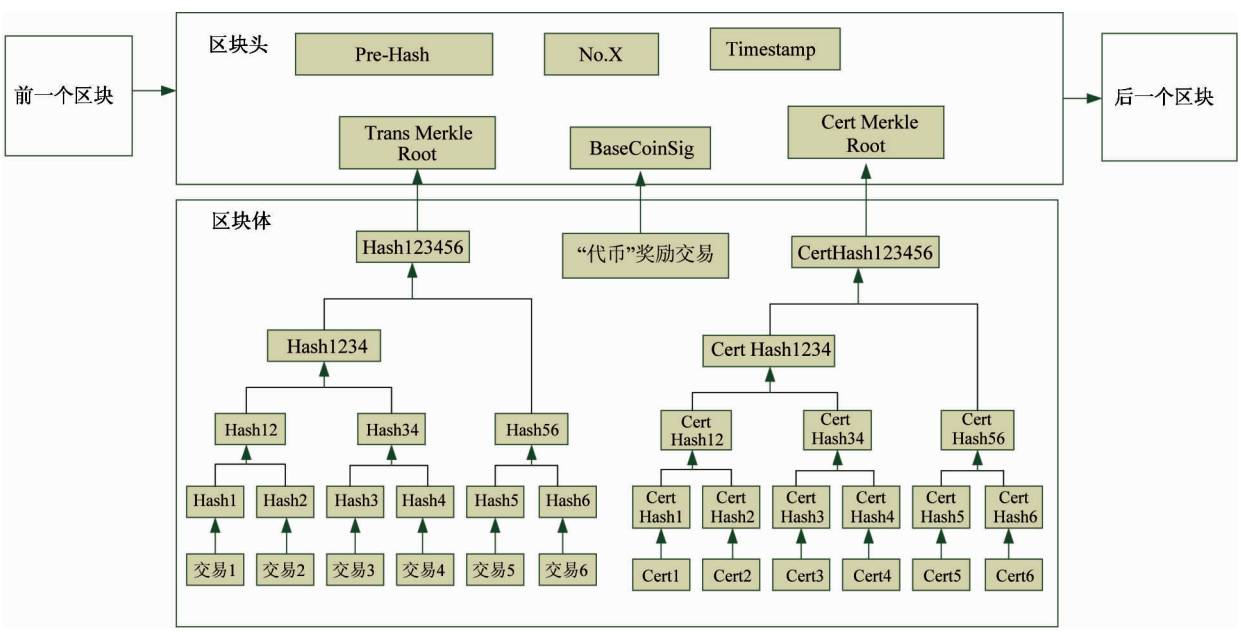

图 4　“一致性哈希共识”的区块链数据结构

3.1.2 网络结构

本方法构成的区块链沿用 P2P 的网络结构,采用广播的方式发布、传递交易单和区块。

3.2 引入 PKI/CA 体系

为避免女巫攻击(Sybil attack)影响公平性,本方法要求区块链网络参与节点采用唯一的公私钥地址参与交易,摒弃比特币交易采用的随机公私钥,唯一的公私钥地址采用 PKI/CA 体系管理,即区块链参与节点首先在 CA 中心注册获得合法的数字证书(数字证书中包含公钥,私钥在证书之外由参与节点采用本地隐私管理的方式保存),CA 中心提供基于数字证书序列号的查询功能,并可验证数字证书合法性。

此方法牺牲了区块链交易的部分隐私性(相对比特币),CA 对数字证书注册信息的管理成为“隐私单点”问题。但是,这也正是某些应用场景所需要的。

本方法通过算法保证了 CA 中心对形成的区块链网络及区块链数据结构不具有任何特殊管理和控制权限,CA 机制不会引入任何额外的区块链方面的安全问题,采用 CA 中心的数字证书机制,引入的只是交易隐私泄露的风险。共识节点在 CA 注册获得唯一合法的数字证书之后,利用 CA 颁发的数字证书参与共识过程,避免了节点随机产生大量的公私钥参与共识而影响公平性。同时,节点通过 CA 的注册,也避免了公有链完全不受监管的状态,使得公有链的节点和数据可以被有限监控。

3.3 共识机制

竞争模式的共识机制,为区块链技术体系提供了算法之外的稳定性和健壮性。基于此,本共识机制依然采用竞争模式的共识,但是并非依靠“算力”竞争,而是以伪随机的方式参与竞争,与“博彩”的思想接近,竞争的目标依然是“建块权利”及内含的“代币奖励”,获得“代币”奖励的方式是基于一致性哈希算法的随机的“博彩”中奖机制。

每个参与节点通过 CA 中心注册获得唯一的数字证书,用该证书对应的公私钥完成交易过程的签名及验签。

3.3.1 新块创建

假设当前区块链网络的节点总数为 n, 当前区块链高度为 h, 也称末尾区块的编号为 h, 待建区块编号为 $h+1$。本方法规定,每隔 T 时间产生新区块。T 的大小是可设调整的数值,大小取决于网络带宽、单位时间的申请交易量。具体步骤如下。

(1)在每个新 T 时段之内,每个节点广播自己的数字证书序列号,并收集网络内的其他节点数字证书序列号,每个节点验证数字证书序列号的正确性并转发。在 T 时段结束时,每个节点将收集到的所有数字证书序列号依据数字顺序排序,生成数字证书序列号集合 $Ce(h+1)\{S_1, S_2, S_3, S_4, \cdots, S_n\}$,使用数字证书集合 $Ce(h+1)$ 生成 Cert Merkle 树,写入待建区块的区块头和区块体数据结构中,成为待建区块的一部分。

(2)在每个新 T 时段之内,每个节点可发布新交易单并广播,每个节点接收网络上的新交易单,验证正确性并转发。在 T 时段结束时,每个节点将收集到的所有新交易单依据交易单的哈希摘要进行排序,生成 Trans Merkle 树,写入待建区块的区块头和区块体的数据结构中,成为待建区块的一部分。

(3)在 T 时段结束时,每个节点以当前区块链高度 h 之前的第 N 个区块(编号为:$(h-N)$)的区块体内记录的数字证书序列号作为节点集合 $Ce(h-N)\{S_1, S_2, S_3, S_4, \cdots, S_n\}$,利用一致性哈希算法计算集合 $Ce(h\text{-}N)$ 中所有节点在哈希环上的映射位置,将每个 S_i 视为一致性哈希算法中的“node”(见 1.2 节),形成集合 $Ce(h-N)$ 在哈希环上的映射,记为 $R(h-N)$ 环;每个区块链节点以当前待建区块头的 Pre-Hash、No. $h+1$、Timestamp、Transaction Merkle Root 作为输入,计算哈希摘要 $HD(h+1)$,将 $HD(h+1)$ 视为一致性哈希算法中的“data”(见 1.2 节),通过一致性哈希算法计算“data”在 $R(h-N)$ 环上的映射位置 $L(h-N)$, 找到该位置对应的处理“node” S_i(见 1.2 节),该“节点”即数字证书序列号(为叙述方便,以上的一致性哈希计算过程称为 $AW(h+1)$ 计算)。本节点判断 S_i 是否是本节点注册的数字证书,如果不是,则退出本次竞争创建新区块的过程,等待接收新区块 $h+1$, 同时进入下一个区块的数据结构收集过程,即进入下一个区块的

步骤(1)和步骤(2);相反,如果 S_i 是本节点注册的数字证书序列号,则本节点获得当前区块的创建权利,同时获得该区块包含的“代币”奖励(见步骤(4))。

(4) S_i 的持有节点进入创建新区块的过程,将“代币”奖励交易单的输出地址指向本节点的数字证书 S_i 对应的公钥地址,“代币”奖励的数值 Re 设置依照式(3):

$$Re(h+1) = B \times |Ce(h+1)| \tag{3}$$

其中,B 代表一个固定常数,$|Ce(h+1)|$ 代表当前待建区块内中数字证书序列号集合包含的总个数。将“代币”奖励交易单数据写入待建区块的区块体,对该代币奖励进行哈希摘要运算获得 Hm,用数字证书 S_i 对应的私钥对 Hm 进行数字签名形成 $SIG_i(Hm)$,将 $SIG_i(Hm)$ 写入待建区块 $h+1$ 的区块头,该节点广播新区块 $h+1$。

以上过程请参照图5。

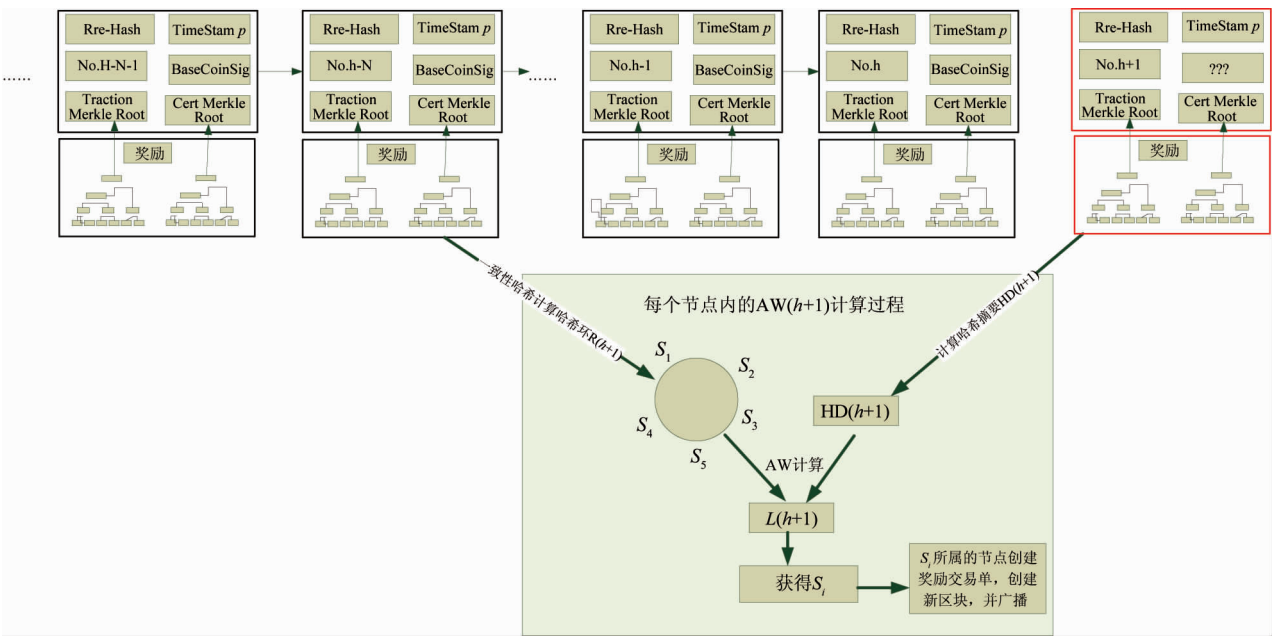

图5　新区块创建的 AW 计算过程

(5) 其他节点接收到广播的新区块,独立执行以下验证。

验证 0　检查新区块引用的上一个区块是否存在且有效。

验证 1　验证该区块包含的交易单的正确性。

验证 2　验证区块头、区块体其他数据项的正确性和完整性。

验证 3　验证“代币”奖励交易单的输出指向地址与当前区块链末尾的区块中包含的“代币”奖励交易单输出指向地址不相同(为了避免攻击行为,不允许连续两个区块的奖励交易单发送给同一个数字证书上的公钥地址)。

验证 4　重新计算 $HD(h+1)$ 和 $R(h-N)$,以及对应的哈希环上的映射位置 $L(h-N)$,验证 $L(h-N)$ 与本节点待建区块计算的 $L(h-N)$ 是否一致。

验证 5　验证该区块 $h+1$ 的“代币”奖励交易单输出指向公钥地址与 $L(h-N)$ 对应节点数字证书 S_i 的公钥地址是否一致。

验证 6　验证本节点的数字证书编号 S_j 是否包含在新建区块的区块体内。

验证 7　验证“代币”奖励交易单的数字签名正确性。

当以上所有验证通过后,该节点接受新区块并链入区块链末尾。

(6) 接受新区块的节点开始下一个时段的区块“竞争”创建过程,返回步骤(1)。

3.3.2　初始链的形成

从 3.3.1 节叙述的区块创建过程可知，最初的编号是 1 至 N 的区块内是无法产生“代币”奖励交易单的，所以编号 1 至 N 的区块体内无法包含其他的任何交易单，也就是说，编号 1 至 N 的区块无法实现一致性哈希的共识过程，这些区块的区块体内只能包含所有节点的数字证书序列号集合。因此在产生区块编号 1 至 N 之间的区块时，需要采用其他的共识机制，例如采用 POW 共识，每个节点通过对确定难度值哈希值的随机碰撞，产生全网共识，将每个节点广播的数字证书序列号列表在编号 1 至 N 的区块内形成一致的记录过程。当然也可采用其他的共识机制，或者采用线下的共识方式。

编号第 $N+1$ 号的区块只包含一个“代币”奖励交易单，AW 计算的输入为 Pre-Hash、No. $N+1$、Timestamp、Transaction Merkle Root，其中 Transaction Merkle Root 包含的交易数据为空。此后，从第 $N+2$ 号区块开始，每个区块的创建应用 3.3.1 节的过程和验证过程。

3.3.3　健壮性分析

（1）新区块创建攻击

交易单的链接和转发依据比特币系统的数据结构和方式，采用数字证书对应的公私钥对进行“签名和验证”的转发过程。因此，基于非对称秘钥机制的安全性，恶意节点是无法盗用别的节点的交易单（基于假设 1），也就是任何恶意节点无法直接盗用包含“代币”奖励交易单在内的所有其他人的交易单，否则无法通过其他节点的区块验证 1。

在确定新区块的创建者时，每个节点使用数字证书集合 $Ce(h-N)$ 通过一致性哈希算法计算在哈希环上的映射时，因为这个集合的输入数据来自被全网验证的历史区块中，所以，任何节点都无法操纵和伪造集合 $Ce(h-N)$，否则，将无法通过其他节点的区块验证 5（如果恶意节点通过修改区块编号 $h-N$ 的区块头，而强制修改集合 $Ce(h-N)$，这相当于“双花攻击”，将在后面讨论）。同时，任何节点对待建区块中属于自己的交易单的修改（基于假设 1，无法篡改其他数字证书签名转发的交易单，否则不能通过区块验证 1），例如，在交易单的保留字段中加入随机值或在交易单的付款与找零之间做连续变化，从而尝试将待建区块的 $AW(h+1)$ 计算结果指向自身的数字证书序列号，在这种情形下，基于一致性哈希算法的特点，恶意节点的攻击行为需要付出的算力难度，相当于暴力碰撞概率为 $\frac{1}{n}$ 的随机事件（n 表示网络节点的数量），其新块创建攻击的示意如图 6，而且需要在有限的时间 T 内完成，就算恶意节点能做到，也无法通过大多数诚实节点的区块验证 4。因此，计算待建区块的“代币”奖励获得者（即对应的数字证书序列号）是随机的，任何节点无法操纵“代币奖励”的发放，代币奖励的发放成为一个去中心化的随机事件。

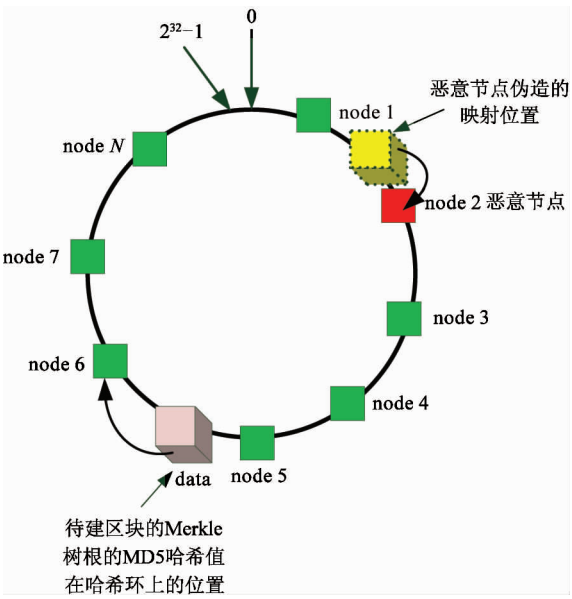

图 6　恶意节点伪造当前区块的共识计算，实现“新块创建攻击”

恶意节点也无法代替合法创建者创建新区块，因为它无法完成对该“代币”奖励交易单的“正确”签名，从而它发布的新区块无法通过区块验证 7。本方法避免了别的节点恶意代替合法创建者对新区块的创建，目标是避免恶意节点剔除待建区块内包含的少量数字证书编号，如果恶意节点能这样做，它将提高自己未来获得“代币”奖励的概率，这个待建区块仍然能通过大多数诚实节点的验证。但是，合法创建者不具有这样的动机，因为合法创建者更希望包含更多的数字证书序列号，（通过式（3））可以获得更多的“代币”奖励。同时，待建区块被尽可能

多的节点通过验证后,该“代币”奖励就获得了尽可能多的认可,相对于未来“代币”奖励的概率部分提高,这对合法创建者更有利。

(2)“双花攻击”

假设恶意节点试图改变区块链历史的某个区块,恶意节点无法改变其他人的交易单,无法伪造任何交易单(基于假设1),其能做的只是“双花”攻击,即改变自己历史的某个交易单。假设当前区块链的总高度为 h,恶意节点试图篡改区块编号 X。将已经在编号为 X 的区块中记录的、由恶意节点自己“签名转发”的交易单进行改变(或剔除),从而产生“双花攻击”,区块号 X 至 h 号的区块包含的区块头和区块体都会发生改变,从而使得 X 至 h 号区块全部失效(无法通过验证2),恶意节点不得不从 X 号区块开始,重新计算之后的所有区块。从 X 区块开始,恶意节点试图操作区块 X 的 $AW(X)$ 计算,由于区块 X 内的交易单被篡改,$AW(X)$ 计算的计算结果在区块编号 $X-N-1$ 的哈希环 $R(X-N-1)$ 上的映射位置必将发生变化,该区块的“代币”奖励交易单将会无效(不能通过区块验证5),为了使该区块重新可以通过区块验证5,恶意节点试图产生“合法”的“代币”奖励交易单。

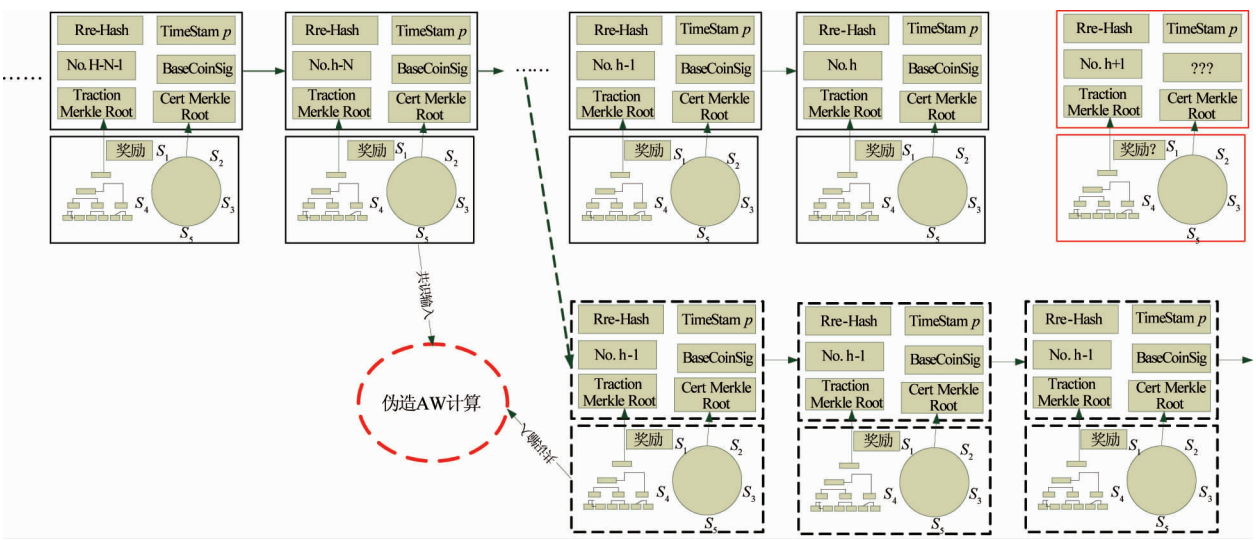

图7　恶意节点伪造共识计算,实现“双花攻击”

基于假设1,恶意节点只能想办法将 $AW(X)$ 计算的结果指向自己注册的数字证书序列号,从而使区块 X 篡改之后的“代币”奖励交易单指向自己的公钥地址,恶意节点要想达到这个目标,有两个可能的选择:

1)修改区块编号 $X-N-1$ 包含的数字证书序列号,即在该区块 $X-N-1$ 包含的数字证书集合 $Ce(h-N-1)\{S_1, S_2, S_3, S_4, \cdots, S_n\}$ 中,循环尝试剔除某个或某些数字证书序列号,然后循环尝试计算 $AW(X)$。这将造成区块 $X-N$ 号至当前区块 h 的区块头全部发生变化,从而使得从 $X-N$ 号区块开始至当前区块 h 的区块链失效。因此,恶意节点不得不从 $X-N$ 号节点开始重新尝试计算 $AW(X-N)$,如果恶意节点依然“递归”选择修改 $X-2N-1$ 号区块的数字证书编号列表 $Ce(X-2N-1)$,将会造成对整条区块链的“递归”失效,因此恶意节点最终不得不重建整条区块链。恶意节点如果重建整条区块链,需要遍历可能的数字证书组合和交易单组合,或修改自己“签名转发”的交易单,使得整条区块链的“代币”奖励交易单的输出地址全部指向自己的数字证书公钥地址,基于假设4,这种重建方式将会使得恶意节点消耗海量算力资源去循环暴力尝试。就算恶意节点侥幸成功,基于区块验证3,重建的区块链也不可能被区块链网络所接受。

2)为了避免产生1)中所述的情况,恶意节点只能篡改区块 X 中属于自己“签名转发”的交易单信息。为了获得足够随机值搜索空间,恶意节点可能进行的篡改包括连续调整该交易单的付款与找零之

间比值,或者在交易单的保留字段加入随机值等,以便使得区块 X 多次地循环尝试 $AW(X)$ 计算,能够将计算结果在哈希环 $L(X-N-1)$ 上指向恶意节点自身的数字证书序列号。从而使得区块 X 生成“合法”的“代币”奖励交易单,这将使恶意节点付出海量算力消耗。基于一致性哈希算法的特点,其面对的算力难度,相当于暴力碰撞概率为 $\frac{1}{n}$ 的随机事件(n 表示网络节点的数量)。假设恶意节点能在有效的时间内完成这种计算难度,即成功伪造区块 X 的 $AW(X)$ 计算,但是区块 $X+1$ 需要完成同样的重建过程。由此可知,区块 X 至当前区块 h 的篡改重建存在算力攻击成功的可能性,其示意在图 7 中描述,但这需要恶意节点在足够短的时间内消耗海量的算力。为了避免这种极端情况的发生,本区块链区块的创建协议附加了验证 3,由区块验证 3 保证了区块 $X+1$ 将无法获得验证通过,恶意节点的篡改区块链无法突破编号 $X+1$。

综上所述,当 $h-X$ 大于 2 时,任何恶意节点即使付出海量算力的代价,“双花攻击”也是无法成功,也就是说,区块 X 之后新链接的区块个数超过 2 时,即可完成对区块 X 内交易单的确认,这优于比特币算力共识所定义的 6 个区块确认间隔。

3.3.4 节点的加入与离开

新节点选择加入交易过程,首先需要在 CA 中心注册获得合法的数字证书及对应的公私钥对,之后,新节点就利用数字证书,可即时参与到区块链网络的交易接收与转发的过程,节点也可选择在任意时刻不再参与交易的接收与转发过程。新节点如果只选择加入交易过程,则不需要下载区块链数据,只需查询区块链 P2P 网络的区块链数据即可。因此新节点加入与离开交易过程不需要区块链网络做任何调整与感知。

新节点也可随时加入一致性哈希过程支配的“博彩”过程,新加入的节点首先在当前区块创建时段 T 内广播自己的数字证书序列号,网络内其他节点在建块过程中,会将此序列号包含进入当前区块,当前区块经过 3.3.1 节的过程创建成功后,新节点的数字证书序列号被写入了最新的区块体内。此后新区块不断被创建和确认,当新创建区块持续到之后的第 $(N+1)$ 个区块时,新节点就已经获得了参与当前区块的“代币”奖励过程,新节点获得了一定的概率获得当前区块的“代币”奖励,概率值为 $\frac{1}{n}$。同样,节点也可在任何时段选择不再参与未来的“代币”奖励“博彩”,节点只需要从当前待建区块开始,不再广播自己的数字证书序列号即可。

4 相关讨论

4.1 公平性

为了叙述方便,本方法在叙述过程中忽略了一致性哈希算法存在的公平性问题。将所有的数字证书序列号集合 $Ce\{S_1, S_2, S_3, S_4, \cdots, S_n\}$ 映射到一致性哈希环上后,不能保证每个数字证书序列号分得的哈希区间长度完全一致,而每个哈希区间长度代表了节点的“中奖”概率。因此,本方法无法保证每次的哈希区间长度完全公平,但是,通过引入确定的、变化的“扰动”,可以避免持续的不公平性。例如,将每个数字证书序列号附加上当前所在区块的时间戳构成一个新的集合 $Ce\{S_1+T, S_2+T, S_3+T, S_4+T, \cdots, S_n+T\}$,将该集合的每个元素哈希值作为一致性哈希的“node”,可以使得每次计算获得的哈希环节点映射发生不可预测的随机变化,如图 8 所示,这样就避免了持续的“中奖”概率不均衡性。

显然,这样的确定“扰动”不会影响本方法的正确性和可验证性。

4.2 安全性

4.2.1 验证 4 的安全性及副作用

不难发现,3.3.1 节中的区块验证 4 是一个“过分”严格的验证,该验证的存在,可能会造成新区块不能成功创建。

当区块链的整体网络环境不稳定时,不能保证每个节点在 T 时段内收到完全一致的交易单,但是每个节点收到的交易单都是合法的。此时,可能会造成任何一个节点在进行 AW 计算时,其结果都是指向某个对手节点,造成了当前新区块没有任何一个节点能作为创建节点。对于该问题,可以考虑的方式包括:

(1)整个系统放弃区块验证 4,当区块链网络规

模足够大,诚实节点总数占比 50% 以上时,造成的风险是可控的。在没有验证 4 的情况下,恶意节点对新区块的创建权利进行攻击需要付出海量算力(见前文描述),且区块链网络节点数量 n 越大,恶意节点耗费的算力就会越多,但是诚实节点产生新区块却不需要耗费算力。这种情况下,诚实节点的新区块会更快地在区块链网络内广播,更容易被区块链网络中的多数节点接受。由此可知,恶意节点攻击成功率仍是极低的。

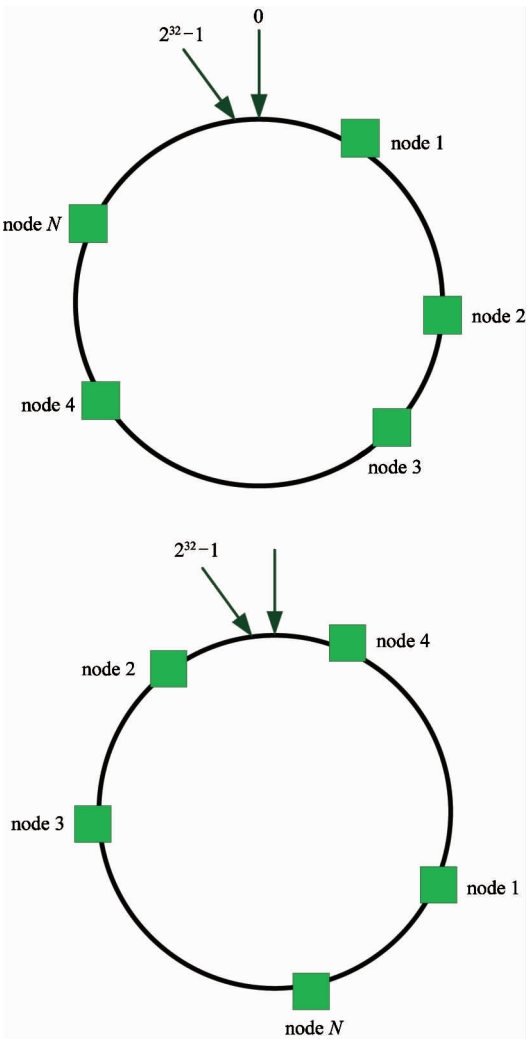

图 8　利用当前区块时间戳,使得数字证书编号在哈希环上的映射位置随机变化

但是,如果没有验证 4,在一个新块周期 T 内,多个节点可能会生成多个合法的新块,这将导致区块链的临时分叉。由于每个分叉在每个周期 T 都会固定产生一个新的区块,因此,与 POW 共识不同的是,在 CBH-Consensus 共识机制下,多个分叉是不能通过竞争长度来"获胜"的。因此,在没有验证 4 的情况下,在新块创建过程的步骤 5 中需加入验证 8 以合并多个分支。

验证 8　当节点从 P2P 网络接收到每个新块时,将验证是否发生分叉。如果发生分叉,节点将会下载多个分叉,当前节点对每个分叉进行以下计算。

$$BranchWeight = \sum_{i=h-b}^{i=h+1}\left[\left(\frac{NumT_i}{1+ownT_i}\right)\times Re(i)\right] \tag{4}$$

其中 b 表示分支块的数量,i 表示当前区块编号,$NumT_i$ 表示当前区块中包含的交易的总数量,$ownT_i$ 表示在当前块中由代币奖励获得者"签名转发"的交易单总数量,$Re(i)$ 表示当前块中的"代币"奖励交易单中包含的"代币"数量。比较每个分支的 $BranchWeight$ 值,最高值的分支是成为当前多个分叉的获胜者,如果 $BranchWeight$ 最高的分叉多于一个,则当前节点随机选择其中的一个作为获胜分叉,此时网络中仍然暂时存在多个分叉。在下一个区块创建周期中,节点将继续按式(4)比较每个分叉的 $BranchWeight$ 值,直到只有一个分支为止。

(2) 保留区块验证 4,这在小规模网络环境及节点数量较少时是可行的。通过调整区块的创建时段 T,调整网络的运行参数,可以使得每个节点在每个 T 时段内收到的交易单数据不一致的概率降低。如果发生异常情况,仍然造成了某个新区块创建者不能确定时,可以延长时段 T,然后,每个节点随机将部分自己收到的交易单进行广播或转发,这样,每个节点将会最终获得一致的交易单数据。这在小规模的网络环境是可行的,当区块链节点数超过某个阈值时,此方法将会造成网络通信负载提高。

4.2.2　验证 3 的安全性

3.3.1 节中的区块验证 3 避免了高性能、高算力节点的暴力攻击行为,但是,无法避免多个高性能、高算力节点的同谋攻击行为。因此,基于网络环境及网络规模的考虑,可以适当调整验证 3,即验证当前新区块的"代币"奖励交易单的输出指向地址与当前区块链末尾及末尾之前 M 个区块中包含的"代币"奖励交易单输出指向地址全都不相同。M 值代表了网络中可能存在的高算力节点同谋个数的可

能性。

4.2.3 参数设置

3.3.1 节中新块创建过程中,在做 *AW* 计算时,读取当前区块链末尾之前第 *N* 个区块中的数字证书序列号列表,作为当前 *AW* 计算的一致性哈希环映射“node”, *N* 的设置源于经验值,考虑区块链网络的安全环境,设置为 2 ~ 6 之间的某值即可。

4.3 区块数据负载

与一般的区块链系统相比,本区块链系统的区块数据结构增加了数字证书序列号的集合,假设每个序列号长度为 8 字节,在参与“博彩”节点数量为 10 万级时,每个区块的额外数据量低于 1MB,这与目前比特币的区块上限一致,即与当前主流公有链的交易数据负载处于同一数量级。因此,本文新增的数字证书信息对网络带宽及共识节点的存储产生的额外负载在当前的区块链优化技术下是可控的。

4.4 隐私

本方法要求每个节点采用 CA 注册的数字证明完成交易过程的签名和验签工作,建议采用两级证书方式,即身份注册证书和交易证书。交易证书由注册证书生成,交易证书不含有用户的注册隐私信息。在交易过程中使用交易证书及交易证书的序列号完成区块链网络内的共识参与过程。由 CA 中心管理注册证书与交易证书的关联关系。在此基础上,区块链中的交易行为隐私保护取决于 CA 的可信性。但是,不管如何,CA 对区块链数据结构和交易过程的安全性不构成任何特殊威胁。

5 结 论

区块链系统中,新区块创建过程的哈希算力共识 POW 被认为是最去中心化、最安全的共识机制,其明显的缺陷是资源浪费及交易确认时间长。本文实现了基于一致性哈希算法的区块链共识机制 CHB-Consensus,仍然通过哈希算力不可伪造性来保障历史区块的安全,任何恶意节点进行“双花”攻击或新区块创建攻击时,需要耗费海量的哈希算力,但是,本共识算法使得诚实节点产生新区块时不用耗费算力。本文方法没有牺牲去中心化和安全性优势,与比特币系统基于同样的安全性假设,解决了类似比特币系统哈希算力共识的资源浪费问题。本文详细分析了该共识机制可能存在的攻击行为和过程,给出了完善但可调整的验证策略。本文最后分析了其公平性、安全性、效率和隐私方面的问题。CA 的存在使得本系统有隐私泄露的风险,这取决于 CA 的可信性与可靠性。

参考文献

[1] 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] Lamport L, Shostak R, Pease M, The Byzantine Generals problem[EB/OL]. https://www.microsoft.com/en-us/research/wp-content/uploads/2016/12/The-Byzantine-Generals-Problem.pdf: microsoft,1982

[3] Fan J, Yi L T, Shu J W. Research on the technologies of Byzan-tine system[J]. *Journal of Software*, 2013, 24(6):1346-1360

[4] Nelson M. The byzantine general's problem: an agreement protocol for distributed system[EB/OL]. http://www.drdobbs.com/cpp/the-byzantine-generals-problem/206904396: drdobbs,2008

[5] Lamport L. The weak byzantine generals problem[J]. *Journal of the ACM* (*JACM*), 1983, 30(3): 668-676

[6] Fedotova N, Veltri L. Byzantine generals problem in the light of P2P computing[C]. In: Proceedings of the International Conference on Mobile & Ubiquitous Systems: Networking & Services, San Jose, USA, 2006. 1-5

[7] REISCHUK R. A new solution for the byzantine generals problem[J]. *Decision Support Systems*, 1985, 1(2):182

[8] Nakamoto S. Bitcoin: a peer-to-peer electronic cash system [EB/OL]. https://bitcoin.org/bitcoin.pdf: bitcoin, 2008

[9] Karger D, Lehman E, Leighton T, et al. Consistent hashing and random trees: distributed caching protocols for relieving hot spots on the world wide web.[C]. In: Proceedings of the 29th ACM Symposium on Theory of Computing, El Paso, USA, 1997. 654-663

[10] 于雷, 金岩. 区块链全局账本数据的拆分技术研究[J]. 高技术通讯, 2017, 27(11-12): 875-888

[11] Dwork C, Naor M. Pricing via processing or combatting junk mail[C]. In: Proceedings of the Annual Interna-

tional Cryptology Conference, Santa Barbara, USA, 1992. 139-147

[12] Back A. Hashcash-a denial of service counter-measure [C]. In: Proceedings of the Usenix Annual Technical Conference, Monterey, USA, 2002. 1-10

[13] Douceur J R. The sybil attack[C]. In: Proceedings of the 1st International Workshop on Peer-to-Peer Systems, London, UK, 2002. 251-260

[14] King S, Nadal S. PPCoin: Peer-to-peer crypto-currency with proof-of-stake [EB/OL]. http://peercoin. net/assets/paper/peercoin-paper-nl. pdf: peercoin, 2012

[15] Larimer D. Delegated proof-of-stake[R]. White Paper, 2014

[16] Castro M, Liskov B. Practical byzantine fault tolerance [EB/OL]. http://dts-web1. it. vanderbilt. edu/ ~ dowdylw//courses/cs381/castro. pdf,2017-4-15

Research on blockchain consensus based on consistent Hash algorithm

Yu Lei[*,**], Zhao Xiaofang[*], Jin Yan[*], Hu Bin[*,**]

([*]Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190)

([**]University of Chinese Academy of Sciences, Beijing 100049)

Abstract

The consensus protocol in the blockchain enables the nodes in the network that do not trust each other to agree on the transaction status of the entire network. The current consensus protocols have contradictions in three aspects of decentralization, security, and energy consumption and cannot be optimized simultaneously. In response to the above issues, a new blockchain consensus protocol is designed in this paper, and is based on the consistent Hash algorithm, called CHB-Consensus. This consensus protocol does not consume any extra computational power resources when creating new blocks by honest nodes, however malicious nodes need to pay massive computational power resources for attacking new block creation right or double spend. Blockchain networks formed by CHB-Consensus are based on the same safety hypotheses as bitcoin systems, so CHB-Consensus saves huge amounts of power without sacrificing decentralization and safety. This paper analyzes the possible attacks and gives a rigorous but adjustable validation strategy. The CHB-Consensus protocol introduces the certificate authority (CA) system, which does not have any special management or control rights over the blockchain network and the blockchain data, but at the risk of privacy breaches, depending on the credibility and reliability of CA system. This paper analyzes the robustness of CHB-Consensus and the optimization strategy corresponding to different network environments.

Key words: blockchain, consensus protocol, consistent Hash, low energy consumption, decentralization