工程科学学报, 第 41 卷, 第 11 期: 1361−1373, 2019 年 11 月
Chinese Journal of Engineering, Vol. 41, No. 11: 1361−1373, November 2019
DOI: 10.13374/j.issn2095-9389.2019.03.26.004; http://journals.ustb.edu.cn

区块链技术及其研究进展

朱　岩✉, 王巧石, 秦博涵, 王中豪

北京科技大学计算机与通信工程学院, 北京 100083
✉通信作者, E-mail: zhuyan@ustb.edu.cn

摘　要　从区块链的设计和需求出发, 阐明了区块链技术中的基本概念与特征及其基础架构; 其次, 以比特币为例详细介绍了区块链中各种机制, 包括: 区块结构与防篡改机制、交易结构与脚本语言、交易人员身份鉴别机制以及网络高效交易传播机制等; 而且, 按照证明类、拜占庭类、传统共识类及混合共识类等类型, 详细描述了当前几种主流的区块链共识算法; 此外, 对智能合约的概念、组织结构及模块关系以及执行方式与过程进行了讨论; 最后, 对区块链面临的主要安全挑战进行了总结, 从而达到系统地把握区块链技术发展和趋势的目的.

关键词　区块链; 去中心化; 共识算法; 智能合约; 分布式系统
分类号　TP319

Survey of blockchain technology and its advances

ZHU Yan✉, *WANG Qiao-shi*, *QIN Bo-han*, *WANG Zhong-hao*

School of Computer & Communication Engineering, University of Science and Technology Beijing, Beijing 100083, China
✉ Corresponding author, E-mail: zhuyan@ustb.edu.cn

ABSTRACT　With the rapid development of e-commerce and network finance involving the Internet, hundreds of millions of online transactions are being carried out on the Internet every moment. Guaranteeing the security of these transactions and realizing the secure storage, exchange, and sharing of massive transaction data have become paramount. Blockchain is a practical technology recently proposed to solve the above problems. Through P2P network technology, distributed ledger technology, asymmetric cryptography, consensus mechanism, and smart contract technology, blockchains can ensure data integrity, nonrepudiation, privacy, consistency, and other security protections. Hence, it has attracted wide attention from academia and industry in recent years. Starting from the design and demand of blockchains, this paper first expounds the basic concepts, features, and typical architecture in the current blockchains. Taking Bitcoin as an example, this paper also explored the various proposed structures and the corresponding mechanisms, including block storage structure and tamper-proof mechanism, transaction structure and scripting language, trader identification mechanism, and efficient network transaction propagation mechanism. Moreover, several current mainstream blockchain consensus algorithms were described according to the categories of proof-mode, Byzantine-type, traditional consensus, and hybrid consensus. In addition, the latest developments in smart contracts were discussed from some aspects, including concepts, organizational structure, the relationship among modules, as well as execution approaches and processes. Finally, the main security challenges faced by blockchains were summarized in order to systematically grasp the developments and trends of blockchain technology.

KEY WORDS　blockchain; decentralization; consensus algorithm; smart contract; distributed system

收稿日期: 2019−03−26
基金项目: 国家重点研发计划资助项目(2018YFB1402702); 国家自然科学基金资助项目(61972032)

区块链(blockchain)的概念首次出现在 2008 年中本聪(Satoshi Nakamoto)发表的《Bitcoin: A peer-to-peer electronic cash system》[1]. 该文提出以区块链技术为基础的比特币(Bitcoin)系统构架, 该构架记录着所有元数据和加密交易信息, 从而建立了一个采用点对点(P2P)技术的分布式电子现金系统[2], 使得在线支付的双方不用通过第三方金融机构而直接进行交易. 随着比特币网络多年来的稳定运行与发展, 比特币在全球流行起来. 同时, 比特币的底层技术逐渐引起了产业界的广泛关注[3], 并被命名为"区块链"技术, 使之与比特币技术区分开, 这也是区块链技术发展的第二阶段.

近几年, 随着对区块链技术的不断探索和演化, 出现了一些新的技术改进. 其中, 针对原有交易脚本不足以处理复杂事务的问题, 一种被称为"智能合约"的新型区块链构架被提出, 也被称为第三代区块链技术[4]. 目前, 区块链作为颠覆性的创新技术, 已渗透到金融、资产[5]、版权[6]、法律[7]和医疗[8]等各种领域中, 成为新的业务增长动力.

本文首先从区块链的设计和需求出发, 阐明了区块链技术中的基本概念与特征及其基础架构; 其次, 以比特币为例详细介绍了区块链中各种机制, 包括: 区块结构与防篡改机制、交易结构与脚本语言、交易人员身份鉴别机制、以及网络高效交易传播机制等; 而且, 按照证明类、拜占庭类、传统共识类、及混合共识类等类型, 详细描述了当前几种主流的区块链共识算法; 此外, 对智能合约的概念、组织结构及模块关系、以及执行方式与过程进行了讨论; 最后, 对区块链面临的主要安全挑战进行了总结, 从而达到系统地把握区块链技术发展和趋势的目的.

与区块链技术的相关综述性文献 [9] 相比, 本文更加侧重于从技术细节来阐释区块链特征、当前进展和未来趋势. 例如, 文献 [9] 侧重于从数据、记账、协议、经济、技术等多角度介绍区块链概念和性质, 并详细介绍了区块链的优势, 如去中心化、去信任化、可追溯性、集体维护性、安全性、开放性、匿名性等; 文献 [10] 和 [11] 更加侧重于区块链面临的安全挑战和问题, 如 51% 攻击、分叉问题、扩展性问题、交易延时问题、现有政策问题、成本问题等; 文献 [5] 和 [12] 则侧重于区块链的应用, 如区块链在金融领域、通信领域、医疗领域、政府管理领域的应用等.

1　区块链基础架构

区块链具有的去中心化、不可篡改性、数据透明性、交易可溯源性以及匿名性等特性保证了交易活动可以在任何时间和任何地点进行, 突破了传统贸易在时空上的限制, 同时也为交易双方创造了更多的交易机会[13].

区块链的这些特性源于它的基础架构, 区块链基础架构模型如图 1 所示. 一般来说区块链系统分为五个部分, 分别是数据层、网络层、共识层、激励层以及合约层.

数据层是区块链通过使用各种密码学技术, 例如非对称加密、默克尔树(Merkle tree)以及哈希函数等创建的数据存储格式, 用以保证区块链数据稳定性和可靠性.

网络层将区块链底层的 P2P 网络组织起来, 并且快速让交易在网络中扩散, 以确保能够及时的验证交易的正确性.

共识层主要实现了整个网络中的高度分散的

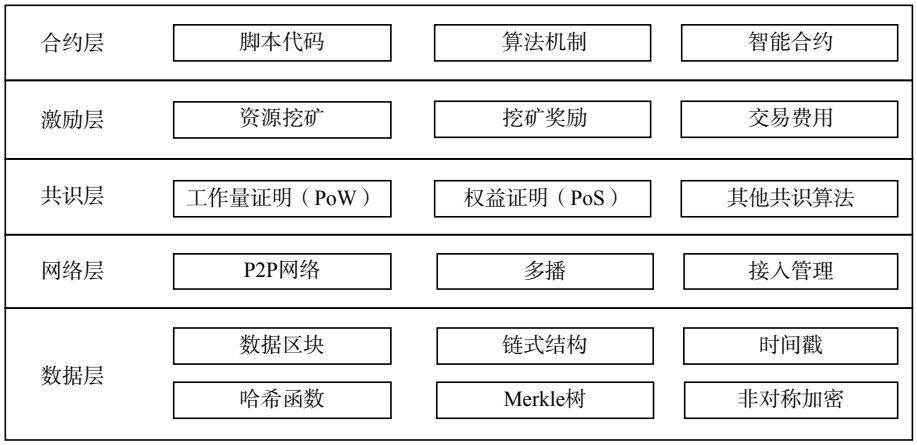

图 1　区块链基础架构

Fig.1　Architecture of a blockchain

节点对交易和数据快速地达成共识，确保全网记账的一致性.

激励层的主要功能是提供一定的激励方式，去鼓励网络中每个节点积极参与区块链中区块的生成和验证工作，以保证区块链的稳定运行.

合约层是在这些基础上提供的一种用于编写可执行代码的接口，利用该接口可以开发基于区块链的各种实际应用.

本文从数据层出发，第二、三节介绍区块链的数据区块、链式结构、Merkle 树和交易等. 第四节介绍区块链网络，第五节介绍区块链共识和挖矿过程，第六节介绍区块链合约层中的智能合约和脚本，第七节介绍区块链当前面临的挑战和机遇，第八节为总结和展望.

2　区块链数据结构

区块链本质上是分布式存储系统[14]，采用了记账式存储模型，以记账形式记录资产的发行、变更、交易和注销等，因此也可称为分布式记账（distributed ledger）系统[15−16]. 在存储结构上，数据区块是区块链的基本构成单元，由区块头（block header）、交易数量（number of transactions）和区块体三部分组成[17]，区块体的内容就是交易列表（list of transactions），包括 Coinbase 交易和常规交易（regular transaction），如图 2 所示.

区块头保存着各种用于连接上一个区块的信息、各种用来验证区块的信息以及时间戳[18]等信息，交易数量用于声明区块体中具体的交易个数，区块体包含了该区块中的所有交易信息.

区块头主要包括：版本号（nVersion）、前一个区块的哈希值 (hashPrevBlock)、当前区块工作量证明的目标难度值（nBits）、当前区块的生成时间（nTime）、用于工作量证明算法的随机数（nNonce）以及用于验证区块体交易的哈希默克尔树树根 (hashMerkleRoot)[19]，具体结构如图 3 所示.

区块链中通常采用哈希函数（SHA256）进行哈希运算，即将任意长度的原始数据经过哈希运算后转换为 256 位（32 字节）的二进制来统一存储和识别. 由图 3 所示，每个区块将其区块头部信息进行两次（Double）哈希运算所得到的结果称为该区块哈希值. 通常任何一段信息只能有唯一的哈希值，即便是改变信息中的任何一个比特位，就会引起整个哈希值出现巨大的差别，而哈希值碰撞（不同的输入信息产生相同哈希值）的概率是极低的，因此区块哈希值可以安全地被认为是区块的唯一标识符.

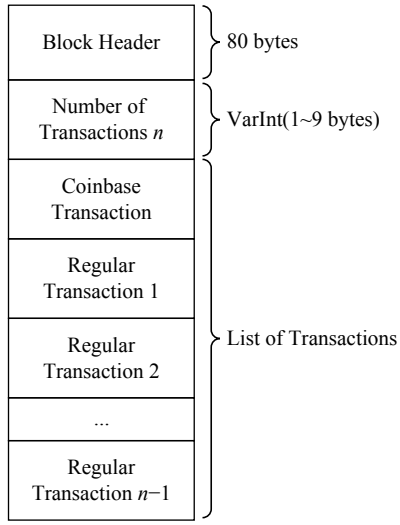

图 2　块结构

Fig.2　Structure of block in blockchain

区块头中的 hashPrevBlock 字段包含区块链中前一个区块的哈希值，由此不同区块根据 hashPrevBlock 字段的内容依次链接起来形成区块链. 由于该长串链条由每个节点所认可，任何对区块信息的修改都会导致后续哈希值的变化，而通过链条的溯源机制，很快就可以发现问题. 因此，区块链十分安全.

区块中的 hashMerkleRoot 字段存放了由区块体中所有交易构成的默克尔树的根节点的值，默克尔树是一种典型的二叉树[20]，它包含：根节点、中间节点以及叶子节点. 其中，非叶子节点的值是

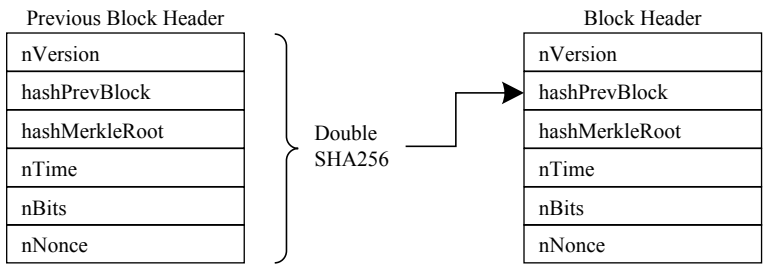

图 3　相邻区块之间关系

Fig.3　Relationship between two adjacent blocks

其所有子节点值的哈希值, 而叶子节点存储了该区块内的所有交易的哈希值, 一个交易对应一个叶子节点. 如图 4 展示了一个简单的基于区块链的默克尔树的数据结构.

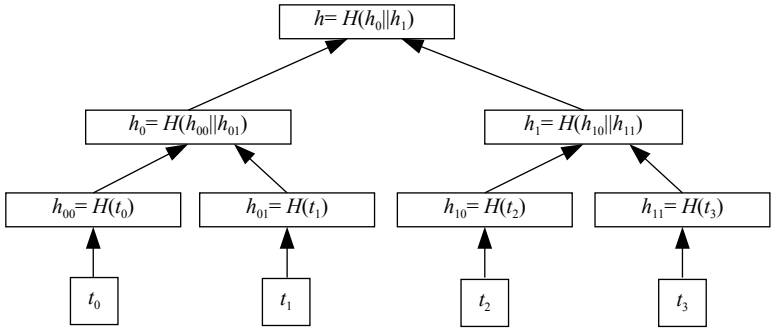

图 4　默克尔树

Fig.4　Merkle tree

图 4 中, t_0,t_1,t_2,t_3表示交易, "||"表示连接运算符, $H(x)$用于表示双 SHA256 函数, 即

$$H(x)=\mathrm{SHA256}(\mathrm{SHA256}(x)) \tag{1}$$

哈希值h被称为树的哈希根或默克尔根. t_0,t_1,t_2,t_3四笔交易对应默克尔树的叶子节点, 使用哈希函数$H(x)$对每笔交易进行计算, 分别得到哈希值 $h_{00}=H(t_0)$、$h_{01}=H(t_1)$、$h_{10}=H(t_2)$和 $h_{11}=H(t_3)$, 由于默克尔树中非叶子节点的值为其所有子节点的值串联形成的字符串的哈希值, 所以通过对每个哈希值进行两两合并哈希, 分别形成哈希值h_0和h_1, 最后h_0与h_1进行两两合并哈希, 得到本区块所对应的默克尔树的根h, 存储在该区块的区块头中. 另外默克尔树的一个显著特点是可以具有很好的溯源性[20].

综上所述, 与传统分布式数据库相比较, 上述区块链的存储方式提供了对交易数据一致性检验和完整性验证功能的支持. 区块链中每个保存完整数据的节点记录了从创世区块到当前区块的所有交易数据, 哈希链表能验证各节点中的数据是一致的, 任何改变都会以密码验证方式被发现. 其次, 默克尔树保证了交易记录不会被恶意篡改.

3　区块链交易

区块链技术的核心基础是对交易(transaction)的支持, 通过区块链交易可实现数字资产的创建、转移、变更、终止等过程. 本小节以比特币为例对其交易结构和交易脚本进行研究.

在比特币系统中一个完整的交易由以下的元素构成的: 交易版本 (nVersion), 输入 (tx_vin), 输出 (tx_vout) 和锁定时间 (nLockTime), 由图 5 所示. 其中输入 (tx_vin) 字段中的输入交易按照数组的形式存储在 vin[] 中, #vin 用来表示输入交易的个数, 输出 (tx_vout) 字段中的输出交易按照数组的形式存储在 vout[] 中, #vout 用来表示输出交易的个数.

交易又分为两种, 一种是 Coinbase 交易, 这种交易只有输出段没有输入段. 另一种是常规交易, 它可以有多个输入和多个输出. 比特币常规交易中的每个输出都是下一笔交易的输入来源, 每一笔交易的输入也能追溯到上一个交易的输出, 所以每一笔交易都可以进行向前溯源, 从而找到每笔交易的所有历史记录.

交易脚本语言是为保障比特币中的交易安全而提出, 它是一种基于栈的脚本语言(scripting language), 具有简单、紧凑、容易理解等特点, 并且其是非图灵完备的, 不支持循环结构.

当一笔比特币交易被验证时, 每一个输入(vin)中包含交易序列号(txid)和解锁脚本(scriptSig), 其中解锁脚本与输入引用的输出(vout)中的锁定脚本(scriptPubKey)同时执行, 从而验证(Verify)这笔交易是否有效, 如图 6 所示. 解锁脚本是一个"解决"或满足被锁定脚本在一个输出上设定的花费条件的脚本, 它将允许输出被消费. 锁定脚本是一个放置在输出上面的花费条件, 它指定了今后花费这笔输出必须要满足的条件. 每一个比特币验证节点会通过同时执行锁定和解锁脚本来验证一笔交易, 如果解锁脚本满足锁定脚本条件, 则输入有效. 所有输入都是独立验证的, 作为交易总体验证的一部分.

经常使用的 5 类脚本分别是:

(1) Pay-to-Public-Key-Hash (P2PKH)

a) scriptPubKey: OP_DUP OP_HASH160 <pubkeyHash> OP_EQUALVERIFY OP_CHECKSIG

字段名称			类型	描述
nVersion			Int（4 bytes）	交易格式版本（目前为1）.
tx_vin	#vin		VarInt（1~4 bytes）	vin中的交易输入的数量.
	vin[]	hash	unit256（32 bytes）	上一个交易的双SHA256哈希值.
		n	VarInt（4 bytes）	由hash指定的交易内的交易输出的索引.
		scriptSigLen	VarInt（1~9 bytes）	scriptSig字段的长度，以字节为单位.
		scriptSig	CScript（Variable）	用于满足交易输出（hash，n）的花费条件的脚本.
		nSequence	unit（4 bytes）	交易输入序列号.
tx_vout	#vout		VarInt（1~4 bytes）	vout中的交易输出的数量.
	vout[]	nValue	unit256（32 bytes）	存储输出要花费的比特币数量.
		scriptPubkeyLen	VarInt（4 bytes）	scriptPubkey字段的长度，以字节为单位.
		scriptPubkey	VarInt（1~9 bytes）	定义了交易输出所需条件的脚本.
nLockTime			unsigned int（4 bytes）	交易的锁定时间，一旦超过锁定时间，交易就被锁定计入区块中.

图 5　常规交易结构

Fig.5　Structure of regular transaction

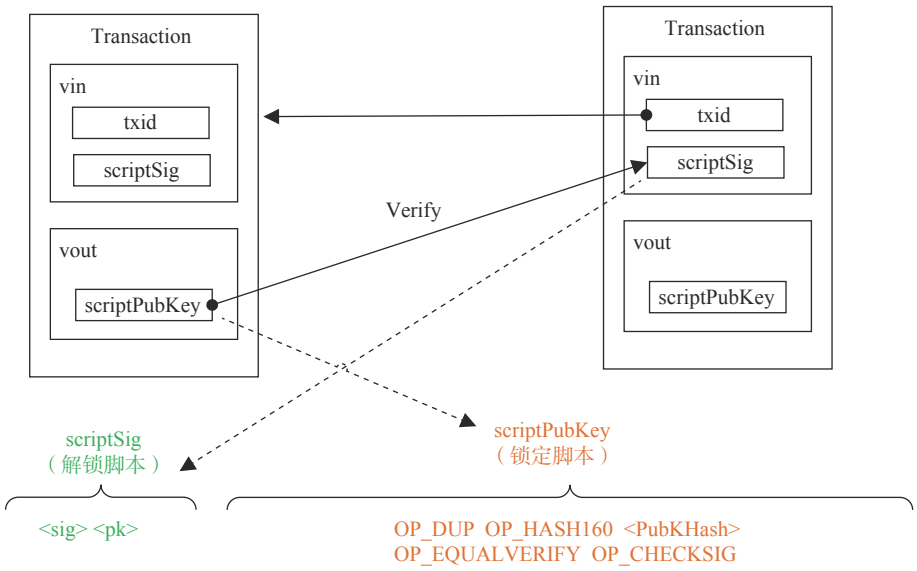

图 6　比特币交易脚本

Fig.6　Bitcoin transaction script

b）scriptSig: <signature> <pubkey>

P2PKH（Pay-to-Public-Key-Hash）是最常用的交易脚本，用于向公钥地址支付数字资产. 如图 6 所示，解锁脚本由<sig>签名与<pk>公钥组成，锁定脚本是由一连串堆栈命令和<pubkeyHash>公钥哈希组成，公钥哈希由两个哈希函数生成，大小 20 字节；比特币地址实际是由该公钥哈希进行 Base58check 编码而来，所以必须拥有该地址的私钥才能将锁定脚本解锁.

（2）Pay-to-Public-Key (P2PK)

a）scriptPubKey: <pubkey> OP_CHECKSIG

b）scriptSig: <signature>

P2PK 是 P2PKH 的简化形式，但不再用于新交易，因为 P2PKH 脚本更安全（公共密钥在输出用

完之前不会显示).

(3) Multi-Signature (MultiSig)

a) scriptPubKey: M <pubkey 1>...<pubkey N> N OP_CHECKMULTISI

b) scriptSig: OP_0 <signature 1>...<signature M>

Multisig 允许在几个地址之间共享数字资产的控制. 在创建脚本时, 可以指定控制资产的公钥, 以及签署有效支出交易所需的密钥数量.

(4) Pay-to-Script-Hash (P2SH)

a) scriptPubKey: OP_HASH160 <scriptHash> OP_EQUAL

b) scriptSig: <signatures> <script>

P2SH 是一段包含了其他脚本哈希值的脚本, 任何想要花费 P2SH 类型输出的交易需要提供匹配该哈希值的脚本.

(5) Data Output (OP_RETURN)

a) scriptPubkey: OP_RETURN <data>

b) NO scriptSig

Data outputs 常被用于向区块链中添加额外数据, 一般数据大小不超过 40 字节.

4　区块链网络

区块链建立在非中心化的、对等点对点(P2P)网络基础上, 可支持全球范围内任意人员自由接入退出[21]. 网络中的资源和服务分散在所有节点上, 信息的传输和服务的实现都直接在节点之间进行, 无需中间环节和服务器的介入, 避免了可能的瓶颈.

区块链按技术类型可分为公有链和联盟链.

公有链在网络拓扑上服从小世界模型(small world model)[22]. 该模型具有特征路径长度较小、聚合系数较大的特点, 在这样的网络中, 数据只需要经过较少节点(6 度原则)就可达到目的节点, 从而保证了交易信息在大规模区块链网络(10 万以上节点)中传播的高效性. 公有链网络拓扑的“高聚集度”和“短链”特征使得区块链可以支撑世界各地的海量用户进行大规模、并发地交易, 及时地将交易数据通过记账节点生成区块的方式存储, 并实现全网内的数据同步, 为保证区块链数据的健壮性、完整性和一致性奠定了网络基础.

联盟链基于全连通网络, 所有节点都是互连的. 每个节点能够管理较大规模用户, 包括, 用户个人信息、权限、密钥(包括公/私密钥)等. 与公有链任何人可随时加入和撤离不同, 联盟链中节点通常数目固定、接入管理更加严格.

5　共识算法

5.1　共识算法分类

共识算法(consensus algorithm)是指在多方协同环境下使所有参与方对任务执行结果达成一致(共识)的算法[23]. 共识算法多应用于确保分布式系统数据一致, 在区块链中引入共识算法最早是为了解决新交易块加入哈希链表中可能出现的“块冲突”问题[24], 也就是同时多个块被不同的块创建者加入到哈希链表中而引起的链表分叉(forking)问题[2], 它可能会导致双重花费(double spending)与交易无效的风险.

区块链共识算法根据设计思想可分为以下几类:

(1)证明类: 核心思想是建块节点需证明自己具有某种能力或完成了某种事情才能合法建块, 通常共识方式是完成一些难以解决却易于验证的难题去竞争建块的权利[25]. 常见的有工作量证明(proof of work, PoW)[26], 权益证明(proof of stake, PoS)[27] 等.

(2)拜占庭类: 以拜占庭协议为基础设计整个算法, 建块节点通常是由其他节点投票选举或从所有符合一定条件的节点中随机选举. 常见的有实用拜占庭容错算法(practical Byzantine fault tolerance, PBFT)[28], Algorand 算法[29] 等.

(3)传统共识类: 将传统分布式系统的一致性算法应用于区块链系统. 通常算法共识效率较高, 但不支持拜占庭容错, 即不考虑恶意篡改和伪造数据的拜占庭节点. 典型的有 Raft 算法[30].

(4)混合类共识: 使用多种共识算法的混合体来选择建块节点. 比如 PoW 和 PoS 混合的 Casper 算法[31], Raft 与 PBFT 混合的 Tangaroa 算法[32] 等.

以下几小节简要阐述包括工作量证明, 权益证明, 实用拜占庭容错和 Algorand 在内的几类代表性区块链共识算法.

5.2　工作量证明(PoW)

PoW 是比特币中使用的共识算法, 其核心思想是通过节点的计算能力, 即“算力”来竞争建块权和奖励. 算法关键是在区块头中加入不同的随机值, 计算区块头哈希值, 直到此哈希值小于或等于目标值, 解决此问题的过程称为挖矿(mining). 挖矿可分为两个步骤, 第一步为获取当前的目标值. 目标值T是一个 256 位的二进制数字, 图 3 中 nBits 是T的 32 位紧凑表示, 它可被 8 个 16 进制数 $0xh_0h_1h_2h_3h_4h_5h_6h_7$编码, T的计算公式[33] 为:

$$T = 0xh_2h_3h_4h_5h_6h_7 \times 2^{8\times(0xh_0h_1-3)} \quad (2)$$

最初的目标值在创世区块中被设置为0x1D00FFFF, 之后每2016个区块会后会重新计算目标值, 新目标值T_{new}的计算与建2016个块花费的总时间t_{sum}和新目标值的上一个目标值T_{old}有关. $14\times24\times60\times60$是14 d的总秒数, t_{sum}是花费的实际秒数, 则T_{new}的计算公式[33]为:

$$T_{new} = \frac{t_{sum}}{14\times24\times60\times60} \times T_{old} \quad (3)$$

第二步为变换随机值, 即图3中的nNonce, 使得H(Block Header)$\leqslant T$.

挖到矿的节点可以创建一个块并获得一定数量的比特币奖励. PoW具体步骤如下:

(1)矿工节点对一段时间内全网待处理的交易验证并将通过验证的交易打包, 然后计算这些交易的hashMerkleRoot;

(2)计算当前目标难度值T, 将目标难度值与hashMerkleRoot等其他字段组成区块头, 并将80字节的区块头作为PoW算法的输入;

(3) 不断变更块头中的随机数, 对变更后的区块头做双重SHA256哈希运算, 与目标值做比对, 如果小于等于目标值, 即PoW完成;

(4)矿工节点将上述区块向全网广播, 其他节点将验证其是否符合规则, 如果验证有效, 则将接收此区块, 并附加在已有区块链之后, 而后进入下一轮.

5.3　权益证明(PoS)

PoS是点点币(PPCoin)[34]使用的共识算法, 其核心思想是在PoW的基础上, 为减轻PoW计算哈希的工作量, 使用币龄作为一个变量来影响参与挖矿的难度, 挖矿难度同区块链中矿工拥有的代币数量与持有代币时间的乘积成反比.

挖矿主要方式为先获得当前节点的余额及代币持有时间和公开的目标值, 计算挖矿的真实目标值后, 再进行计算哈希值的步骤. PoS的执行过程与PoW类似, 只是多了记录币龄, 利用币龄计算真正目标值的过程.

5.4　实用拜占庭容错算法(PBFT)

PBFT是开源项目Hyperledger[35]使用的共识算法之一, 它是拜占庭算法的改进, 降低了算法的复杂程度, 相较于原算法更具实用性. 算法中有$3f+1$个节点, 可以容忍f个拜占庭错误的节点整个算法按照以下阶段来进行操作[28,36]:

(1)请求阶段: 客户端向主服务器节点发送请求;

(2)预准备阶段: 主节点接收请求后, 给请求赋值一个序列号, 主节点向其他服务器节点广播预准备消息, 其他服务器节点最初确定是否接受请求;

(3)准备阶段: 服务器节点选择接受请求, 则向其他所有服务节点广播准备消息, 并从其他节点接受准备消息, 在收到$2f+1$条消息后, 如果大多数节点接受请求, 则进入提交阶段;

(4)提交阶段: 处于提交状态的每个节点都向服务器中的所有其他节点发送提交消息. 同时, 如果服务器节点收到$2f+1$条提交消息, 则可以认为大多数节点达成共识以接受该请求. 然后, 节点执行请求消息中的指令, 并发送响应;

(5)响应阶段: 客户端等待来自不同节点的响应, 若有$f+1$个响应相同, 则该响应即为算法的一致结果.

5.5　基于Algorand的共识协议

Algorand是一种新型的拜占庭共识算法. Algorand设计目标是成为一种低能耗, 低分叉概率, 可扩展性好, 抗攻击能力强的共识算法, 为了达到这个目标, 采用了以下技术[29]:

(1)可验证随机函数(verifiable random function, VRF)[37]. VRF是一种伪随机函数, 能提供证明其输出正确性的公开可验证证据. 给定输入值x,密钥SK可以得到函数值$y=F_{SK}(x)$和证明P. 使用x、P和公钥PK, 每个人不需要SK就可检查$y=F_{SK}(x)$的正确性, 且无法得到SK的信息. 利用VRF和设定的目标值组合成"抽签"函数代替挖矿解决区块链建块需要大量计算哈希值所造成的能源浪费问题.

(2)委员会投票. 从全体用户中随机选取委员会成员, 让委员会成员进行建块, 验块等操作, 而非全体成员参与共识, 减轻网络负担, 增加了系统的可扩展性, 而且投票协商选出块的方式相较于各节点独自建块、广播块的行为更不容易造成"块冲突", 降低了分叉概率.

(3)加权用户. 为防止敌手大量制造"假名", 在选举中占据优势, 算法将用户权益与拥有资产相关联, 设定每个成员需有资产才有权被选为委员会成员, 且被选为委员会成员的可能性与资产数量正相关, 分散资产给"假名"的方式不能提高被选为委员会成员的概率, 攻击者在选举中不能占据优势.

(4)加密抽签和参与者替换. 为防止敌手攻击委员, 以加密抽签的方式选举委员, 且每次投票后需重新抽签. 加密抽签是指抽签结果只有抽签者

自己知道, 即委员会成员在参与共识以前, 其他节点无法知晓节点的委员身份, 所以敌手无法进行事前攻击; 委员会成员在参与共识, 投出票之后, 再次投票需要重新抽签, 这使敌手的事后攻击失去了意义.

表 1 给出常见的 5 种共识算法性能对比, 可见共识算法各有优势, 比如 PoW、PoS 及委托权益证明(DPoS)的执行速度和吞吐量(TPS)低于 PBFT 和 Raft, 但可扩展性强于两者; Raft 不支持拜占庭容错, 安全性方面低于 PBFT, 但 TPS 高于 PBFT. 事实上, 共识算法性能在效率、可扩展性和安全性方面确实存在博弈的过程, 难以全面提高, 由此才有各种共识算法共存, 应用于不同的场景, 而不是一种共识算法取代其他所有算法.

表 1　共识算法对比[36]

Table 1　Comparison of consensus algorithm[36]

共识算法	执行速度	可扩展性	拜占庭容错	TPS	代表应用
PoW	>100 s	强	<1/2	<100	比特币
PoS	<100 s	强	<1/2	<1000	点点币
DPoS	<100 s	强	<1/2	<1000	比特股
PBFT	<10 s	弱	<1/3	<2000	Hyperledger
Raft	<10 s	弱	不支持	>10000	Etcd

6　智能合约

6.1　智能合约概述

智能合约(smart contract)这个术语由跨领域法律学者尼克萨博(Nick Szabo)在 1995 年提出[38], 他给出的智能合约的定义是“一套以数字形式定义的承诺(promise), 包括合约参与方可以在上面执行这些承诺的协议[39]. ”

广义的智能合约指的是: 数字形式的可自动执行的协议; 狭义上的智能合约一般是指: 部署在区块链上可以自动执行的代码. 智能合约大致分为合约模块、执行模块和区块链模块三个模块, 如图 7 所示.

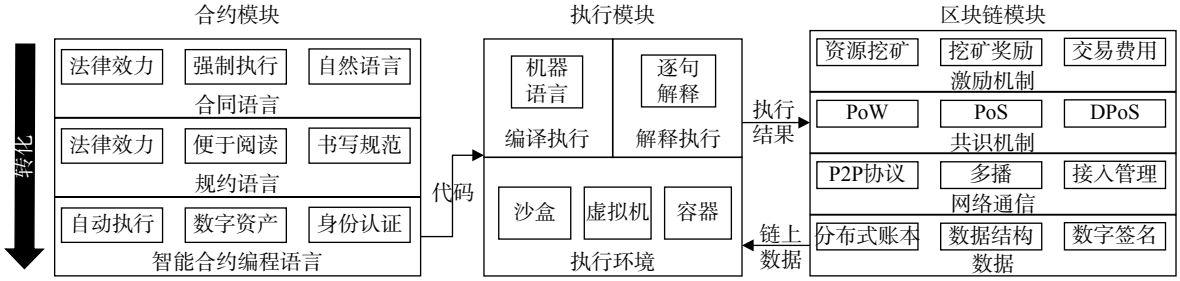

图 7　智能合约执行过程

Fig.7　Execution process of smart contracts

(1) 区块链模块: 区块链层封装了支持智能合约运行, 赋予智能合约去中心化、不可篡改等特性的关键技术.

(2) 执行模块: 执行层封装了智能合约的执行环境.

(3) 合约模块: 合约层封装了参与方从沟通协商到编写出一份智能合约中的语言.

6.2　区块链层与智能合约关系

智能合约执行是基于事件触发机制的, 区块链中保存智能合约的一个中间状态, 当有外部事件时如账户调用、定时器自动调用等, 调用者会先在本地执行合约, 通过从区块链中获取代码、合约状态, 计算出执行结果. 然后, 将调用参数与计算结果广播至全网, 由所有挖矿节点执行验证, 在达成共识之后与其他交易组装成区块, 记录在区块链中.

区块链与智能合约结合主要体现在以下几点:

(1) 智能合约采用区块链的共识机制, 由所有挖矿节点运行智能合约程序, 程序运算结果经过 P2P 网络传播并达成共识上传到区块链上.

(2) 区块链为智能合约提供存储空间. 区块链中的数据是经过共识的且不可篡改的, 保证了智能合约数据来源的可信性.

(3) 区块链的激励机制促使挖矿节点参与运行

和验证智能合约程序.

由于智能合约的验证过程需要全部挖矿节点运行智能合约, 一旦出现无限循环或是过于复杂的程序代码, 会占用大量计算资源甚至导致区块链网络崩溃, 且在公有链中这点很容易成为恶意节点的攻击方式, 需要采取一定限制措施. 如: 以太坊采用了收取交易费的方式来对合约的执行进行限制, 当用户支付的交易费不足时, 合约程序自动停机并回滚.

6.3　执行模块

目前各个平台智能合约的执行方式与执行环境有所不同, 下面主要以比特币、以太坊、Hyperledger Fabric 三个平台介绍.

比特币内置一套的基于栈的脚本执行引擎对脚本进行解释执行. 图 8 是解锁脚本对锁定脚本的解锁过程, 具体步骤如下:

	Combined Script(组合脚本)	Stacks（栈）
1	<sig>,<pk>,OP_DUP,OP_HASH256,<pk_hash>,OP_EQUALVERIFY,OP_CHECKSIG	
2	<pk>,OP_DUP,OP_HASH256,<pk_hash>,OP_EQUALVERIFY,OP_CHECKSIG	<sig >
3	OP_DUP,OP_HASH256,<pk_hash>,OP_EQUALVERIFY,OP_CHECKSIG	<pk >, < sig >
4	OP_HASH256,<pk_hash>,OP_EQUALVERIFY,OP_CHECKSIG	<pk >, < pk >, < sig >
5	<pk_hash>,OP_EQUALVERIFY,OP_CHECKSIG	SHA256d(pk), < pk >, < pk >, < sig >
6	OP_EQUALVERIFY,OP_CHECKSIG	<pk_hash >, SHA256d(pk), < pk >, < pk >, < sig >
7	OP_CHECKSIG	<pk >, < sig >
8		True

图 8　交易有效性校验过程

Fig.8　Transaction validity verification process

(1) 将解锁脚本和锁定脚本构成联合脚本, 从左至右依次入栈;

(2) 将<sig>入栈, 置于栈顶;

(3) 将<pk>入栈, 置于栈顶;

(4) 执行 OP_DUP 操作, 复制<pk>并置于栈顶;

(5) 对复制的<pk>值进行 OP_HASH256 哈希计算操作, 记做 SHA256(pk),置于栈顶;

(6) 将<pk_hash>入栈, 置于栈顶;

(7) 对 <pk_hash>和 SHA256(pk) 进行比较操作, 如果值一致, 则相继出栈;

(8) 将签名<sig>和公钥<pk>进行匹配, 如果匹配成功, 则会返回 True.

从图 8 可知, 比特币将解锁脚本与锁定脚本拼接后, 使用堆栈语言按后进先出的顺序执行, 最后完成检查, 如果返回结果为 TRUE, 代表解锁成功, 用户可以使用该笔输出作为新交易的输入进行交易. 如果返回为 FALSE 或者执行过程中出现问题, 则该交易无效.

以太坊智能合约属于编译执行, 先将智能合约转化为字节码, 再运行在以太坊虚拟机(Ethereum virtual machine, EVM)上. 以太坊虚拟机是由以太坊开发的, 且有许多针对智能合约的设计, 是一个完全独立的沙盒, 合约代码可对外完全隔离并在 EVM 内部运行.

Hyperledger Fabric 是 Linux 基金会所主导的 Hyperledger(超级账本)的项目之一. Fabric 使用现有的一种名为 Docker 的容器技术来支持智能合约功能, 理论上可以用任何语言来编写. Docker 也是独立的沙盒, 相较于 EVM 更加轻量, 但对于智能合约执行流程的控制没有 EVM 强.

6.4　合约模块

创建合约是将参与方协商达成一致的结果编译成计算机可执行代码. 在这一过程中, 需要考虑法律和金融等专家与程序员沟通问题、合约代码与法律对接的问题、代码逻辑安全性问题及智能合约语言表达能力问题等.

合约层包含三类合约语言:

(1) **合同语言**: 合同语言采用自然语言, 语法多样, 书写自由, 结合约定俗成的概念以及法律中的规定确保合同的法律效力.

(2) **智能合约编程语言**: 与传统编程语言类似, 具有表达能力强、语义明确和难以验证的特性, 阅读与编写需要编程领域专业知识.

(3) **规约语言**: 介于合同语言与智能合约编程语言之间的一种中间语言, 保留自然语言的可读性, 同时规范书写方式, 增强语义明确性, 能自动转化到智能合约编程语言.

目前智能合约编程语言有很多, 如以太坊的 Solidity、HyperLedger 采用的 Go 语言等, 各自的代码风格不同.

Hyperledge Fabric 智能合约的结构分为以下四部分: main、init、query 和 invoke 函数.

(1) **main 函数**: 作为程序的入口;

(2) **init 函数**: 在智能合约首次被部署时调用, 负责所有初始化工作;

(3) **query 函数**: 负责所有的查询;

(4) **invoke 函数**: 负责执行函数的调用.

以太坊支持多种智能合约语言, 以太坊智能合约语言是图灵完备的, 使得开发者能够创建任意的基于共识、可扩展、标准化、特性完备、易于开发和协同的应用, Solidity 语言与传统面向对象语言类似, 其结构大致如下:

(1) **变量定义**: 合约中的全局变量是存储在区块链中的, 这里定义了合约的属性以及需要记录的信息;

(2) **事件定义**: 定义需要记录的事件及参数;

(3) **修饰器定义**: 修饰器用来修饰函数, 判断函数执行条件;

(4) **函数定义**: 具体函数实现.

合同语言写法自由, 且依赖于常识与法律, 很难直接向智能合约编程语言转化. SPESC[40] 是一种类自然语言的智能合约规约语言, 试图通过类自然语言、规范化书写化的结构与书写、半自动化向 Solidity 转换的方式解决不同领域专家沟通、法律效力以及部分逻辑安全问题. SPESC 结构如下:

(1) **角色定义**: 定义参与方的角色, 同时定义角色属性与可执行动作;

(2) **变量定义**: 定义合约中需要记录的信息;

(3) **条款定义**: 定义角色对应的权利或义务, 包含角色、动作、执行条件、资金转移、后置条件等信息.

SPESC 条款的表示类似于自然语言, 如:

term no2 Seller **can** abort

when before buyer did pay

表示卖家可以在买家付款前终止合约.

6.5 面向隐私的智能合约框架

区块链中的所有交易、合约和用户余额都可被链上的用户公开查看. 比如在竞标合约当中, 竞标参与方不想让其他人知道自己的出价, Solidity 的技术文档[41] 提出参与方可以在智能合约中通过加密手段出价, 最后竞标结束再公布价格. 但是, 如果要保证竞标者不违约, 需要竞标者在投标时向合约转入资金, 然而资金的转移也是公开的, 于是只能设置一个最高价或是设置一些虚假出价, 竞标者提前将最高价转入合约, 但这种方式显然有不合理之处.

Kosba 等[42] 提出一种可以自动生成加密协议的工具 Hawk, 通过在区块链上存储金融交易的特殊加密形式, 从而保证链上数据只有拥有者可见, 但其分布式沙箱的构造仍是一个未解决的问题.

还有一些人通过零知识证明的方法扩展交易协议[43], 构建货币体系[44], 以此来加强匿名性, 从而保证交易的隐私性, 但结构针对性较强, 缺乏普适性.

在文献 [45] 和 [46], 多方安全计算被应用到了比特币脚本中, 保证交易的忠实性和隐私性, 文献 [47] 中多方安全计算则被推广到了一般的智能合约中, 封装了多方安全计算的计算过程与网络通信, 上层智能合约编程人员只需将计算符号替换为安全计算符号而无需考虑实现问题.

6.6 智能合约存在的问题

智能合约在以以太坊为代表的区块链推动下, 有了很大的发展, 技术日益成熟, 但是发展过程中依旧存在一些问题[48].

(1) **安全问题**: 2016 年 5 月, 以太坊最大众筹项目 The DAO 被攻击者利用程序中的递归调用漏洞获取了大量以太币, 这个问题引发了大众对于智能合约安全性和法律问题的思考. Luu 等[49] 提出了检查智能合约漏洞的工具 Oyente, 文中利用工具检查了以太坊中包含 3068654 以太币、价值三千万美金的 19366 个合约, 其中有 8833 个合约至少包含一种安全漏洞.

(2) **法律问题**: The DAO 事件对法律也提出了巨大挑战. 首先, 界定“漏洞”困难, 智能合约与实际合约无法显示转换; 其次是由于区块链不可更改, 智能合约难以撤销. 目前也有一些对于智能合约从法律的角度上讨论模板与标准化格式, 讨论了法律与参数、代码联系以及数据标准与合约特点的层面.

(3) **效率问题**: 区块链系统本身存在的吞吐量低、交易延迟、能耗过高、容量和带宽限制等性能问题极大限制了智能合约的执行效率, 且由于智能合约顺序执行, 当交易增多以后, 将导致交易确认很慢, 且交易费会变高.

7 挑战与机遇

尽管目前区块链技术日益成熟, 但是仍然存在一些问题. 比如: 资源浪费问题, BitCoin 中的挖矿步骤需要计算机系统进行大量的计算, 造成时

间、电力、物力等资源浪费. 此外，面对大量的互联网业务，系统吞吐量（TPS）也是一项重要性能指标，它是指系统每秒钟能够处理的交易数量. 假如 TPS 每秒并发太低，很容易造成网络拥堵严重，从而使得区块链在高价值的高并发业务领域无法落地. 比如，由于 TPS 每秒并发太低，BitCoin 和 Ethereum 都存在交易费用高、确认时间长、扩展性差的问题. 随着技术的进步，TPS 已经由 BitCoin 的 7、LiteCoin 的 56、Ripple 的 1500，逐渐接近 7000，使得基于区块链的大宗交易成为可能.

此外，分叉问题也是一个亟待解决的问题，即存在多个节点因缺乏同步机制而各自自行建块，区块链某一后续区块会出现多个合法块，进而导致其他节点会在不同的块后继续建块，从而出现链分叉. 分叉会导致节点上的数据不一致问题，造成正常交易失败或双重花费等问题. 为了保持节点们数据一致性，区块链一般采用的是“最长链原则”，即长度最长的链为合法链（主链），因为通常长度更长的区块链受更多节点的认同，消耗的计算力也更多. 然而，“最长链原则”并没有从根本上解决问题，为了确认一笔交易，必须等到该交易所在块后建了足够多的块才能确认，这会带来很大的交易延迟，对于小额交易是很难接受的，而且即便确认也不是最终确认，如果敌手有超过区块链网络中 50% 的算力，就有可能颠覆之前确认过的交易.

随着智能合约的兴起，由脚本语言和执行机制引发的安全问题也日益严重，最为著名的是 MT.Gox 加密货币失窃事件. 在这个事件中，攻击者利用交易脚本中的指令可替换性，逃避了对脚本的语义检查，但却可以通过不同指令编码实现货币的转移，即由一个钱包地址转向另外一个钱包地址. 通过这种方法，攻击者攻击了位于日本的 MT.Gox 交易所，导致总计 74.4 万个比特币失窃，价值约为 3.5 亿美金. 由此可见，尽管有各种安全机制，但交易脚本或智能合约的安全漏洞依然是不可避免的，还有待于开发更加完善和有效的检测和保障技术. 除此之外，区块链还面临着各种传统安全攻击的威胁，例如，DoS 攻击、Sybil 攻击等. 因此，改进和增强区块链安全性仍然是任重道远的挑战性任务.

目前，上述问题在学术界有了新进展，如文献 [50] 中 Bitcoin-NG 区块链协议尝试在不牺牲性能情况下提高区块链的吞吐率，基本方式是将比特币的区块链操作分解成两部分：首领选择（leader election）和交易序列化（transaction serialization）. 该协议介绍了两种类型的区块：用于首领选择的关键区块（Key-Block）和包含账本记录的微区块（Micro-block）. 相较于比特币 PoW 来选择当前区块，Bitcoin-NG 使用 PoW 来选择关键区块，生成关键区块的节点成为首领，可写入多个微区块. 由于这些块生成间隔很短，能总体上提高区块链的吞吐率.

为了解决区块链为降低分叉概率而限制性能的问题，文献中 [51] 的提到了一种基于 blockDAG 的协议，由于有向无环图 DAG 的引用，与传统区块链一个块只有一个父亲相比，blockDAG 中区块可引用多个父辈，因而可支持“分叉”结构并可异步并发写入多个交易，再通过排序达成共识，因此 blockDAG 协议与链式协议相比具有更高扩展性.

针对 Bitcoin 确认交易的时间以小时计，文献 [29] 中提出 Algorand 机制尝试以分钟为单位确认交易，该机制使用“抽签”函数代替 PoW，即通过可验证随机函数来选取建块者，再通过一种新型的拜占庭（Byzantine）协议扩展到多方. 这使得 Algorand 无需重复计算哈希值就能低延迟地在新块上达成共识，同时通过基于资产的授权控制防止 Sybil 攻击.

8　总结和展望

区块链是一次互联网技术的大变革，它使得人们看到全球性的协同计算正成为可能，区块链的实际应用给互联网金融、贸易、司法等各行各业带来了观念、措施、制度上的革新. 在区块链带来巨大科技创新和思维创新的背景下，区块链的进一步推进也正日益引起科研工作者的浓厚兴趣. 同时，一系列新的问题还有待进一步从理论层面和应用实践中得到解决和验证.

参　考　文　献

[1] Nakamoto S. Bitcoin: a peer-to-peer electronic cash system[J/OL]. *Bitcoin* (2018−06−10)[2019−03−01]. https: //bitcoin.org/bitcoin.pdf

[2] Decker C, Wattenhofer R. Information propagation in the bitcoin network//*IEEE P2P* 2013 *Proceedings*. Trento, 2013: 1

[3] Crosby M, Nachiappan P P, Verma S, et al. Blockchain technology: Beyond bitcoin. *Appl Innovation Rev*, 2016(2): 6

[4] Xu X W, Pautasso C, Zhu L M, et al. The blockchain as a software connector//2016 13*th Working IEEE/IFIP Conference on Software Architecture* (*WICSA*). Venice, 2016: 182

[5] Pilkington M. *Blockchain Technology: Principles and Applications*. Northampton: Edward Elgar Publishing, 2016

[6] An R, He D B, Zhang Y R, et al. The design of an anti-counterfeiting system based on blockchain. *J Cryptologic Res*, 2017, 4(2): 199

[7] Tian H B, He J J, Fu L Q, et al. A privacy preserving fair contract signing protocol based on block chains. *J Cryptologic Res*, 2017, 4(2): 187

[8] Mettler M. Blockchain technology in healthcare: The revolution starts here//2016 *IEEE* 18*th International Conference on e-Health Networking*, *Applications and Services* (*Healthcom*). Munich, 2016: 1

[9] Wang Y D, Li L, Hu D. A literature review of block chain. *J China Univ Min Technol Social Sci*, 2018(3): 74
(王元地, 李粒, 胡谍. 区块链研究综述. 中国矿业大学学报:社会科学版, 2018(3): 74)

[10] Lin I C, Liao T C. A survey of blockchain security issues and challenges. *Int J Network Security*, 2017, 19(5): 653

[11] Zheng Z B, Xie S A, Dai H N, et al. Blockchain challenges and opportunities: a survey. *Int J Web Grid Services*, 2018, 14(4): 352

[12] Randall D, Goel P, Abujamra R. Blockchain applications and use cases in health information technology. *J Health Med Informat*, 2017, 8(276): 2

[13] Yuan Y, Wang F Y. Blockchain: The state of the art and future trends. *Acta Autom Sinica*, 2016, 42(4): 481
(袁勇, 王飞跃. 区块链技术发展现状与展望. 自动化学报, 2016, 42(4): 481)

[14] Goldwasser S, Micali S. Probabilistic encryption. *J Comput Syst Sci*, 1984, 28(2): 270

[15] Evans D S. Economic aspects of bitcoin and other decentralized public-ledger currency platforms[J/OL]. *SSRN* (2014−04−15) [2019−03−01]. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2424516

[16] Garay J, Kiayias A, Leonardos N. The bitcoin backbone protocol: Analysis and applications//*Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Berlin, 2015: 281

[17] Barbosa M, Farshim P. On the semantic security of functional encryption schemes//*International Workshop on Public Key Cryptography*. Berlin, 2013: 143

[18] Lamport L. The part-time parliament. *ACM Trans Comput Syst*, 1998, 16(2): 133

[19] Szydlo M. Merkle tree traversal in log space and time//*International Conference on the Theory and Applications of Cryptographic Techniques*. Berlin, 2004: 541

[20] Jakobsson M, Leighton T, Micali S, et al. Fractal Merkle tree representation and traversal//*Cryptographers' Track at the RSA Conference*. Berlin, 2003: 314

[21] Shen X, Pei Q Q, Liu X F. Survey of block chain. *Chin J Network Inf Security*, 2016, 2(11): 00107-1
(沈鑫, 裴庆祺, 刘雪峰. 区块链技术综述. 网络与信息安全学报, 2016, 2(11): 00107-1)

[22] Adamic L A. The small world web//*International Conference on Theory and Practice of Digital Libraries*. Berlin, 1999: 443

[23] Zheng Z B, Xie S A, Dai H N, et al. An overview of blockchain technology: Architecture, consensus, and future trends//2017 *IEEE International Congress on Big Data* (*BigData Congress*). Honolulu, 2017: 557

[24] Baliga A. Understanding blockchain consensus models[J/OL]. *PersistentSystems*(2017−04)[2019−03−01].https://pdfs.semanticscholar.org/da8a/37b10bc1521a4d3de925d7ebc44bb606d740.pdf?_ga=2.21200635.1919538867.1522092864-1798624458.1520283070&source=post_page

[25] Bach L M, Mihaljevic B, Zagar M. Comparative analysis of blockchain consensus algorithms//2018 41*st International Convention on Information and Communication Technology*, *Electronics and Microelectronics* (*MIPRO*). Opatija, 2018: 1545

[26] Jakobsson M, Juels A. Proofs of work and bread pudding protocols//*Secure Information Networks*. Boston: Springer, 1999: 258

[27] Kiayias A, Russell A, David B, et al. Ouroboros: A provably secure proof-of-stake blockchain protocol//*Annual International Cryptology Conference*. Santa Barbara: Springer, 2017: 357

[28] Castro M, Liskov B. Practical byzantine fault tolerance// *Proceedings of the* 3*rd Symposium on Operating Systems Design and Implementation*. New Orleans, 1999: 173

[29] Gilad Y, Hemo R, Micali S, et al. Algorand: Scaling byzantine agreements for cryptocurrencies//*Proceedings of the* 26*th Symposium on Operating Systems Principles*. Shanghai, 2017: 51

[30] Ongaro D, Ousterhout J. In search of an understandable consensus algorithm//2014 *Annual Technical Conference* ({*USENIX*} {*ATC*} 14).San Diego, 2014: 305

[31] Lowe G. Casper: a compiler for the analysis of security protocols. *J Comput Security*, 1998, 6(1-2): 53

[32] Copeland C, Zhong H X. Tangaroa: a byzantine fault tolerantraft[J/OL]. *Stanford University Press* (2018−04−10)[2019−03−01]. http://www.scs.stanford.edu/14au-cs244b/labs/projects/copeland_zhong.pdf

[33] Gervais A, Karame G O, Wüst K, et al. On the security and performance of proof of work blockchains//*Proceedings of the* 2016 *ACM SIGSAC Conference on Computer and Communications Security*. Vienna, 2016: 3

[34] King S, Nadal S. PPCoin: peer-to-peer crypto-currency with proof-of-stake[J/OL]. *Bitcoin*(2012−08−19)[2019−03−01].https://bitcoin.peryaudo.org/vendor/peercoin-paper.pdf

[35] Cachin C. Architecture of the hyperledger blockchain fabric// *Workshop on Distributed Cryptocurrencies and Consensus Ledgers*. Switzerland, 2016

[36] Du M X, Ma X F, Zhang Z, et al. A review on consensus algorithm of blockchain//2017 *IEEE International Conference on Systems*, *Man*, *and Cybernetics* (*SMC*). Banff, 2017: 2567

[37] Micali S, Rabin M, Vadhan S. Verifiable random functions//40*th Annual Symposium on Foundations of Computer Science* (*Cat. No. 99CB37039*). New York, 1999: 120

[38] Linnhoff-Popien C, Schneider R, Zaddach M. *Digital Marketplaces Unleashed*. Berlin: Springer, 2018

[39] Szabo N. Smart contract[J/OL]. *Phonetic Sciences*[2018−05−30] [2019−03−01]. http://www.fon.hum.uva.nl/rob/Courses/InformationInSpeech/CDROM/Literature/LOTwinterschool2006/szabo.best.vwh.net/smart.contracts.html

[40] He X, Qin B H, Zhu Y, et al. SPESC: a specification language for smart contracts//2018 *IEEE* 42*nd Annual Computer Software and Applications Conference* (*COMPSAC*). Tokyo, 2018: 132

[41] Ethereum. Solidity by Example[DB/OL]. *Ethereum Revision* (2016)[2019−03−02]. https://solidity.readthedocs.io/en/latest/solidity-by-example.html#blind-auction

[42] Kosba A, Miller A, Shi E, et al. Hawk: The blockchain model of cryptography and privacy-preserving smart contracts//2016 *IEEE Symposium on Security and Privacy* (*SP*). San Jose, 2016: 839

[43] Miers I, Garman C, Green M, et al. Zerocoin: Anonymous distributed e-cash from bitcoin//2013 *IEEE Symposium on Security and Privacy*. Berkeley, 2013: 397

[44] Sasson E B, Chiesa A, Garman C, et al. Zerocash: Decentralized anonymous payments from bitcoin//2014 *IEEE Symposium on Security and Privacy*. San Jose, 2014: 459

[45] Andrychowicz M, Dziembowski S, Malinowski D, et al. Secure multiparty computations on bitcoin//2014 *IEEE Symposium on Security and Privacy*. San Jose, 2014: 443

[46] Andrychowicz M, Dziembowski S, Malinowski D, et al. Fair two-party computations via bitcoin deposits//*International Conference on Financial Cryptography and Data Security*. Berlin, 2014: 105

[47] Zhu Y, Song X X, Xue X B, et al. Smart contract execution system over blockchain based on secure multi-party computation. *J Cryptologic Res*, 2018, 6(2): 246
(朱岩, 宋晓旭, 薛显斌, 等. 基于安全多方计算的区块链智能合约执行系统. 密码学报, 2018, 6(2): 246)

[48] Wang H, Song X F, Ke J M, et al. Blockchain and privacy preserving mechanisms in cryptocurrency. *Netinfo Security*, 2017, 7: 32

[49] Luu L, Chu D H, Olickel H, et al. Making smart contracts smarter//*Proceedings of the* 2016 *ACM SIGSAC Conference on Computer and Communications Security*. New York, 2016: 254

[50] Eyal I, Gencer A E, Sirer E G, et al. Bitcoin-NG: a scalable blockchain protocol//13*th* {*USENIX*} *Symposium on Networked Systems Design and Implementation* ({*NSDI*} 16). Santa Clara, 2016: 45

[51] Sompolinsky Y, Zohar A. PHANTOM: a scalable blockDAG protocol. *IACR Cryptology ePrint Archive*, 2018, 2018: 104