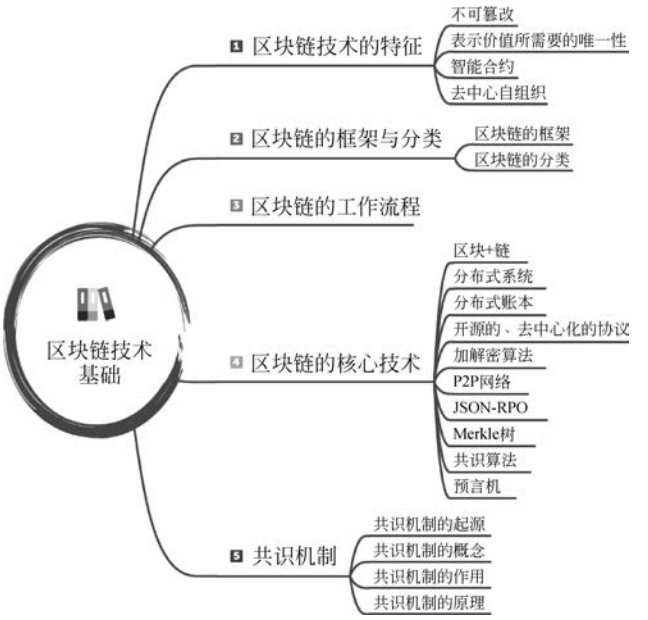
# 第3章

# 区块链技术基础

**本章思维导图**




2009年出现的比特币使得一切关于货币、互联网和价值传递的问题，同时也是困扰网络创造者的棘手问题，似乎迎来了解决方案，巧妙融合P2P网络、密码学、共识算法等已有技术的比特币优雅地解决了在互联网上产生、存储、传递和交换价值的问题。在这个系统中，数据以区块(block)为单位产生和存储，按时间顺序连接成链式结构(chain)，彼此独立的节点共同参与数据的验证、存储、维护，具有不可篡改、公开透明的特点。blockchain被翻译成“区块链”，它建立了在不可信网络中进行信息和价值的传递、交换的可信机制。

2010年2月，中本聪曾在论坛中发帖称，20年内比特币要么归零，要么无比强大。经过极客、技术布道者、加密货币爱好者乃至敏锐资本的推广和发展，比特币从籍籍无名发展到誉满天下，由其衍生的区块链技术开枝散叶，必将带领人类迈入新纪元。而针对区块链技术，各国政府、大资本、技术先驱甚至普罗大众，已慢慢建立起共识——继蒸汽机、电力、互联网之后，区块链或许是下一代颠覆性的核心技术，支持互联网从信息互联网向价值互联网进化。

网络的发展虽然日新月异，但是信息传播和价值传递的突破性进展都不是偶然事件，从研究者和爱好者夜以继日地尝试、探索，到普通用户的应用反馈，都要经历漫长而艰苦的演

变过程。随着技术和需求趋于稳定成熟，每个新的发展阶段都可以在前一个阶段的基础上建设和创新，穿过历史重重迷雾，回看信息互联网的发展历程，我们应当对建设和使用价值互联网有充足的心理准备，虽然技术是现成的，但大规模的普及应用不可能一蹴而就。那么，区块链究竟是一门怎样的技术，竟有如此魅力？俗话说："外行看热闹，内行看门道"，接下来让我们一探究竟。

区块链技术的特征

## 3.1 区块链技术的特征

在第 2 章通过对比特币和以太坊这两个主要系统的介绍，讨论了区块链的进化和基本理论之后，本章来看看区块链的特征与用途，尝试回答"区块链有什么用"这个问题。答案就藏在区块链的四个基础特性中。

在讲解了以太坊带来的变化后，区块链特征以及与其相关的应用已经较为清晰地展现出来。这四个基础特征分别是：不可篡改，不可复制的唯一性，智能合约，去中心自组织或社区化，如图 3-1 所示。


图 3-1 一张图看懂区块链：从基础到应用

区块链不仅仅影响技术层面，它还将从经济、管理、社会等层面带来变化，它可能改变人类交易的方式，包括货币、账本、合同、协同等，这些将在后续章节中讨论。

接下来分别讨论区块链的这四个基础特性。

### 3.1.1 不可篡改

区块链最容易被理解的特性是不可篡改。

不可篡改是基于"区块＋链"(block＋chain)的独特账本而形成的：存储交易数据的区块按照时间顺序持续加到链的尾部，要修改一个区块中的数据，就需要重新生成它之后的所有区块。

共识机制的重要作用之一是使得修改大量区块的成本极高，几乎是不可能实现的。以采用工作量证明机制的区块链网络（如比特币、以太坊）为例，只有拥有51%或以上的算力才可能重新生成所有区块以篡改数据。但是，破坏数据并不符合拥有大算力的玩家的自身利益，这种实用设计增强了区块链上的数据可靠性。

通常，在区块链账本中的交易数据可以视为不能被“修改”，它只能通过被认可的新交易来“修正”，修正的过程会留下痕迹，因此说区块链是不可篡改的（篡改是指用作伪的手段改动或曲解）。

在现在常用的文件和关系数据库中，除非采用特别的设计，系统本身是不记录修改痕迹的。区块链账本采用的是与文件、数据库不同的设计，借鉴了现实中的账本设计——留存记录痕迹。因此，我们无法不留痕迹地“修改”区域链账本，而只能“更正”区域链账本，如图3-2所示。

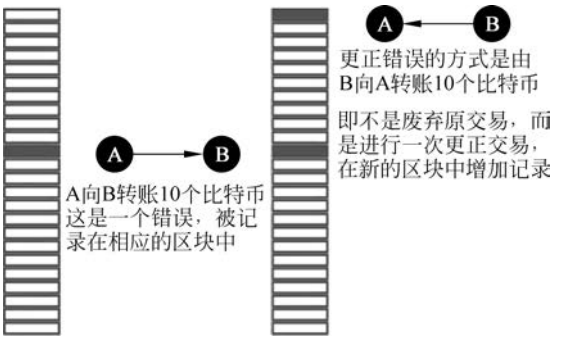



图3-2　区块链账本“不能修改、只能更正”

区块链的数据存储被称为“账本”（leger，总账），这是非常符合其实质的名称。区块链账本的逻辑和传统的账本相似。例如，A可能因错误转了一笔钱给B，这笔交易被区块链账本接受，记录在其中。更正错误的方式不是直接修改账本，将它恢复到这个错误交易前的状态，而是进行一笔新的更正交易，B把这笔钱转回给A。当新交易被区块链账本接受，错误就被修正。所有的更正过程都记录在账本之中，有迹可循。

将区块链投入使用的一类设想正是利用它的不可篡改特性。在区块链系统中建立的电子合同，是没办法让一些中心化的组织来更改合同中的条款，让每一次交易都停留在双方买卖的第一笔记录上。随着区块链技术的发展，被开发出来的智能合约，与不可篡改这项技术进行结合，可以让所有进行交易的买卖双方，自动根据合约内容、时间进行交易。同时，还不能随意篡改双方的交易记录。

2018年3月，在网络零售集团京东发布的《区块链技术实践白皮书》中，区块链技术（分布式账本）的三种应用场景是：跨主体协作，需要低成本信任，存在长周期交易链条。这三个应用场景所利用的都是区块链的不可篡改特性。多主体在一个不可篡改的账本上协作，降低了信任成本。区块链账本中存储的是状态，未涉及的数据的状态不会发生变化，且越早的数据越难被篡改，这使得它适用于长周期交易。

## 3.1.2　不可复制的唯一性

不管是可互换通证（ERC20），还是不可互换通证（ERC721），又或者是其他提议中的通

证标准，以太坊的通证都展示了区块链的一个重要特征：表示价值所需要的唯一性。

在数字世界中，最基本的单元是比特，比特的根本特性是可复制。但是价值不能被复制，价值必须是唯一的。之前已经讨论过，这正是矛盾所在：在数字世界中，很难使一个文件是唯一的，至少很难普遍地做到这一点。这正是现在我们需要中心化的账本来记录价值的原因。

在数字世界中，用户没法像拥有现金一样，手上拿着钞票，而是需要银行等信用中介，用户的钱是由银行账本帮忙记录的。

比特币系统带来的区块链技术可以说第一次把“唯一性”普遍地带入了数字世界，而以太坊的通证将数字世界中的价值表示功能普及开来。

2018 年年初，中国的两位科技互联网企业领袖不约而同地强调了区块链带来的“唯一性”。腾讯公司主要创始人、CEO 马化腾说：“区块链确实是一项具有创新性的技术，用数字化表达唯一性，区块链可以模拟现实中的实物唯一性。”

百度公司创始人、CEO 李彦宏说：“区块链到来之后，可以真正使虚拟物品变得唯一，这样的互联网跟以前的互联网是非常不一样的。”

对于通证经济的探讨和展望正是基于在数字世界中、在网络基础层次上区块链提供了去中心化的价值表示和价值转移的方式。在以以太坊为代表的区块链 2.0 时代，出现了更通用的价值代表物——通证，从而由区块链 1.0 的数字现金时期进入数字资产时期。

### 3.1.3 智能合约

从比特币到以太坊，区块链最大的变化是“智能合约”，如图 3-3 所示。比特币系统是专为一种数字货币而设计的，它的 UTXO 和脚本也可以处理一些复杂的交易，但有很大的局限性。而维塔利克创建了以太坊区块链，他的核心目标都是围绕智能合约展开的：一个图灵完备的脚本语言，一个运行智能合约的虚拟机（EVM），以及后续发展出来的一系列标准化的、用于不同类型通证的智能合约等。

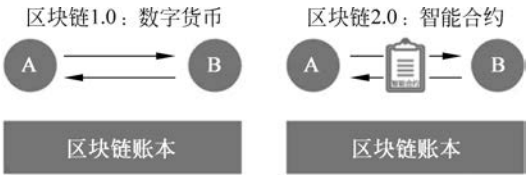



图 3-3 区块链 2.0 的关键改进是“智能合约”

智能合约的出现使得基于区块链的两个人不仅可以进行简单的价值转移，而且可以设定复杂的规则，由智能合约自动、自治地执行，这极大地扩展了区块链的应用可能性。

当前把焦点放在通证的创新性应用上的项目，在软件层面都是通过编写智能合约来实现的。利用智能合约，可以进行复杂的数字资产交易。

在讨论区块链的进化过程时，我们介绍了智能合约的几种实例，在此不再赘述。这里再借维塔利克的讨论，重复一下我们认同的智能合约的软件性质——它相当于一种特殊的服务端后台程序（daemon）。在以太坊白皮书中，维塔利克写道：“（合约）应被看成存在于以太坊执行环境中的‘自治代理’（autonomous agents），它拥有自己的以太坊账户，收到交易信息，它们就相当于被捅了一下，然后它就自动执行一段代码。”

智能合约的执行流程如图 3-4 所示。

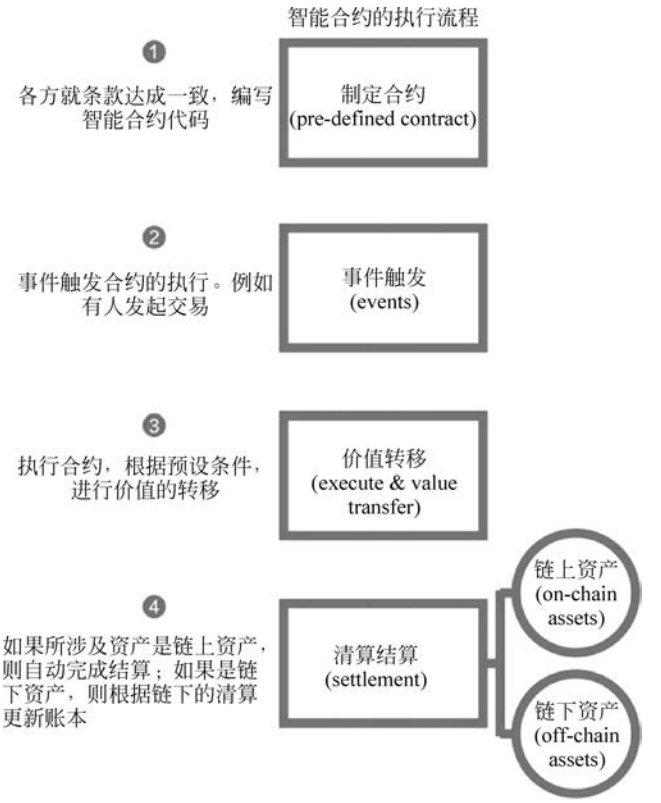



图 3-4　智能合约的执行流程

## 3.1.4　去中心自组织

区块链的第四大特征是去中心自组织。到目前为止，主要区块链项目的自身组织和运作都与这个特征紧密相关。很多人对区块链项目的理想或期待是，它们成为自治运转的一个社区或生态。

匿名的中本聪在完成比特币的开发和初期的迭代开发之后，就完全从互联网上消失了。但他创造的比特币系统持续地运转着：无论是比特币这个加密数字货币，比特币协议（即比特币的发行与交易机制），比特币的分布式账本、去中心化网络，还是比特币矿工和比特币开发，都在去中心化、自组织地运转着。

我们可以合理地猜测，在比特币之后出现了众多修改参数分叉形成的竞争币、硬分叉形成的比特币现金（BCH），可能都符合中本聪的设想。中本聪选择了“失控”，这里的失控可视为自治的同义词。

到目前为止，以太坊项目仍在维塔利克的“领导”之下，但正如本章一开始讨论的，他是以领导一个开源组织的方式引领着这个项目，就像 Linus 领导开源的 Linux 操作系统和 Linux 基金会一样。

维塔利克可能是对去中心自组织思考得较多的人之一，他一直强调和采用基于区块链的治理方式。2016 年以太坊的硬分叉是他提议的，但需要通过链上的社区投票、获得通过

方可实行。在以太坊社区中，包括 ERC20 等在内的众多标准是社区开发者自发形成的。

在《去中心化应用》一书中，作者西拉杰·拉瓦尔（Siraj Raval）还从另一个角度进行了区分，这个区分有助于我们更好地理解未来的应用与组织。他从两个维度来看现有的互联网技术产品：一个维度是在组织上是中心化的，还是去中心化的；另一个维度是在逻辑上是中心化的，还是去中心化的。

他认为“比特币在组织上去中心化，在逻辑上集中”。而电子邮件系统在组织上和逻辑上都是去中心化的，如图 3-5 所示。

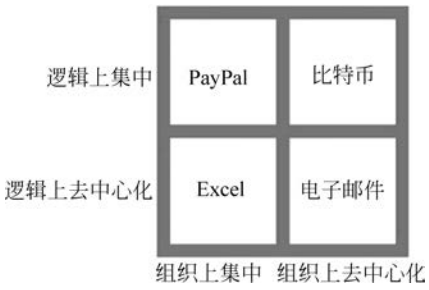



图 3-5　比特币在组织上去中心化，在逻辑上集中

在设想未来的组织时，我们心中的理想原型常是比特币的组织，即完全去中心化的自治组织。但在实践过程中，为了保证效率、能够推进，我们又会略微往中心化组织靠拢，最终找到一个合适的平衡点。

现在，在通过以太坊的智能合约创建和发放通证，并以社区或生态方式运行的区块链项目中，不少项目的理想状态是类似于比特币的组织，但实际情况是介于完全的去中心化组织和传统的公司之间。

在讨论区块链的第四个特征去中心自组织时，其实我们已经在从代码的世界往外走，涉及人的组织与协同了。现在，各种讨论和实际探索也揭示了区块链在技术之外的意义：它可能作为基础设施支持人类的生产组织和协同的变革。这正是区块链与互联网完全同构的又一例证，互联网也不仅仅是一项技术，它改变了人们的组织和协同方式。

总的来说，以太坊把区块链带入了新的阶段。在讨论以太坊时，如果要总结两个关键词的话，那么这两个关键词分别是智能合约和通证；而如果只能选一个的话，笔者会选择“通证”。笔者会更愿意从互联网的历史中找寻它的意义，重复之前的类比：作为价值表示物的通证，它的角色类似于 HTML。在有了 HTML 之后，建成什么样的网站完全取决于我们的想象力。

现在，很多人迫不及待地试图进入区块链 3.0 阶段，即不再仅仅把区块链用于数字资产的交易，而是希望将区块链应用于各个产业和领域中，从互联网赋能走向区块链赋能，从“互联网＋”走向“区块链＋”。继续将信息互联网的发展历程作为对照来展望未来，信息互联网最早是用来传递文本信息的，但它真正的爆发是由于后来出现的电子商务、社交、游戏以及和线下结合的 O2O——也就是应用。未来真正展现区块链价值的也将是各种现在尚未知的应用。

# 3.2　区块链的框架与分类

## 3.2.1　区块链的框架

关于区块链的框架，已有不少学者和专家进行过阐述，其中比较有代表性的是发表于2016 年第 4 期《自动化学报》的文章《区块链技术发展现状与展望》，该文首次将区块链的框架划分为六层结构，认为区块链系统由数据层、网络层、共识层、激励层、合约层和应用层组成。其中，数据层作为最底层封装了数据区块以及相关的数据加密和时间戳等技术；网络层包括分布式组网机制、数据传播机制和数据验证机制等；共识层主要封装网络节点的各类共识算法；激励层将经济因素集成到区块链技术体系中来，主要包括经济激励的发行机制和分配机制等；合约层主要封装各类脚本、算法和智能合约，是区块链可编程特性的基础；应用层封装了区块链的各种应用场景和案例。本书对区块链框架进行梳理优化，认为区块链的框架应划分为四层，如图 3-6 所示，包括底层数据层、网络通信层、共识验证层和业务应用层。

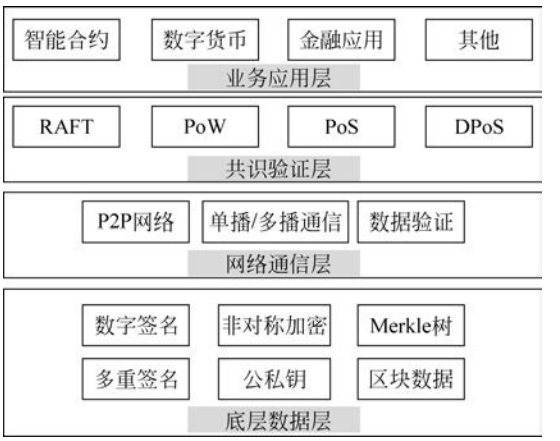



图 3-6　区块链框架

**1. 底层数据层**

底层数据层是最底层的技术，是一切的基础。它主要实现了两个功能：一个是相关数据的存储，另一个是账户和交易的实现与安全。数据的存储主要基于 Merkle 树，通过区块的方式和链式结构实现。账号和交易的实现基于数字签名、Hash 函数、非对称加密技术、多重签名等多种密码学算法和技术，保证了交易在去中心化的情况下能够安全地进行。

**2. 网络通信层**

网络通信层主要实现网络节点的连接和通信，包括点对点技术、单播/多播通信和验证技术，是没有中心服务器、依靠用户群交换信息的互联网体系。与有中心服务器的中央网络系统不同，对等网络的每个用户端既是一个节点，又有服务器的功能，具有健壮性、去中心化等特点。

**3. 共识验证层**

共识算法是区块链体系的核心。共识验证层主要实现全网所有节点对交易和数据达成

一致，防范拜占庭攻击、女巫攻击、51%攻击等共识攻击。因为其应用场景不同，所以已经出现了多种有特色的共识机制，如下所示。

PoS（Proof of Sake，权益证明）。原理是：节点获得区块奖励的概率与该节点持有的代币数量和时间成正比，在获取区块奖励后，该节点的代币持有时间清零，重新计算。但由于代币在初期分配时人为因素过高，容易导致后期贫富差距过大。

DPoS（Delegate Proof of Stake，股份授权证明）。原理是：所有节点投票选出 100 个（或其他数量）委托节点，区块完全由这 100 个委托节点按照一定算法生成，类似于美国的议会制。

Casper 投注。原理是：以太坊下一代的共识机制，每个参与共识的节点都要支付一定的押金，节点获取奖励的概率和押金成正比，如果有节点作恶则押金要被扣掉。

PBFT（Practical Byzantine Fault Tolerance，实用拜占庭容错）。原理是：与一般公有链的共识机制主要基于经济博弈原理不同，PBFT 主要基于异步网络环境下的状态机副本复制协议，本质上是由数学算法实现了共识，因此区块的确认不需要像公有链一样在若干区块之后才安全，可以实现出块即确认。

在共识机制中，用户激励主要实现区块链资产的发行和分配机制（例如以太坊定位以太币为平台运行的燃料，可以通过挖矿获得，每挖到一个区块固定奖励 5 个以太币，同时运行智能合约和发送交易都需要向矿工支付一定的以太币）。

**4. 业务应用层**

基于区块链技术，可以构建种类极其丰富的业务应用，如游戏、网上购物、视频、旅游咨询、火车票、汽车、直播媒体、时尚信息网上社区、知识产权、音乐、房地产照片、保险保单、贵重金属（如黄金）、股权、票据、域名、商标、数字资产、虚拟货币，等等；甚至可以构建以脚本为基础的智能合约，该合约赋予账本可编程的特性，通过虚拟机的方式运行代码，实现智能合约的功能，如以太坊虚拟机（EVM）。同时，这一层通过在智能合约上添加能够与用户交互的前台界面，形成去中心化的应用（DAPP）。

## 3.2.2 区块链的分类

**1. 根据网络范围分类**

根据网络范围，区块链可以划分为公有链、私有链和联盟链。

1）公有链

所谓公有就是指完全对外开放，任何人都可以任意使用，没有权限的设定，也没有身份认证，不但可以任意参与使用，而且产生的所有数据都可以任意查看，完全公开透明。比特币就是一个公有链网络系统，大家在使用比特币系统的时候，只需要下载相应的软件客户端，就可以执行创建钱包地址、转账交易、挖矿等操作，这些功能都可以自由使用。公有链系统完全没有第三方管理，依靠的就是一组事先约定的规则，这些规则要确保每个参与者在不信任的网络环境中能够发起可靠的交易事务。通常来说，凡是需要公众参与，需要最大限度地保证数据公开透明的系统，都适用于公有链，如数字货币系统、众筹系统、金融交易系统等。

这里需要注意，在公有链的环境中，节点的数量是不固定的，节点的在线与否也是无法控制的，甚至不能确定某节点是不是一个恶意节点。在 3.3 节讲解区块链的一般工作流程

的时候，将提出一个问题，在这种情况下，如何知道数据是被大多数节点写入确认的呢？实际上，在公有链环境下，这个问题没有很好的解决方案，目前最合适的做法就是通过不断地相互同步，最终网络中由大多数节点同步的区块数据所形成的链就是被承认的主链，这也称为最终一致性。

2）私有链

私有链是与公有链相对的一个概念，所谓私有就是指不对外开放，仅仅在组织内部使用的系统，如企业的票据管理、账务审计、供应链管理等，或者一些政务管理系统。私有链在使用过程中，通常是有注册要求的，即需要提交身份认证，而且具备一套权限管理体系。读者可能会有疑问，比特币、以太坊等系统虽然都是公有链系统，但如果将这些系统搭建在一个不与外网连接的局域网中，不就成了私有链了吗？从网络传播范围来看，可以这样说，因为只要这个网络一直与外网隔离，就只能是自己在使用，只不过由于使用的系统本身并没有任何身份认证机制以及权限设置，因此从技术角度来说，这种网络只能算是使用公有链系统的客户端搭建的私有测试网络。例如，以太坊就可以用来搭建私有链环境，通常这种情况可以用来测试公有链系统，当然也适用于企业应用。

在私有链环境中，节点数量和节点的状态通常是可控的，因此一般不需要通过竞争的方式来筛选区块数据的打包者，可以采用更加节能、环保的方式，如在3.2.1节中提到的PoS、DPoS、PBFT等。

3）联盟链

联盟链的网络范围介于公有链和私有链之间，通常使用在有多个成员角色的环境中，如银行之间的支付结算、企业之间的物流等，这些场景往往都是由不同权限的成员参与的。与私有链一样，联盟链系统一般也具有身份认证和权限设置，而且节点的数量往往也是确定的，对于企业或者机构之间的事务处理很适用。联盟链并不一定完全管控，有些数据是可以对外公开的，就可以部分开放，如政务系统。

由于联盟链一般用在明确的机构之间，因此与私有链一样，节点的数量和状态也是可控的，并且通常也采用更加节能、环保的共识机制。

**2. 根据部署环境分类**

1）主链

所谓主链，也就是部署在生产环境的真正的区块链系统，软件在正式发布前会经过很多内部的测试版本，用于发现可能存在的bug，并且用来内部演示以便于查看效果，直到最后才会发布正式版。主链也可以说是由正式版客户端组成的区块链网络，只有主链才是会被真正推广使用的，各项功能的设计也都是相对最完善的。另外，有些时候区块链系统会由于种种原因产生分叉，如挖矿的时候临时产生的小分叉等，此时将最长的那条原始的链称为主链。

2）测试链

测试链很好理解，就是开发者为了方便大家学习使用而提供的用于测试的区块链网络，如比特币测试链、以太坊测试链等。当然，并不是说只有区块链开发者才能提供测试链，用户也可以自行搭建测试链。测试链中的功能设计与生产环境中的主链是可以有差别的，例如主链中使用工作量证明算法进行挖矿，在测试链中可以更换算法以便更方便地进行测试。

**3. 根据对接类型分类**

1）单链

能够单独运行的区块链系统都可以称为“单链”，如比特币主链、测试链，以太坊主链、测试链，莱特币的主链、测试链，超级账本项目中的 Fabric 搭建的联盟链等，这些区块链系统拥有完备的组件模块，自成一个体系。大家要注意，有些软件系统，如基于以太坊的众筹系统或金融担保系统等，这些只能算是智能合约应用，不能算是一个独立的区块链系统，应用程序的运行需要独立的区块链系统的支撑。

2）侧链

侧链是一种区块链系统的跨链技术，这个概念主要是由比特币侧链发起的。随着技术发展，除了比特币，出现了越来越多的区块链系统，每一种系统都有自己的优势特点，那么如何将不同的链结合起来，打通信息孤岛，彼此互补呢？侧链就是实现这个目标的一项技术。

以比特币为例，比特币系统主要是用来实现数字加密货币的，且业务逻辑也已固化，因此并不适用于实现其他的功能，如金融智能合约、小额快速支付等。然而比特币是目前使用规模最大的一个公有区块链系统，在可靠性、去中心化保证等方面具有相当大的优势，那么如何利用比特币网络的优势来运行其他区块链系统呢？可以考虑在现有的比特币区块链之上，建立一个新的区块链系统，新的系统可以具备很多比特币没有的功能，如私密交易、快速支付、智能合约、签名覆盖金额等，并且能够与比特币的主区块链进行互通。简单来说，侧链是以锚定[①]比特币为基础的新型区块链。锚定比特币的侧链目前有 ConsenSys 的 BTCRelay、Rootstock 和 BlockStream 的元素链等。大家要注意，侧链本身就是一个区块链系统，并且侧链并不一定要以比特币为参照链。侧链是一个通用的技术概念，例如以太坊可以作为其他链的参照链，也可以本身作为侧链与其他的链去锚定。实际上，抛开链、网络这些概念，侧链就是在不同的软件之间互相提供接口，增强软件之间的功能互补，侧链的示意图如图 3-7 所示。

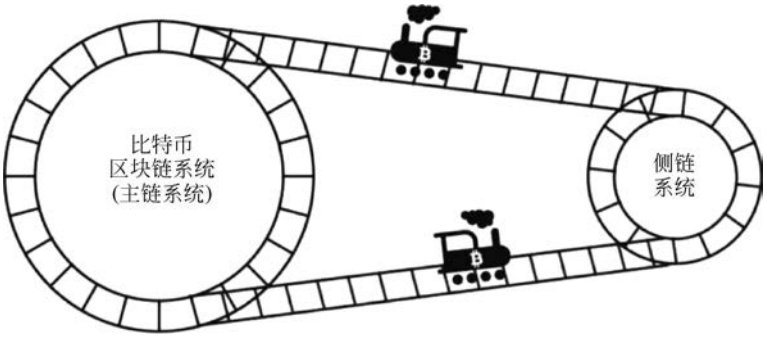



图 3-7　侧链示意图

通过这个简单的示意图可以看到，区块链系统与侧链系统本身都是一个独立的链系统，两者之间可以按照一定的协议进行数据互动，通过这种方式，侧链能起到扩展主链功能的作

① 锚定(anchoring)是指人们倾向于把对未来的估计和采用过的估计联系起来，同时易受他人建议的影响。当人们对某件事的好坏进行估测的时候，其实并不存在绝对意义上的好与坏，一切都是相对的，关键在于如何定位基点。基点定位就像一只锚，如果它确定了，评价体系也就确定了，好坏也就评定出来了。

用，很多在主链中不方便实现的功能可以在侧链中实现，而侧链再通过与主链的数据交互增强自身的可靠性。

3）互联链

如今我们的生活可以说几乎离不开互联网，仅仅互通互联，带来的能量已经如此巨大。

区块链也是这样，目前各种区块链系统不断涌现，有的只是实现了数字货币，有的实现了智能合约，还有的实现了金融交易平台；有些是公有链，有些是联盟链；等等。这么多链，五彩缤纷，功能各异，脑洞大开，不断涌现更新颖的应用。那么，这些链系统如果能够彼此互联会发生什么样的化学反应呢？从最初的数字货币到未来可能实现的区块链可编程社会，它们不单单会改变生活服务方式，还会促进社会治理结构的变革，如果说每一条链都是一根神经的话，一旦互联起来，就像神经系统一般，将会给我们的社会发展带来更高层次的智能化。

另外，从技术角度来讲，区块链系统之间的互联可以彼此互补，每一类系统都会有优势和不足，彼此进行功能上的互补，甚至彼此进行验证，可以大大提高系统的可靠性及性能。

## 3.3 区块链的工作流程

区块链的工作流程主要包括以下几个环节。

（1）发送节点将新的数据记录向全网进行广播。

每个发送数据的节点均有区块链地址，地址是解决公钥过长的方案。下面以比特币为例，介绍比特币地址的生成过程。比特币是建立在密码学基础上的，先利用椭圆加密算法（ECC）来产生比特币的私钥和公钥，由私钥可以计算出公钥，公钥的值经过一系列数字签名运算可以得到比特币的地址。其步骤如下。

① 随机选取一个 32 字节的数，作为私钥。

② 使用椭圆曲线加密算法（ECC）计算私钥所对应的公钥。

③ 计算公钥的 SHA-256 Hash 值。

④ 取上一步结果，计算 RIPEMD-160 Hash 值。

⑤ 取上一步结果，前面加入地址版本号（如比特币主网版本号“0x00”）。

⑥ 取上一步结果，计算 SHA-256 Hash 值。

⑦ 取上一步结果，再次计算 SHA-256 Hash 值。

⑧ 取上一步结果的前 4 字节（8 位十六进制数）。

⑨ 把这 4 字节加在步骤⑤的结果后面，作为校验（这就是比特币地址的十六进制形式）。

⑩ 用 base58 表示法变换地址（这就是最常见的比特币地址形式），如 16UwLL9Risc3QfPqBUvKofHmBQ7wMtjvM。

（2）接收节点对收到的数据记录信息进行检验，如检验记录信息是否合法，通过检验后，数据记录将被纳入一个区块中。

区块中会记录区块生成时间段内的交易数据，区块主体实际上就是交易信息的合集。每一种区块链的结构设计可能不完全相同，但大体上分为区块头（header）和区块体（body）两部分。区块头用于链接到前面的块并且为区块链数据库提供完整性保证，区块体则包含了经过验证的、区块创建过程中发生的价值交换的所有记录。区块结构有以下两个非常重

要的特点。

① 一个区块上记录的交易是在上一个区块形成之后、该区块被创建之前发生的所有价值交换活动，这个特点保证了数据库的完整性。

② 在绝大多数情况下，一旦新区块完成后就被加入区块链的最后，此区块的数据记录就再也不能被改变或删除，这个特点保证了数据库的严谨性，即无法被篡改。

顾名思义，区块链就是区块以链的方式组合在一起，以这种方式形成的数据库称为区块链数据库。区块链是系统内所有节点共享的交易数据库，这些节点基于价值交换协议参与到区块链的网络中。

(3) 全网所有接收节点对区块执行共识算法(PoW、PoS 等，详细内容将在后文进行介绍)。

(4) 区块通过共识算法处理后被正式纳入区块链中存储，全网节点均表示接受该区块，而表示接受的方法就是将该区块的随机散列值视为最新的区块散列值，新区块的制造将以该区块链为基础进行延长。

## 3.4 区块链的核心技术

区块链保证数据安全、不可篡改及透明性的技术包括以下几方面。

### 3.4.1 区块＋链

关于如何建立一个严谨的数据库，区块链的办法是将数据库的结构进行创新，把数据分成不同的区块，每个区块通过特定的信息在逻辑上链接到上一区块的后面，前后顺序连接，呈现一套完整的数据，这也是“区块链”这个名字的来源。区块(block)的定义是：在区块链技术中，数据以电子记录的形式被永久存储，存放这些电子记录的文件就称为“区块”。区块是按时间顺序一个个先后生成的，每一个区块记录着它在被创建期间发生的所有价值交换活动，所有区块汇总起来构成一个记录合集。

那么区块链是如何工作的呢？由于每一个区块的块头都包含了前一个区块的交易信息压缩值，这就使从创世区块(第一个区块)直到当前区块连接在一起形成了一条长链。如果不知道前一个区块的交易信息压缩值，就没有办法生成当前区块，因此每个区块必定按时间顺序跟随在前一个区块之后。这种所有区块包含前一个区块引用的结构让现存的区块集合形成了一条数据长链。“区块＋链”结构提供了一个数据库的完整历史。从第一个区块开始，到最新产生的区块为止，区块链上存储了系统全部的历史数据。区块链提供了数据库内每一笔数据的查找功能。区块链上的每一条交易数据，都可以通过“区块链”的结构追本溯源，一笔笔进行验证。区块(完整历史)＋链(完全验证)＝时间戳。这是区块链数据库的最大创新点。区块链数据库让全网的记录者在每个区块中都盖上一个时间戳来记账，表示这个信息是在这个时间写入的，形成了一个不可篡改、不可伪造的数据库。

### 3.4.2 分布式系统

分布式系统的定义：一个硬件或软件组件分布在不同的网络计算机上，彼此之间仅仅通过网络消息传递进行通信和协调的系统。一个标准的分布式系统在没有任何特定业务逻

辑约束的情况下，都会有如下两个特征。

（1）分布性。

分布式系统中的多台计算机都会在空间上随意分布，同时，计算机的分布情况也会随时变动。

（2）对等性。

分布式系统中计算机没有主从之分，组成分布式系统的所有计算机节点都是对等的。副本（Replica）是分布式系统对数据和服务提供的一种冗余方式。在常见的分布式系统中，为了对外提供高可用的服务，我们往往会对数据和服务进行生成副本的处理，即数据副本和服务副本。

**1. 分布式与集中式的区别**

分布式架构在价格成本、自主研发、兼容性、伸缩扩展性方面有比较显著的优势，支持按需扩展。

在集中式架构下，为了应对更高的性能和更大的数据量，往往只能向上升级到更高配置的计算机，如升级更强的CPU、升级多核、升级内存、升级存储等，一般这种方式称为Scale Up，但单机的性能永远都有瓶颈，随着业务量的增长，只能通过Scale Out的方式来支持，即横向扩展出同样架构的服务器。

**2. CAP（最终一致性）**

分布式系统绕不开CAP理论，即Consistency（一致性）、Availability（可用性）、Partition tolerance（分区容错性），三者不可兼得，如表3-1所示。

一致性（C）：在分布式系统中的所有数据备份，在同一时刻是否有同样的值（等同于所有节点都是最新的数据）。

可用性（A）：在集群中一部分节点发生故障后，集群整体是否还能响应客户端的读写请求（对数据更新具备高可用性）。

分区容错性（P）：以实际效果而言，分区相当于对通信的时限要求。系统如果不能在时限内达成数据一致性，就意味着发生了分区的情况，必须就当前操作在C和A之间做出选择。

**表3-1　CAP理论**

| 选择 | 说　明 |
| --- | --- |
| CA | 放弃分区容错性，加强一致性和可用性。其实就是传统的单机数据库的选择 |
| AP | 放弃一致性（这里的一致性是强一致性），追求分区容错性和可用性。这是很多分布式系统设计时的选择，例如很多NoSQL系统就是如此 |
| CP | 放弃可用性，追求一致性和分区容错性。基本不会做这种选择，网络问题会直接使整个系统不可用 |

比特币采用P2P协议进行节点之间的数据传输，放弃了CAP中的Consistency，采用了A和P两个维度。因为放弃了Consistency这个属性，所以就产生了拜占庭将军问题，即这么多节点如何达成数据一致。拜占庭军队都是由一个个小分队组成，每个小分队都由一个将军负责，将军们通过号令兵传达一系列行动，但是当其中出现一些叛将故意破坏号令时该怎么办？

分布式存储系统和拜占庭将军问题一样，很难实现一致性，对于比特币开放式的、全球化部署的系统集群更是如此。所以比特币放弃了强一致性，没有中心节点，并实现了 P2P 通信，这样，整个集群中的服务器发生故障或离开，或者新的服务器加入集群，对整个集群都不会产生影响。

### 3.4.3 分布式账本

分布式账本指交易记账由分布在不同地方的多个节点共同完成，而且每个节点记录的是完整的账目，因此它们都可以参与监督交易合法性，同时也可以共同为其作证。与传统的分布式存储有所不同，区块链的分布式存储的独特性主要体现在以下两个方面。

一是区块链的每个节点都按照块链式结构存储完整的数据，而传统分布式存储一般是将数据按照一定的规则分成多份进行存储。

二是区块链的每个节点存储都是独立的、地位相等的，依靠共识机制保证存储的一致性，而传统分布式存储一般是通过中心节点向其他备份节点同步数据。没有任何一个节点可以单独记录账本数据，从而避免了单一记账人由于被控制或者被贿赂而记假账的可能性。由于记账节点足够多，理论上讲除非所有的节点都被破坏，否则账目就不会丢失，从而保证了账目数据的安全性。

### 3.4.4 开源的、去中心化的协议

有了“区块＋链"的数据之后，接下来就要考虑记录和存储的问题了。应该让谁来参与数据的记录，又应该把这些盖了时间戳的数据存储在哪里呢？在当今中心化的体系中，数据都集中记录并存储在中央计算机上。但是区块链结构设计精妙的地方就在于此，它并不赞同把数据记录存储在中心化的一台或几台计算机上，而是让每个参与数据交易的节点都记录并存储所有的数据。

① 关于如何让所有节点都能参与记录，区块链的办法是：构建一整套协议机制，让全网每个节点在参与记录的同时也来验证其他节点记录结果的正确性。只有当全网大部分节点（甚至所有节点）都同时认为这个记录正确时，或者所有参与记录的节点都比对结果、一致通过后，记录的真实性才能得到全网认可，记录数据才被允许写入区块中。

② 关于如何存储“区块链”这套严谨的数据库，区块链的办法是构建一个分布式架构的网络系统，让数据库中的所有数据都实时更新并存储在所有参与记录的网络节点中。这样即使部分节点损坏或被黑客攻击，也不会影响整个数据库的数据记录与信息更新。区块链根据系统确定的开源的、去中心化的协议，构建了一个分布式的架构体系，让价值交换的信息通过分布式传播发送给全网，通过分布式记账确定信息数据内容，盖上时间戳后生成区块数据，再通过分布式传播发送给各个节点，实现分布式存储。

从硬件的角度讲，区块链的背后是由大量的信息记录存储器（如计算机等）组成的网络，那么这一网络如何记录发生在网络中的所有价值交换活动呢？区块链设计者没有为专业的会计记录者预留一个特定的位置，而是希望通过自愿原则来建立一套人人都可以参与记录信息的分布式记账体系，从而将会计责任分散化，由整个网络的所有参与者来共同记录。区块链中每一笔新交易的传播都采用分布式的结构，根据 P2P 网络层协议，消息由单个节点直接发送给全网所有的其他节点。区块链技术让数据库中的所有数据均存储于系统所有的

计算机节点中并实时更新。完全去中心化的结构设置使数据能实时记录，并在每个参与数据存储的网络节点中更新，这极大地提高了数据库的安全性。

通过分布式记账、分布式传播、分布式存储这三大“分布”可以发现，没有人、没有组织，甚至没有哪个国家能整体控制这个系统。系统内的数据存储、交易验证、信息传输过程全部都是去中心化的。在没有中心的情况下，大规模的参与者达成共识，共同构建了区块链数据库。可以说，这是人类历史上第一次构建了一个真正意义上的去中心化系统。甚至可以说，区块链技术构建了一套永生不灭的系统——只要不是网络中的所有参与节点在同一时间集体崩溃，数据库系统就可以一直运转下去。现在已经有了一套严谨的数据库，也有了记录并存储这套数据库的可用协议，那么当将这套数据库运用于实际社会时，要解决的一个最核心的问题是：如何使这个严谨且完整存储的数据库变得可信赖，可以在互联网无实名的背景下成功防止诈骗。

### 3.4.5 加解密算法

加密算法分为可拟和不可逆两种。可逆加密算法又分为两大类：对称式和非对称式。

对称式加密的特点是：加密和解密使用同一个密钥，通常称为“Session Key”。

非对称式加密的特点是：加密和解密使用的不是同一个密钥，而是两个密钥，一个称为“公钥”，另一个称为“私钥”。它们两个必须配对使用，信息用其中一个密钥加密后，只有用另一个密钥才能解开，否则不能打开加密文件。这里的“公钥”是指可以对外公布的，“私钥”则只能由持有人本人知道。

常见加密算法的分类如表 3-2 所示。

**表 3-2 常见加密算法**

| 分类 | | 常见的加密算法 |
|---|---|---|
| 不可逆加密算法 | | SHA-256、SHA-512、MD5、HMAC、RIPE-MD、HAVAL、N-Hash、Tiger |
| 可逆加密算法 | 对称加密算法 | DES、3DES、DESX、Blowfish、IDEA、RC4、RC5、RC6、AES |
| | 非对称加密算法 | RSA、ECC、Diffie-Hellman、El Gamal、DSA |
| 常见的 Hash 算法 | | MD2、MD4、MD5、HAVAL、SHA、SHA-1、HMAC、HMAC-MD5、HMAC-SHA1 |

非对称加密算法 RSA 在加密和解密的过程中分别使用两个密码，两个密码具有非对称的特点。

① 加密时的密码（在区块链中称为“公钥”）是全网公开可见的，所有人都可以用自己的公钥来加密一段信息（信息的真实性）；

② 解密时的密码（在区块链中称为“私钥”）是只有信息拥有者才知道的，加密过的信息只有拥有相应私钥的人才能够解密（信息的安全性）。简单总结：区块链系统内，所有权验证机制的基础是非对称加密算法。常见的非对称加密算法包括 RSA、El Gamal、Diffie-Hellman、ECC（椭圆曲线加密算法）等。

可以看出，从信任的角度，区块链实际上是用数学方法解决信任问题的产物。过去，人们解决信任问题可能依靠熟人社会的“老乡”，或传统互联网中的交易平台“支付宝”。而区块链技术中，所有的规则事先都以算法程序的形式表述出来，人们完全不需要知道交易的对

方是“君子”还是“小人”，更不需要求助中心化的第三方机构来进行交易背书，而只需要信任数学算法就可以建立互信。区块链技术的背后，实质上是算法在为人们创造信用，达成共识背书。

区块链系统内所有权验证机制的基础是非对称加密算法。在区块链系统的交易中，非对称密钥的基本使用场景有两种：①公钥对交易信息加密，私钥对交易信息解密。私钥持有人解密后，可以使用收到的信息。②私钥对信息签名，公钥验证签名。通过公钥签名验证的信息，确认为私钥持有人发出。节点始终都将最长的区块链视为正确的链，并持续以此为基础验证和延长它。如果有两个节点同时广播不同版本的新区块，那么其他节点在接收到该区块的时间上将存在先后差异，它们会在率先收到的区块基础上进行工作，但也会保留另外一条链，以防后者变成长的链。该僵局的打破需要共识算法的进一步运行，如果其中的一条链被证实为较长的一条，那么在另一条分支链上工作的节点将转换阵营，开始在较长的链上工作。以上就是防止区块链分叉的过程。

### 3.4.6 P2P 网络

P2P 网络采用点对点网络传输（对等网络），没有中心服务器，网络中的每个用户端既是客户端，又是服务器端。区块链的点对点技术，简单来讲，就是用户之间可以直接进行转账和交易，不需要经过中间机构的确认和授权。BT 下载就是采用了 P2P 技术来让客户端之间进行数据传输，一来可以加快数据下载速度，二来可以减轻下载服务器的负担。

### 3.4.7 JSON-RPC

远程过程调用（RPC）是一个计算机通信协议。该协议允许运行于一台计算机的程序调用另一台计算机的子程序，而程序员无须额外地为这个交互编程。如果涉及的软件采用面向对象编程，那么远程过程调用也可称作远程调用或远程方法调用，如图 3-8 所示，如 Java RMI。

RPC 的主要功能目标是使构建分布式计算（应用）更容易，在提供强大的远程调用能力时不损失本地调用的语义简洁性。

RPC 分为以下两种。

- 同步调用：客户端等待调用执行完成并返回结果。
- 异步调用：客户端调用后不必等待执行结果返回，但依然可以通过回调通知等方式获取返回结果。若客户端不关心调用返回结果，则变成单向异步调用，单向调用不必返回结果。

客户端和服务器端通过 Socket 调用时需要进行序列化和反序列化，传输的数据是二进制形式。

JSON-RPC 是一个无状态且轻量级的远程过程调用传输协议，其传递内容主要通过 JSON 完成。相对于 REST 通过网址（如 GET /login）调用远程服务器，JSON-RPC 直接在内容中定义了要调用的函数名称（如 {“method”：“getUser”}）。

JSON-RPC 有两个版本，客户端使用的是 2.0 版本。请求中必须要有 jsonrpc、method、params、id 字段，如表 3-3 所示。响应中则必须有 jsonrpc、id 字段，处理成功时还必须有 result 字段，处理失败时必须有 error 字段。

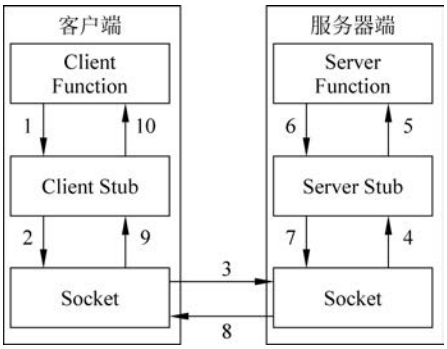



图 3-8　RPC 调用流程

**表 3-3　请求实例的字段说明**

| 请求字段 | 说　　明 |
|---|---|
| jsonrpc | 2.0 |
| method | 调用的方法名 |
| params | 调用的方法所需的参数 |
| id | 客户端的唯一标识 |

请求实例

```
{
    "jsonrpc": "2.0",
    "method": "subtract",
    "params": [56,28],
    "id": 1
}
```

响应：

```
{
    "jsonrpc": "2.0",
    "result": 19,
    "id": 1
}
```

gRPC 使用的是 HTTP 1.0 协议，可以通过 Protobuf 来定义接口。Protobuf 是一套类似 JSON 或者 XML 的数据传输格式和规范，用于在不同应用或进程之间进行通信。通信时所传输的信息通过 Protobuf 定义的 message 数据结构进行打包，然后编译成二进制的码流再进行传输或者存储。

Protobuf 序列化后体积很小，消息大小只有 XML 的 1/10～1/3，解析速度比 XML 快 20～100 倍，支持多种语言。

## 3.4.8　Merkle 树

Merkle 树(Merkle tree)又称哈希树，是一种典型的二叉树结构，由一个根节点、一组中间节点和一组叶子节点组成。Merkle 树最早由 Merkle Ralf 在 1980 年提出，曾广泛用于文件系统和 P2P 系统中。其主要特点如下：

- 底层的叶子节点包含存储数据或其 Hash 值；
- 非叶子节点（包括中间节点和根节点）的内容是其两个子节点内容的 Hash 值。

进一步地，Merkle 树可以推广到多叉树，此时非叶子节点的内容为其所有子节点的内容的 Hash 值。

Merkle 树逐层记录 Hash 值，让它具有一些独特的性质。例如，底层数据的任何变化都会传递到其父节点，沿着路径层层传递，直到树根。这意味着树根的值实际上代表了对底层所有数据的“数字摘要”。

目前，Merkle 树的典型应用场景包括以下 4 种。

**1. 证明某个集合中存在或不存在某个元素**

通过构建集合的 Merkle 树，并提供该元素各级兄弟节点中的哈希值，可以不暴露集合完整内容而证明某元素存在。

另外，对于能够进行排序的集合，可以将不存在元素的位置用空值代替，以此构建稀疏 Merkle 树（sparse Merkle tree）。该结构可以证明某个集合中不包括指定元素。

**2. 快速比较大量数据**

对每组数据排序后构建 Merkle 树结构。当两个 Merkle 树根相同时，则意味着所代表的两组数据必然相同。否则，必然不同。

由于 Hash 计算的过程可以十分快速，预处理可以在短时间内完成。利用 Merkle 树结构能带来巨大的比较性能优势。

**3. 快速定位修改**

以图 3-9 为例，基于数据 $D_0$、$D_1$、$D_2$、$D_3$ 构造 Merkle 树，如果 $D_1$ 中的数据被修改，会影响到 $N_1$、$N_4$ 和 Root。

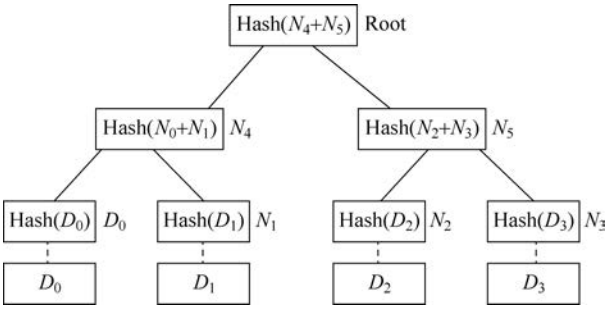



图 3-9 Merkle 树示例

因此，一旦发现某个节点（如 Root）的数值发生变化，沿着 Root→$N_4$→$N_1$，最多通过 $O(\lg N)$ 时间即可快速定位到实际发生改变的数据块 $D_1$。

**4. 零知识证明**

仍以图 3-9 为例，如何向他人证明拥有某个数据 $D_0$ 而不暴露其他信息。由挑战者提供随机数据 $D_1$、$D_2$ 和 $D_3$，或由证明人生成（需要加入特定信息，避免被人复用证明过程）。

比特币区块链的基础数据结构是 Merkle 树。Merkle 树是哈希指针形式的二叉树，每个节点包含其子节点的哈希值，一旦子节点的结构或内容发生变动，该节点的哈希值必然发生改变，因此，如果两个 Merkle 树顶层节点的哈希值相同，我们就认为两棵 Merkle 树是完

全一致的。

以太坊(Ethereum)所使用的 Merkle 树则更为复杂,称为“默克尔帕特里夏树”(Merkle Patricia Tree)。每个以太坊区块头不只是包括一棵 Merkle 树,而是为三种对象设计的三棵树:交易(Transaction)、收据(Receipts,本质上是显示每个交易影响的多块数据)、状态(State)。

### 3.4.9 共识算法

共识算法是区块链中节点保持区块数据一致、准确的基础,现有的主流共识算法包括 PoW、PoS、RCPA 等。以 PoW 为例,是指通过消耗节点算力形成新的区块,是节点利用自身的计算机硬件进行数学计算,实现区块链网络的交易确认并提高其安全性的过程。交易支持者(矿工)在计算机上运行比特币软件,通过不断计算、处理软件提供的复杂的密码学问题来保证交易的进行。作为对他们服务的奖励,矿工可以得到他们所确认的交易中包含的手续费,以及新创建的比特币。在后面章节将会结合实际案例详细介绍共识算法。

### 3.4.10 预言机

在计算机领域,Oracle 的含义就是预言机,区块链上的预言机就是 OracleChain。什么叫作 Oracle? 这个 Oracle 不是指世界上最大的数据库公司 Oracle(甲骨文)。而是天才数学家图灵在 20 世纪中叶提出的思想。现在的计算机都叫图灵机。以太坊的特别之处就是它是一个带有内置的、成熟的图灵完备语言的区块链。

图灵虚构了一种叫作预言机(OracleMachine,又称谕示机)的计算机。预言机具备图灵机的一切功能,并额外拥有一种能力:可以不通过计算而直接得到某些问题的答案,这个过程叫作 Oracle。如果用户要解决的问题是怎样计算都无法得到结果的,那么这个情况下图灵机就成了一堆废铁,只有 Oracle 才能解决。

区块链本身是封闭的,区块链预言机是区块链与外部世界交互的一种实现机制,它在区块链与外部世界间建立一种可信任的桥接机制,使得外部数据可以安全可靠地进入区块链。

受限于区块链的共识模型,智能合约只能调用内部合约,无法直接与外部系统进行交互。在智能合约中执行的逻辑不可以执行区块链之外的任何操作,外部数据进入智能合约的唯一方法是将其置入一个交易中,通过向系统发送一个新的交易来触发区块链状态的更新。

Oracle 适用于以下场景:

- 智能合约需要可信地访问 Web 数据;
- 智能合约通过调用 Open API 使用互联网服务;
- 智能合约需要与外部系统交互;
- 智能合约依赖公共现实事件,如天气、赛事信息、航班信息等。

外部数据源服务在区块链上部署了区块链预言机合约,提供异步查询互联网数据接口供用户合约使用。正常情况下,用户合约调用预言机合约发起查询请求后,预言机合约在 1～3 个区块内就能得到外部数据源服务取回的数据,然后回调用户合约,传入数据,如图 3-10 所示。

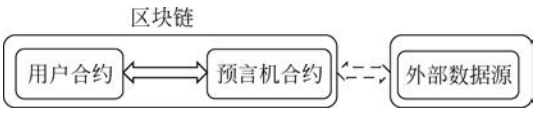



图 3-10　用户合约与预言机合约

共识机制

## 3.5　共识机制

区块链有以下三个基本特征：

- 区块链是一个分布式数据库（系统）；
- 区块链采用密码学保证数据不被篡改；
- 区块链采用共识算法对于新增的数据达成共识。

以上三点可以简单地从哲学角度理解，如图 3-11 所示。

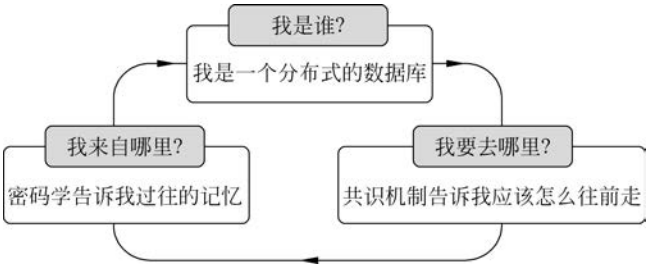



图 3-11　区块链基本特征

区块链技术的伟大之处就是它的共识机制，在去中心化的思想基础上解决了节点间互相信任的问题。区块链拥有众多节点并达到一种平衡状态是因为共识机制。尽管密码学占据了区块链的半壁江山，但是共识机制才是让区块链系统不断运行下去的关键。而要深入理解区块链的共识机制，就避不开一个问题——拜占庭问题。

### 3.5.1　共识机制的起源

现代共识机制的基础于 1962 年被提出。RAND 公司的工程师 Paul Baran 在论文《论分布式通信网络》中提出了加密签名的概念。这些数字化签名不久后就成为了系统对要修改数据或文档的用户进行验证的方法。二十年后，三名学者 Leslie Lamport、Robert Shostak、和 Marshall Pease 发表了一篇关于去中心化系统可靠性问题的论文《拜占庭将军问题》。该论文提出了一个思维实验——拜占庭将军问题。

**1. 拜占庭将军问题**

拜占庭将军问题是容错计算中的一个老问题，由莱斯特·兰伯特等在 1982 年提出。拜占庭帝国为公元 395 年至 1453 年的东罗马帝国，拜占庭城邦拥有巨大的财富，令它的 10 个邻邦垂涎已久，但是拜占庭城邦高墙耸立，固若金汤，任何单个城邦的入侵行动都会失败，入侵者的军队也会被歼灭，使得其自身反而容易遭到其他 9 个城邦的入侵。这 10 个城邦之间也互相觊觎对方的财富并经常爆发战争。

拜占庭的防御能力如此之强，非大多数人一起不能攻破，而且只要其中一个城邦背叛盟

军，那么所有进攻军队都会被歼灭，并随后被其他邻邦所劫掠。因此这是一个互不信任的各邻邦构成的分布式网络，每一方都小心行事，因为稍有不慎就会给自己带来灾难。为了夺取拜占庭的巨额财富，这些邻邦分散在拜占庭的周围，依靠士兵传递消息来协商进攻目的及进攻时间，这些邻邦将军想要攻占拜占庭，但面临的一个困扰，邻邦将军不确定他们之中是否有叛徒，叛徒是否擅自变更了进攻意向或者进攻时间，如图 3-12 所示。

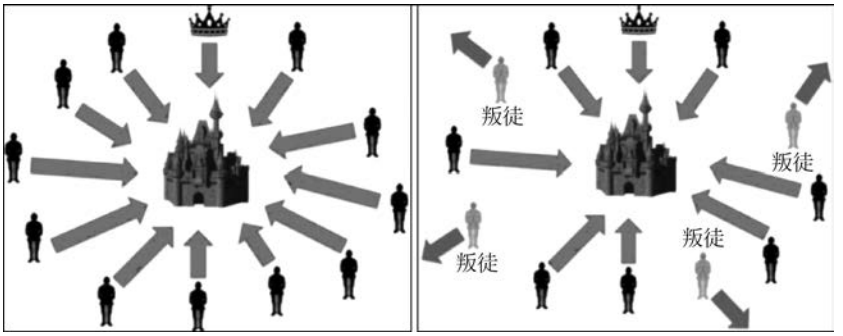



图 3-12　拜占庭将军问题示意

在这种状态下，将军们能否找到一种分布式协议来进行远程协商、达成共识，进而赢取拜占庭城邦的财富呢？

在拜占庭将军问题模型中，对于将军们有两个公认的假设，如图 3-13 所示。

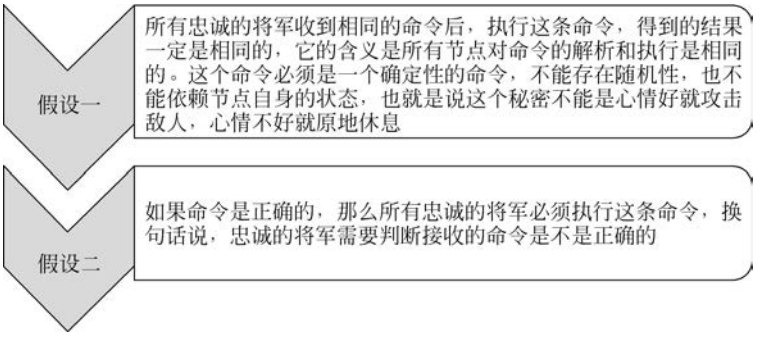



图 3-13　拜占庭将军问题的两条假设

对于将军们的通信，在拜占庭将军问题中也是有默认假设的：点对点通信是没有问题的。也就是说，我们假设 A 将军要给 B 将军发一条命令“M”，那么派出去的传令兵一定会准确地把命令“M”传给 B 将军。

但问题在于，如果每个城邦都向其他 9 个城邦派出一名信使，那么就是这 10 个城邦每个都派出了 9 名信使，也就是说在任何一个时刻有总计 90 次的信息传输，并且每个城市分别收到 9 条信息，可能每一条都写着不同的进攻时间。除此以外，信息传输过程中，如果叛徒想要破坏原有的约定时间，就会自己修改相关信息，然后发给其他城邦以混淆视听，这样的结果是，部分城邦收到错误信息后，会遵循一个或者多个城邦已经修改过的攻击时间，从而违背发起人的本意。这样一来，遵循错误信息的城邦（包含叛徒）将重新广播超过一条信息的信息链，整个信息链会随着他们发送的错误信息，迅速变成不可信的信息，变成一个相互矛盾的纠结体。

针对这个问题，人们主要提出了两种解决方法，一个是口头协议算法，另一个是书面协议算法。

口头协议算法的核心思想是：要求每个被发送的信息都能被正确投递，信息接收者明确知道信息发送者的身份，并且知道信息中是否缺少内容。采用口头协议算法，若叛徒数少于1/3，则拜占庭将军问题可以很容易解决。但是口头协议算法存在着明显的缺点，那就是信息不能溯源。

为解决这个问题，提出了书面协议算法。该算法要求签名不可伪造，一旦被篡改即可发现，同时任何人都可以验证签名的可靠性。

就算是书面协议算法也不能完全解决拜占庭将军问题，因为该算法没有考虑信息传输延迟、签名机制难以实现的问题，且签名消息记录的保存也难以摆脱中心化机构。

这个问题该如何解决？中本聪给出了一个比较好的答案：不是让所有人都有资格发信息，而是给发信息设置了一个条件“工作量”。将军们同时做一道计算题，谁先正确完成，谁才能获得给其他小国发信息的资格。而其他小国在收到信息后，必须采用加密技术签字盖戳，以确认身份。然后再继续做题，做对的人再继续发消息……这种对先后顺序达成共识的算法，开创了共识机制的先河。

**2. 区块链共识机制解决方案**

中本聪所创建的比特币，通过对这个系统做出一个简单的改变解决了这个问题。他为发送信息加入成本，这降低了信息传递的速率，并加入了一个随机元素，以保证在某个时刻只有一个城邦可以进行广播。

中本聪加入的成本是“工作量证明”——挖矿，并且工作量证明是基于一个随机哈希算法进行计算的。这个哈希算法的作用是根据输入进行计算，得到一个64位的由随机数字和字母组成的字符串。

在比特币的世界中，输入数据包括到当前时间点的整个总账。尽管单个哈希值用现在的计算机基本可以及时地计算出来，但是比特币系统接受的工作量证明是无数个64位哈希值中唯一的哈希值，而且这个哈希值的前13个字符均为0，这样一个哈希值是极其罕见、不可能被破解的，并且在当前的算力水平下，需要花费整个比特币网络总算力约10分钟时间才能找到一个。

在一台网络计算机随机地找到一个有效哈希值之前，数十亿个无效值会被计算出来，计算哈希值就需要花费大量时间，增加了发送信息的时间间隔，降低了信息传递速率，而这就是使得整个系统可用的“工作量证明”机制。

而那台发现下一个有效哈希值的计算机，将所有之前的信息汇总，附上它自己的身份信息，以及它的签名等，向网络中的其他计算机进行广播。只要网络中的其他计算机接收到并验证通过这个有效的哈希值和附着在上面的签名信息，它们就会停止当下的计算，使用新的信息更新它们的总账副本，然后把更新后的总账作为哈希算法的输入，开始再次计算哈希值。

哈希计算竞赛从一个新的开始点重新开始……如此这般，网络持续同步着，所有网络上的计算机都使用着同一版本的总账，与此同时，每一次成功找到有效哈希值并进行区块链更新的间隔大概是10分钟，在那10分钟以内，网络上的参与者发送信息并完成交易，并且因为网络上的每一个计算机都使用同一个总账，所有的这些交易和信息都会进入每一份遍布

全网的总账副本。当区块链更新并在全网同步之后，在之前 10 分钟内进入区块链的所有交易也被更新并同步，因此分散的交易记录是在所有的参与者之间进行对账和同步的。

最后在用户向网络输入一笔交易的时候，使用内嵌在比特币客户端的标准公钥加密工具来加密，同时用他的私钥及接收者的公钥为这笔交易签名，这对应于拜占庭将军问题中用来签名和验证消息时使用的“印章”，如图 3-14 所示。因此，哈希计算速率的限制，加上公钥加密，使得一个不可信网络变成一个可信的网络，所有参与者可以在某些事情上达成一致（如攻击时间、一系列的交易域名记录、政治投票系统，或者其他任何需要分布式协议的情况）。

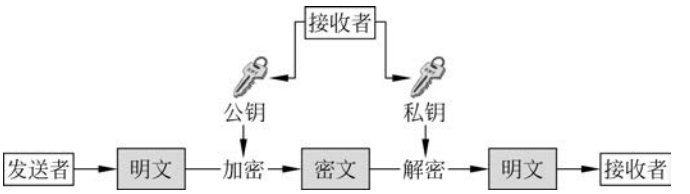



图 3-14　加解密过程

将比特币的共识机制引入拜占庭将军问题，就形成了这样一种情况，城邦 A 向其他 9 个城邦发送进攻相关信息时，直接将相关信息及其发送的时间附加在通过哈希算法加密的信息中，并且加上独属于自己的数字签名，再传递给其他城邦。等其他城邦中相应的计算机已经收到并验证通过这个有效哈希值和附加在上面的签名信息，就会停止他们的计算，而使用新的信息更新他们的总的进攻信息副本，然后把更新的信息区块链作为哈希算法的输入，再发给其他城邦。其他城邦接收消息后，重复此流程，直至所有城邦都收到消息。如此这般，网络持续同步，网络上的所有计算机都使用着同一版本的总账。

如果叛徒想要修改进攻信息来误导其他城邦，其他城邦的计算机会立刻识别到异常信息，同步的虚假信息将不被认可，而依旧会同步大部分共同的信息，这样叛徒就失败了，他无法破坏 10 个城邦当中的大多数节点，也就是至少 6 个节点，这样信息的一致性就得到了保证，完美解决了拜占庭将军问题。

这就是区块链共识机制如此关键的原因：它为一个算法上的难题提供了解决方案，通过不断同步各个节点的信息，使得各分布式节点之间达到一种平衡，保证了绝大多数节点的一致性，即达成了共识。

## 3.5.2　共识机制的概念

由于加密货币多数采用去中心化的区块链设计，节点是各处分散且平行的，所以必须设计一套制度来维护系统的运作顺序与公平性，统一区块链的版本，并奖励提供资源来维护区块链的使用者，以及惩罚恶意的危害者。这样的制度必须依赖某种方式来证明，是谁获得了一个区块的打包权（或称记账权），此人可以得到打包这个区块的奖励；又是谁企图实施危害，此人就会得到一定的惩罚，这就是共识机制。

区块链的共识机制通常包含了两个方面，如图 3-15 所示。

我们经常说的“共识机制”，多数情况下同时包含了共识算法和共识规则，少数情况下单指其中一方，这也是大家在认识上经常存在的误区。

由于点对点网络中存在较高的网络延迟，各个节点所观察到的事务先后顺序不可能完


图 3-15　区块链的共识机制

全一致，因此区块链系统需要设计一种机制，对在差不多时间内发生的事务的先后顺序进行共识。这种对一个时间窗口内的事务的先后顺序达成共识的算法称为“共识机制”。这里解释的只是共识算法，也就是节点依照共识规则达成共识的计算机算法。

而共识规则则是指每个区块链里面都有自己精心设计好的规则性协议，这些协议通过共识算法来保证其得以可靠地执行。如通常所说的比特币的挖矿，就是比特币记账的共识规则，其专业术语为 PoW，即工作量证明。比特币的 PoW 共识规则通过 SHA（Secure Hash Algorithm）系列安全哈希算法之一——SHA-256 来得以可靠地执行。

## 3.5.3　共识机制的作用

区块链的核心是参与者之间的共识（参见图 3-16 方框中的步骤）。共识机制之所以关键，是因为在没有中央机构的情况下，参与者必须就规则及其应用方法达成一致，并同意使用这些规则来接受和记录拟定的交易。

如图 3-16 所示，交易一经创建和发布，即署有交易发起人的签名，签名表示获得授权以支付金钱、订立合同或传输与交易相关的数据指标。交易在签名后即生效并包含执行需要的所有信息。

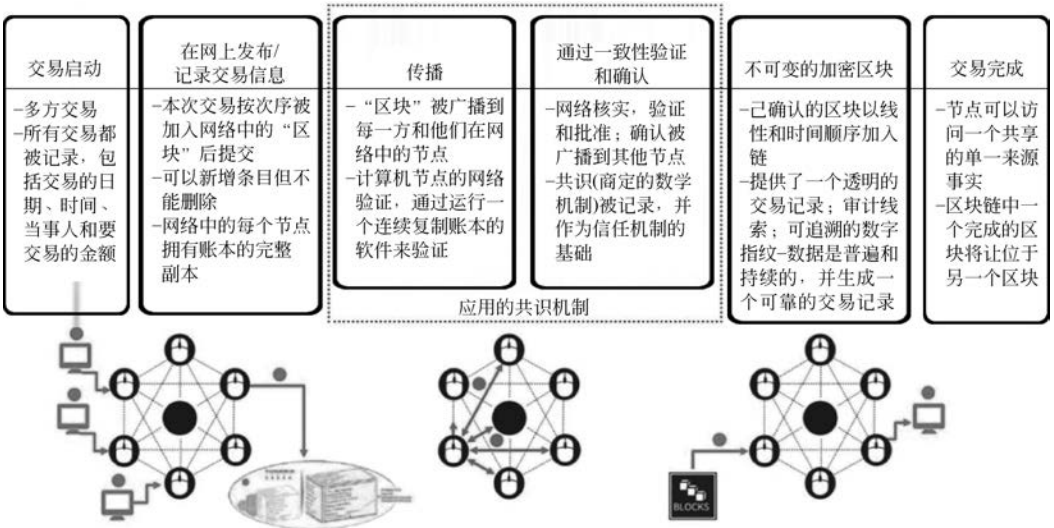



图 3-16　共识机制过程

一旦交易被验证并纳入区块，该交易便会在整个网络中传播。在整个网络达成共识且

网络中的其他节点接受新区块后，该区块就并入区块链中。经区块链记录和足够多的节点确认后，该交易将成为公共账本的永久组成部分，区块链网络中的所有节点会视其为有效。

### 3.5.4　共识机制的原理

共识机制用来决定区块链网络中的记账节点，并对交易信息进行确认和一致性同步。早期的比特币区块链采用高度依赖节点算力的工作量证明机制来保证比特币网络分布式记账的一致性。随着区块链技术的发展和各种竞争币的相继涌现，图 3-17 展示了当前主流的共识机制及其有代表 性的项目。

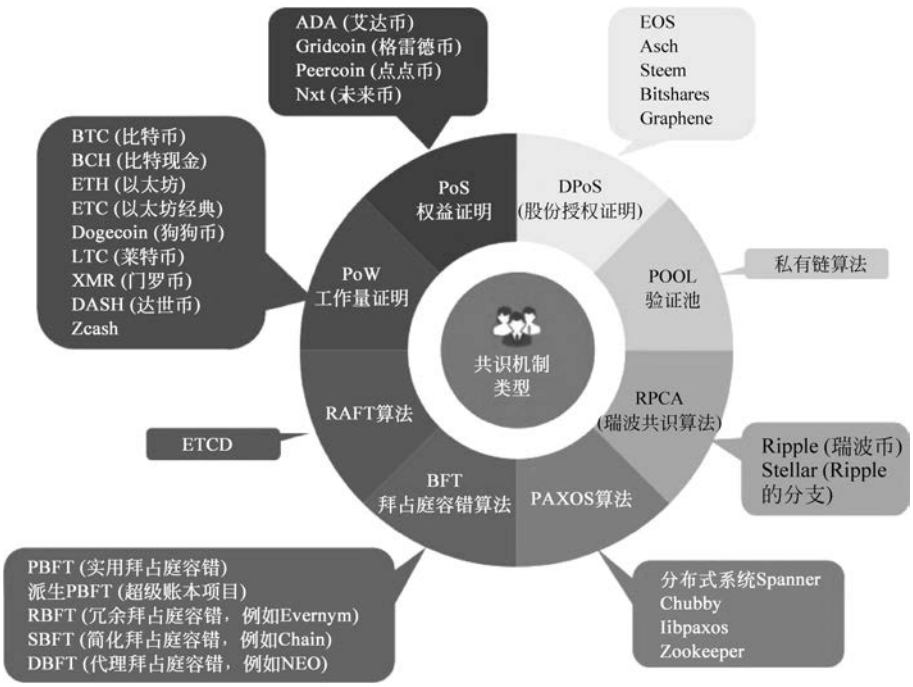



图 3-17　不同共识机制的代表性项目

因技术发展日新月异，图 3-17 所示的是截至 2021 年 12 月主流共识机制算法的代表性项目的概览。本书的目的并不是完整展示当前所有共识机制，而是仅描述那些当前作为创建区块链的技术选项而被热切讨论和探索的机制。本书并不进行学术讨论，所以对于共识机制的具体技术细节并没有深入讲解，仅仅是进行概略性的介绍。以下介绍的共识机制中大部分在区块链和分布式账本产生前已被应用。

**1. PoW**

工作量证明(Proof of Work，PoW，如图 3-18 所示)的主要特点是将解决计算困难问题所需要的计算代价作为新加入区块的凭证和获得的激励收益。


图 3-18　PoW 共识机制

中本聪在其比特币奠基性论文中设计了 PoW 共识机制，其核心思想是通过引入分布式节点的算力竞争来保证数据一致性和共识的安全性。比特币系统中，各节点（即矿工）基于各自的计算机算力相互竞争来共同解决一个求解复杂度很高但容易验证的 SHA-256 数学难题（即挖矿），最快解决该难题的节点将获得区块记账权和系统自动生成的比特币奖励。

该数学难题可表述为：根据当前难度值，通过搜索求解一个合适的随机数（Nonce），使得区块头中各元数据的双 SHA-256 哈希值小于或等于目标哈希值，比特币系统通过灵活调整随机数搜索的难度值，将区块的平均生成时间控制在 10min 左右。

一般来说，PoW 共识的随机数搜索过程如下。

第一步：搜集当前时间段的全网未确认交易，并增加一个用于发行新比特币奖励的 Coinbase 交易，形成当前区块体的交易集合。

第二步：计算区块体交易集合的 Merkle 根，记入区块头，并填写区块头的其他元数据，其中随机数 Nonce 置零。

第三步：将随机数 Nonce 加 1，计算当前区块头的双 SHA-256 哈希值，如果它小于或等于目标哈希值，则成功搜索到合适的随机数并获得该区块的记账权；否则继续执行第三步，直到任一节点搜索到合适的随机数为止。

第四步：如果一定时间内未搜索成功，则更新时间戳和未确认交易集合，重新计算 Merkle 根后继续搜索。

符合要求的区块头哈希值通常由多个前导的 0 构成，目标哈希值越小，区块头哈希值的前导的 0 越多，成功找到合适的随机数并挖出新区块的难度越大。由此可见，比特币区块链系统的安全性和不可篡改性是由 PoW 共识机制的强大算力所保证的，任何对于区块数据的攻击或篡改都必须重新计算该区块及其后所有区块的 SHA-256 难题，并且计算速度必须使得伪造链的长度超过主链，这种攻击难度导致其成本将远超其收益。正是这种机制保证了区块链的数据一致性和不可篡改性，但是同时也带来了资源浪费，甚至由于超大矿池的出现而失去了去中心化的优势。

举个简单的例子。如果算法得到的哈希值总是 0～10 000，而算法要求得到的（哈希值）小于 1，一台计算机如果每秒能够计算一次，那么平均每计算一万次，就有一次的值可能小于 1；或者反过来说，每次计算有万分之一的机会值小于 1。如果有一万个节点同时在计算，那么每秒都有可能有一个节点得到符合条件的结果，得到符合条件结果的节点就出块成功。而每秒得到结果的计算机都可能不一样，这样就获得了足够随机的结果。

图 3-19　PoS 共识机制

**2. PoS**

权益证明（Proof of Stake，PoS，如图 3-19 所示）共识机制的主要特点以权益证明代替工作量证明，由具有最高权益的节点实现新区块加入并获得激励收益。

由于工作量证明机制资源消耗大且计算资源趋于中心化，权益证明机制受到广泛关注。如果把工作量证明中的计算资源视为对区块进行投票的份额，那么权益证明就是将与系统相关的权益作为投票的份额。可以合理假设，权益的所有者更乐于维护系统的一致性和安全性。

假设网络同步性较高，系统以轮为单位运行，在每一轮的开始，节点验证自己是否可通过权益证明被选为代表，只有代表可以提出新的区块。代表在收到的最长的有效区块链后

提出新的待定区块，并将自己生成的新的区块链广播出去，等待确认。下一轮开始时，重新选取代表，对上一轮的结果进行确认。诚实的代表会在最长的有效区块链后面继续工作。如此循环，共同维护区块链。

权益证明机制在一定程度上解决了工作量证明机制能耗大的问题，缩短了区块的产生时间和确认时间，提高了系统效率。权益证明每一轮产生多个通过验证的代表，也就是产生多个区块，在网络同步性较差的情况下，系统极易产生分叉，影响一致性。若恶意节点成为代表，就会通过控制网络通信，形成网络分区，向不同网络分区发送不同的待定区块，就会造成网络分叉，从而可进行二次支付攻击，严重影响系统的安全性。恶意节点也可以对诚实代表进行贿赂，破坏一致性。权益证明的关键在于如何选择恰当的权益，构造相应的验证算法，以保证系统的一致性和公平性。不当的权益会影响系统公平性。例如，PPCoin 采用币龄作为权益的一个因子，若节点在进入系统初期就保持一部分小额交易不用于支付，则其币龄足够大，该节点更容易被选为代表，影响系统公平性。

在 PoS 出现后，一些针对其某个缺点进行改进的新协议相继诞生，它们称作 PoS 的衍生协议，如 PoSV 和 PoA。

PoSV 针对 PoS 中币龄是时间的线性函数这一问题进行改进，致力于消除数字货币持有者的屯币现象。PoSV 意为权益和活动频率证明，是瑞迪币(Reddcoin)目前使用的共识机制，瑞迪币在前期使用 PoW 进行币的分发，后期使用 PoSV 维护网络的长期安全。PoSV 将 PoS 中币龄和时间的线性函数修改为指数级衰减函数，即币龄的增长率随时间逐渐减小，最后趋于零，因此新币的币龄比老币增长得更快，直到达到上限阈值，这样在一定程度上缓解了数字货币持有者屯币的现象。

PoA 意为行动证明，也是 PoS 的一种改进方案。它的本质是通过奖励参与度高的货币持有者而不是惩罚消极参与者来维护系统安全。PoA 将 PoW 和 PoS 结合，主要思想是将 PoW 挖矿生成币的一部分以抽奖的方式分发给所有活跃节点，而节点拥有的股权与抽奖券的数量即抽中概率成正比。

PoS 共识机制的实施过程始终是一个复杂的人性博弈过程。以太坊的 Casper FFG 版 PoS 机制将于以太坊第三阶段 Metropolis 的第二部分 Constantinople(君士坦丁堡)中投入使用，这是一种融合了改进的 PoS 共识和 PoW 共识的混合共识。以太坊 Casper FFG 版本的记账人选择和出块时间都由 PoW 共识完成，PoS 共识在每隔 100 个区块处设置检查点，为交易提供最终确认，这也是 PoW-PoS 混合共识机制优于 PoW 共识机制的原因。

**3. DPoS**

为了进一步加快交易速度，同时解决 PoS 中节点离线也能累积币龄的安全问题，Daniel Larimer 于 2014 年 4 月提出 DPoS。

股份授权证明(Delegated Proof of Stake，DPoS)是 PoS 的一个演化版本，首先通过 PoS 选出代表，进而从代表中选出区块生成者并获得收益。

DPoS 共识机制的基本思路类似于“董事会决策”，即系统中每个股东节点可以将其持有的股份权益作为选票授予一个代表，获得票数最多且愿意成为代表的前 101 个节点将进入“董事会”，按照既定的时间表轮流对交易进行打包结算并且签署(即生产)一个新区块。每个区块被签署之前必须先验证前一个区块已经被受信任的代表节点所签署，“董事会”的授权代表节点可以从每笔交易的手续费中获得收入，同时，要成为授权代表节点必须缴纳一

定的保证金，其金额相当于生产一个区块的收入的 100 倍。授权代表节点必须对其他股东节点负责，如果它错过签署相对应的区块，股东将会收回选票，从而将该节点“投出”董事会。因此授权代表节点通常必须保证 99%以上的在线时间以实现盈利目标。

显然，与 PoW 共识机制必须信任最高算力节点和 PoS 共识机制必须信任最高权益节点不同的是，DPoS 共识机制中每个节点都能够自主决定其信任的授权节点，且由这些节点轮流记账和生成新区块，因而大幅减少了参与验证和记账的节点数量，可以实现快速共识验证。

采用 DPoS 机制的最典型的是 EOS。EOS 系统中共有 21 个超级节点和 100 个备用节点，超级节点和备用节点由 EOS 权益持有者选举产生。区块的生产以 21 个区块为一轮。在每轮开始的时候会选出 21 名区块生产者，前 20 名区块生产者由系统根据网络持币用户的投票数自动生成，最后一名区块生产者根据其得票数按概率生成。所选出的生产者会根据从区块时间戳导出的伪随机数轮流生产区块。

EOS 结合了 DPoS 和 BFT（拜占庭容错算法）的特性，在区块生成后即进入不可逆状态，因而具有良好的稳定性。DPoS 作为 PoS 的变形，通过缩小选举节点的数量以减少网络压力，是一种典型的分治策略：将所有节点分为领导者与跟随者，只有领导者之间达成共识后才会通知跟随者。

DPoS 为了实现更高的效率而设置的代理人制度，背离了区块链世界里人人可参与的基本精神，这也是 EOS 一直受质疑的地方。

**4. RPCA**

瑞波共识算法（Ripple Protocol Consensus Algorithm，RPCA）是一种数据正确性优先的网络交易同步算法，它是基于特殊节点（也称“网关”节点）列表达成的共识。在这种共识机制下，必须首先确定若干个初始特殊节点，如果要新接入一个节点，必须获得至少 51%的初始节点的确认，并且只能由被确认的节点产生区块。

瑞波共识机制的工作原理如下。

第一步：验证节点接收并存储待验证交易，将其存储在本地。本轮共识过程中新到的交易需要等待，在下次共识时再确认。

第二步：由活跃的信任节点发送提议。信任节点列表是验证池的一个子集，其信任节点来源于验证池，参与共识过程的信任节点须处于活跃状态，验证节点与信任节点都需要处于活跃状态，长期不活跃的节点将被从信任节点列表中删除。信任节点根据自身掌握的交易双方额度、交易历史等信息对交易做出判断，并加入提议中进行发送。

第三步：本验证节点检查收到的提议是否来自信任节点列表中的合法信任节点，如果是，则存储；如果不是，则丢弃。

第四步：验证节点根据提议确定认可的交易列表，假定信任节点列表中活跃的信任节点个数为 $M$（如 5），本轮中交易认可阈值为 $N$（百分比，如 50%），则每个超过 $M\times N$ 个信任节点认可的交易都将被本验证节点认可，本验证节点生成认可交易列表。系统为本验证节点设置一个计时器，如果计时器时间已到，本信任节点需要发送自己的认可交易列表。

第五步：账本共识达成。本验证节点仍然在接收来自信任节点列表中信任节点的提议，并持续更新认可交易列表。验证节点认可列表的生成并不代表最终账本的形成以及共识的达成，账本共识只有在每笔交易都获得至少超过一定阈值（如 80%）的信任节点列表的

认可才能达成，达成时交易验证结束，否则继续上述过程。

第六步：共识过程结束，形成最新的账本，将剩余的待确认交易以及新交易纳入待确认交易列表，开始新一轮共识过程。

瑞波共识机制使得一组节点能够基于特殊节点列表达成共识。初始特殊节点列表就像一个俱乐部，要接纳一个新成员，必须由一定比例的该俱乐部会员投票通过。因此，它区别于其他共识机制的主要因素是有一定的“中心化”。

**5. PAXOS**

PAXOS 是一种基于消息传递且具有高度容错性的一致性算法，它将节点分为 3 种类型，如图 3-20 所示。

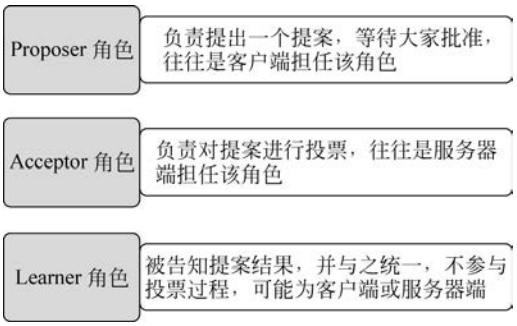



图 3-20　PAXOS 的三种节点类型

基本共识过程是先由 Proposer 提出提案，争取大多数 Acceptor 的支持，如果超过一半的 Acceptor 支持，则发送结案结果给所有人进行确认。如果 Proposer 在此过程中出现故障，可以通过超时机制来解决。在极为凑巧（概率很小）的情况下，每次新的一轮提案的 Proposer 都恰好故障，系统则永远无法达成一致。

第一阶段，Proposer 向网络内超过半数的 Acceptor 发送 Prepare 消息，Acceptor 正常情况下回复 Promise 消息。

第二阶段，在有足够多 Acceptor 回复 Promise 消息时，Proposer 发送 Accept 消息，正常情况下 Acceptor 回复 Accepted 消息。PAXOS 中 3 类角色的主要交互过程发生在 Proposer 和 Acceptor 之间，如图 3-21 所示，其中 1、2、3、4 代表顺序。

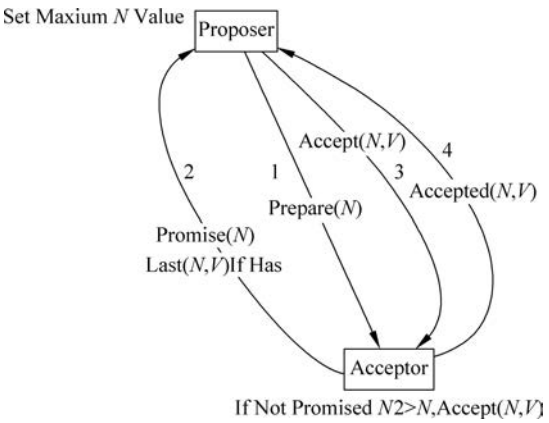



图 3-21　Proposer 和 Acceptor 交互过程

PAXOS 协议用于微信 PaxosStore 中，微信每分钟调用 PAXOS 协议过程的次数达 10 亿量级。PAXOS 协议用于分布式系统中的典型例子就是 Zookeeper，它是第一个被证明的共识算法，其原理基于两阶段提交并进行了扩展。

**6. BFT**

拜占庭将军问题提出后，很多的算法被提出用于解决这个问题，这类算法统称拜占庭容错（Byzantine Fault Tolerance，BFT）算法。BFT 从 20 世纪 80 年代开始研究，目前已经是一个研究得比较透彻的理论，具体实现都已经有现成的算法。

**7. PBFT**

最常用的 BFT 共识机制是实用拜占庭容错（Practical Byzantine Fault Tolerance，PBFT）算法。该算法是 Miguel Castro 和 Barbara Liskov 在 1999 年提出的，解决了原始拜占庭容错算法效率不高的问题，将算法复杂度由节点数的指数级降低到节点数的平方级，使得拜占庭容错算法在实际系统中应用变得可行。

PBFT 算法分为 5 个阶段：请求（Request）、预准备（Pre-prepare）、准备（Prepare）、确认（Commit）、回复（Reply）。其共识过程如图 3-22 所示。

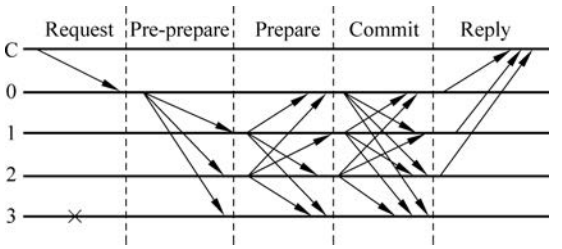



图 3-22　PBFT 共识过程

图 3-22 中 C 为发送请求端，0、1、2、3 为服务端，3 为宕机的服务端，具体步骤如下。

第一步，请求阶段。从全网节点中选举出一个主节点（Leader），这里是 0，新区块由主节点负责生成，请求端 C 发送请求到主节点。

第二步，预准备阶段。每个节点把客户端发来的交易向全网广播，主节点 0 将从网络收集到的、需要放在新区块内的多个交易排序后存入列表，并将该列表向全网广播，扩散至节点 1、2、3。

第三步，准备阶段。每个节点接收到交易列表后，根据排序模拟执行这些交易。所有交易执行完后，基于交易结果计算新区块的哈希摘要，并向全网广播，1→023，2→013，3 因为宕机无法广播。

第四步，确认阶段。如果一个节点收到的 $2f$（$f$ 为可容忍的拜占庭节点数）个其他节点发来的摘要都和自己相同，就向全网广播一条 Commit 消息。

第五步，回复阶段。如果一个节点收到 $2f+1$ 条 Commit 消息，即可提交新区块及其交易到本地的区块链和状态数据库。

这种机制下有一个叫作视图的概念。在一个视图里，一个是主节点，其余的都叫作备份节点。主节点负责将来自客户端的请求排好序，然后按序发送给备份节点。但是主节点可能是有问题的，它可能会给不同的请求加上相同的序号，或者不分配序号，或者让相邻的序号不连续。备份节点有责任来主动检查这些序号 的合法性，并能通过超时机制检测到主节

点是否已经宕机。当出现这些异常情况时，这些备份节点就会触发视图更换协议，选举出新的主节点。

**8. DBFT**

考虑到 BFT 算法存在的扩容性问题，NEO 采用了一种代理拜占庭容错算法（Delegated Byzantine Fault Tolerant，DBFT）。它与 EOS 的 DPoS 共识机制一样，由权益持有者投票选举产生代理记账人，由代理人验证和生成区块，借此大幅度降低共识过程中的节点数量，解决了 BFT 算法固有的扩容性问题。

为了便于在区块链开放系统中应用，NEO 的 DBFT 将 PBFT 中的 C/S（客户机/服务器）架构的请求响应模式，改进为适合 P2P 网络的对等节点模式，并将静态的共识参与节点改进为可动态进入、退出的动态共识参与节点，使其适用于区块链的开放节点环境。

DBFT 算法中，参与记账的是超级节点，普通节点可以看到共识过程，并同步账本信息，但不参与记账。共 $n$ 个超级节点，分为 1 个议长和 $n-1$ 个议员，议长由议员轮流当选。每次记账时，先由议长发起区块提案（拟记账的区块内容），一旦有至少 $(2n+1)/3$ 个记账节点（议长和议员）同意了这个提案，那么这个提案就成为最终发布的区块，并且该区块是不可逆的，即所有其中的交易都是百分之百确认的，区块不会分叉。

**9. RAFT**

RAFT 算法是对 PAXOS 算法的一种简单实现。其核心思想是如果多个数据库的初始状态一致，只要之后进行的操作一致，就能保证之后的数据一致。因此 RAFT 使用日志方式进行同步，并且将服务器分为 3 种角色：Leader、Follower、Candidate，角色之间可以互相转换，如图 3-23 所示。

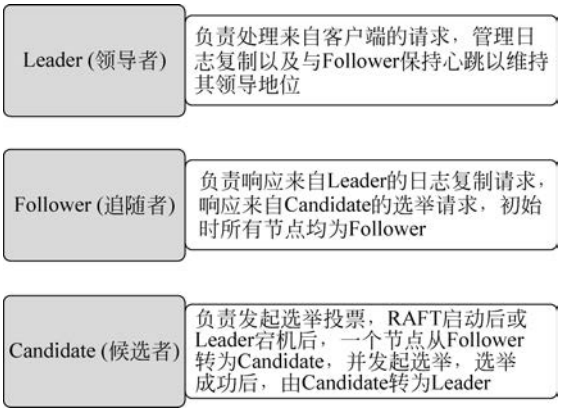



图 3-23 RAFT 的三种角色[①]

RAFT 算法主要包含以下两个步骤。

第一步，选举 Leader。Follower 自增当前任期，转换为 Candidate，对自己投票，并发起投票申请，等待下面 3 种情形之一发生：一是获得超过半数服务器的投票，赢得选举，成为 Leader；二是另一台服务器赢得选举，并接收到对应的心跳，成为 Follower；三是选举超

① 图中的心跳机制是指节点定时发送一个自定义的数据结构（心跳包），让其他节点知道它在线，以确保连接有效性的机制。

时，没有任何一台服务器赢得选举，自增当前任期，重新发起选举。

第二步，Leader 生成日志，并与 Follower 进行心跳同步。Leader 接受客户端请求，更新日志，向所有 Follower 发送心跳信息，并同步日志。所有 Follower 都有选举超时机制，如果在设定时间之内没有收到 Leader 的心跳信息，则认为 Leader 失效，重新选举 Leader。在 RAFT 算法中，日志的流向只有从 Leader 到 Follower，并且 Leader 不能覆盖日志，日志不是最新版本者不能成为 Candidate。

**10. POOL**

POOL（验证池）共识基于传统的分布式一致性技术，并加上了数据验证机制，这种共识机制的主要特点是基于当前成熟的分布式一致性算法（PBFT、PAXOS、RAFT 等）来实现秒级共识验证，是目前在私有链和联盟链中大范围使用的共识机制，此处不再赘述。

除了常见的上述几类共识机制，在区块链的实际应用过程中也衍生出了 PoW＋PoS、行动证明（Proof of activity）等多个变种机制。还存在着五花八门的依据业务逻辑自定义的共识机制，如小蚁的“中性记账”、类似瑞波共识的 Stellar 共识机制、Factom 等众多以“侧链”形式存在的共识机制，这些共识机制各有优劣势。比特币的 PoW 共识机制依靠其先发优势已经形成成熟的挖矿产业链，支持者众多；而 PoS 和 DPoS 等新兴机制则更为安全、环保和高效，从而使得共识机制的选择问题成为区块链系统研究者最不易达成共识的问题。

**11. 混合共识算法及其他**

1）Proof of Luck

美国加利福尼亚大学伯克利分校的研究人员基于 TEEs（Trust Execution Environments，可执行信任环境）算法设计了一种新型的共识机制，运行在支持 SGX 的 CPU 上，来抵御挖矿以及对能源的消耗。该算法包含两个函数：PoLRound 和 PoLMine，其中所有参与者都运行这两个函数，得到以同一区块为祖先的不同区块。PoLMine 会选择一个介于 0 到 1 之间的随机数字（运气），最大数字意味着运气最好，将所持有的区块作为区块链中的下一个区块。由于在 SGX 环境中发生随机数选择，所以不能伪造它。研究人员在论文中使用的是 Intel 公司的 TEE——SGX，基于 Intel 的硬件环境提出了对应的共识协议——POET（Proof Of Elapsed Time，所用时间证明）算法。据 Intel 公司自己的实验数据，该算法可以拓展到数千节点。但是问题是该算法依赖底层 CPU，需要把信任 交给 Intel 公司，这与区块链去中心化的思想相悖。

2）PoDD

在 USENIX 技术研讨会上，一种新的加密数字货币 DDoSCoin 由科罗拉多大学和密歇根大学的研究人员提出，旨在奖励用户使用他们的计算机参与 DDoS 攻击。在 DDoSCoin 中，矿工工作量的计算是依据建立的 TLS 连接，这导致其只适用于已启用 TLS 加密的网站。他们使用的共识机制就是 PoDD（Proof of DDoS），参与 DDoS 攻击会给矿工带来数字货币奖励，矿工便可将货币转换成比特币或其他法定货币，这可以认为是 PoW 的另一种形式。恶意的“DDoS 身份验证”操作是让矿工连接到 Web 服务器。将响应作为链接证据。在现代版本的 TLS 中，服务器在握手过程中签署客户端提供的参数，并在连接的密钥交换中使用服务器提供的值，这允许客户端向其他人证明其已经与服务器通信。此外，服务器返回的签名值对于客户端来说是不可预知且随机分布的。

3）PoB

PoB(Proof of burn)即烧毁证明，和开发比特币的过程也很相似。但 PoB 是通过将货币转移到不可逆转的地址上以销毁货币，而不是投资到计算硬件上。这种转移也叫作“燃烧”，货币被转到了某个很难找到的地址。

创建新区块的人必须为创建新的货币支付费用。这些费用将按照预先规定的比例或者算法转换为新的货币。合约币(XCP)就是通过烧毁比特币而产生的。

## 3.6 习 题

1. 以比特币为例，说明区块链技术的特征。

2. 简述对称加密与非对称加密的异同。

3. 区块链的本质是一个去中心化的分布式账本。那么，所谓的中心化指什么？去中心化的分布式账本与中心化的账户有什么区别？你能讲出几个生活中去中心化和中心化的不同场景吗？

4. 共识算法是区块链技术的重要组成部分，现实世界中共识就是一群人对一件或者多件事情达成一致的看法或协议。那么在区块链的世界中，共识是什么？说出几种共识算法，并说明它们之间的区别。