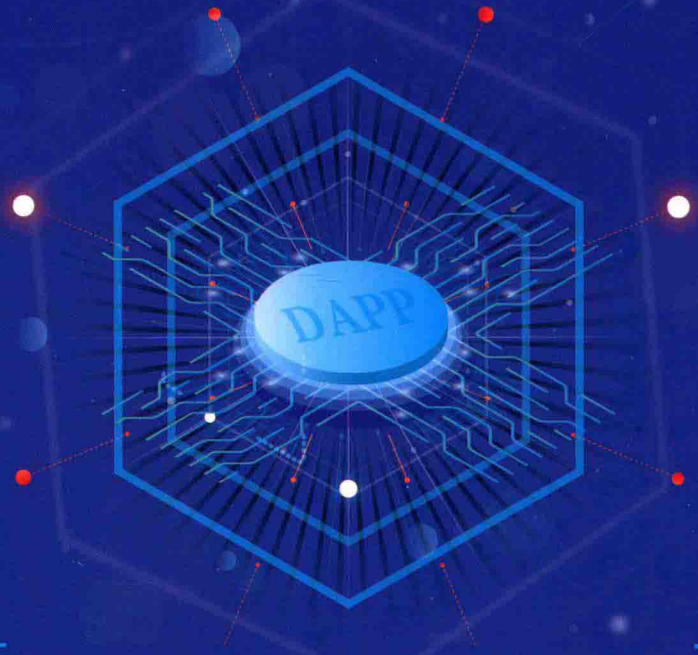
深度剖析公链底层技术和虚拟机底层技术
围绕股票、彩票、投票、众筹等行业创新应用

Broadview®
www.broadview.com.cn

区块链DAPP

开发入门、代码实现、场景应用

李万胜◎著



从底层代码到上层应用场景和业务介绍，全面阐述智能合约的价值
完备的系统架构讲解，从公链到DAPP全流程讲解
深入讲解Solidity语言从编译到部署的机制和原理
深入剖析智能合约编程语言的语法细节及注意事项
以真实可落地的案例，全面展示智能合约所涉及的技术

中国工信出版集团
電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

区块链DAPP

开发入门、代码实现、场景应用

李万胜◎著

電子工業出版社
Publishing House of Electronics Industry
北京•BEIJING

内 容 简 介

本书以 DAPP 的原理和具体实现为主线索，通过对这些知识的讲解，使读者对 DAPP 系统开发有更全面的认识，同时把区块链公链的相关知识融合进来，使得读者不仅能够设计和实现 DAPP 系统，还能大体理解 DAPP 的相关知识。

本书分为 7 章。第 1 章简单介绍了区块链的基础知识，通过一个简单的智能合约的代码，让读者对区块链有更直观的认识；第 2 章讲解了各类集成开发环境的搭建，尤其是 MetaMask 插件，用户需要通过此插件连接以太坊主网后才能使用 DAPP 系统；第 3 章讲解了 Solidity 编译及部署到公链之后的数据表达和函数调用方式；第 4 章介绍了 ABI 接口的技术细节；第 5 章和第 6 章以案例的方式介绍了 DAPP 开发的细节；第 7 章讲解了 DAPP 潜在的风险。

希望本书能为广大系统开发者和投资者提供一些帮助。

图书在版编目（CIP）数据

区块链 DAPP 开发入门、代码实现、场景应用 / 李万胜著. —北京：电子工业出版社，2019.9

ISBN 978-7-121-37375-6

Ⅰ. ①区… Ⅱ. ①李… Ⅲ. ①电子商务－支付方式－程序设计 Ⅳ. ①F713.361.3 ②TP311.1

中国版本图书馆 CIP 数据核字（2019）第 192445 号

责任编辑：董　英　　特约编辑：田学清
印　　刷：三河市鑫金马印装有限公司
装　　订：三河市鑫金马印装有限公司
出版发行：电子工业出版社
　　北京市海淀区万寿路 173 信箱　　邮编：100036
开　　本：787×980　1/16　　印张：15　　字数：341 千字
版　　次：2019 年 9 月第 1 版
印　　次：2019 年 9 月第 1 次印刷
定　　价：79.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888，88258888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式：（010）51260888-819，faq@phei.com.cn。

前　言

最早接触区块链是在 2009 年，当时我正在华为的一个网络安全部门从事研发工作，我在诺基亚工作的同学徐明亮，让我为他的美国同事开发一款比特币钱包应用，我也有幸接触比特币的源代码，并尝试编写区块链的相关应用。

当时我对区块链的理解尚处在代码层面，作为一个没有太多社会经验的程序员，我认为比特币、区块链仅仅作为点对点网络，通过密码学的方式保证数据传输和访问的安全，从性能提高和技术创新的角度看，它并没有先进性可言。当时我的工作内容就涉及网络与安全（密码学属于安全范畴），这个知识背景使得我对比特币的技术没有产生特别浓厚的兴趣。况且其代码完全开源，对我来讲，这样的项目一点竞争优势都没有。

到 2017 年，一种新型区块链公链项目爆发，它就是以太坊。当时国内外的区块链项目如火如荼，基于以太坊发行自己项目的 token 一时风头无二，很多投资机构也积极地参与其中。此时我已经参与过多个创业项目，对商业基本逻辑有了初步的认识，对于创业初期面临的问题也有了较深刻的理解。此时有投资人推荐我从事区块链方向的创业，我也因此重新对比特币和以太坊的设计进行了认真学习。当我研读完以太坊的白皮书之后，深深地被其技术特点及可以解决的问题范畴所震撼，我会在本书第 5 章对其涉及的 ICO、token 和 DAO 做详细的讲解。不同于比特币仅仅通过加密对一个数值进行去中心化的安全处理，以太坊已经可以与现实世界的商业逻辑进行紧密的融合。

从应用场景来说，比特币仅限于金融领域的应用，而以太坊已经超出了这个范畴，其第一个爆款 DAPP——ICO 在近几年全球的创业圈中掀起了巨大的变革浪潮，虽然它还伴随着信息不对称、技术不对称等问题，有很多欺诈项目也打着区块链的旗号作恶，但是如果用这样的技术解决真正有价值的问题，那么其影响也是巨大的。本书第 6 章以彩票为例，从代码到业务模式进行了详细的讲解，通过区块链 DAPP 解决彩票问题有很大的经济价值和社会价值。

从技术角度来说，比特币是一个很难编程的架构，如果需要利用区块链加密安全、公开透明、去中心化的特点，在一般情况下，项目方都需要升级整套比特币源代码，以满足自己项目的技术需求和业务需求。很多分叉币都是以这样的思路升级比特币的某些特性，然后重新建设自己的生态的。而对于以太坊来讲，智能合约虚拟机的引入使得以太坊成为一个可编程的区块链网络，为了达到同

样的目的，项目方只需编写 DAPP 即可，无须重新搭建网络和建设生态。从这个角度来讲，比特币网络类似于诺基亚的功能手机，而以太坊类似于 iPhone 的智能手机。前者不可编程，新特性需要新型号的手机；后者有 App Store，用户可以通过下载 App 满足自定义的手机需求。

本书的另一个目的是希望通过代码及架构设计的讲解，把区块链真正的价值和正确的使用方式传递给更多的读者，使得各类诈骗项目不再能轻易地欺骗更多的群众。在认真研究了以太坊之后，我欣然接受了投资人的建议，成功融资并开始了区块链公链方面的研发。目前我从事的项目主要是对以太坊进行进一步升级。虽然以太坊使得区块链可以编程，但是其操作对象仍然是抽象的数字货币，无法与现实世界中有价值的资产进行关联。我希望通过区块链对带宽流量、分布式存储、CPU 算力共享等 IT 资源进行 token 激励和记账管理，这些资源在现实世界中是有价值的，且其价值是可衡量的。

在项目开发过程中，我整理了很多底层的区块链架构的技术文档，包括一些智能合约相关的技术文挡，因为这些资料相对较少，并且区块链热潮使得很多人对此类知识相对渴望，因此这些技术文档的读者逐渐多了起来。2018 年，出版社的朋友希望我整理一些资料，编写成书，让更多的人可以读到这些技术文档。我觉得这对于区块链开发者和投资人来说都是一件非常有意义的事情，因此我欣然答应，这是本书的写作背景。

因为当前区块链尚处在发展初期，包括以太坊的编程语言 Solidity，其版本仍然未达到 release 版本，因此很多知识可能会发生变化，希望读者及时跟进官方的资料。欢迎读者对本书表述不合理的地方提出建议或意见，我一定虚心接受。本书的代码会放在 GitHub 上：https://github.com/9992800/Dapp-on-Ethereum。我仅以本书抛砖引玉，希望更多的科技人才加入区块链行业中来，一起促进区块链的良性发展。同时希望本书对投资人有所帮助，提高其分辨骗局项目的能力。

再次感谢出版社对我的信任，以及朋友和亲人对我事业的支持。

李万胜

目 录

第 1 章 智能合约概述 1

1.1 区块链基础知识 1

1.1.1 交易 1

1.1.2 区块 4

1.1.3 链 4

1.1.4 挖矿 5

1.1.5 共识算法 6

1.1.6 分叉 7

1.1.7 攻击 8

1.2 以太坊智能合约 9

1.2.1 以太坊 9

1.2.2 EVM 12

1.2.3 智能合约 13

1.2.4 DAPP 14

1.3 简单的智能合约 15

1.3.1 示例 1 16

1.3.2 示例 2 17

1.4 小结 18

第 2 章 开发环境搭建 20

2.1 Remix 的使用 20

2.1.1 编程界面 20

2.1.2 运行环境 22

2.1.3 其他设置 24

2.2 Ethereum Wallet 的安装与使用 26
2.2.1 安装 26
2.2.2 部署合约 28
2.2.3 调试 32
2.2.4 Ethereum Wallet 小结 38
2.3 Ganache + Truffle 的安装与使用 38
2.3.1 Ganache 38
2.3.2 Truffle 40
2.3.3 安装总结 45
2.4 MetaMask 的配置与使用 45
2.5 小结 47
2.6 课后练习 47

第 3 章 Solidity 编程语法 48

3.1 Solidity 前导知识 48
3.2 智能合约的基本构成 50
3.2.1 状态变量 50
3.2.2 函数 50
3.2.3 函数修饰器 50
3.2.4 事件 51
3.2.5 结构体 51
3.2.6 枚举类型 52
3.3 Solidity 数据类型 52
3.3.1 值类型 53
3.3.2 引用类型 58
3.3.3 左值操作类型 65
3.3.4 类型转换原则 66
3.4 全局变量和单位 68
3.4.1 单位 68
3.4.2 全局变量和函数 69
3.5 控制逻辑与表达式 71
3.5.1 控制语句 71
3.5.2 函数调用 71

3.5.3 通过 new 关键字创建合约 73
3.5.4 赋值 74
3.5.5 作用范围与声明 75
3.5.6 异常处理 76
3.6 智能合约 78
3.6.1 创建合约 78
3.6.2 可见范围和 getter 80
3.6.3 函数修饰符 83
3.6.4 状态常量 85
3.6.5 函数 86
3.6.6 events 91
3.6.7 继承 93
3.6.8 抽象合约 97
3.6.9 接口 98
3.6.10 库 99
3.6.11 using for 102
3.7 Solidity 汇编语言 103
3.7.1 内嵌式汇编 103
3.7.2 独立汇编 109
3.8 小结 110
3.9 课后练习 110

第 4 章 ABI（应用程序二进制接口） 111

4.1 接口调用的基本原理 111
4.2 函数调用与参数封装 115
4.2.1 数据封装的基础知识 115
4.2.2 函数选择 116
4.2.3 参数封装 117
4.2.4 封装示例 118
4.3 ABI 接口 JSON 描述 129
4.4 小结 132
4.5 课后练习 133

第 5 章　ICO、token 和 DAO 134

5.1　ICO 是第一个爆款 DAPP 134
5.1.1　被扭曲了的 ICO 135
5.1.2　传统众筹与以太坊众筹 135
5.1.3　众筹的变种 ICO 143
5.2　token 144
5.2.1　token 的基本元素 144
5.2.2　改进 token 150
5.3　ICO 158
5.4　DAO 162
5.4.1　创建 DAO 合约 162
5.4.2　使用 DAO 的方式 170
5.4.3　模拟股票监管的 DAO 合约 174
5.5　ICO 认知误区与防骗指南 175
5.6　小结 176
5.7　课后练习 177

第 6 章　DAPP 完整实战 178

6.1　投票 178
6.1.1　Truffle 默认案例讲解 178
6.1.2　修改为投票系统 187
6.1.3　实战小结 193
6.2　誓言上链 193
6.3　区块链彩票 199
6.3.1　业务架构 199
6.3.2　业务代码实现 201
6.4　小结 219
6.5　课后练习 219

第 7 章　智能合约安全与公链技术简介 220

7.1　合约溢出攻击实例 220
7.2　智能合约安全漏洞与建议 224

7.2.1 合约漏洞 224
7.2.2 安全建议 226
7.3 常见的公链安全问题 227
7.3.1 双花攻击 227
7.3.2 女巫攻击 228
7.3.3 日食攻击 229
7.3.4 DDoS 攻击 230
7.4 小结 230

第1章 智能合约概述

智能合约是运行在区块链公链上的一种代码，该代码由 Solidity 编写，并通过区块链的智能合约虚拟机来执行，以达到对区块链编程的目标。为了更好地理解智能合约的运行环境，本章将讲解区块链公链的基本概念。可以将区块链公链理解为操作系统，Solidity 是编写该操作系统应用程序的编程语言，智能合约虚拟机则是编程语言编译之后的代码运行环境。本章除介绍区块链公链的基础知识外，还会讲解智能合约与区块链公链的交互方式，以及智能合约虚拟机的系统架构。

本章主要涉及的知识点有：

- 区块链公链的常用术语和基本知识。
- 智能合约虚拟机的系统架构。
- DAPP 与区块链交互的方式。
- 智能合约编程语言 Solidity 的基本语法结构。

1.1 区块链基础知识

本节将简单介绍区块链公链的基本概念和技术架构，从交易的产生到“区块”这个名称的产生，从挖矿到区块链攻击，从公链基础功能到基于公链操作系统的可编程环境。本节将从不同的角度，对公链的基本技术术语进行简明阐述，对公链底层技术感兴趣的读者可以根据本文的介绍，检索相关资料，比特币和以太坊在官网都有详细的技术文档和资料，感兴趣的读者可以自行检索，本节仅讲解与 DAPP 开发相关的技术点。

1.1.1 交易

区块链通常被理解为超级账本，账户与账户之间可以通过交易来完成转账，只是这种转账方式与传统的银行转账有很大的不同。

（1）这是一个完全去中心化的金融系统，区块链账户不需要使用者到银行机构或者其他部门申请，因为整个系统中没有这样的中心化部门来管理账户信息，使用者只需要根据一种非对称加密算法来生成一个密钥对，其公钥作为账户地址，也就是常说的区块链钱包地址，这个地址可以在网络中广播，允许网络中所有的账户获取和使用，如图 1.1 所示。

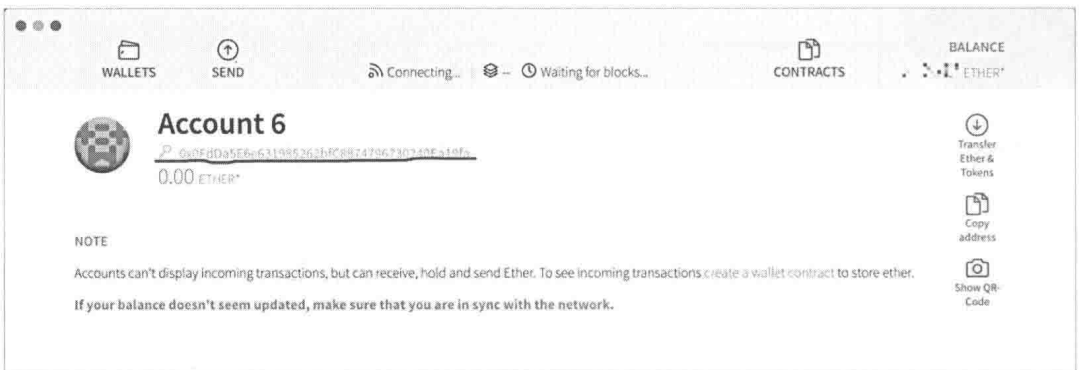


图 1.1　以太坊钱包的地址

该账户地址是公开的。私钥作为转账交易的签名和密码，由使用者私人保管，每次转账时，需要通过私钥签名来证明交易的合法性。在现实使用中，私钥往往非常长，而且非常复杂，因此区块链钱包会将私钥进行对称加密，通过使用者输入人类能够理解的密码作为私钥加密的密钥，将区块链密钥以密文的形式保存在磁盘空间里。但是如果忘记解密密钥的密码，用户就失去了对账户的操作权限，也就是说即使账户有余额并且可以查看，也无法进行转账和使用，这就是社交媒体经常报道的丢失比特币的情况。

（2）这是一个完全无中心的账本系统，其交易方式与传统的交易系统有着本质的差别，传统的银行系统中，当 A 转账 100 元给 B 时，银行从数据库中将 A 的账户扣除 100 元，同时在 B 的账户下增加 100 元。在这个过程中，数据库是在银行的严密保护下操作的，扣除与增加金额的操作需要很高的安全级别才能进行，并且查询余额用的数据库和真正写入的数据大多数情况下是分离的。尤其需要注意的是这个过程生成的交易记录至少有 2 条：1 条是从 A 的账户下扣除金额的记录，另 1 条是在 B 的账户下增加金额的记录。

而发生在区块链上的交易则完全不一样，在区块链上每一笔交易都是一条转账记录，如果该交易成功被整个区块链网络认可则转账成功，并且会将转账记录存储在区块链的数据库里面，每个区块链节点都可以访问和操作这个数据库，并且任何人都可以查询交易双方的账户信息。对于比特币网络来说，BTC 的转账，除了转给对方，还要将账户下的余额转给自己，这样就会生成多条记录，这样做是因为 BTC 没有账户余额的设计。

如图 1.2 所示，Bob 给 Alice 转 0.5 BTC 时，会生成 2 条交易记录，1 条记录是 Bob→Alice，另

1 条是 Bob→Bob，这些交易使用了非对称加密技术的加密和验证签名技术，这个过程属于区块链公链的设计范畴，不在本书讨论范围内。需要指出的是，这样可以在没有任何中心化银行提供服务并且没人掌握交易接收者或者发送者密钥的情况下，实现公开、安全的转账交易。即使交易数据公开在网络中传播，也无法被篡改，即无人可以将 Bob 转账的 0.5 BTC 修改为 5 BTC，因为任何数据的变更都会导致数据的 Hash 值变化，进而导致数据的签名无法被校验通过，进而无法被全网认可，也就是说伪造任何报文和交易，都是无法被网络认可的。这样，比特币在没有银行并且公开交易信息的情况下也能转账。

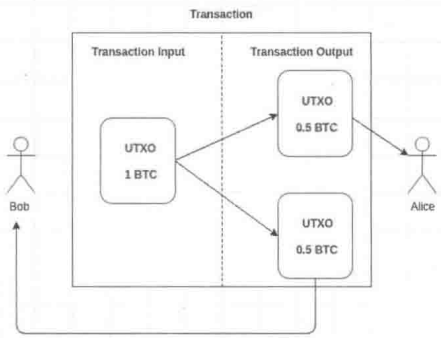


图 1.2　比特币网络交易

（3）发生在区块链上的交易，是存储在区块链网络中的所有全节点（拥有所有交易数据的区块链节点叫作全节点）上的，并且也是公开可查的，用户不仅可以查询某个账户下的所有交易，还可以查看其账户余额，这与银行的交易记录也是截然不同的。图 1.3 所示为比特币区块链浏览器，这些查询系统被称作区块链浏览器。与银行不同的是，即使能看到区块链交易的所有细节及所有账户的余额，也无法将账户与现实世界中的人联系起来，这与银行系统的账户设计完全不同。

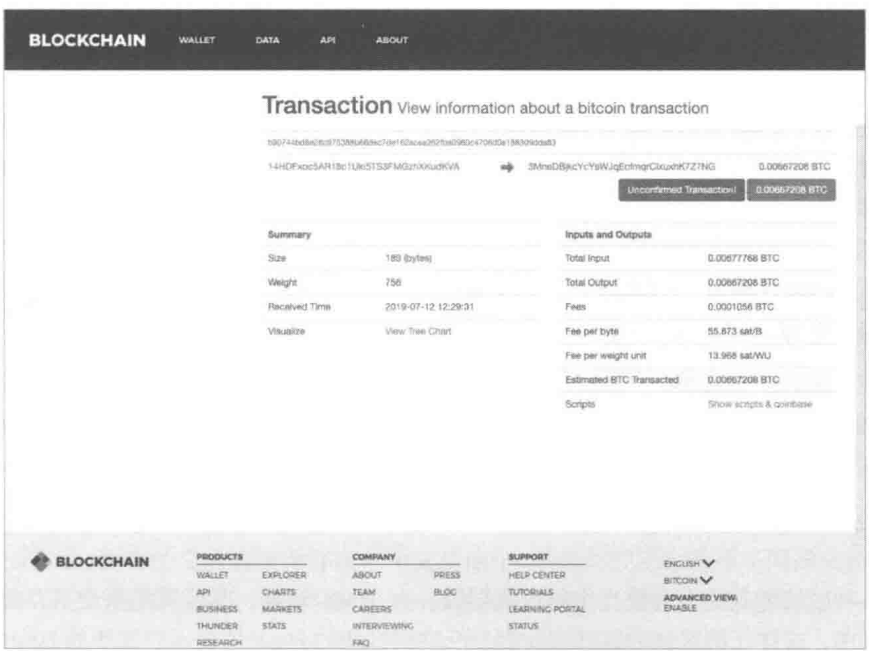


图 1.3　比特币区块链浏览器

1.1.2 区块

所有的交易信息都会形成一个结构化的账本，它们会被区块链的节点（矿工）按照一定的方式和时间间隔组织起来，存储在区块链节点中。这个用于存储交易信息的结构体就是区块，除了交易信息，区块还要存储一些额外的信息以保证交易数据的完整性和可靠性，区块数据结构如图 1.4 所示，该结构图仅仅列举了一些关键信息，不同的公链有各自不同的设计。关于区块的生产间隔，不同的区块链网络有不同的设定，比如以太坊出块的时间间隔约为 15s，而比特币网络则需要 10min 才生产一个区块。

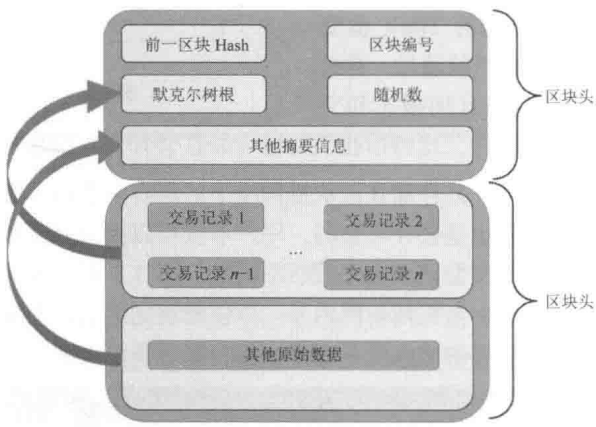


图 1.4 区块数据结构

区块数据包含区块头与区块体，区块体存储具体的交易及交易相关的原始数据，区块头存储的是原始数据的 Hash 信息，任何对原始数据的修改，都会引起区块头 Hash 值的变化，这样对任何信息的篡改都很容易被察觉到并被验证为假数据。除了交易原始数据和对应的 Hash 信息，还有一些为了维持区块安全性和有效性的附加信息，比如区块高度、区块难度、矿工地址等；不同的公链系统，还有不同的额外信息，比如以太坊会有状态变更的原始数据和对应的默克尔树信息、收据信息和账户余额信息等。

因此，区块的作用就是将不同时间阶段内的交易数据按照一定的格式和数量，打包成结构化数据，方便存储和管理。只有被打包到区块中并且被全公链网络认可的交易，才能算真正的有效交易，此时账户下面的数字货币的数量才会真正地发生增减，如果交易失败则转账无效。

1.1.3 链

区块头和区块体数据也会被当作输入数据做一次 Hash 运算，其运算结果会被存储在下一个区块的区块头中，这样任何区块内容的修改都会反映到区块的 Hash 值上，而区块的 Hash 值又是下一个区块的输入数据，它又会被当作新区块的数据参与一次新区块的 Hash 运算，随着时间的推移和

交易量的增加，所有的区块会通过保存前一个区块的 Hash 运算结果的方式组成一条链。

图 1.5 所示为区块链数据结构，对这个链的数据进行任何篡改，都需要修改此链的所有数据及对应的 Hash 值，只有这样篡改数据才能被校验为正确的数据。这需要非常强大的算力来支持，同时需要比较长的时间才能完成，而在篡改数据的过程中，还会不断有新交易加入区块链，因此从工程实现的角度来讲，这种篡改需要消耗极大量的能源并需要组织巨大量的计算资源，从经济的角度来讲，进行篡改不如进行算力挖矿划算，完全得不偿失。

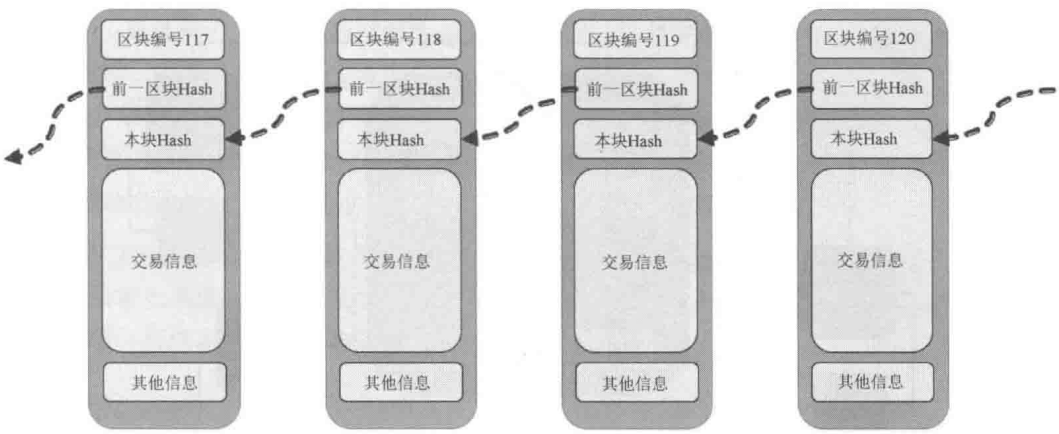


图 1.5　区块链数据结构

将交易打包成数据块，再将数据块以 Hash 值的方式组织成链式结构，就是区块链定义的来源。这种结构有点类似于计算机数据结构中的单向链，不同的是，这个链式结构不再局限于单个计算机，而是由分布在全世界的节点构成，任何人都可以公开查询，但又无法任意修改。

1.1.4　挖矿

区块在被增加到区块链之前，并不是所有区块都可以生成区块数据，也不是所有区块数据都能被增加到区块链成为最新的数据，这个过程有一定的门槛，需要筛选出一个值得信任的节点来生成数据，然后由其他节点来验证其生成数据的有效性。这个生产区块的过程会得到数字货币的激励，因此很多节点会加入生产区块的竞争。如果某个节点生产的区块数据得到了其他节点的验证，则其他节点会将最新的区块存储在本地，然后加入下一个数据块的生产竞争，这个过程被称作挖矿，而生成数据的节点被称作矿工。

如图 1.6 所示，所有的交易会被保存在一个交易池中，不同的矿工会选择不同的交易进行打包，然后制作属于自己的区块，假设当前将要生成的区块编号是 121，因为区块链网络是一个点对点的网络，没有中心节点进行协调，所以这些矿工节点无法感知到彼此的存在。

在矿工将交易打包成区块之后，还需要查找一个数字 n，这个数字 n 需要满足图 1.6 中的不等式，得到该数字之后将其放入打包的区块数据中，然后将该区块数据广播到区块链网络中，如果该区块被认可，则这个矿工就成了区块高度 121 的出块人，而网络中的节点将新生产的区块存储在本地，并将其链接到其他区块数据之后，基于此生产下一个区块。计算有效数字 n 的过程是需要付出算力的，而这个算力付出的过程是值得信赖的，这就是区块链为何可以在没有中心管理者的情况下正常运作，且任何人都可以在无须信任的情况下参与区块链业务，正所谓“In math we trust”。

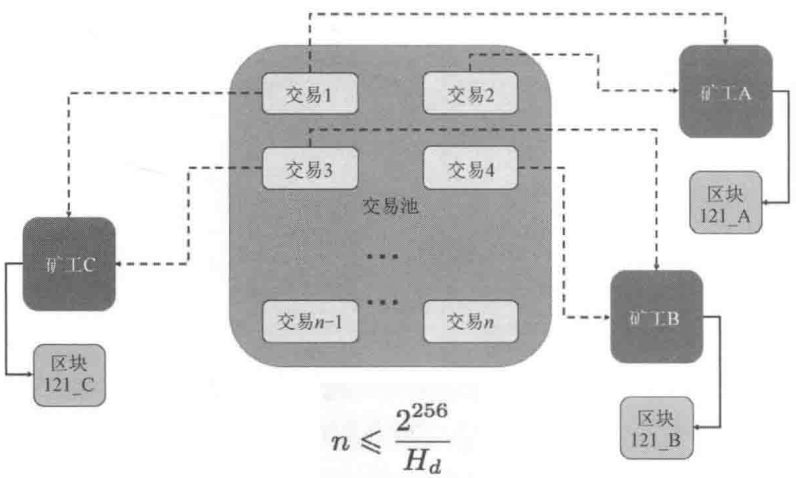


图 1.6　矿工从交易池中选择交易打包，并通过寻找满足不等式的数据 n 来挖矿

成为出块人就可以成功拿到奖励，奖励分为系统的奖励和交易中的交易手续费，在每一笔交易数据中，转账人都可以手动设置手续费，这些手续费用于奖励矿工打包的工作。在转账时设置的手续费越高，转账时间就越短，转账速度就越快，这是因为矿工们会优先选择手续费高的交易进行打包，而如果交易的手续费过低，很有可能会因为没有矿工愿意为其打包而导致转账失败。

1.1.5　共识算法

在挖矿的过程中，矿工需要付出算力来查找一个满足条件的数字，这种算力的付出是无法伪造的，必须付出相应的算力和电力之后才能得到正确的数字。当某一个节点发现该数字之后，其他节点可以很快验证该数字的有效性，验证并不用花费太多的算力和能源。也就是说图 1.6 中查找满足不等式条件的数字的过程是困难的，但是验证不等式的过程是简单的。这种为了持续生成区块而被所有网络节点认可的方案就叫作共识算法，而付出算力来证明自己工作的共识算法被称作 PoW（Proof of Work）。

除了 PoW 这种共识算法，还有一些其他的共识算法，它们的特征都是一样的，全网节点都共同

认可并遵守该算法，并且该算法产生的结果无法被伪造，但是可以被轻易验证。目前比较流行的共识算法有 DPoS、BFT 和 PoST 等。

- DPoS 算法类似于民主选举的运行模式，该算法根据数字货币持有的数量对区块事务进行投票管理，并且轮流来选举出块人进行出块，笔者认为这种弱中心化的方案与区块链的根本价值观背道而驰，但是在编写本书之时，此方案仍然流行。
- BFT 是著名的拜占庭容错机制，该机制的产生有着有趣的背景故事，读者可以自行查找相关细节。简单来讲，该方案通过多次通信交互来区分恶意节点和诚实节点，接收诚实节点的区块数据，丢弃恶意节点发送的数据和消息。
- PoST 是存储公链的共识算法之一，即统计节点有效存储的数据的大小和时长，将其作为节点的算力，来竞争成为出块节点，算力越大成为出块节点的概率越高，算力越大生成恶意数据的动力就越小，这是通过一种经济手段约束恶意行为的共识算法。

以上共识算法只是众多公链共识算法中的一小部分，是目前比较流行的几种方案。关于算法的技术细节，读者如果感兴趣，可以以本节的内容为线索，进行深入的探究。由于这些技术已经超出了本书的范畴，所以本书仅做简单论述，不再详细讲解。

1.1.6　分叉

因为整个区块链系统是点对点的对等网络，没有统一的中心机构协调各个节点的行为，所以在生产区块时，各个节点的行为都是相互独立的，很有可能同时由多个矿工在同一区块高度生产出 2 个以上的区块来。这些区块打包的交易很可能是不一样的，同时满足条件的数字 n 不是唯一的，多个矿工之间生产的数字 n 是不一样的，但是同样是满足不等式的。在这种情况下，网络中的其他节点很可能同步到不同的区块数据，并且这些数据在数学上都是合法的、有效的。当不同的节点中的不同的区块作为当前最新区块时，就会存在分叉的情况，即不同的矿机对同一高度的区块生产了内容不一样的新区块，并且这些矿工都找到了满足不等式的数字 n。

如图 1.7 所示，矿工 A 和 B 同时对编号 121 的区块进行打包并签名，它们同时广播了自己的区块到系统中，不同的节点有些会收到矿工 A 的区块数据，然后验证通过，并加入本地的账本，有些会收到矿工 B 的区块，这样整个系统的账本就会出现不一致的情况。为了解决这个问题，区块链采用了一种长链抛弃短链的决策方式：即当矿工 C 挖出了第 122 块数据，并将数据指向了矿工 B 的 121 区块时，因为矿工 B 所在的区块链更长，凝聚了更多的算力，既是最安全的，也是付出算力和资源最多的，此时矿工 A 所在的区块链在区块 120 位置发生的分叉将会被抛弃。区块 121_A 内所包含的交易也将失败，如果有一笔交易既被区块 121_A 打包，又被区块 121_B 打包，那么即使区块 121_A 被抛弃，该交易仍然成功；如果交易只在 121_A 区块中，则交易会在短链被抛弃前显示成功，在短链被抛弃后显示失败，失败后交易双方的账户余额会恢复到转账之前的状态。