

## 第12讲 | 深入区块链技术（四）：PoW共识

2018-04-20 陈浩

深入浅出区块链

[进入课程 >](#)



讲述：黄洲君

时长 11:34 大小 5.30M



上一篇文章中，我们谈到了区块链其实就是一种分布式系统，它在技术上并没有跳出分布式系统的理论框架，只是给出了一种不同于计算科学领域的解决方案。今天，我们就来重点聊聊区块链的这种解决方案：PoW 共识机制。

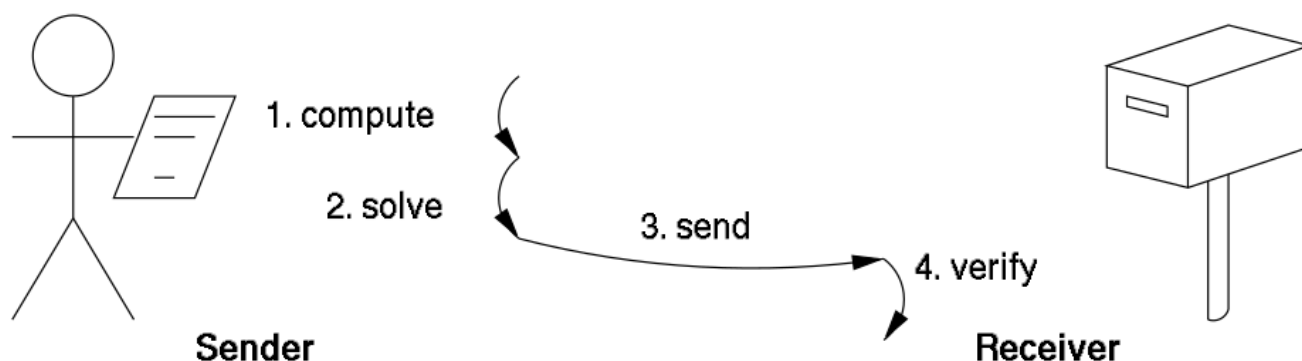
### PoW 工作量证明

因为比特币采用了 PoW 共识机制，所以这个概念才得以被广泛传播。PoW 全称 Proof of Work，中文名是工作量证明，PoW 共识机制其实是一种设计思路，而不是一种具体的实现。

PoW 机制其实早在 1997 年就被提出了，它早期多被应用在抵抗滥用软件服务的场景中，例如抵抗垃圾邮件（所以 PoW 在邮件服务系统会有所涉及）。

我们借用维基百科的一张图来解释一下 PoW 机制是如何用在这个场景中的。

为了防止垃圾消息泛滥，接收者并不直接接受来自任意发送者的消息，所以在一次有效的会话中，发送者需要计算一个按照规则约定难题的答案，发送给接受者的同时，需要附带验证这个答案，如果这个答案被验证有效，那么接受者才会接受这个消息。



可以看出 PoW 的核心设计思路是提出一个计算难题，但是这个难题答案的验证过程是非常容易的，这种特性我们称之为计算不对称特性，我们在“浅谈区块链共识机制”中举的 24 点游戏的例子就具备了□计算不对称特性。

## 如何理解区块链 PoW

上面介绍了一般的 PoW 是什么，那么区块链上的 PoW 又是如何设计的呢，我们还是以比特币为例子来讲一讲，这个部分会有代码演示，如果你在收听音频，可以点击文稿查看。

在分析拜占庭将军问题的时候可以看出，如果所有节点在同一时刻发起提案，那么这个系统的□记账过程将会非常的复杂混乱，为了□降低具有□提案权的节点数量，采用工作量证明不失为一个好办法。

所以我们需要构造一个计算不对称的难题，这个难题在比特币中被选定为□以 SHA256 算法计算一个目标哈希，使得这个哈希值符合前 N 位全是 0。

□举个例子，假设我们给定一个字符串“geekbang”，我们提出的难题是，计算一个数字，与给定的字符串连接起来，使这个字符串的 SHA256 计算结果的前 4 位是 0，这个数字我们称作 nonce，□比如字符串“geekbang1234”，nonce 就是 1234，我们要找到符合条件的 nonce。

我们以 Python 代码作为示例。

```
1
2 #!/usr/bin/env python
3 import hashlib
4
5 def main():
6     base_string = "geekbang"
7     nonce = 10000
8     count = 0
9     while True:
10         target_string = base_string + str(nonce)
11         pow_hash = hashlib.sha256(target_string).hexdigest()
12         count = count + 1
13         if pow_hash.startswith("0000"):
14             print pow_hash
15             print "nonce: %s  scan times: %s" % (nonce, count)
16             break
17         nonce = nonce + 1
18
19 if __name__ == '__main__':
20     main()
```

代码中，我规定了基础字符串是 "geekbang"，nonce 从 10000 开始自增往上搜索，直到找到符合条件的 nonce 值。

我们计算的结果放在图中，你可以点击查看。

```
1 # 前 4 位是 0
2 0000250248f805c558bc28864a6bb6bf0c244d836a6b1a0c5078987aa219a404
3 nonce: 68828  scan times: 58829
4 # 前 5 位是 0
5 0000067fc247325064f685c32f8a079584b19106c5228b533f10c775638d454c
6 nonce: 1241205  scan times: 1231206
7 # 前 7 位是 0
8 00000003f41b126ec689b1a2da9e7d46d13d0fd1bece47983d53c5d32eb4ac90
9 nonce: 165744821  scan times: 165734822
```

可以看出，每次要求哈希结果的前 N 位多一个 0，计算次数就多了很多倍，当要求前 7 位都是 0 时，计算次数达到了 1.6 亿次。这里我同时截图了操作系统当时 CPU 的负载，可以看到单核 CPU 负载长时间达到 100%。

```
× root@iZuf693neutho... №1 × mvs@iZuf693neutho... №2

1 [|||||||||||||||||||||||||||||||||||||100.0%] Tasks: 37, 78 thr; 2 running
2 [                                                                ] Load average: 0.64 0.20 0.06
3 [|                                                                ] Uptime: 161 days(!), 05:57:26
4 [                                                                ]
Mem[|||||||||||||||||||||||||||||||||195]
Swp[                                                                ]

PID USER      PRI  NI  VIRT   RES   SHR  S  CPU% MEM%   TIME+  Command
32011 mvs         20   0 31132  7388  4632  R  99.5  0.1   0:52.90 python ./pow-hash-test.py
```

通过上述程序，希望你对区块链 PoW 机制有个直观的了解。□由于结果只能暴力搜索，而且搜索空间非常巨大，作弊几乎不可能，另外符合条件的 nonce 值也是均匀分布在整个空间中的，所以哈希是一个非常公平且粗暴的算法。

以上代码的基本逻辑就是 PoW 挖矿过程，搜索到一个目标值就会获得记账权，□距离上一次打包到现在未确认的交易，矿工就可以一次性将未确认的交易打包并广播了，并从 Coinbase 获得奖励。

□实际挖矿的基本步骤如下。

1. 生成 Coinbase 交易，并与其他所有准备打包进区块的交易组成交易列表，□并生成默克尔哈希；
2. 把□默克尔哈希及其他相关字段组装成区块头，将区块头（Block Header）作为工作量证明的输入，区块头中包含了前一区块的哈希，区块头一共 80 字节数据；
3. 不停地变更区块头中的随机数即 nonce 的数值，也就是暴力搜索，并对每次变更后的的区块头做双重 SHA256 运算，即 SHA256(SHA256(Block\_Header)))，将结果值与当前网络的目标值做对比，如果小于目标值，则解题成功，工作量证明完成。

如果更深程度去理解的话，PoW 机制是将现实世界的物理资源转化成区块链上虚拟资源的过程，这种转化为区块链提供了可信的前提。

## PoW 挖矿的发展历程

好了，现在我们知道了，PoW 的过程其实就是计算一个难题解的过程。

在区块链的发展史上，PoW 经历了大致两个阶段。分为早期□分散挖矿阶段和中心化矿池挖矿阶段。我们目前处于第二个阶段，并且将会长期处于这个阶段。

早期分散挖矿是中本聪的愿景，期望是 1CPU=1 票，所以如果 CPU 挖矿，那么将会是非常理想化的情况，而现实的情况是，SHA256 只需要非常简单的重复计算逻辑，它不需要复杂的逻辑控制。

那么 CPU 这种重控制逻辑，轻重复计算的计算单元来搞这么低端的暴力计算非常吃力不讨好，大部分人的第一反应肯定是用 GPU 呀，非常正确。

所以这个时期，出现了 GPU 挖矿，它的效率是 CPU 的十几甚至上百倍，那么 1CPU=1 票的逻辑就被打破了，挖矿工具的改变让人们意识到挖矿技术也是极大改进的。除了 GPU 挖矿，我们还有 ASIC 芯片挖矿，这部分内容我们在讨论挖矿算法分类时会详细讲解。

同期我们也慢慢进入到了中心化挖矿阶段。中心化挖矿很好理解，算力如果越分散，也就意味着竞争越激烈，如果某个节点计算出答案了，那么也意味着其他矿工这段时间的工作量几乎都白费了，投入了物理资源结果零收益，可以说是负收益。

那怎么办呢？思路就是把分散的算力汇聚到一个池子里面，这个池子我们称作矿池，就像四面八方的小溪流最终汇总成一条大江一样。

矿工参与到某个矿池，相当于矿工把算力租给矿池了，与其他矿工联合挖矿，最后看起来矿池这个节点的算力就会很大，获得记账权的概率就越大，如果这个矿池计算出了答案，将获得 Coinbase 的奖励，矿池就会按既定的分配比例打给每一位参与的矿工。

我们借用一下《精通比特币》一书中的部分图来看一下：



FPGA 出现的时间比较短暂，最终人们开发出了 ASIC 专业芯片来计算 SHA256，这就是我们常说的专业矿机。

专业矿机的出现加速了 PoW 挖矿的中心化过程，因为购买专业矿机需要额外的时间和精力，配置运行还有一定的门槛，普通人也只能从专业机构手里购买专业矿机。

所以这些专业矿机直接就是数字货币印钞机，生产专业挖矿芯片的商业公司几乎成了数字货币的货币发行司，这不得不说到市值直逼英伟达的比特大陆公司，它用的就是专业生产数字货币挖矿芯片。

新的数字货币开发者们为了防止情况重演，不断发明新的挖矿算法。有名的有 Scrypt、X11、SHA-3，不过这些依然是计算困难型的挖矿算法，依然没有逃脱出现专业矿机的命运。

这里不得不提到以太坊的 PoW 挖矿算法：ETHASH，ETHASH 是 Dagger-Hashimoto 的修改版本，它是典型的内存困难型挖矿算法。直到如今，也没有芯片厂商设计出挖矿芯片。

正如我们上文所说，因为工作量证明要求的组件从计算资源转变为内存资源，而对内存的高要求使得矿工必须加内存。

在专业矿机上加一块内存的收益□与在 GPU 上加一块内存获得的收益是差不多的，所以厂商并没有研发内存困难型专业矿机的动力，没有专业矿机的出现，这从某种程度上也缓解了算力中心化的问题。

## PoW 的优势和劣势

PoW 共识的内在优势在于可以稳定币价，因为在 PoW 币种下，矿工的纯收益来自 Coinbase 奖励减去设备和运营成本，成本会驱使矿工至少将币价维持在一个稳定水平，所以攻击者很难在短时间内获得大量算力来攻击主链。

PoW 共识的外在优势是目前□它看起来依然是工业成熟度最高的区块共识算法，所以在用户信任度上、矿工基础上都有很好的受众。

PoW 共识最大的缺点是非常消耗计算资源，耗电耗能源，这一点也一直为人们所诟病。因为每次产生新的区块都会让相当一部分工作量证明白白浪费了，也就是将计算资源浪费了。

目前来看这个是无解的，只要是 PoW 共识，一定会遇到计算资源浪费的问题。不过人们也想了一些改进方案，早期如素数币，近期有比原币，它们都号称深度学习友好型的工作量证明方法。

从理论上来看，PoW 会一直有 51% 算力攻击的问题，即攻击者只需要购买超过全网 51% 算力设备，即可发起“双花攻击”，甚至“重放攻击”等多种高收益攻击，这个问题目前没有解决方案。

除了 51% 攻击，PoW 共识还有自私挖矿的问题，自私挖矿是一种特殊的攻击类型，不会影响区块链正常运转，但是会形成矿霸，间接造成 51% 攻击，我们就曾经遇到过这样的自私挖矿攻击。

PoW 共识机制是一种简单粗暴的共识算法，它不要求高质量的 P2P 网络资源，它可以为公链提供稳定有效的记账者筛选机制。同时它也面临了挖矿中心化严重的问题，这也促使人们研究出了新的共识机制，我们留到下一篇讲解。

## 总结

今天我介绍了 PoW 工作量证明，并且使用 Python 语言演示了一遍基于 SHA256 的挖矿算法工作过程，又介绍了发展历程和算法分类，最后提到了 PoW 的优势和缺陷。相信你对 PoW 机制的理解可以更加深入了。

PoW 工作量证明的挖矿过程是否可以替换成有意义的算法呢，历史上是否有过类似创新？你可以调查一下，我们一起分享。




# 深入浅出区块链

你的区块链入门第一课

陈浩 元界 CTO



新版升级：点击「请朋友读」，10位好友免费读，邀请订阅更有**现金**奖励。



上一篇 第11讲 | 深入区块链技术（三）：共识算法与分布式一致性算法

下一篇 第13讲 | 深入区块链技术（五）：PoS共识机制

## 精选留言 (13)

写留言



阿痕

2018-04-21

3

PoS和DPoS就是对pow不环保的改进，不过也带来了新的问题，所以不可能三角还是有一定道理的



沃野阡陌

2018-04-21

2

什么是coinbase奖励？

展开

作者回复: 系统奖励给挖出块矿工的交易类型



suns

2018-07-26

1

老师 现在是不是出了一个ant e3型号的asic矿机是针对ethash的

展开

作者回复: 是的，但效率提升只有两倍，很明显不像其他ASIC提升千倍以上。所以谈不上威胁，毕竟内存困难型的还是吃内存的。



八神

2018-10-01

1

交易是用收款方的公钥加密的，除了收款方可以查看，别人无法看，校验这笔交易的人该

如何校验呢？必须是收款方才能校验吗？

作者回复: 你好，并不是用收款方的公钥加密，而是使用收款方公钥的哈希作为地址，通过构造一个交易脚本，使得收款方能辨别并且必须出具签名才能操作。交易都是公开的，所有人都可以验证。具体:发送者通过地址构造一个ScriptPubKey的“锁”，收款方必须构造一个ScriptSig的“钥匙”，才能花费这些币。



**Calios**

2018-09-26



python代码示例在iPhone上看不到，有其他同学有这个问题么？

展开 ∨



**锐**

2018-06-20



自私挖矿是个什么概念？老师解释一下呢

展开 ∨



**skevy**

2018-04-28



举个例子，假设我们给定一个字符串“geekbang”，我们提出的难题是，计算一个数字，与给定的字符串连接起来，使这个字符串的 SHA256 计算结果的前 4 位是 0，这个数字我们称作 nonce，比如字符串“geekbang1234”，nonce 就是 1234，我们要找到符合条件的 nonce。

...

展开 ∨

作者回复: 这个由挖矿算法的细节决定的，一般是变化的，通过历史记录计算出来的，例如以太的 DAG生成。



**塞翁**

2018-04-28



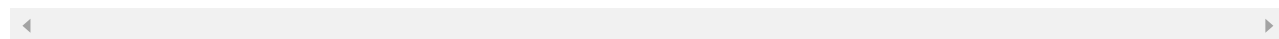
生成 Coinbase 交易，并与其他所有准备打包进区块的交易组成交易列表，□并生成默克尔哈希；

这段中，怎么保证收到的交易列表的完整性，如果漏收交易，所有运算都白费了？或者说，这个交易列表完整性是怎么判断的？纯粹按照时间？

展开 ▾

作者回复: 不存在完整性这一说的，有就有，没有就不会被打包了，完全取决于记账节点与你的网络是否可以路由通，如果不通说明网络分叉了。

也就是纯粹的先后来后到，一起打包。



ylshuibug...

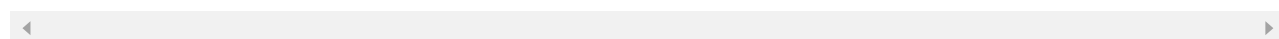
2018-04-23



为什么矿池的特殊在于它可以产生新的区块

展开 ▾

作者回复: 代码中约定的



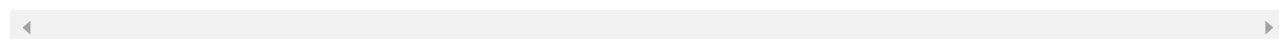
小5

2018-04-23



看了下精通比特币电子书，这些基本的概念里面都有讲，什么时候能上一些网上搜不到的东西

作者回复: 基础概念其实是相同的，这个计算机教材入门讲都差不多是类似的，不过为了小白读起来容易花了不少功夫哇。下一篇开始逐渐深入了。希望继续关注哈



熊猫

2018-04-22



自私挖，全节点，双花攻击和重放攻击能分别讲解是什么意思吗？

展开 ▾

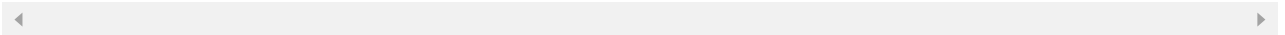
作者回复: 全节点是一个独立节点，可以自己验证交易并且可以挖矿的节点。

自私挖矿是指矿工先通过较大算力积累优势，挖出多的块的同时不广播，等别人广播一个新区块的时候，自己一下广播2个甚至更多区块，让别人一直处于被分叉的状态，自己成为矿霸，可以

形成100%出块率。

双花攻击就是指一个币可以花费两次，属于数据库一致性问题。

重放攻击与传统it的重放不同，是指硬分叉的两条链的交易都是合法的，可以在a生成交易后去b上去花费。



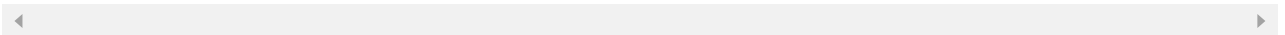
良辰美景

2018-04-21



我最近听得到吴军老师说比特币交易成本极高。去中心化带来了大量资源的浪费。如果有好的解决办法是个暴富的机会哦

作者回复: PoW是的，其他算法PoS，DPoS是环保的。



不了峰

2018-04-21



「ETHASH 是 Dagger-Hashimoto 的修改版本，它是典型的内存困难型挖矿算法。」请问以太坊的挖矿，矿机是占用大量内存还是占用大量cpu？

我其实是想问，一台主机上能不能部署多个挖矿机，比如，门罗和ETH，我以为一个是消耗cpu，一个是消耗内存。

展开 ∨

作者回复: 可以的，正常矿机就是部署6~8张显卡。目前cpu挖矿的币种极少。

