

区块链技术（2）-区块链技术细节-哈希算法

2018-04-03 陈皓



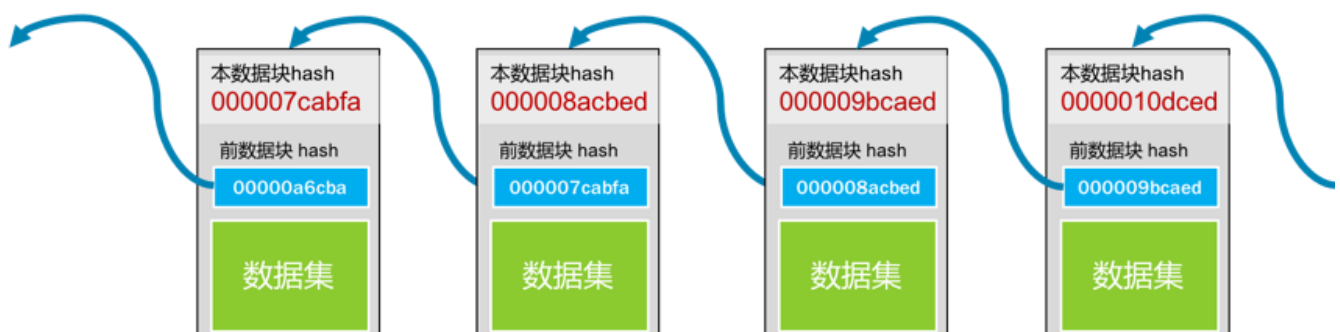
区块链技术（2）-区块链技术细节-哈希算法

朗读人：柴巍 11'13" | 5.14M

对于计算机来说，区块链就像一个单向链表，一个数据块中保存着三个信息。

- 真正的数据。
- 自己的地址（或是 ID）。
- 前一个数据块的地址。

这样，通过追溯前一个块的地址，把所有的数据块存成了一条链。所以，我们叫其 BlockChain。如下图所示。



每个数据块的 " 地址 " 的编码使用了计算机上的一个算法，计算机圈内人士把这个算法叫 Secure Hash。有人音译为 " 安全哈希 "，也有人意译为 " 安全散列 "。在计算机应用中，hash 算法主要有几个功能。

- 用来生成唯一标识一个数据块的 ID（身份证），这个 ID 几乎不能重复。
- 用来做数据的特征码。只要数据中一个 bit 的数据出现更改，那么整个 hash 值就完全不一样了。而且数据学上保证了，我们无法通过 hash 值反推回原数据。

于是，很多公司在互联网上发布信息或软件的时候，都会带上一个 Checksum（校验码）。你只要把整个文件的数据传入到那个特定的 hash 算法中，就会得到一串很长的字符串。如果和官方发布的 Checksum 字符串不一样，那么就说明信息或文件内容被人更改或是信息残缺了。因此，也被应用在 " 数字签名 " 中。

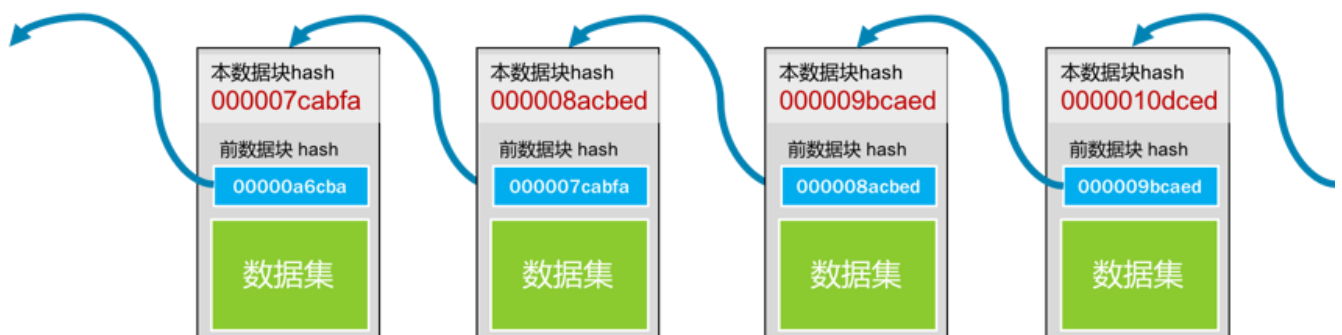
在计算机世界里，有两个很著名的 hash 算法，一个叫 MD5（[Wikipedia 链接](#)），一个叫 SHA-2（[Wikipedia 链接](#)），区块链用的是 SHA-256 这个算法。

下面是一个示例。

- 对 "chen hao" 这个信息计算 MD5 值得到 9824df83b2d35172ef5eb63a47a878eb（一个 16 进制数）。
- 如果对 "chen hao" 做一个字符的修改，将字母 "o" 改成数字 "0"，即 "chen ha0"，计算出来的 MD5 值就成了 d521ce0616359df7e16b20486b78d2a8。可以看到，这和之前的 MD5 值完全不一样了。

于是，我们就可以利用 hash 算法的这个特性来对数据做 " 数字签名 "。也就是说，我将 " 数据 " 和其 " 签名 "（hash 计算值）一起发布，这样可以让收到方来验证数据有没有被修改。

我们再来看上面那个区块链的图。



对于第一块数据，我们把其 " 数据集 " 和 " 前数据块的 hash 值 00000a6cba " 一起做 hash 值，得到本区块的地址 000007cabfa。然后，下一个区块会把自己的数据和 000007cabfa 一起做 hash，得到 000008acbed 这个哈希值.....如此往复下去。

根据 " 被 hash 的数据中有一个 bit 被修改了，整个 hash 就完全不一样了 " 这个特性，我们知道：

- 如果前置数据块中的数据改了，那么其 hash 就会完全不一样了，也就是说你的 ID 或地址就变了，于是别人就找不到这个数据块了；
- 所以，你还要去修改别人数据块中指向你的地址，但是别人数据块中指向你的地址（ID/hash）变了，也会导致他自己的地址（ID/hash）随之变化。因为他用你的地址生成了自己的地址，这样一来，你就需要把其他人的地址全部改掉。

在这样的连锁反应下，你想要偷偷修改一个 bit 的难度一下就提高很多。所以，在区块链的世界里，越老的区块越安全也越不容易被人篡改，越新的区块越不安全也越容易被人篡改。

比特币的 hash 算法

下面我来简单介绍一下，比特币中区块链的一些细节。下图是区块链的协议格式。

Magic Number (4)		Block Size (4)		
Version (4)		Previous Block Hash (32)		
		Merkle Root(32)		
		Timestamp (4)		
Difficulty Target (4)		Nonce (4)		
Transaction Counter (Variable : 1-9)				
	Transaction List (Variable : Upto 1 MB)			

BLOCK HEADER

其中 Version，Previous Block Hash，Merkle Root，Timestamp，Difficulty Target 和 Nonce 这六个数据字段是区块链的区块数据协议头。后面的数据是交易数据，分别是：本块中的交易笔数 H 和交易列表（最多不能超过 1MB，为什么是 1MB，后面会说）。

下面我来说一下区块头中的那六个字段的含义。

- Version：当前区块链协议的版本号，4 个字节。如果升级了，这个版本号会变。
- Previous Block Hash：前面那个区块的 hash 地址。32 个字节。

- Merkle Root : 这个字段可以简单理解为是后面交易信息的 hash 值 (后面具体说明一下) 。 32 个字节。
- Timestamp : 区块生成的时间。这个时间不能早于前面 11 个区块的中位时间 , 不能晚于 " 网络协调时间 "——你所连接的所有结点时间的中位数。 4 个字节。
- Bits : 也就是上图中的 Difficulty Tagrget , 表明了当前的 hash 生成的难度 (后面会说) 。 4 个字节。
- Nonce : 一个随机值 , 用于找到满足某个条件的 hash 值。 4 字节。

对这五字段进行 hash 计算 , 就可以得到本区块的 hash 值 , 也就是其 ID 或是地址。其 hash 方式如下 (对区块头做两次 SHA-256 的 hash 求值) :

```
SHA-256(SHA-256 (Block Header))
```

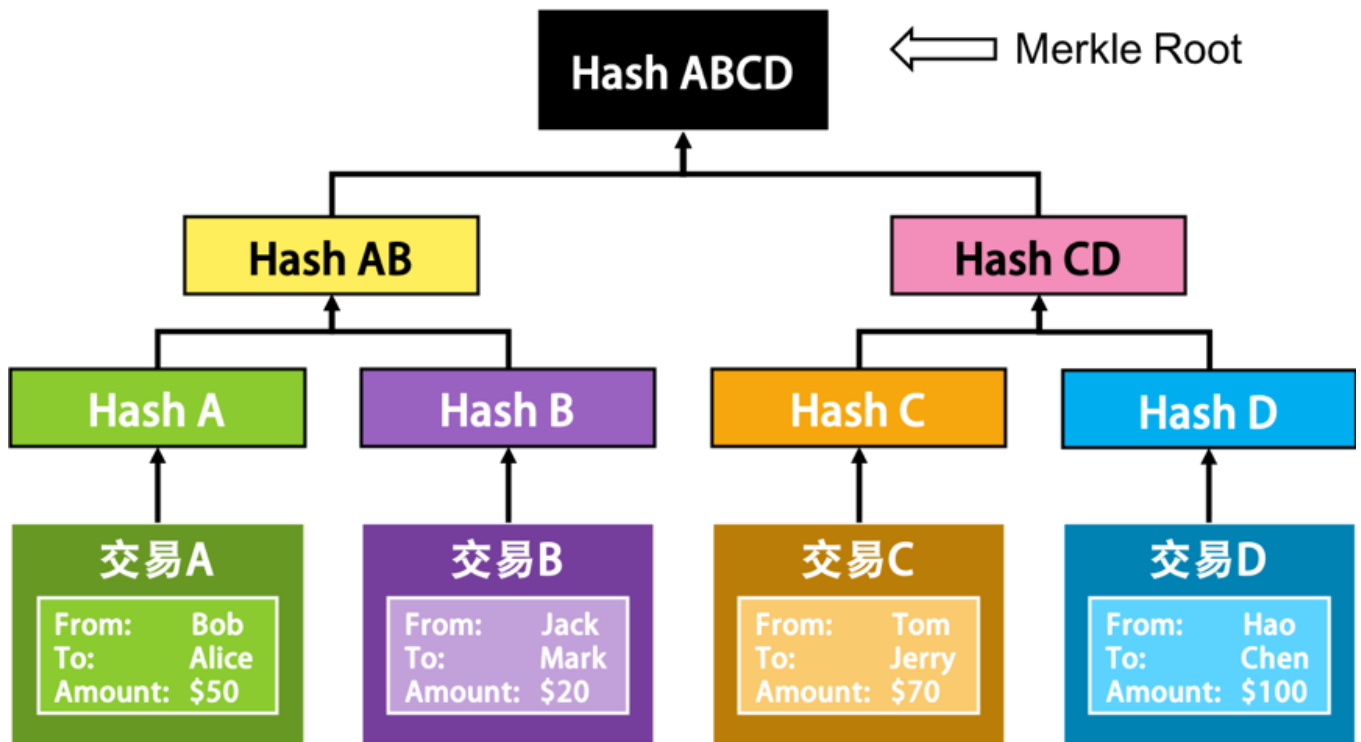
当然 , 事情并没有这么简单。比特币对这个 hash 值是有要求的 , 其要求是那个 Bits 字段控制的 , 然后你可以调整 Nonce 这个 32 位整型的值来找到符合条件的 hash 值。我们把这个事情叫做 " 挖矿 " (在下一篇中 , 我们会详细讲一下这个事) 。

关于 Merkle Root

前面说到过 , 可以简单地将 Merkle Root 理解为交易的 hash 值。这里 , 我们具体说一下 , 比特币的 Merkle Root 是怎么计算出来的。

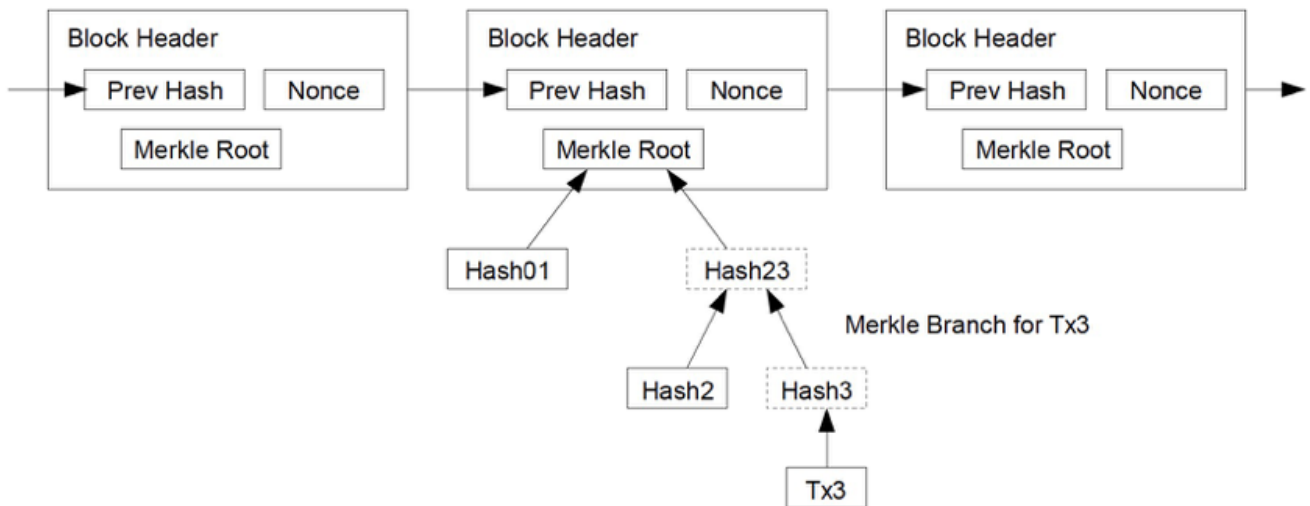
首先 , 我们知道 , 比特币的每一笔交易会有三个字段 , 一个是转出方 , 一个是转入方 , 还有一个是金额。那么 , 我们会对每个交易的这三个字段求 hash , 然后把交易的 hash 做两两合并 , 再求其 hash , 直到算出最后一个 hash 值 , 这就是我们的 Merkle Root。

我画了一个图展示一下这个过程。



上面的示意图中有四笔交易，A 和 B 的 hash 成了 Hash-AB，C 和 D 的 hash 成了 Hash-CD，然后再做 Hash-AB + Hash-CD 的 hash，得到了 Hash-ABCD，这就是 Merkle Root。整个过程就像一个“二叉树”一样。

下图是一个区块链的示意图，来自[比特币的白皮书](#)。



为什么要这样做呢？为什么不是把所有的交易都放在一起做一次 hash 呢？这不也可以让人无法篡改吗？这样做的好处是——我们把交易数据分成了若干个组。就像上面那个二叉树所表示的一样，我们可以不断地把这个树分成左边的分支和右边的分支，因为它们都被计算过 hash 值，所以可以很快地校验其中的内容有没有被修改过。

这至少带来三个好处。

1. 大量的交易数据可以被分成各种尺寸的小组，这样有利于我们整合数据和校验数据。

2. 这样的开销在存储和内存上并不大，然而我们可以提高校验一组数据的难易程度。
3. 在 P2P 的无中心化网络上，我们可以把大量数据拆成一个一个小数据片传输，可以提高网络的传输速度。

最后，需要说一下的是，以太坊有三个不同的 Merkle Root 树。因为以太坊要玩智能合约，所以需要更多的 Merkle Root。

- 一个是用来做交易 hash 的 Merkle Root。
- 一个是用来表示状态 State 的。因为一个智能合约从初始状态走到最终状态需要有若干步（也就是若干笔交易），每一步都会让合同的状态发生变化，所以需要保存合同的状态。
- 还有一个是用来做交易收据的。主要是用来记录一个智能合约中最终发生的交易信息。在 StackExchange 上的问题 "[Relationship between Transaction Trie and Receipts Trie](#)" 中有相应的说明，你可以前往一看。

以太坊称其为 Merkle Patricia Tree（具体细节可参看其[官方的 Wiki](#)）。

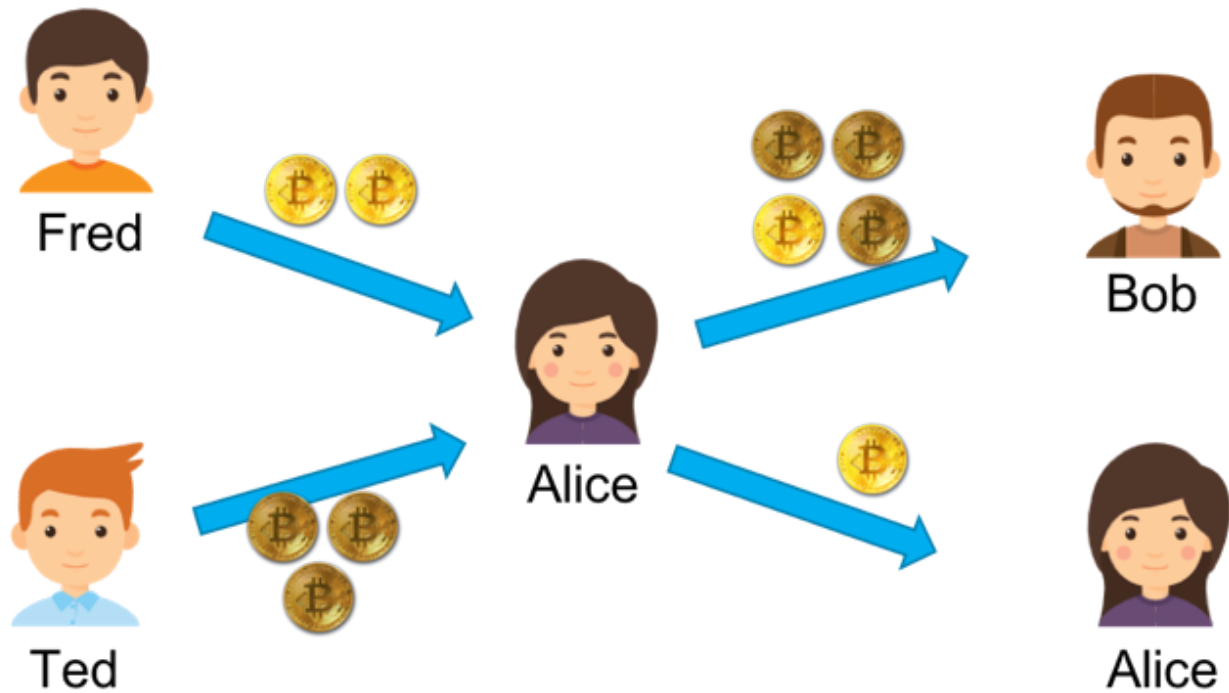
比特币的交易模型

比特币区块中的交易数据，其实也是一个链。为了讲清楚这个链，我们需要了解一下比特币交易中的两个术语，一个是 input，一个是 output，也就是交易的支出方（input）和收入方（output）。

在比特币中，一个交易可以有多个 output，也就是说我可以把一笔钱汇给多个人，但一个 output 只能对应一个源的 input，还有一个条件就是，output 跟 input 的总数要吻合。

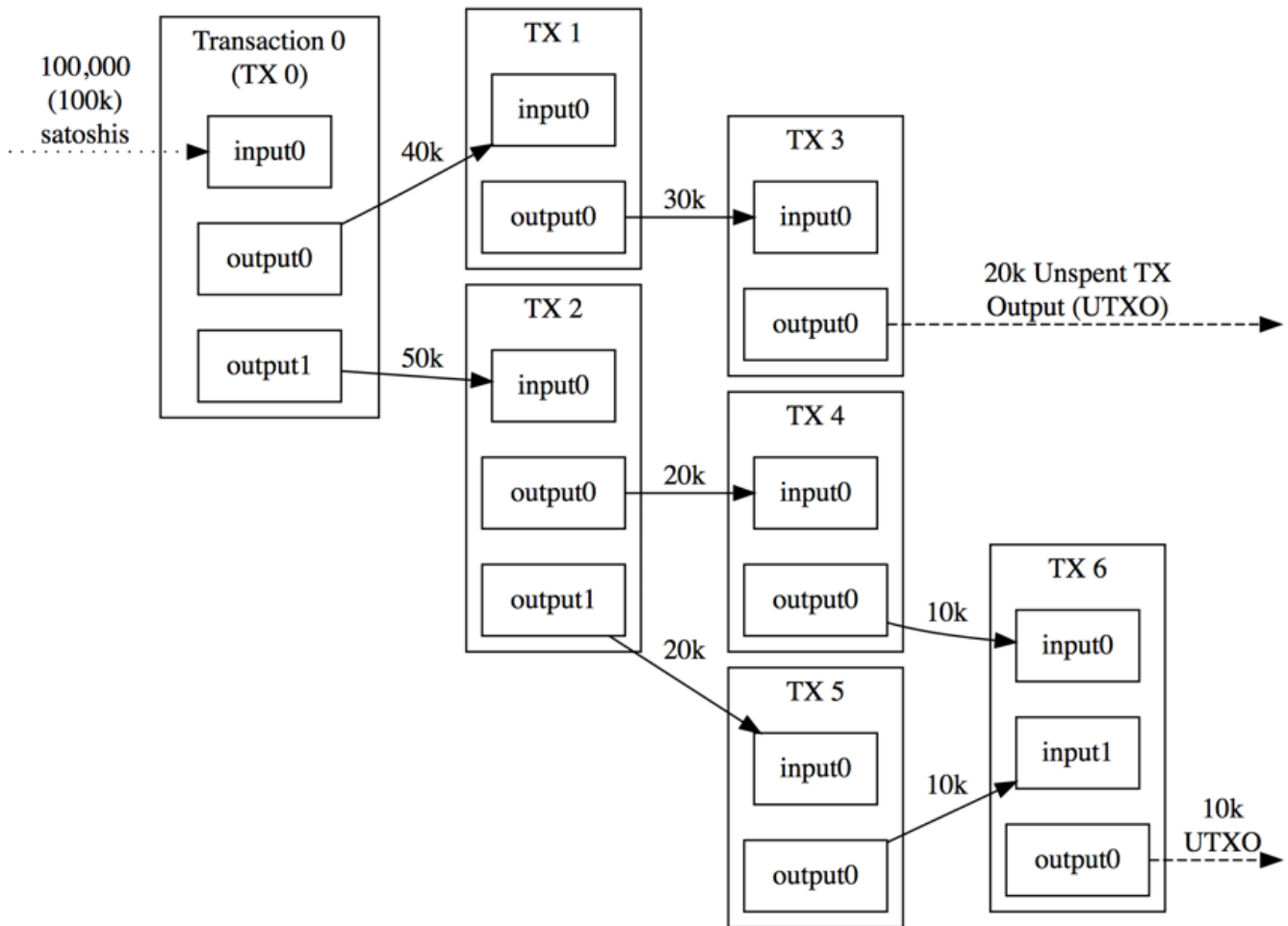
这里举个例子。假设，Fred 给了 Alice 2 个比特币，Ted 给了 Alice 3 个比特币，这个时候，Alice 有 5 个比特币。然而，大比特币的世界里是没有余额的，所以，对于 Alice 来说，她只有两个没有花出去的交易，一个是 2 个比特币，一个是 3 个比特币。这在比特币中叫 UTXO（Unspent Transaction Output）。

此时，如果 Alice 想要转给 Bob 4 个比特币，她发现自己的两个交易中都不够，也不能拆开之前的那两个比特币交易，那么她只能把交易 2 和交易 3 当成 input，然后把自己和 Bob 当成 output，Bob 分得 4 个，她自己分 1 个。这样的交易才平衡。



于是，一笔交易可能会包含大量的 Input 和 Output。因为比特币没有 " 余额 " 的概念，所以需要通过多个 input 来凑，然后 output 这边还需要给自己 " 找零 "，给矿工 " 小费 "。

这样一来，在比特币交易中，交易会开成一个链，也就是你把钱给了我，我又给了张三，张三给了李四.....就这样传递下去，形成了一个交易链。因为还没有花出去，所以就成为了 UTXO，而系统计算你有没有钱可以汇出去时，只需要查看一下你的 UTXO 就可以了。



Triple-Entry Bookkeeping (Transaction-To-Transaction Payments) As Used By Bitcoin

(图片来源：<https://bitcoin.org/en/developer-guide>)

UTXO 因为没有账户和余额的概念，所以可以并行进行多笔交易。假如你有多个 UTXO，你可以进行多笔交易而不需要并行锁。然后其还有匿名性的特征，你可以隐藏自己的交易目的地（通过设置的多个 output），而且没有余额意味着是没有状态的。要知道你有多少个比特币，只需要把 UTXO 的交易记录统计一下就可以知道了。但这也让人比较费解，而且也不利于应用上的开发。以太坊则使用了余额的方式。

在这篇文章中，我先讲述了什么是区块链？它是如何得名的，它的核心原理是什么。随后分享了比特币的 hash 算法，以及 Merkle Root 是如何计算出来的。最后，介绍了比特币的交易模型。希望对你有帮助。（这篇文章中图片很多，很难用音频体现出来，所以没有录制音频，还望谅解。）

文末给出了《区块链技术》系列文章的目录，希望你能在这个列表里找到自己感兴趣的内容。

- [区块链技术（1）- 区块链的革命性及技术概要](#)
- [区块链技术（2）- 区块链技术细节：哈希算法](#)
- [区块链技术（3）- 区块链技术细节：加密和挖矿](#)
- [区块链技术（4）- 去中心化的共识机制](#)
- [区块链技术（5）- 智能合约](#)

- 区块链技术 (6) - 传统金融和虚拟货币



版权归极客邦科技所有，未经许可不得转载

精选留言



杜小琨

所以，为什么是1M？

2018-04-03

作者回复

我居然写漏了.....😓

👍 12

1) 中本聪经常在代码里放一些不说明的事，这个就是一个。

2) 区块有大小限制容易理解，一个是为了更好地去中心化，因为如果尺寸下限，算力大的可以打包更多的交易，就中心化。另一个是为了节省网络带宽。

3) 为什么是1M而不是2M，这是中本聪设置的，但他并没有做出解释。

2018-04-06



逆行

一次交易完成后，Merkle Root 就会改变，那么每次交易都要重写整个区块链表，那效率太低了吧，是不是我错过了什么细节？

2018-04-03

作者回复

Merkle Root 是一组交易的层层hash的结果。

2018-04-06

👍 4



jimmy

👍 3

近期在研究event-sourcing架构，比特币的交易感觉有点像event-sourcing的味道，记录的只是事件，无状态，天然适应分布式，不知道这个类比是否有点牵强

2018-04-03

| 作者回复

是很牵强

2018-04-06



Nelson

👍 3

"假如你有多个 UTXO，你可以进行多笔交易而不需要并行锁。"这一句没懂，不会出现一个UTXO被使用两次情况吗？

2018-04-03

| 作者回复

不同的交易在不同的UTXO上并行，在同一个上不行。如果用余额来做，则无法并行，上一笔不完成，下一笔无法执行。

2018-04-06



晓聪

👍 2

"" 对这五字段进行 hash 计算，就可以得到本区块的 hash 值 ""

数了下，前面介绍了6个，这里写的五个，是笔误吗？

2018-04-29



修炼

👍 2

然而对比特币没兴趣

2018-04-03



Donald

👍 1

为什么是对区块头做两次 SHA-256 的 hash 求值？

2018-04-14



郎哲

👍 1

赞赞赞讲的通俗易懂

2018-04-03



i

👍 0

既然一个区块可以存放至多四千笔交易，那什么时候才能生成新区块？够四千笔交易还是其他别的生成逻辑？

2018-06-03



龚极客

👍 0

Merkle Root是否至多三层？因为每个hash4个字节*7 = 28 < 32。如果这样，那么超过3层怎么处理？

2018-04-19

作者回复

没有啊，hash无论hash多长的字符，总是得到一样的长度。你说的“hash4个字节*7=28”是什么意思？

2018-04-19

**怀兵**

0

关于以太坊state存储的部分有点模糊，表达成是账户状态的存储，而非合同状态，可能更好一些

2018-04-18

**总指挥**

0

生成Merkle Root 过程理解的，但它的目的还是不能理解，如果是要验证完整性为什么不直接全部tx来哈希？小弟不才，望各位解答。

2018-04-13

作者回复

文中已经讲了。可以分块整理

2018-04-13

**吴天**

0

UTXO不太理解 我有多少可交易的比特币是系统从哪里验证的？从区块链追溯下去还是另外有一个存储记录？

2018-04-10

**湖心亭看雪**

0

同问为什么是1M?

2018-04-05

**WTF**

0

下载软件通过校验码验证真伪这个小常识，可以避免很多李鬼事件，就值回票价。

2018-04-05

作者回复

好习惯

2018-04-08

**Dylan**

0

最近在研究ETH的MPT树，看到这豁然开朗很多了

2018-04-04

**浪迹天涯**

0

期待下一期的挖矿的技术细节，HashCash和命中概率问题。

2018-04-04



WX

0

浅显易懂，点赞

2018-04-04



李书德

0

这个严格来说是说比特币而不是区块链的，换作以太坊很多就不一样了

2018-04-03

作者回复

区块链的总体结构是一样的

2018-04-06



风的叹息

0

单笔交易和区块是什么关系？记录在数据集里面之一？然后同时进行的交易呢？广播，这里的时序问题怎么处理的，不知后文有没有解答

2018-04-03

作者回复

一个区块里有多笔交易，一个区块最大1M，一笔交易平均250个字节，于是一个区块4000笔交易，一个区块平均生成时间10分钟，所以每秒6.66667笔交易。

交易没有时序问题，因为一笔交易的支出必需来自（Unspent Transaction Output），所以，如果你无法对同一个“未花交易”操作两次，而对一个“未花交易”的操作未被确认前是会产生新的未花交易，于是你也就无法进行下笔操作。

广播的时序？为什么会有时序问题呢？一个区块里有前一个区块的地址，另外，算力太大了，10分钟才能产生一个，完全是超低并发应用。

2018-04-06