通信网络数理基础与应用大作业

院（系）名 称：信息与通信工程学院

学 生 姓 名：蒋佳宁

北京邮电大学

**2022**年**7**月

目 录

[第1章 比特币中的密码学原理 3](#_Toc109816028)

[1.1 哈希函数 3](#_Toc109816029)

[1.2 两种加密体系 3](#_Toc109816030)

[1.3 数字签名过程 3](#_Toc109816031)

[第2章 比特币中的数据结构 5](#_Toc109816032)

[2.1 哈希指针 5](#_Toc109816033)

[2.2 Merkle tree(默克尔树) 5](#_Toc109816034)

[第3章 比特币的协议 7](#_Toc109816035)

[3.1 比特币区块的具体介绍 7](#_Toc109816036)

[3.2 比特币的共识协议—PoW 7](#_Toc109816037)

[3.3 PoW的实现—挖矿机制 8](#_Toc109816038)

# 第1章 比特币中的密码学原理

1.1 哈希函数

区块链使用的哈希函数(cryptographic hash function)是SHA-256。哈希函数需要满足的三个条件：

1）可以接收任意长度的字符串作为输入；

2）产生固定长度的输出；

1. 高效可计算的，即计算时间在合理范围内。

哈希函数的三个性质：

* collision resistance
* hiding
* puzzle friendly

1.2 两种加密体系

通信的加密体系可以分为两种：对称加密体系和非对称加密体系。

对称加密体系是指加密和解码都是使用同一个密钥的算法，成为对称加密。通信的过程是：甲方选择某一种加密规则对信息明文进行加密，而乙方使用同一种规则进行解密。

非对称加密体系中加密和解密使用的是不同的密钥。通信过程是：通信的乙方生成两把密钥(公钥和私钥)。公钥是公开的，私钥是保密的。之后，甲方获取到乙方的公钥对明文进行加密，乙方收到加密后的消息后用自己的私钥进行解密。

在比特币系统中，我们使用的是非对称加密体系。每一个BTC账户都有一对公私钥，BTC开户的过程就是在本地创立一对公钥和私钥的过程。公钥就相当于用户的银行账户，转账只需要知道公钥就可以了。私钥相当于账户密码，知道了私钥就可以将账户上的钱转走。

1.3 数字签名过程

发送报文时，发送方用一个哈希函数从报文文本中生成报文摘要，然后用发送方的私钥对这个摘要进行加密，这个加密后的摘要将作为报文的数字签名和报文一起发送给接收方，接收方首先用与发送方一样的哈希函数从接收到的原始报文中计算出报文摘要，接着再公钥来对报文附加的数字签名进行解密，如果这两个摘要相同、那么接收方就能确认该报文是发送方的。

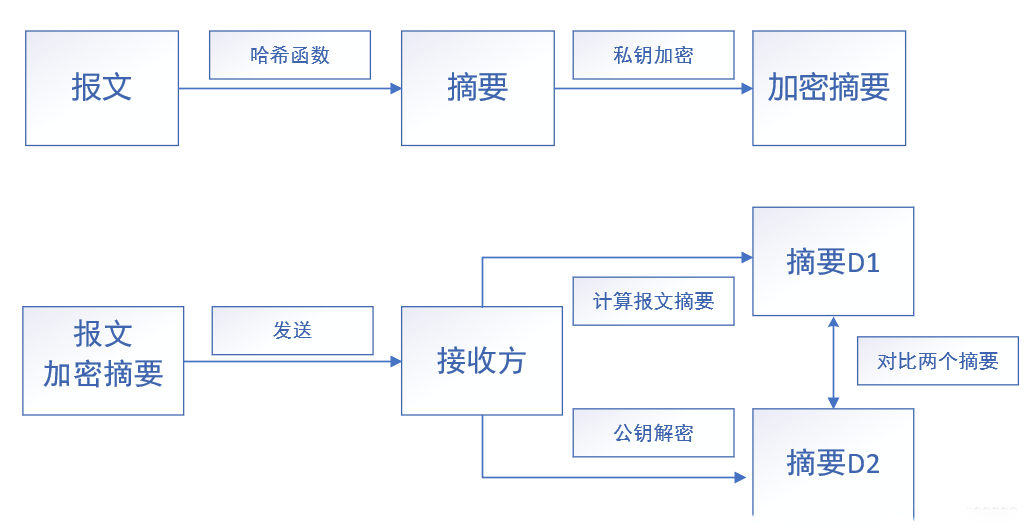


图1.1 数字签名实现过程

数字签名的作用：

* 确定消息的不可抵赖性：因为他人假冒不了发送方的私钥签名。发送方是用自己的私钥对信息进行加密的，只有使用发送方的公钥才能解密。
* 数字签名能保障消息的完整性：一次数字签名采用一个特定的哈希函数，它对不同文件产生的数字摘要的值也是不相同的。如果在传输中途 报文+加密摘要 的组合消息被修改了，那么接收方最终计算出的两个摘要D1和D2必定是不相等的。

# 第2章 比特币中的数据结构

2.1 哈希指针

1）普通指针P：存储某个结构体在内存中的地址(起始位置)。

2）哈希指针H()：既存储某个结构体在内存中的地址，也存储对结构体内容的哈希值。好处在于不仅可以找到结构体的位置，还可以检测结构体中的内容有没有被篡改。

3）区块链与普通链表的区别：使用哈希指针代替了普通的指针。换句话说，区块链就是用哈希指针代替普通指针的链表。

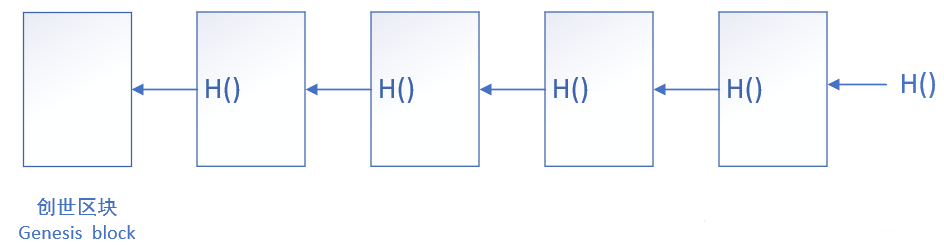


图2.1 比特币区块链的链式结构

这种结构的作用：可以建立一个防篡改日志(tamper-evident log)。系统的多数节点只需要记录最后一个区块(Most recent bolck)的哈希值，前面任意一个区块被篡改都能被检测出来，因为修改了某个区块的内容，会导致它与后面区块保存的哈希值对不上了，那个后面区块的哈希值就也要改，然后再往后的就也要改。类似于多米诺骨牌效应，前面的修改一定会通过哈希指针传递到最后一个区块。

2.2 Merkle tree(默克尔树)

Merkle tree本质上是用哈希指针代替普通指针的二叉树。

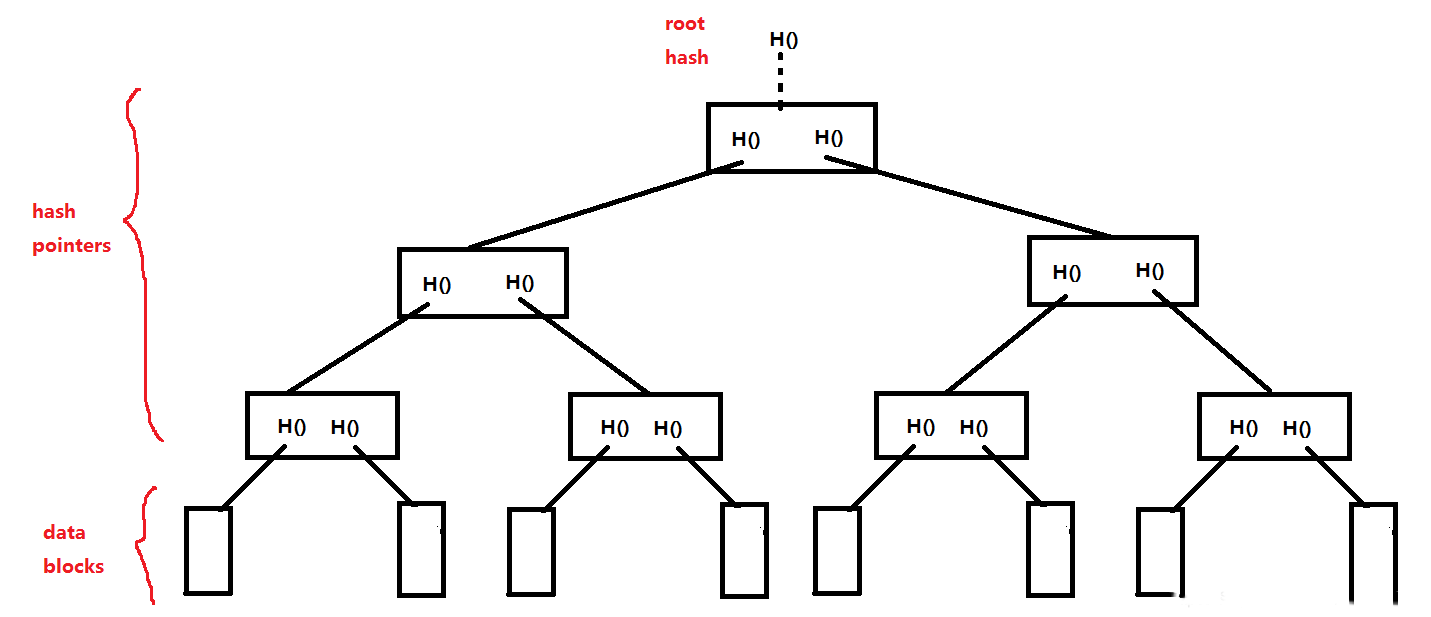


图2.2 默克尔树的结构图

Merkle Tree和普通二叉树的区别就是用哈希指针代替普通指针，最底下的一层叶子结点是数据块，其上的若干层非叶子结点都是存储哈希指针的。一个非叶子结点中的两个哈希值拼在一起，再取一个哈希值，也就是指向该结点的父结点存储的哈希指针的哈希值。对根节点也可以取一个哈希值，称为根哈希值（root hash）。

和区块链一样，在Merkle Tree中，只要记录下根哈希值，就能检测出对树中任何部位的修改，也就是用根哈希值保护了整棵树上没有篡改。

第3章 比特币的协议

3.1 比特币区块的具体介绍

比特币区块链连起来的是块头，块身挂在区块上，哈希指针和块身没有直接联系（间接联系就是通过Merkle Tree的根哈希建立的）。

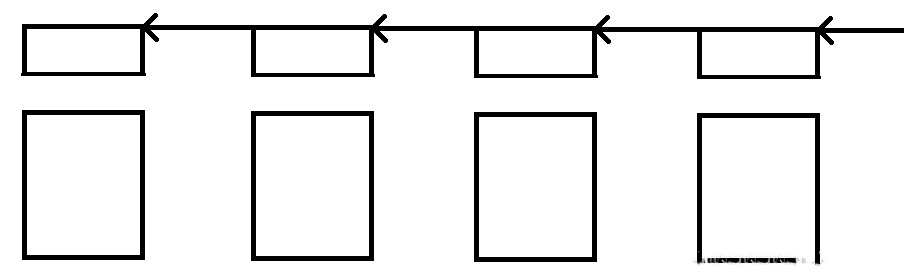


图3.1 比特币区块链

* 区块头：块头里保存的是区块的宏观的信息：

① 用的是比特币的哪个版本的协议 version

② 指向前一个区块块头的哈希指针

③ 整棵Merkle Tree的根哈希值

④ 挖矿的难度目标阈值target

⑤ 挖矿用的随机数nonce，要使得

* 区块体：包含交易列表，是一颗Merkle Tree.

3.2 比特币的共识协议—PoW

每个账户都可以发布交易，区块链可以看做账本，那么发布的交易应该写在哪个区块里呢？交易广播给每个区块，每个人都在自己本地的区块链上写入交易，如何保证写入后的一致性？也就是说账本的内容要取得分布式的共识（distributed consensus）。

比特币系统中的共识要考虑到有些结点是有恶意的，假设系统中大部分结点是友好的，有恶意的结点占少数。

比特币系统中不是用账户来投票，而是用计算力来投票，每个结点都可以在本地组装出一个候选区块，把它认为合法的区块放在这个区块里，然后就开始尝试各种nonce值（4 byte），使得H(block header)≤target。如果某个结点找到了符合要求的nonce，也就获得了记账权——往比特币去中心化的账本（区块链）里写入下一个区块的权力，其它结点收到这个区块之后，要验证这个区块的合法性（如检查target的编码nBits域设置的是不是符合比特币协议规定的难度要求、检查带nonce的块头哈希值是不是小于target、检查块身中的每个交易是否都有合法的签名、检查每个交易都没有双花等）。

3.3 PoW的实现—挖矿机制

PoW就是要求全节点不断尝试区块块头中的nonce和extra nonce的值，使得。这个过程我们称为挖矿。显然目标阈值target越小，则挖矿的难度就越大。所以调整挖矿难度就是在调整target，以调整目标空间在整个输出空间中所占的比例。

比特币中使用的哈希函数是SHA-256，产生的哈希值是256位的，所以整个输出空间是，调整目标空间所占的比例，在这个问题里直观的来看就是最后得到的哈希值前面有多少位0（这只是通俗直观来看的），这个0越多显然值就越小，也就是挖矿难度越大了。挖矿难度和目标阈值的大小成反比，具体可见以下公式：



上式中常量difficulty\_1\_target是指挖矿难度difficulty=1时所对应的目标阈值target的值。挖矿难度最小就是1，所以这个常量也就是target允许的最大值。

系统中的总算力越来越强，如果挖矿难度保持不变，那么平均出块时间会越来越短，这会造成一些问题。假设平均出块时间减小到了1秒钟，也就是每隔1秒左右就有一个新的区块携带一些列交易被发布到比特币网络上，而在比特币网络上这个区块传播给大多数结点可能就要几十秒。如果有两个结点几乎同时发布了区块，那么就会出现分叉：

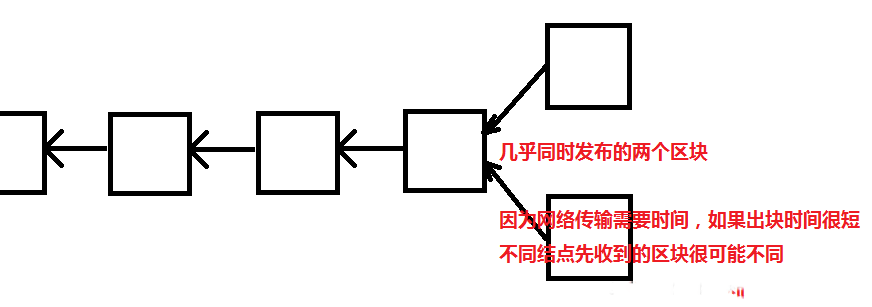


图3.2 区块链因挖矿难度过低导致的分叉

分叉过多对比特币系统达成共识没有好处，并且会危害比特币系统的安全性。正常情况下，因为大部分结点是诚实的，有恶意的结点想要在6个确认后拿这段时间集中算力算出的新链去覆盖掉最长合法链是很难的，因为诚实结点也都在集中算力扩展最长合法链。如果出块时间很短，就会导致分叉过多（因为相比于出块时间，可以认为网络上传输的时间变长了），这样诚实结点的算力就被分散了，这时恶意结点要进行51% attack很可能就不需要50%以上的算力了，可能百分之十几就足够了，这样大大降低了比特币系统的安全性。

为了避免上述情况，比特币协议中规定，每隔2016个区块（大约每2个星期）要重新调整一下目标阈值target，具体的迭代更新公式是：



这里expected time就是预期的两次调整的间隔时间，即2016乘以10分钟(大约两周)；而actual time是系统中产生最近的2016个区块实际的花费时间。为了避免系统中出现某些意外情况，导致系统出现非常大的波动，每次对目标阈值target的调整最大不能超过4倍，最小不能小于1/4。也即上式中的actual time / expected time即便超过4了也按4使用，即便小于 1/4也只按1/4使用。