1. **哈希函数的三个条件---P1**

哈希函数的三个性质：

* 1. collision resistance：发生哈希碰撞是不可避免的，原因是哈希函数的输入空间是无限大的，而输出空间是有限的，以SHA-256为例，输出只有种可能性。如果当输入的可能性大于时，发生哈希碰撞就是必然的。

所谓的**collision resistance**是指我们**没有什么高效的方法去人为的制造哈希碰撞**，即知道一个y，没有特定的找到x的方法使得H(x)=H(y)。如果一定要找的话只能用brute-force的方式，即遍历输入空间不断计算哈希值，直到找到恰好碰撞的那个x。

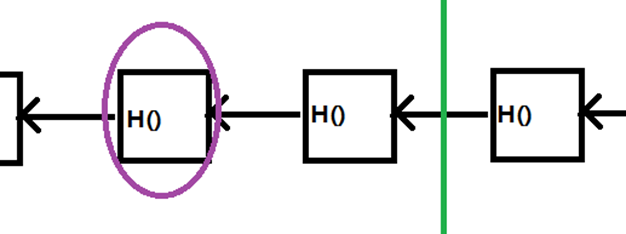
利用**collision resistance**可以用来对一个信息求digest，用来检测对这个信息的篡改。因为如果这个信息的内容被改掉了，那么哈希值必然就会改变，**没有专门的只修改信息内容而不改变其哈希值的手段**。

* 1. hiding(单向性)：**哈希函数的计算过程是单向的**，是不可逆的。即可以，但无法。这表示哈希值 H ( x )没有泄露有关输入x的任何信息。
  2. puzzle friendly: 我们想要计算出的哈希值落在某个范围内(就是说系统需要一个具有指定特征的哈希值)，那没有特定的办法去构造输入。比如，想要计算出的哈希值前面有k个连续的0，也没有特定的办法去构造输入。甚至无法知道什么样的输入更有可能得到具有特定特征的哈希值，要想得到这种输入就只能一个一个去试，没有捷径。

1. **数字签名 –P2**
2. 对称加密体系的缺点：通信双方必须事先商议好加密规则，但这一过程如果在网络传输很容易被第三方窃取，如果线下的话又会导致不便利。
3. 数字签名的作用：
   1. 防止第三方伪造发送方身份发送虚假消息
   2. 防止第三方获取报文并修改后再发送给接收方
4. **数据结构—链式结构—P3**

与普通的链式结构不同，区块链采用的不是一般的指针，而是哈希指针。

比如只保存了下图中绿线右侧的区块，这些是比较新产生的区块，然后如果要用到紫色圈圈出来的区块，可以向别人要。保存的哈希指针在这里可以校验给的区块是不是正确的



1. **数据结构–Merkle tree(默克尔树)**

比特币区块的块头一般会存储有关于这个区块的重要信息：比如区块产生时间、难度阈值、区块序号、前一区块的哈希值，还有就是默克尔树的根哈希值。

比特币的块身就是存储区块中所有的交易TX。

为什么要把区块分成块头和块身两部分？原因也很简单，因为如果这样混在一起的话，对于轻结点(轻结点就是用户手机中的比特币钱包)来说，存储负担就比较大，具体大到什么什么程度呢？大概是有1000倍容量差距。

那么默克尔树到底有什么神奇之处呢？我们可以简单介绍一下默克尔树的作用。~~

1. **交易过程与协议—比特币的交易**

比特币系统中，节点获取铸币权的方式就是通过挖矿，最初的时候没挖出一个区块可以获得50 BTC ，后面就是每4年就减半(其实是每21W个区块就减半)，也就是25 BTC ,到现在为止是每挖出一个区块奖励12.5 BTC 。我们可以利用收敛级数简单算一下比特币的总量：



最后的结果是2100W BTC，这就是整个比特币系统的BTC总数。

1. **系统实现—基于交易的账本模式**

比特币是基于交易的账本模式，还有一些系统是基于账户的模式(account-based ledger)，比如以太坊就是用状态树MPT显式的记录每个账户中有多少个币。

|  |
| --- |
| 交易费的具体作用：  仅仅为获得记账权的结点给予出块奖励是不够的，获得记账权的结点为什么要把某些交易记下来？这样做对他有什么好处呢？一个结点完全可以只打包自己的交易，记录别人的交易不仅要去验证其合法性，而且一个区块装的交易多了，在网络上传输的带宽也会比较多，在网络上传输的速度也会慢。这里的差额作为记账费就解决了这个给别人记账的动机的问题。 |

比特币系统的这种基于交易的模式，隐私保护性比较好，但会带来一些代价，如转账交易要说明币的来源(币是从之前的哪个交易的哪个输出中来的)以防止双花攻击。为此，比特币设计了另一种数据结构—UTXO，可以快速探明交易中币的来源。

|  |
| --- |
| 比特币系统中使用的脚本语言很简单，唯一能访问的内存空间就是一个栈，这点和通用脚本语言的区别很大。  合并输入输出脚本其实就是进行特定数据的入栈和出栈，只要结束时栈是空的就能说明脚本被正确执行了，那么交易就是合法的。 |

1. **挖矿机制—挖矿设备与趋势**

最早时候大家都是用普通计算机来挖矿，因为计算哈希值的操作只用到通用CPU中的很少一部分指令，其他资源都是闲置的，随着挖矿难度提高，用通用计算机上的CPU挖矿很快就无利可图了。

后来大家都采用GPU开始挖矿，但GPU主要用来做通用的大规模并行计算，用来挖矿还是会有不少浪费，而且GPU的噪音很大，其中很多部件还是浪费了（如用于浮点数计算的部件）。

最后，为了更进一步提高挖矿的性价比，专门研究了为了挖矿而设计的芯片，这些芯片没有多余的电路，干不了别的事。而且通常为某一种加密货币设计的ASIC芯片只能挖这一种加密货币的矿。

研制挖特定加密货币的ASIC芯片需要一定周期，但和研制通用芯片的速度相比已经是非常快的了，如研制比特币挖矿的ASIC芯片大约用一年的时间。不过加密货币的价格变化是比较剧烈的，曾经就发生过比特币价格在几个月内下跌80%，因为加密货币多变的价格，这些挖矿设备的研制风险也是很大的。

由于单个矿工挖矿的收益是很不稳定的，平均出块时间10分钟是对于比特币系统中的所有矿工而言的。一个矿工用一个矿机挖出矿的时间可能要很久，并且除了挖矿之外还要承担全结点的其它责任。

所有就有人开创矿池将很多矿工组织起来，一般的架构就是一个矿主（pool manager）全结点去驱动很多矿机，下属矿工只负责计算哈希值，全结点的其它职能只由矿主来承担。有了收益以后再大家一起分配。

这种大型矿池的出现就严重威胁了比特币设计的去中心化的初衷，只要某个大型矿池联合底下矿工发动51%攻击，就会导致整个比特币系统的安全机制失效

如今的矿池的算力还算比较分散，有好几家矿池在竞争，但一个集体的算力完全可以潜伏分散在不同矿池中，等到攻击时再集中起来，因为矿工要转换矿池是很容易的。