

区块链技术与应用

谢文进

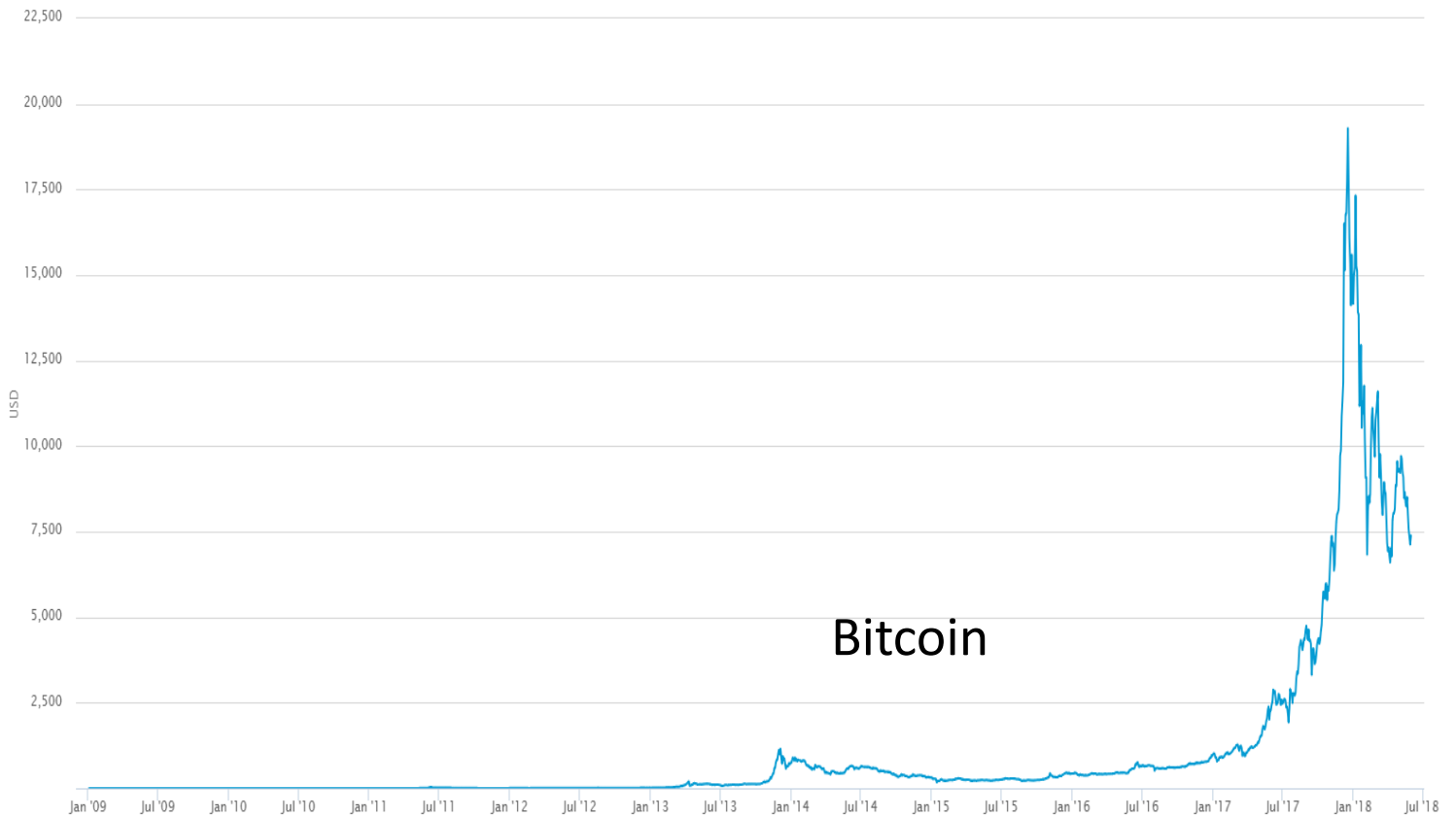
2023 年 4 月 11 日

1 课堂笔记

01-课程简介

Market Price (USD)

source: blockchain.info



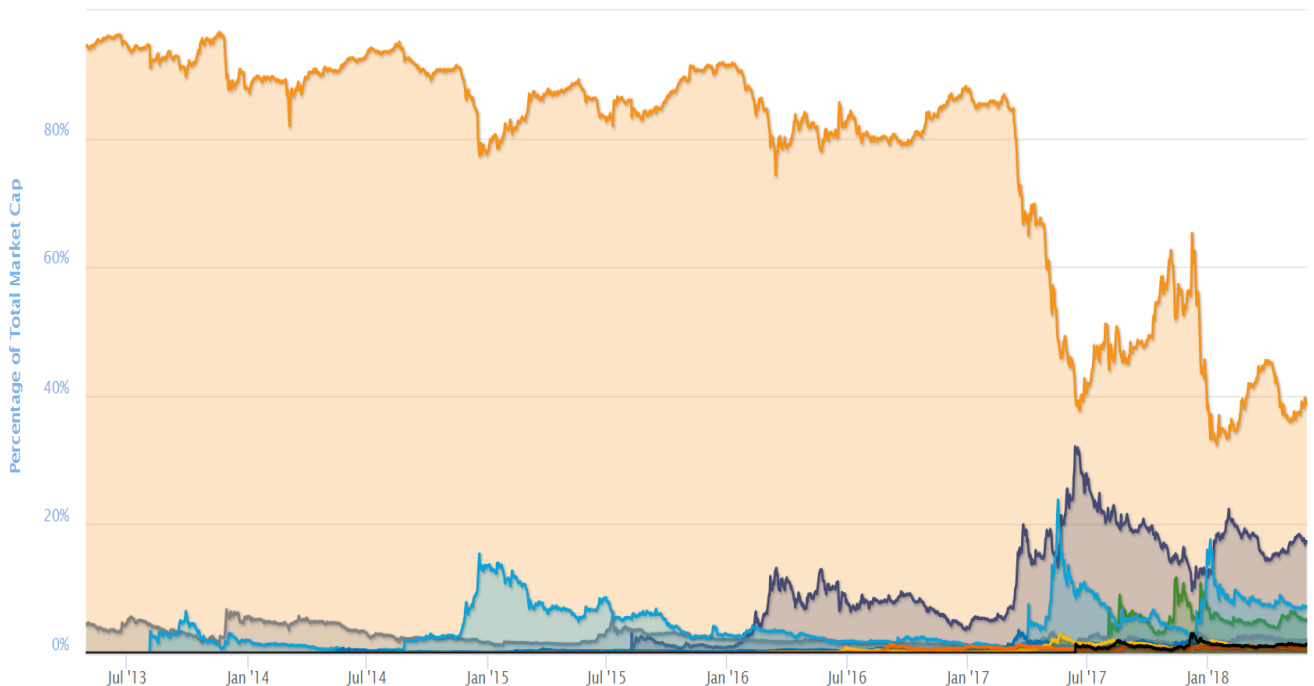
Percentage of Total Market Capitalization (Dominance)

Source: coinmarketcap.com

Overlapping Stacked

Zoom 1d 7d 1m 3m 1y YTD ALL

From Apr 29, 2013 To Jun 1, 2018



Bitcoin Ethereum Bitcoin Cash Litecoin Ripple Dash NEM Monero IOTA NEO Others

课程基本信息

- 假设已在本科阶段学过基本的数据结构和算法，掌握编程的基本技能
 - 数组、链表、二叉树、哈希函数
- 参考资料
 - BitCoin and Cryptocurrency Technologies
A Comprehensive Introduction
 - 以太坊白皮书、黄皮书、源代码
 - Solidity文档
- 公开课形式：全程录像

课程大纲：比特币

- 比特币
 - 密码学基础
 - 比特币的数据结构
 - 共识协议和系统实现
 - 挖矿算法和难度调整
 - 比特币脚本
 - 软分叉和硬分叉
 - 匿名和隐私保护

课程大纲：以太坊

- 以太坊
 - 概述：基于帐户的分布式账本
 - 数据结构：状态树、交易树、收据树
 - GHOST协议
 - 挖矿：memory-hard mining puzzle
 - 挖矿难度调整
 - 权益证明
 - Casper the Friendly Finality Gadget(FFG)
 - 智能合约
- 总结与展望

02-BTC-密码学原理

加密货币 (Crypto-currency) 中主要用到密码学中的哈希函数和签名。

1. 哈希函数

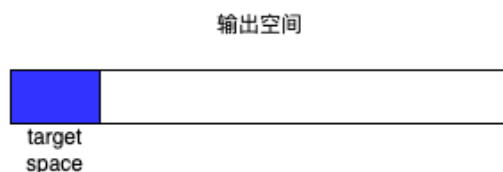
哈希函数 (hash function) 有两个性质:

- **collision resistance (or collision free)**: 不能在多项式时间内找到 $x \neq y$, 使得 $H(x) = H(y)$ 。这条性质说明没有办法篡改内容而又不被检测出来。
- **hiding**: $x \rightarrow H(x)$ 是单向的, 不可逆的。这里有个前提是要求输入空间大, 取值分布均匀。

没有哪个哈希函数在数学上证明是 collision resistance 的, 但是可以找到哈希碰撞的方法, 例如 MD5 就被攻破了。

可以用哈希函数的 hiding 性质做 digital commitment, 也就是 digital equivalent of a sealed envelope。比如预测股市, 先将结果放在信封里, 不能提前公布预测结果, 因为预测的结果可能会影响股市, 接着将信封交给第三方保证不被篡改, 等到开盘再打开验证。如果使用哈希函数, 可以先公布哈希值 $H(x)$, 等到要验证时, 再拿出之前写好的 x 。哈希函数 hiding 性质的前提是输入要足够大, 分布均匀, 如果输入不够大, 可以在 x 后面拼接随机数 $H(x||nonce)$ 。

bitcoin 中还要求哈希函数有 **puzzle friendly** 性质。由于哈希值的计算事先是不可预测的, 可以设置一个 puzzle, 比如要求计算出的哈希值前 k 个都为 0, 形如 $00 \cdots 0xx \cdots x$ 。这其实就是挖矿, 找一个随机数 nonce, 使得 $H(\text{block header}) \leq \text{target}$, block header 中含有可调节的 nonce。



输出空间很大, 但是目标空间很小, 这只能一个一个试 nonce 的值。这也是工作量证明 (proof of work)。虽然挖矿很难, 但是验证比较容易 (difficult to solve, but easy to verify)。

2. 签名

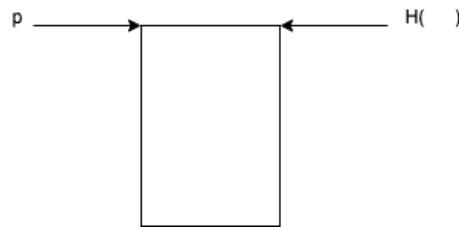
在比特币中开户其实就是创建一个**公私钥对** (public key, private key)。公钥相当于银行账户, 私钥相当于密码。在进行交易时, 用私钥进行签名, 说明是本人进行的, 发布交易时也要发送公钥, 可供他人进行验证。

在这个过程中, 产生公私钥对和签名时需要一个好的随机源 (a good source of randomness)。

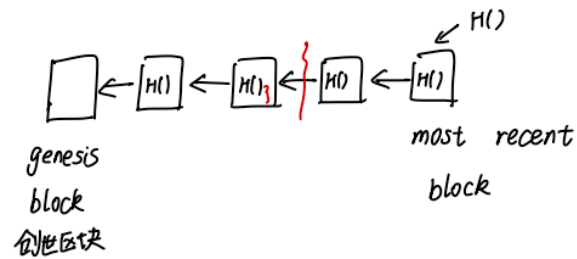
03-BTC-数据结构

哈希指针

hash pointer



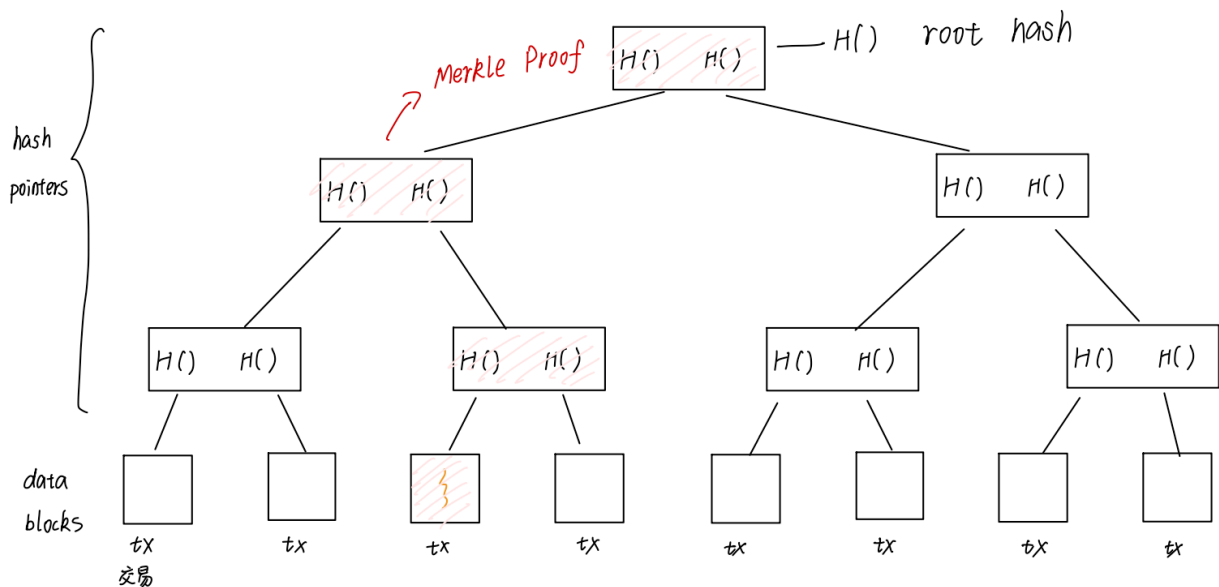
Block chain is a linked list using hash pointer.



实现 **tamper-evident log**，只要保存最后一个值，就知道前面有没有修改。

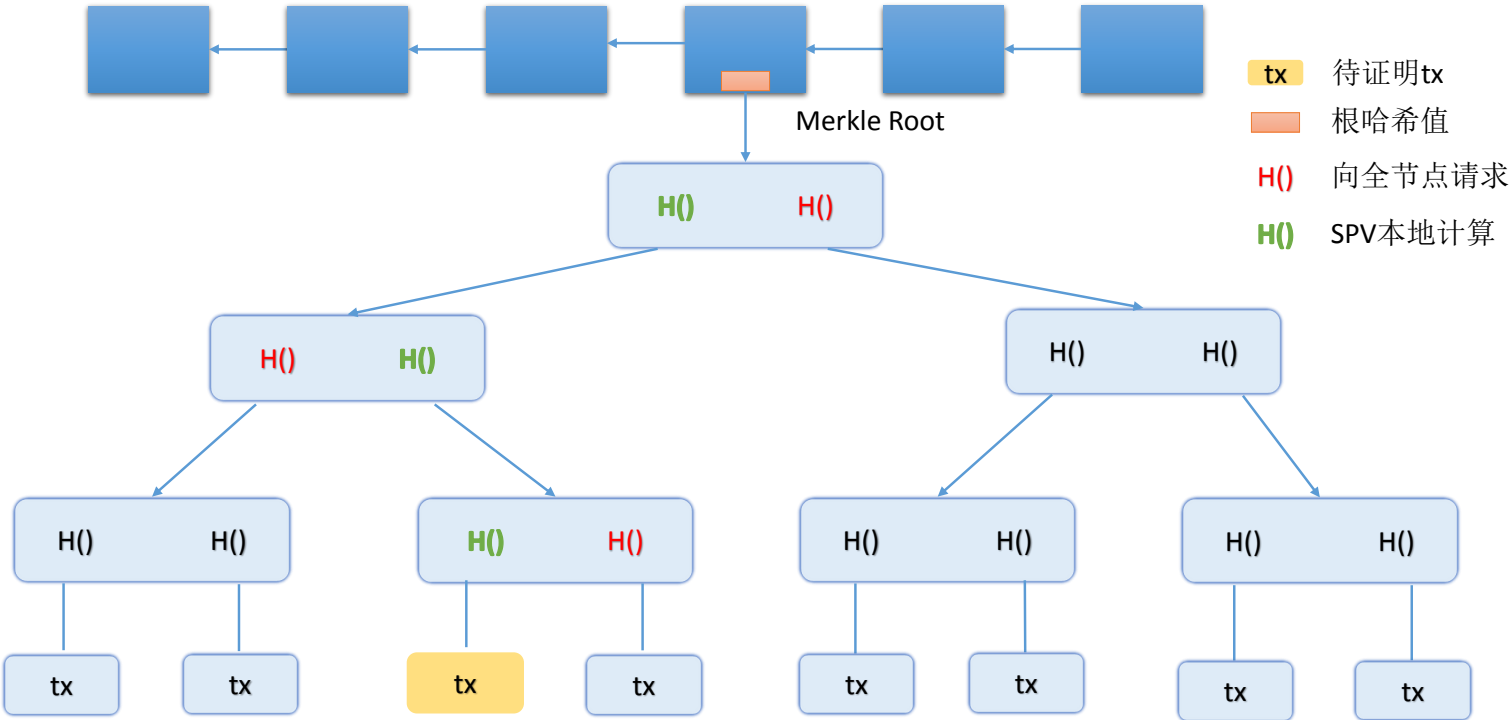
Merkle tree

Merkle tree 是用 Hash 指针代替普通指针的二叉树。



- block header (有 root hash)
- block body (有交易列表)

Merkle tree 的作用是可以提供 Merkle proof，证明包含某种交易 (**proof of membership / proof of inclusion**)，是 $O(\log(n))$ 的时间复杂度。但是如果证明某个交易不包含在 Merkle tree 中，如果进行所有叶子节点的遍历，时间复杂度是 $O(n)$ ，这时可以考虑采用 sorted Merkle tree，时间复杂度降为 $O(\log(n))$ 。



数据结构无环可以用 hash pointer。如果有环会出现冲突。

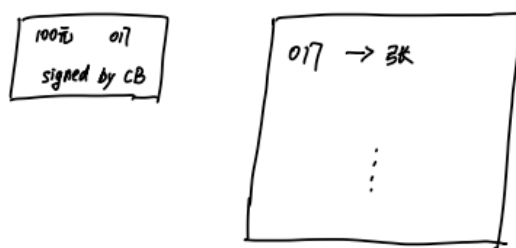


04-BTC-协议

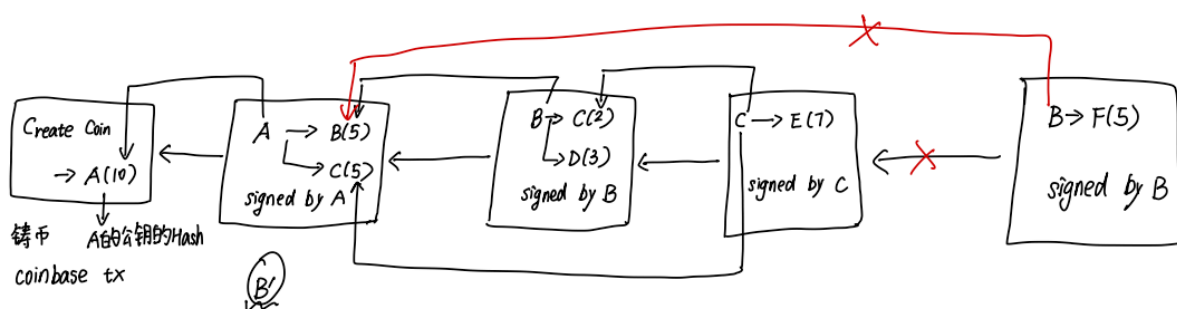
如果中央银行要发售电子货币，每张纸币可以由中央银行签名。



这会出现双花攻击 (double spending attack)，因为这些电子货币其实就是代码，可以进行复制，我花出去一张 100 的，我可以复制很多张，继续花费。可以考虑维护一个数据库，给发行的每张货币一个编号，然后记录该货币属于谁，不过这样维护数据库就很麻烦了，而且也并不是去中心化的。



比特币中的交易：

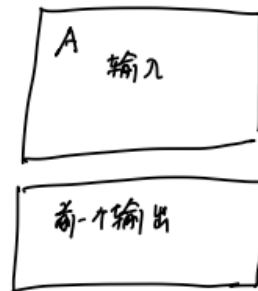


A 要给 B 转 5 个比特币，A 需要知道 B 的地址。B 也需要知道 A 的公钥，为了验证 A 的签名。在上图中，在交易时不仅要说明转账的地址，也要说明币的来源。在图中第二个区块中

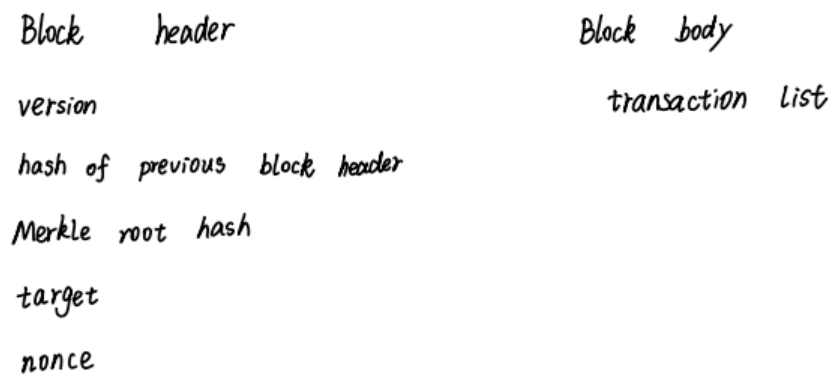
- 输入：币的来源、A 的公钥
- 输出：B 的公钥的哈希

这里就能避免双花，例如图中最后 B 想给 F 转 5 个比特币，可以看到 B 的 5 个比特已经花出去了，经过检查是不能再使用的。

脚本验证 (Bitcoin Script): 将输入与前一个输出拼接在一起，看是否能正确运行。



每个块可以有很多交易，组织成 Merkle tree。



挖矿: $H(\text{block header}) \leq \text{target}$. 区块链的另一种画法，因为其实哈希指针计算的是前一个 block header 中的哈希。



- full node: fully validating node, 全节点记录所有信息
- light node: 轻节点

账本的内容要取得分布式的共识 (distributed consensus)。

FLP impossibility result: 在异步系统中，网络传输时延没有上限，哪怕系统中只有一个成员是 faulty，也没法达成共识。

CAP Theorem: 任何一个分布式系统，以下 3 个性质中最多满足 2 个。

- C: Consistency 一致性

- A: Availability
- P: Partition tolerance

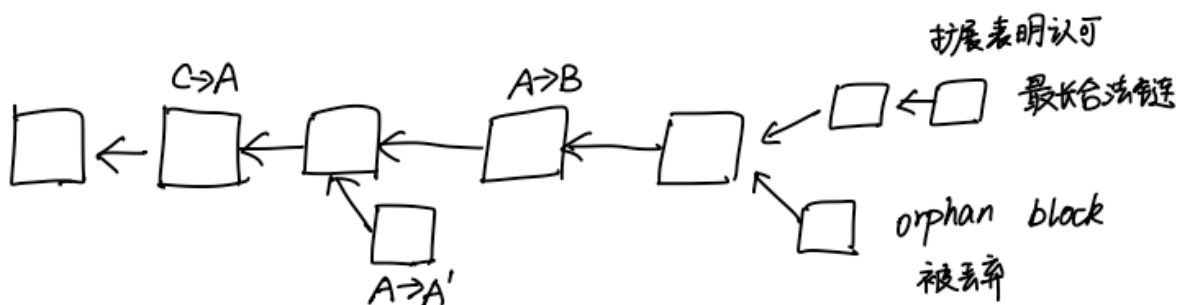
Paxos 协议满足 Consistency 性质。

Consensus in BitCoin

重要的问题是：成员 (membership) 中谁拥有投票权。联盟链 (hyperledger fabric) 是只有加入链的成员有投票权。

女巫攻击 (sybil attack)：攻击方产生超过一半的用户。

如果节点能解决 puzzle 问题，可以获得记账权，有权利发布下一个节点。



longest valid chain: 最长合法链

forking attack: 分叉攻击

block reward: 出块奖励

mining: 挖矿

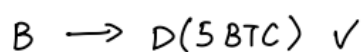
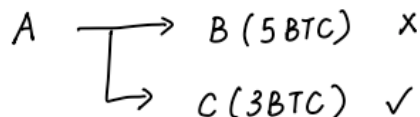
miner: 矿工

coinbase transaction 是产生币的唯一方法。每隔 21 万个区块，区块奖励减半，50BTC→25BTC→12.5BTC.

05-BTC-实现

比特币是基于交易记录的模式 (transaction-based ledger)。

UTXO: Unspent Transaction Output 还未花掉的交易的输出，为了检测双花，包含产生交易的哈希值以及在交易中的第几个。



B 的 5BTC 已经花出去了，因此不在 UTXO 中。

total inputs = total outputs 所有输入 = 所有输出，但有时候不完全相等，例如所有输入为 1BTC，所有输出为 0.99BTC，这其中的差值是交易费 (transaction fee)。比特币中大约每 4 年区块奖励会减半，因此到后期的奖励可能大部分来自于交易费。

$$\frac{21 \text{ 万区块} \times 10 \text{ 分钟}}{60 \text{ 分钟} \times 24 \text{ 小时} \times 365 \text{ 天}} \approx 4 \text{ 年}$$

还有一种模式是基于账户的模式 (**account-based ledger**), 例如以太坊。

Block Example

Block #529709


Summary	
Number Of Transactions	686
Output Total	4,220.46616378 BTC
Estimated Transaction Volume	651.93844862 BTC
Transaction Fees	0.12458867 BTC
Height	529709 (Main Chain)
Timestamp	2018-06-29 06:17:26
Received Time	2018-06-29 06:17:26
Relayed By	BTC.com
Difficulty	5,077,499,034,879.02
Bits	389508950
Size	333.53 kB
Weight	1160.618 kWU
Version	0x20000000
Nonce	3897564446
Block Reward	12.5 BTC

Hashes	
Hash	<u>0000000000000000000030f1531dbfa069037188c5048b23f0cb979ce8728ed8c5</u>
Previous Block	<u>0000000000000000000000841c59a4679d8e707152724f5b195b66b47397d65175e</u>
Next Block(s)	
Merkle Root	6f73d36264f05e0c55c4703f858e56a282931e9673bcefc302b676dc2b378

source: blockchain.info



Block header

```
13  /** Nodes collect new transactions into a block, hash them into a hash tree,
14   * and scan through nonce values to make the block's hash satisfy proof-of-work
15   * requirements. When they solve the proof-of-work, they broadcast the block
16   * to everyone and the block is added to the block chain. The first transaction
17   * in the block is a special one that creates a new coin owned by the creator
18   * of the block.
19   */
20  class CBlockHeader
21  {
22  public:
23      // header
24      int32_t nVersion;
25      uint256 hashPrevBlock;
26      uint256 hashMerkleRoot;
27      uint32_t nTime;
28      uint32_t nBits;
29       uint32_t nNonce; 232个可能
```

source: bitcoin/src/primitives/block.h



Block headers are serialized in the 80-byte format described below and then hashed as part of Bitcoin's proof-of-work algorithm, making the serialized header format part of the consensus rules.

Bytes	Name	Data Type	Description
4	version	int32_t	The block version number indicates which set of block validation rules to follow. See the list of block versions below.
32	previous block header hash	char[32]	A SHA256(SHA256()) hash in internal byte order of the previous block's header. This ensures no previous block can be changed without also changing this block's header.
⇒ 32	merkle root hash	char[32]	A SHA256(SHA256()) hash in internal byte order . The merkle root is derived from the hashes of all transactions included in this block, ensuring that none of those transactions can be modified without modifying the header. See the merkle trees section below.
4	<u>time</u> 区块产生时间	uint32_t	The block time is a Unix epoch time when the miner started hashing the header (according to the miner). Must be strictly greater than the median time of the previous 11 blocks. Full nodes will not accept blocks with headers more than two hours in the future according to their clock.
4	<u>nBits</u> 目标值	uint32_t	An encoded version of the target threshold this block's header hash must be less than or equal to. See the nBits format described below.
⇒ 4	<u>nonce</u>	uint32_t	An arbitrary number miners change to modify the header hash in order to produce a hash less than or equal to the target threshold . If all 32-bit values are tested, the time can be updated or the coinbase transaction can be changed and the merkle root updated.

The hashes are in **internal byte order**; the other values are all in little-endian order.
source: bitcoin.org



Transaction View information about a bitcoin transaction

99ffa15c268a5aee6a0a0381ec266654fa283fe7cdd7ced583c57738e8dadd3

No Inputs (Newly Generated Coins)



1C1mCxRukix1KfegAY5zQQJV7samAciZpv - (Unspent)
Unable to decode output address - (Unspent)

12.62458867 BTC
0 BTC

1 Confirmations

12.62458867 BTC

Summary

Size 243 (bytes)

Weight 864

Received Time 2018-06-29 06:17:26

Reward From Block 529709

Scripts [Hide scripts & coinbase](#)

Visualize [View Tree Chart](#)

CoinBase 可写Hash值, 可写别的

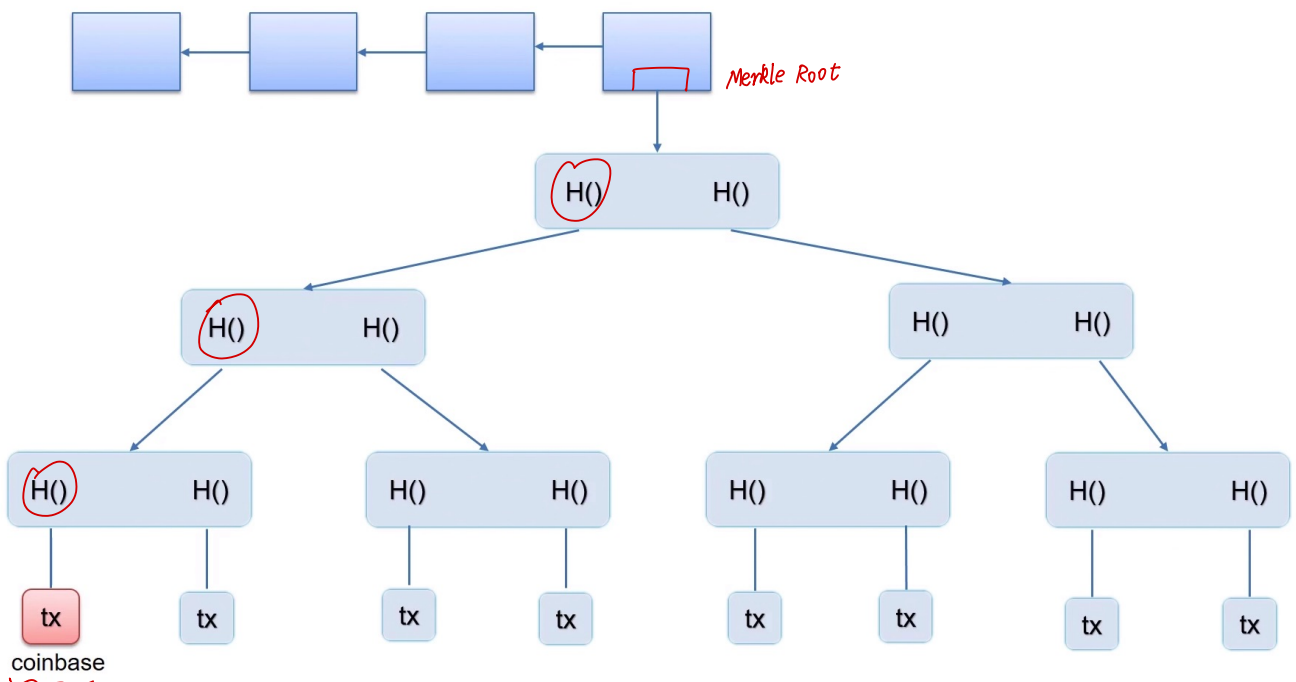
032d150804f6ce355b672f4254432e434f4d2fabe6d6d1dcbc17dbdbcb4e98221c0b5f06459f705df3854da8e496d3175096abe84babe0100000000000008566bf194da9010000000000
(decoded) 5[g/BTC.COM/5mm]8Tm1u jM

Output Scripts

DUP HASH160 PUSHDATA(20)[78ce48f88c94df3762da89dc8498205373a8ce6f] EQUALVERIFY CHECKSIG

RETURN PUSHDATA(36)[aa21a9ede6643c66fadd737364679ab3c22e03fd44264643ae6692287e54cdcb71b1a0ca]
(decoded) d

source: blockchain.info



⇒ 2⁹⁶

Transaction View information about a bitcoin transaction

3f0d94dc733a614114a930a470241e0d99ea6966f99f1fa6f895396a6645f		
14BHEP2sNRUj5jSxgBjyszza87psR2Uv (0.00408688 BTC - Output)	1Kqj3Qiqbd1ah9G9imm8comNDKgxVvLVF - (Unspent)	0.0396 BTC
1K7q5sqzbT1tdd6MtZDhb9E4jJCShiUcR (0.04758063 BTC - Output)	17Eu6pyMUJZHoaEKj3u8k2D3izdw9QYRT - (Unspent)	0.01019031 BTC
1 Confirmations		0.04979031 BTC

Summary		Inputs and Outputs	
Size	373 (bytes)	Total Input	0.05166751 BTC
Weight	1492	Total Output	0.04979031 BTC
Received Time	2018-06-29 06:13:26	Fees	0.0018772 BTC
Lock Time	Block: 529708	Fee per byte	503.271 sat/B
Included In Blocks	529709 (2018-06-29 06:17:26 + 4 minutes)	Fee per weight unit	125.818 sat/WU
Confirmations	1 Confirmations	Estimated BTC Transacted	0.0396 BTC
Visualize	View Tree Chart	Scripts	Hide scripts & coinbase

配对

Input Scripts 输入

ScriptSig: PUSHDATA(71) [304402200e902feaf8e49ea467bbc3d9034bdf33fedfbfb662e75e8822372a3c0871e102204176d40c1781ca6f465d47db3628e2610f53d5a54ceb24e6118296ae71df5e5201] PUSHDATA(33)[03b339dc9b56131ad95753f69958086fcc2c686bad4f69ea0b9bc9e564315c1e515]
ScriptSig: PUSHDATA(72) [3045022100eade6baa42d09d279033581a593340780181aa0dd8a7fd2d5b6162acd81ca4d022074bd1a62c6138505823efae9ec62d4dd16e57c454a245be25f961a6e8144930201] PUSHDATA(33)[02ceddac2ca907c010dfea74dc3c4bc27a17dca0a2e8e8993cdccc243c818279ed]

Output Scripts 输出

DUP HASH160 PUSHDATA(20)[cea9fb4638839ad9ebc9c8382dd03e266aaeb65] EQUALVERIFY CHECKSIG
DUP HASH160 PUSHDATA(20)[4471a74ad7287cbbf6e6d1c092b22b55c886c1de] EQUALVERIFY CHECKSIG

source: blockchain.info



Hash puzzles

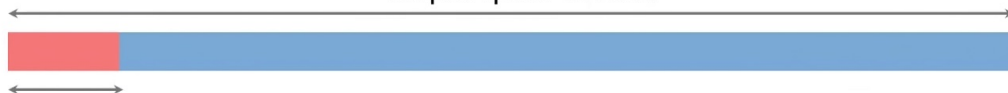
To create block, find nonce s.t.
 $H(\text{nonce} \parallel \text{prev_hash} \parallel \text{tx} \parallel \dots \parallel \text{tx})$ is very small

没有tx信息

只用到Block Header

nonce
prev_h
Tx
Tx

Output space of hash



Target space

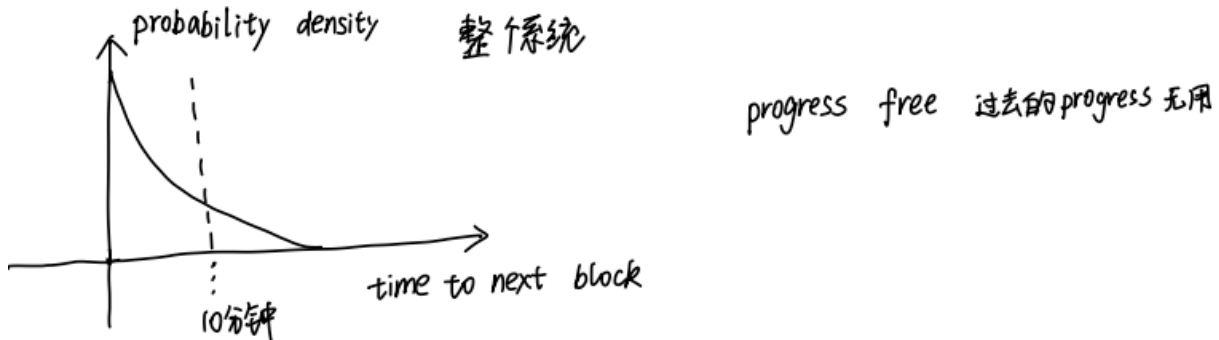
If hash function is secure:
only way to succeed is to try enough nonces until you get lucky

这是比特币教材配套的一页PPT。大家看看有问题吗？



每次挖矿尝试看作是 **Bernoulli trial**: a random experiment with binary outcome. Bernoulli process: a sequence of independent Bernoulli trials. Bernoulli process 具有无记忆性 (**memoryless**)。大量的 Bernoulli process 可用 Poisson process 近似。

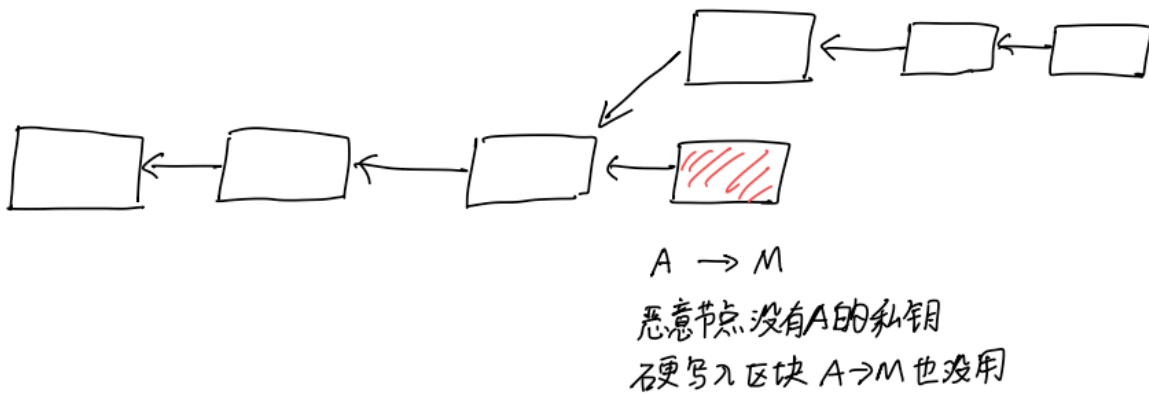
出块时间服从指数分布 (exponential distribution), 也具有无记忆性。



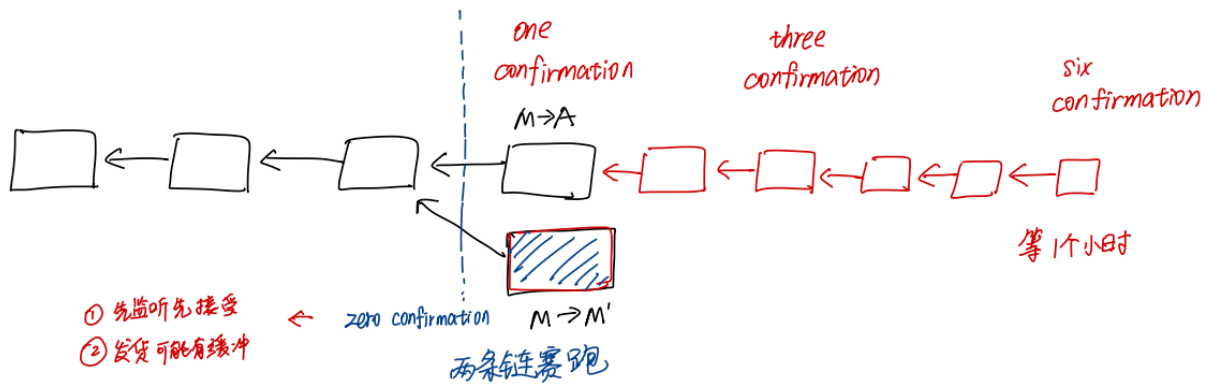
产生比特币数量构成几何序列 (geometric series)

$$\begin{aligned}
 & 21万 \times 50 + 21万 \times 25 + 21万 \times 12.5 + \dots \\
 & = 21万 \times 50 \times \left(1 + \frac{1}{2} + \frac{1}{4} + \dots \right) = 2100万 \\
 & \qquad \qquad \qquad \underbrace{\qquad \qquad \qquad}_{\frac{1}{1-\frac{1}{2}}=2}
 \end{aligned}$$

BitCoin is secured by mining.



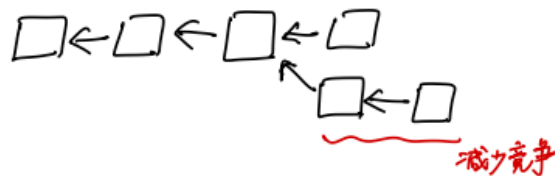
double spending



防范: 多等几个区块/确认 confirmation

区块链是不可篡改的账本 irrevocable ledger

selfish mining 的分叉攻击, 挖一长串, 然后一下发布, 这么做是有风险的。



06-BTC-网络

上层是 application layer: BitCoin Block chain. 下层是 network layer: P2P Overlay Network. 比特币中各节点是对等的。

比特币网络设计原则: simple, robust(鲁棒), but not efficient.

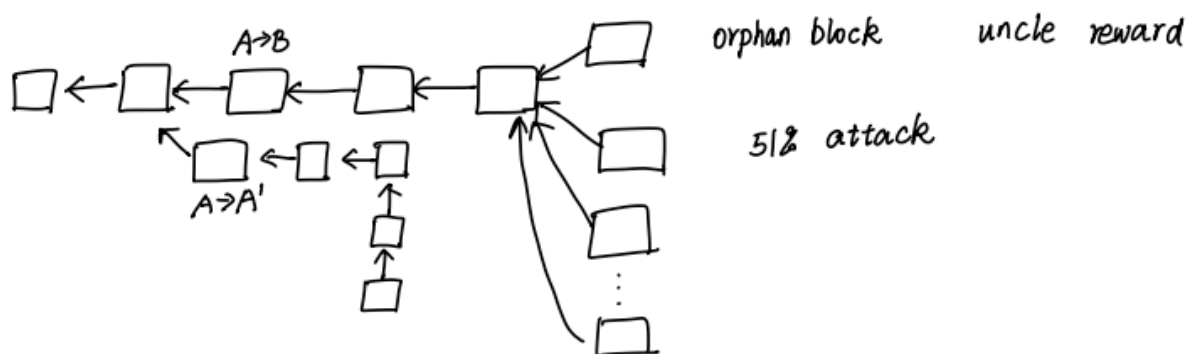
flooding, 邻居节点的选取是随机的, 因此两个节点可能地理上相隔很远, 网络传输慢, 并不是非常有效。全节点维护一个等待上链的集合, 第一次听到转发, 会在该集合中删除该交易, 后续不转发。区块大小限制在 1M。网络传播是尽力交付 (best effort)。

07-BTC-挖矿难度

$H(\text{block header}) \leq \text{target}$, 挖矿难度

$$\text{difficulty} = \frac{\text{difficulty_1_target}}{\text{target}}$$

难度 1, target 更大。



出块时间越短，出现分叉的可能更多。

以太坊协议 ghost，平均出块时间需要保持稳定。

调整挖矿难度：

$$\text{target} = \text{target} \times \frac{\text{actual time}}{\text{expected time}}$$

其中，actual time 是最近挖出 2016 区块的时间，期望时间 expected time = 2016 × 10，理想的挖出 2016 个区块的时间是两周。难度最大增长 4 倍，最小也是缩小 4 倍。

08-BTC-挖矿

09-BTC-比特币脚本

10-BTC-分叉