# Bitcoin Basics

-- Bitcoin Protocol and Consensus

**LING Zong, Ph. D.**
**Senior Software Engineer / Scientist**
**IBM Almaden Research Center**
**San Jose, California, U.S.A.**

# Overview

- **Bitcoin Concepts**
- **Consensus Build-up**
- **Mining Overview**
- **Cryptocurrency Mining**

# Basic Concepts - What is Bitcoin?

- **Cryptocurrency: "A digital currency in which encryption techniques are used to regulate the generation of units of currency and verify the transfer of funds, operating independently of a central bank."**
  - Built upon a combination of computer science, cryptography, and economics
- **Bitcoin is a cryptocurrency**
  - “Bitcoin” can refer to:
    - Bitcoin (uppercase) - the protocol, software, and community
    - bitcoins (conventionally lowercase) - the unit
- **Community terminology**
  - **"crypto"** - cryptocurrencies, Ethereum
  - **"private blockchain"** - private blockchains, permissioned ledgers, large financial institutions
  - "**blockchain**" - umbrella term

加密货币:一种数字货币，使用加密技术来规范货币单位的产生和验证资金的转移，独立于中央银行运行。
〇计算机科学、密码学和经济学的结合
比特币是一种加密货币
〇比特币可以指:
比特币(大写)-协议、软件和社区
比特币(通常小写)-单位
社区的术语
〇加密货币，以太坊
〇“私有区块链”——私有区块链、授权账本、大型金融机构
〇"区块链"——总括性术语

# Implications of blockchain technology

Altcoins (Dash, Dogecoin, Litecoin)
比特币2.0 /以太坊-在金融领域之外应用区块链
汇款——绕开传统的银行基础设施
寄5美分的钱到世界上任何一个地方
做你自己的银行 - 100%正常运行时间
当前热门话题：治理和区块链
块大小的辩论
对金融影响感兴趣的银行
私有区块链——降低传统银行基础设施的成本+结算时间

- **Altcoins (Dash, Dogecoin, Litecoin)**
- **Bitcoin 2.0 / Ethereum - applying blockchains outside of finance**
- **Remittances - circumvent traditional banking infrastructure**
  - Send money anywhere in the world for 5 cents
- **Be your own bank - 100% uptime**
- **Current hot topics: Governance and “blockchain”**
  - Block size debate
  - Banks taking interest in financial implications
- **Private blockchains - reduce costs + settlement times in traditional banking infrastructure**

# Basic Concepts - Identity in Bitcoin

用假名汇款
〇假名=地址=公钥
加密原语
〇数字签名方案(ECDSA椭圆曲线数字签名算法)
· 公钥/私钥对:比如电子邮件地址+密码
〇单向哈希函数(SHA-256)
比特币隐藏在大量的公钥中
〇用户可以生成任意多对密钥
〇示例地址:1Ft93erwVzTH8bsoH26NAj98tw98X2upB4
〇2^160可能的地址
(1,461,501,637,330,902,918,203,684,832,716,283,019,655,932,542,976个地址)
〇地球上的沙粒:2^63
〇2的126次方实际上只有2的160次方的0.0000000058%

- **Send money between pseudonyms**
  - pseudonym == address == public key
- **Cryptographic primitives**
  - digital signature scheme (ECDSA: Elliptic Curve Digital Signature Algorithm)
    - public key/private key pair; **like email address + password**
  - one-way hash function (SHA-256)
- **Bitcoin is hidden in the large amount of public keys**
  - Users can generate arbitrarily many key pairs
  - Example Address: 1FtQU9X78hdshngJiCBw9tbE2MYpx87eLT
  - 2^160 possible addresses (1,461,501,637,330,902,918,203,684,832,716,283,019,655,932,542,976 addresses)
  - Grains of sand on earth: 2^63
  - 2^126 is actually only 0.0000000058% of 2^160

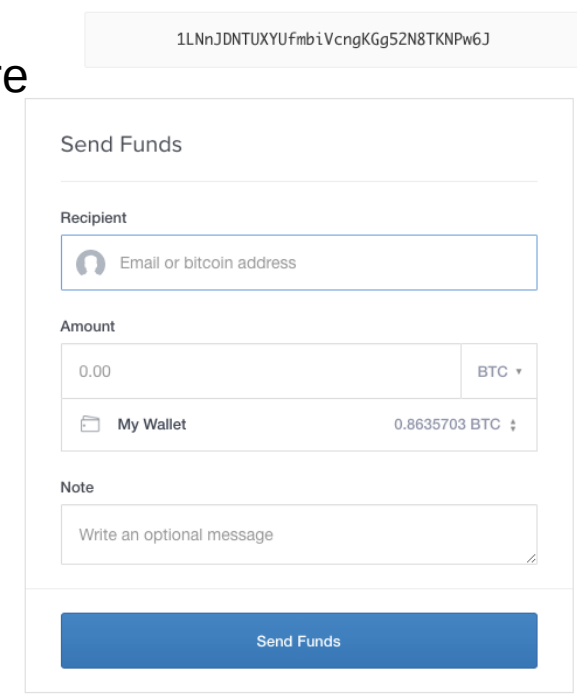
一个比特币交易-基本版本

# A Bitcoin Transaction - Basic Version

- **Bitcoin exists as software**
  - Transactions are conducted through wallet software
  - Wallet creation generates a Bitcoin address
- **To receive money, you share your address**
  - Sender specifies address and amount
- **The transaction is broadcast to the network, where "miners" verify it and add it to the transaction history**

比特币以软件的形式存在
〇通过钱包软件进行交易
〇钱包创建生成一个比特币地址
为了收到钱，你分享你的地址
〇寄件人指定地址和金额
事务被广播到网络中，“矿工”验证它并将其添加到事务历史中

1LNnJDNTUXYUfmbiVcngKGg52N8TKNPw6J



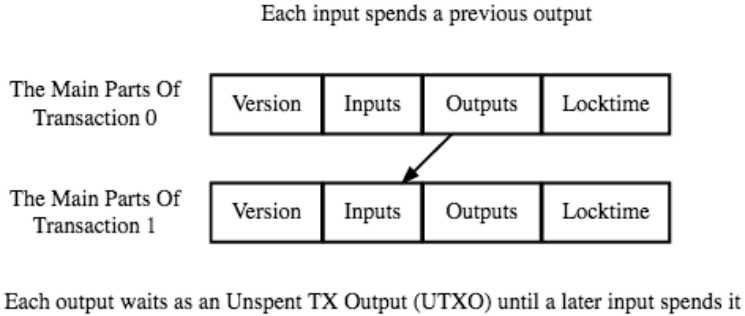

Coinbase interface

基本概念-交易

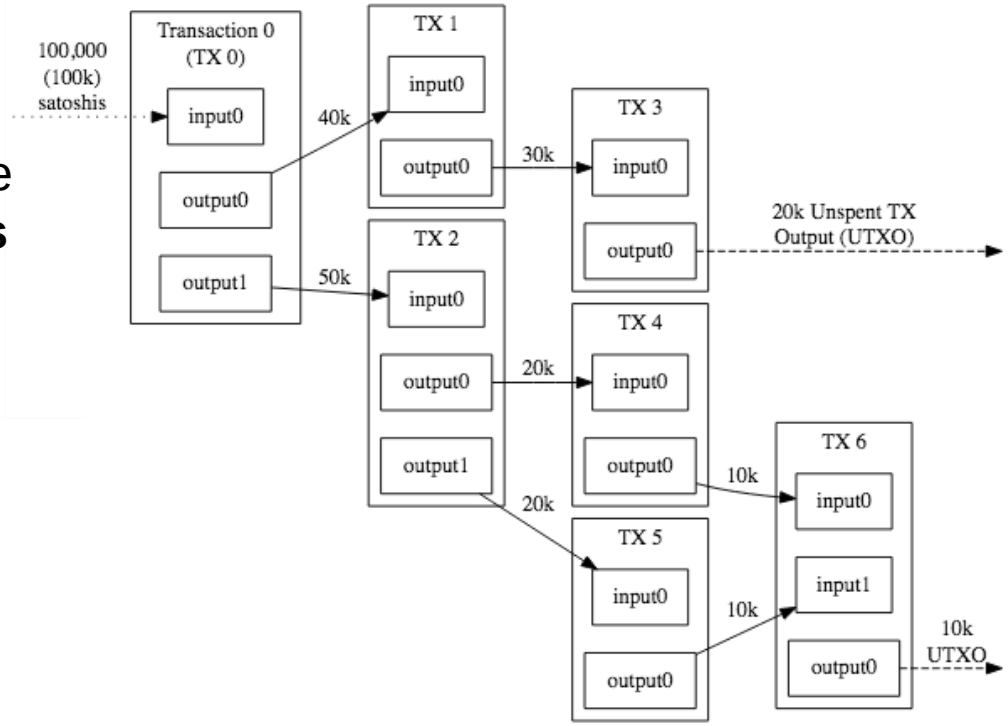
# Basic Concepts - Transactions

将输入地址映射到输出地址
〇产出只能用一次
典型的tx:一个输入，两个输出
费用是隐式的

- **Maps inputs addresses to output addresses**
  - Outputs can only be spent once
- **Typical tx: one input, two outputs**
- **Fees are implicit**




Each output waits as an Unspent TX Output (UTXO) until a later input spends it




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

Source: Bitcoin Developer Guide

# Basic Concepts - Blocks + Blockchain

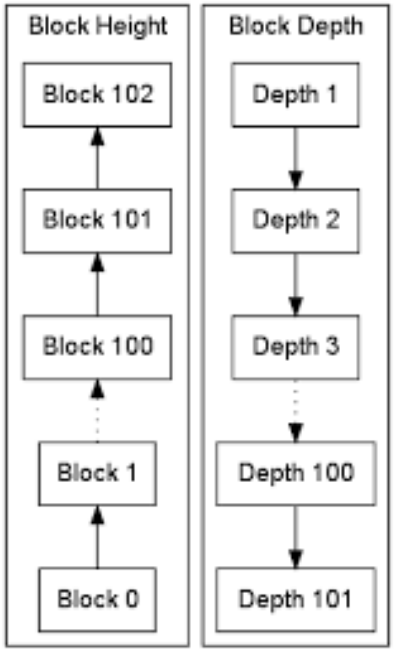
## Blocks

块
包含一组有序的事务
事务的时间戳是不可变的
每个块引用前一个块
每个块都有高度和深度(确认符)
目前428 k块
区块链
整个系列的块被“锁”在一起

- **Contains an ordered bunch of transactions**
  - Timestamps the transactions, are immutable
- **Each block References a previous block**
- **Each block has height and depth (confirmations)**
  - Currently 428k blocks

## Blockchain

- **The entire series of blocks 'chained' together**




Block Height Compared To Block Depth

Source: Bitcoin Developer Guide

# Transaction View information about a bitcoin transaction

447cb6623db32b5f28c94ac10551802075f053208fe995204a145197e2904bb9

| | | |
|---|---|---|
| 3Nxwenay9Z8Lc9JBiywExpnEFiLp6Afp8v (44,000 BTC - Output) | 3LrLWTSdd69oZVVQ6dtWaAAaBLn7N3rRjz - (Spent) | 333.33328889 BTC |
| | 3QkXtcSWJA9w77eCujnMBKDWFe7F7zwxTg - (Spent) | 333.33328889 BTC |
| | 3Qd7hXZoZ1iyXZznrbdUwUQBxHMmujdqhJ - (Spent) | 333.33328889 BTC |
| | 3ECJwvx9VgfotcUuEJMVNvmWnTGVMk179L - (Spent) | 333.33328889 BTC |
| | 3BuQmbmdce3e31GEovq5SgowLdfMgJzLDE - (Spent) | 333.33328889 BTC |
| | 3NwKLjJjzXSnBFQWokXRgBG3JeuF3bsnfE - (Spent) | 333.33328889 BTC |
| | 3GEaT8ZRXELcjMSFvGro6eZcC5S1LSLZuN - (Spent) | 333.33328889 BTC |
| | 35DVAzDtZDKAU94kFT9sxoscnuLCTxgwYc - (Spent) | 333.33328889 BTC |
| | 3Nxwenay9Z8Lc9JBiywExpnEFiLp6Afp8v - (Unspent) | 38,000 BTC |
| | 35mwqShnStDro6uEB4bmsgbyBo8en6Byfm - (Spent) | 333.33328889 BTC |
| | 39pvSqfNcUosc8RGVWxyzKM3ny96a3uSkW - (Spent) | 333.33328889 BTC |
| | 39QNJSgQg5JnBXAtbF8ezkDn72VqWdPZPJ - (Spent) | 333.33328889 BTC |
| | 3L9qAGBQLbXkFAB2GpijnJXPScSVjuiJio - (Spent) | 333.33328889 BTC |
| | 37WSkANPVUQ8uuktf8hv671CejRtBtQ4tJ - (Spent) | 333.33328887 BTC |
| | 3EEwPZZ6pYRJSotCz9RBoVYPRnoWyGWEka - (Spent) | 333.33328889 BTC |
| | 3C4ABC7iPcAAKBh6SJXfvUSDBew3abCtw3 - (Spent) | 333.33328889 BTC |
| | 3HpQozfTzoXAsHf87m2mwJXUQ14LVtLgK4 - (Spent) | 333.33328889 BTC |
| | 337RfngTLRTpU7RT9sKWQWDdmfcdmWnugi - (Spent) | 333.33328889 BTC |
| | 3P2eoKr3vAeZhJcTzon3VFkv5r7DqSXW9G - (Spent) | 333.33328889 BTC |
| | | 43,999.9992 BTC |

| Summary | |
|---|---|
| Size | 1055 (bytes) |
| Received Time | 2016-08-30 11:45:03 |
| Included In Blocks | 427512 ( 2016-08-30 11:51:09 + 6 minutes ) |
| Confirmations | 854 Confirmations |
| Relayed by IP | 5.39.93.85 (whois) |
| Visualize | View Tree Chart |

| Inputs and Outputs | |
|---|---|
| Total Input | 44,000 BTC |
| Total Output | 43,999.9992 BTC |
| Fees | 0.0008 BTC |
| Estimated BTC Transacted | 333.33328887 BTC |
| Scripts | Hide scripts & coinbase |

# Basic Concepts - UTXO analogy

UTXOs表示“未使用的交易输出”
未使用比特币的全球集合
"我花了这个比特币"而不是"我花了一个比特币"
类似于雅浦群岛的雷石
雷的石头从未动过
相反:就所有权变更达成一致

**UTXOs stands for "Unspent Transaction Outputs"**

- Global set of unspent bitcoins
- "I'm spending THIS bitcoin," not "I'm spending A bitcoin."

**Analogous to Rai Stones of the Yap Islands**

- Rai Stones never moved
- Instead: Agreed on change of ownership

Source: Wikipedia

比特币是由中本聪(Satoshi Nakamoto)在2009年创造的
➢首次分散的、不可靠的交易系统
〇一个只需要互联网连接的低成本金融系统
Nakamoto解决了双支出问题
〇防止某人将同样的资产花费两次
〇解决方案吗?区块链+ PoW

# Recap - The Innovation of Satoshi Nakamoto

**Bitcoin was created by Satoshi Nakamoto in 2009**

- **First ever decentralized, trustless system for transactions**
  - A low cost financial system that only requires an internet connection
- **Nakamoto solved the Double Spending problem**
  - Prevent someone from spending the same asset twice
  - Solution? The blockchain + PoW

Dorian Satoshi Nakamoto
(not actually Satoshi Nakamoto)

# Overview

- Bitcoin Concepts
- Consensus Build-up
- Mining Overview
- Cryptocurrency Mining

建立准备:拜占庭将军的问题

# Build up prep: **Byzantine** Generals Problem

**Group of generals surrounding a city must vote and agree on a plan of action**

包围城市的将军们必须投票并同意一项行动计划

**Constraints:**

约束:
将军们在地理上是分开的;
必须使用信使，信使可能会失败
将军可能是忠诚的，也可能是故意叛国的
假设大多数将军是忠诚的
“拜占庭容错”的实现，如果忠诚的将军们一致同意战略
在比特币中，这是一份关于交易历史的协议

- **Generals are physically separated; must use messengers**
  - Messengers may fail
- **Generals may be loyal or intentionally traitorous**
- **Assume majority of generals are loyal**
- **"Byzantine Fault Tolerance" achieved if loyal generals have unanimous agreement on strategy**

**In Bitcoin, this is an agreement on the history of transactions**

In this version, if Alice wants to send a bitcoin to Bob, she should write and sign this message: "I, Alice, am giving Bob one bitcoin."

在这个版本中，如果爱丽丝想发送一个比特币给鲍勃，她应该写并签署这个消息：“我，爱丽丝，给鲍勃一个比特币。

# v1

## Alice writes and signs a message describing her transaction

Alice编写并签署了一条描述她的事务的消息


“I, Alice, am giving Bob one bitcoin.”

Next, Alice should send the transaction to the everyone running the Bitcoin software.
= Now everyone knows that Bob has one more bitcoin and Alice has one less.

接下来，爱丽丝应该把交易发送给所有运行比特币软件的人。现在大家都知道Bob多了一个比特币，Alice少了一个。

# v1

## Alice sends her message to the world

This first version has one major flaw:
Alice could keep sending the same transaction five times.

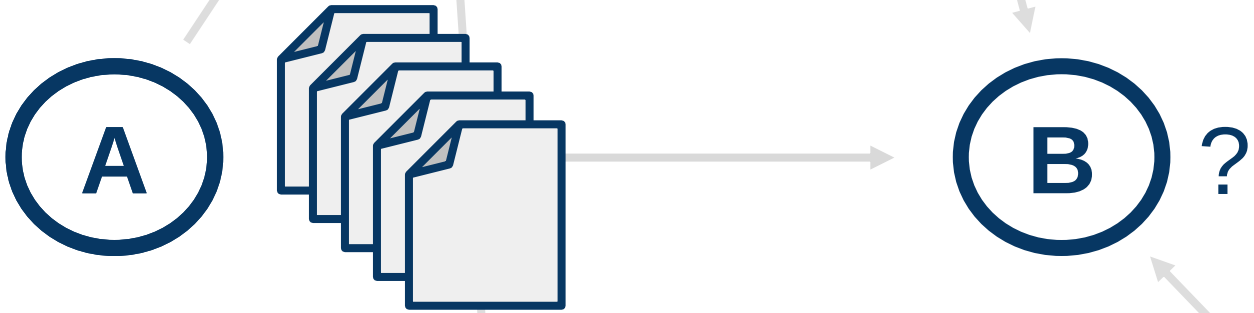
= What would that mean? Does Bob now have five different bitcoins or five duplicates of the same bitcoin?

第一个版本有一个主要的缺陷:Alice可以连续发送5次相同的交易。那是什么意思?鲍勃现在有5个不同的比特币还是5个相同比特币的副本

# v1

## Alice sends five identical messages

Alice发送了5条相同的信息



In version 2, we’re going to solve that problem of double spending by introducing serial numbers to make bitcoins uniquely identifiable.

在版本2中，我们将通过引入序列号来让比特币具有唯一的可识别性来解决重复消费的问题。
如果爱丽丝想给鲍勃发送一个比特币，她应该发送这样的信息“我，爱丽丝，给鲍勃一个比特币，序列号是8732。”这样，每个比特币Alice只能花一次。

Now if Alice wants to send a bitcoin to Bob, she should send the message "I, Alice, am giving Bob one bitcoin, with serial number 8732."

This way, Alice can only spend each bitcoin once.

# v2

## Introducing uniquely identifiable serial numbers

引入唯一可识别序列号

There is a problem:

有一个问题：
=这些序列号是从哪里来的?
=我们如何管理谁拥有哪些比特币?
=有了序列号，Bob可以确保Alice不会两次给他发送相同的比特币，但是他怎么能确定这个比特币一开始就是属于她的呢

= Where would these serial numbers come from?
= And how do we manage who owns which bitcoin?
=With serial numbers, Bob can make sure that Alice doesn't send him the same bitcoin twice, but how can he be sure that the bitcoin belonged to her in the first place?

# v2

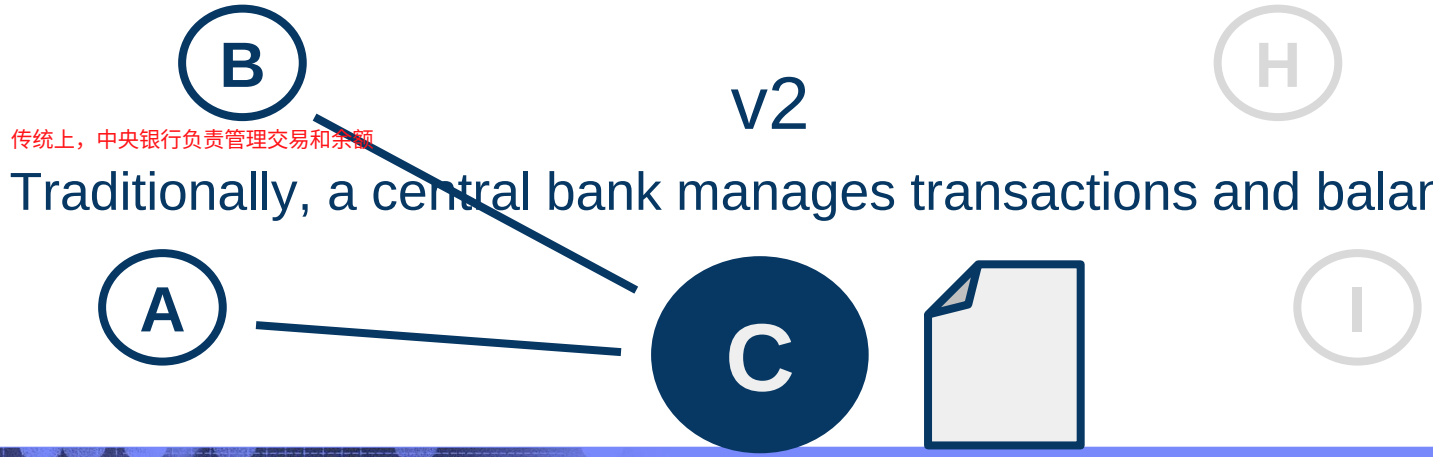
## Where do serial numbers come from?

序列号从何而来?


?

To make version 2 work, there needs to be a trusted source of serial numbers.
The traditional source is a bank.
This bank would provide serial numbers for bitcoins, keep track of who owns which bitcoins, and verify that transactions are legitimate.
Now when Alice sends her transaction to Bob, he can check with the bank that the bitcoin actually belonged to her and that it is unique.

要使版本2工作，需要有一个可信任的序列号来源。传统的来源是银行。这家银行将提供比特币的序列号，追踪谁拥有哪些比特币，并验证交易是否合法。现在，当爱丽丝将她的交易发送给鲍勃时，他可以向银行确认比特币实际上是属于她的，而且它是唯一的。

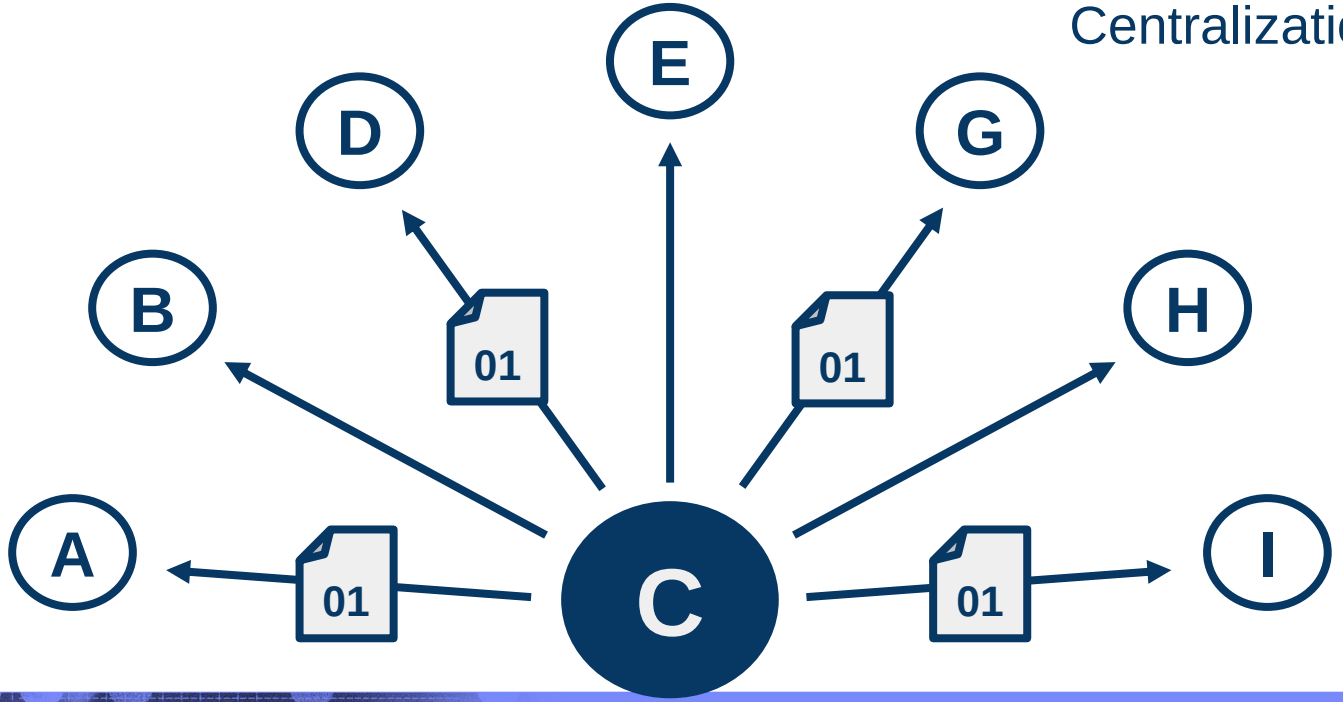


There is a problem:
version 2 did solve the issue of duplication, but it lost the decentralized nature of version 1, where transactions were announced to everyone without a bank.

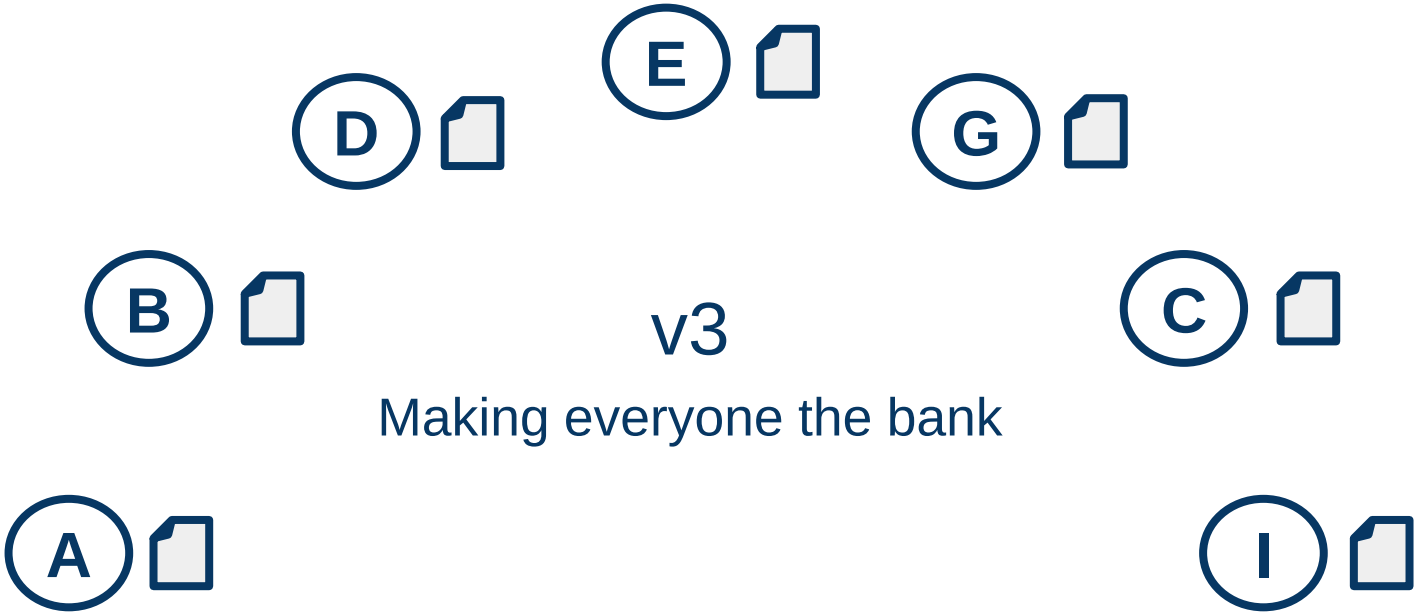
有一个问题:版本2确实解决了复制的问题，但它失去了版本1的分散化本质，在版本1中，交易是不通过银行向所有人宣布的

v2

Centralization

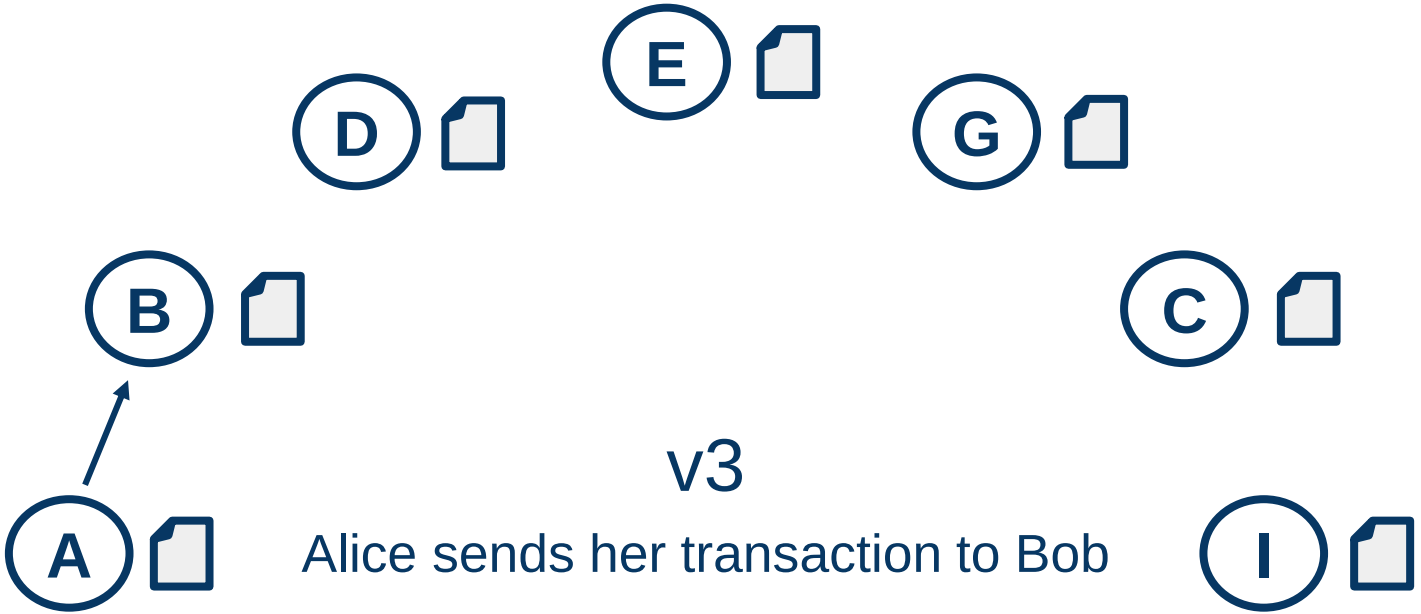


In version 3, we're going bring back that decentralized structure by making everyone the bank.
**Now everyone has a complete record of all transactions.**
In Bitcoin, this is called the **blockchain**.
在第三版中，我们将恢复分散化的结构，让每个人都成为银行。现在每个人都有了所有交易的完整记录。在比特币中，这被称为区块链



Now when Alice sends her transaction to Bob, he can check **his copy** of the blockchain to be sure that the bitcoin actually belonged to Alice.

现在，当Alice将她的交易发送给Bob时，Bob可以检查他的区块链副本以确定比特币实际上属于Alice

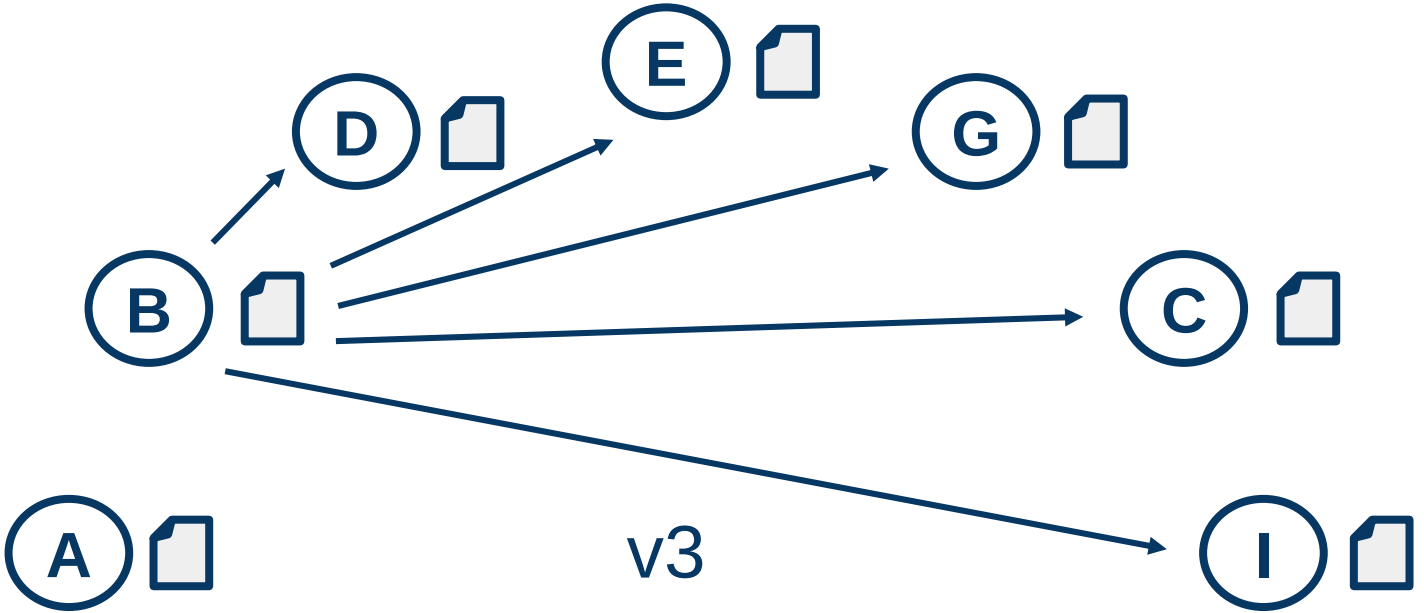



v3

Alice sends her transaction to Bob

If that works out, Bob announces the transaction to the world and everyone updates their copy of the block chain.

如果这可行的话，Bob向全世界宣布交易，每个人都更新他们的区块链副本。




Bob announces the transaction to the world

There is a problem:

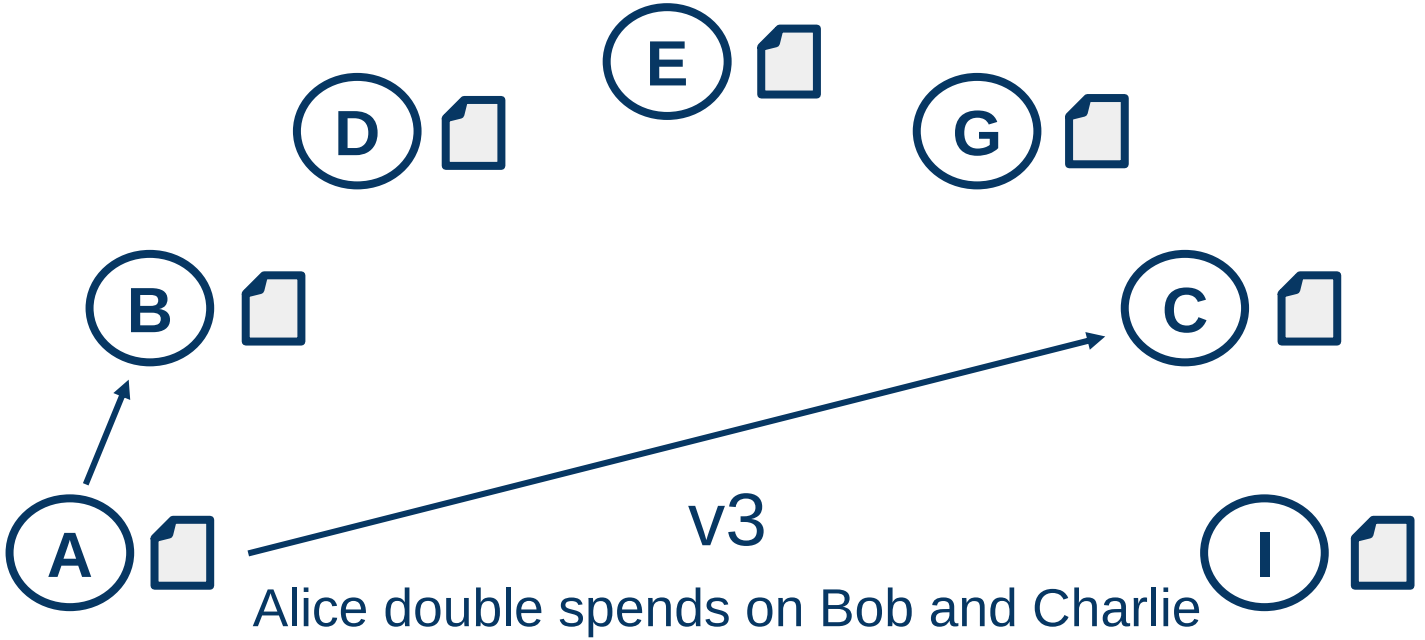
有一个问题:如果Alice同时将她的交易发送给Bob和Charlie怎么办?双方都会发现比特币属于爱丽丝，接受交易并向全世界宣布。其他人应该如何更新他们的区块链副本?很明显，鲍勃和查理不能拥有相同的比特币，所以我们有一个问题。

What if Alice sends her transaction to Bob and Charlie simultaneously?

Both will find that the bitcoin belonged to Alice, accept the transaction, and announce it to the world.

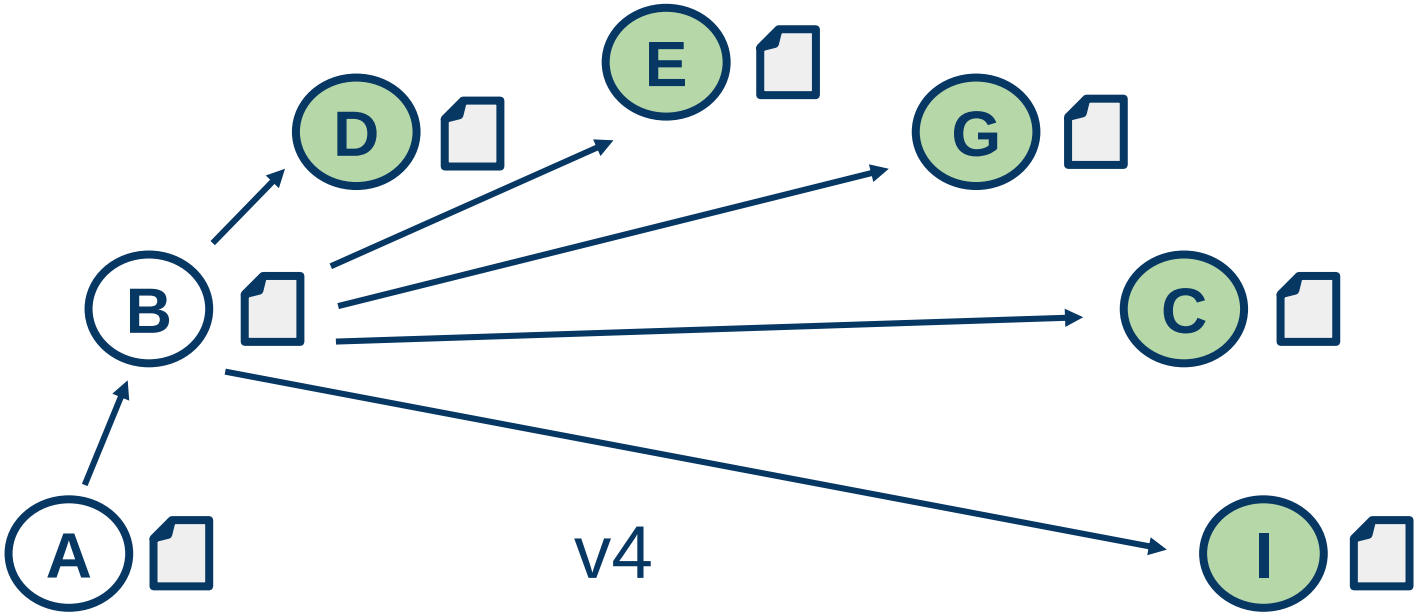
How should other people update their copies of the block chain?

Obviously, both Bob and Charlie can't own the same bitcoin, so we have a problem.



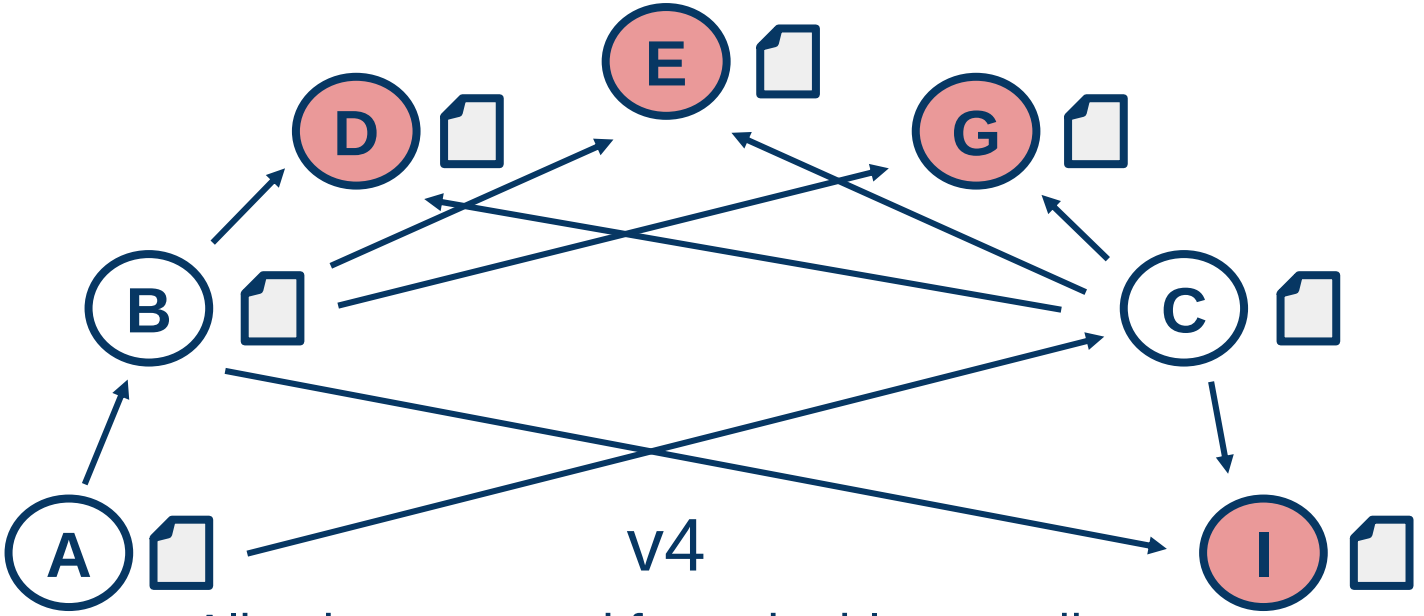
We can begin to solve that problem of double spending in version 4 by giving everyone the power to verify a transaction.
Now when Alice sends her transaction to Bob, Bob shouldn’t try to verify the transaction alone.
Rather, he should broadcast the possible transaction to the entire network of bitcoin users, and ask them to help verify it.
If enough users verify the transaction, Bob can accept the bitcoin, and everyone will update their block chain.

在版本4中，我们可以通过赋予每个人验证交易的能力来解决重复消费的问题。现在，当Alice向Bob发送交易时，Bob不应该单独尝试验证交易。相反，他应该向整个比特币用户网络广播可能的交易，并请他们帮忙核实。如果有足够多的用户验证了交易，Bob就可以接受比特币，每个人都将更新自己的区块链。



This way, if Alice tries to double spend on Bob and Charlie, other bitcoin users will notice and reject the transactions.

这样的话，如果Alice试图在Bob和Charlie身上花重复的钱，其他比特币用户就会注意到并拒绝交易。

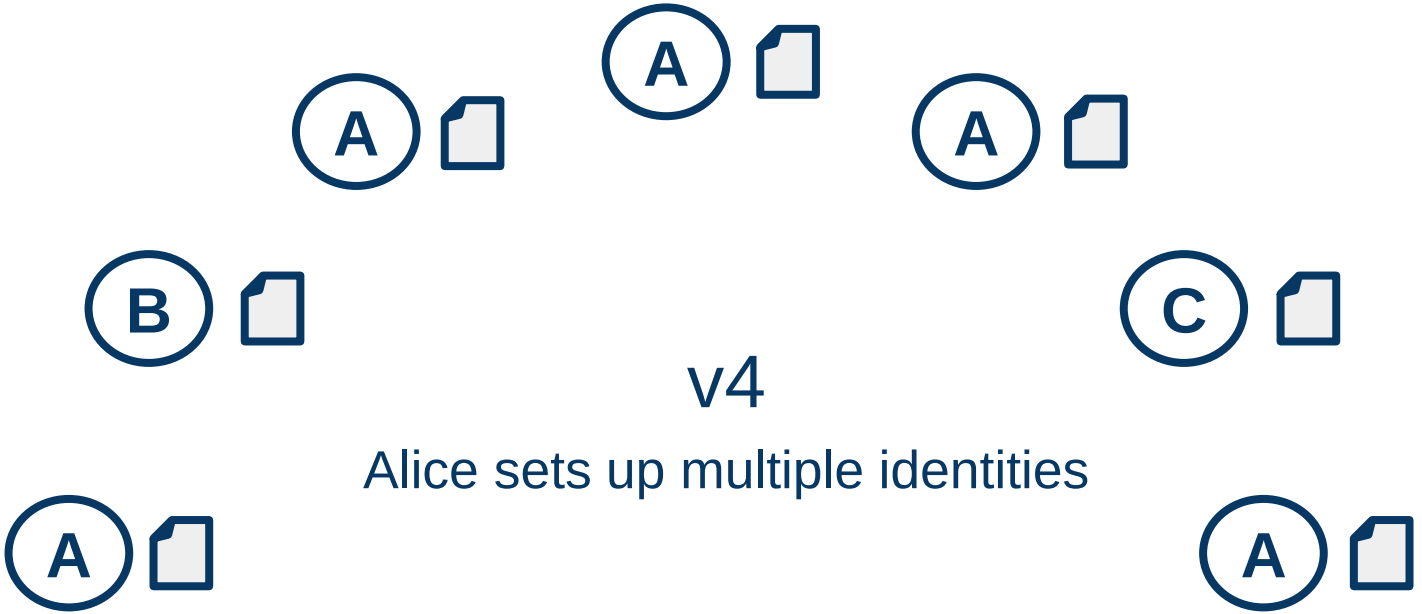



Alice is prevented from double spending 艾丽丝被禁止重复消费

There's a problem with this approach:

这种方法存在一个问题:爱丽丝可以通过接管比特币网络，在鲍勃和查理身上花重复的钱。她可以使用一个自动系统来建立大量独立的身份，从而淹没比特币网络。

Alice could double spend on Bob and Charlie by taking over the bitcoin network.
She can use an automated system to set up a large number of separate identities that overwhelm the bitcoin network.

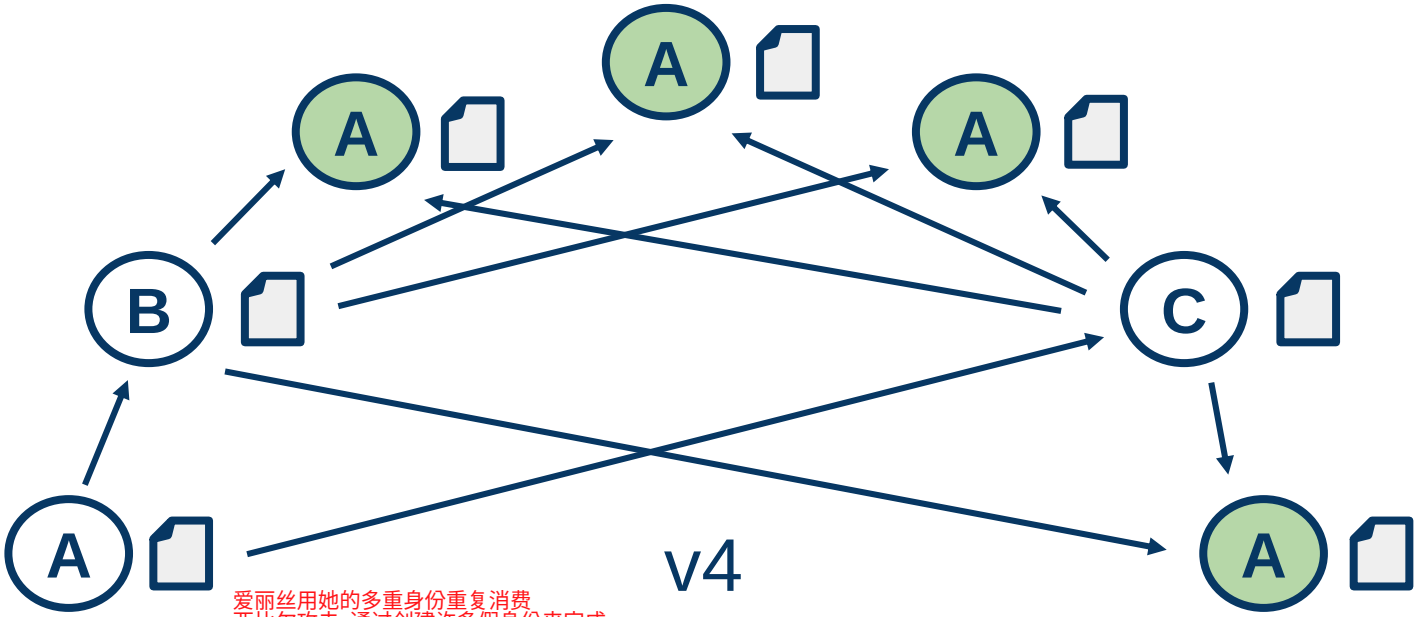



v4

Alice sets up multiple identities

Now when Bob and Charlie ask the network to verify their transactions,
Alice’s many identities swamp the network and announce to both Bob and Charlie that the transactions are fine, fooling them into accepting the same bitcoin.

现在，当鲍勃和查理要求网络验证他们的交易时，爱丽丝的许多身份淹没了网络，并向鲍勃和查理宣布交易很好，欺骗他们接受相同的比特币。



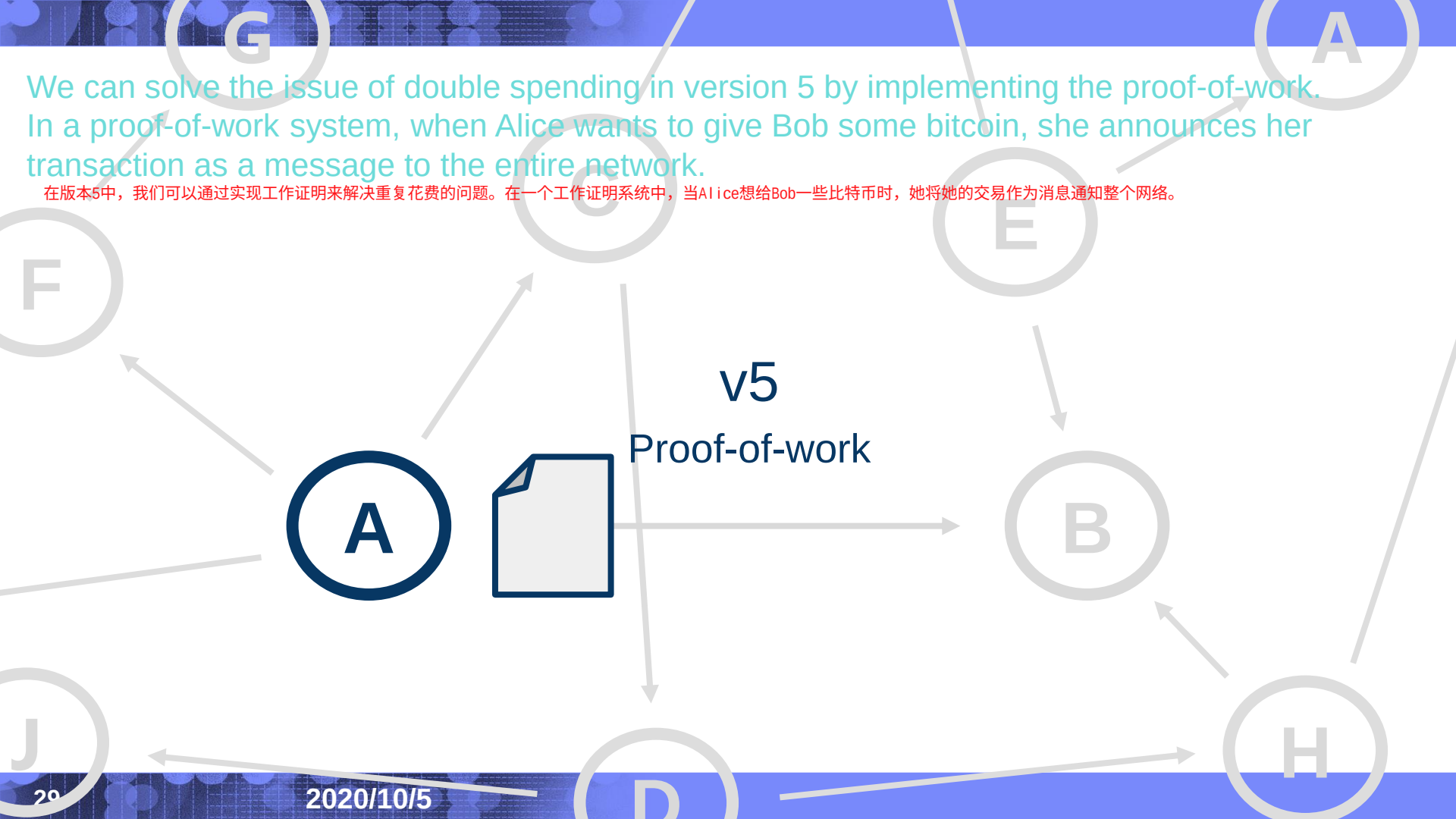

爱丽丝用她的多重身份重复消费
西比尔攻击:通过创建许多假身份来完成

Alice double spends with her multiple identities

**Sybil Attack**: Done by creating many fake identities

We can solve the issue of double spending in version 5 by implementing the proof-of-work. In a proof-of-work system, when Alice wants to give Bob some bitcoin, she announces her transaction as a message to the entire network.

在版本5中，我们可以通过实现工作证明来解决重复花费的问题。在一个工作证明系统中，当Alice想给Bob一些比特币时，她将她的交易作为消息通知整个网络。



As other users receive Alice's transaction message, they add it to a list of pending transactions they've been told about, but haven't yet been verified by the network.

当其他用户收到Alice的交易消息时，他们会将其添加到一个已告知但尚未通过网络验证的挂起交易列表中。任何用户都可以维护他们自己的挂起事务列表

Any user can maintain their own list of pending transactions

# v5

## Pending transactions

1. I, Tom, am giving Sue one bitcoin, with serial number 3920.
2. I, Sydney, am giving Cynthia one bitcoin, with serial number 1325.
3. I, Alice, am giving Bob one bitcoin, with serial number 1234.

If David wants to verify these pending transactions in a proof-of-work system, he has to do three things:

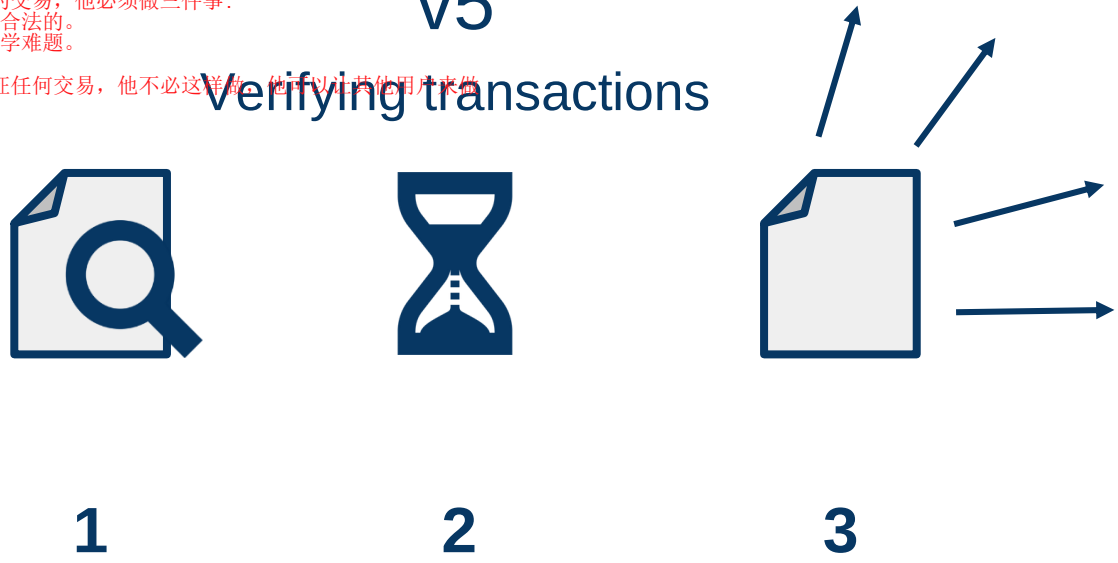
- First David has to check his copy of the block chain to make sure the transactions are legitimate.
- Second, his computer has to use resources to solve a hard mathematical puzzle.
- And third, he has to announce the block of transactions to the network.

Before we take a closer look, remember that if David doesn't want to verify any transactions, he doesn't have to—he can just let other users do that!

如果David想要在工作验证系统中验证这些悬而未决的交易，他必须做三件事:
首先，大卫必须检查他的区块链副本，以确保交易是合法的。
其次，他的计算机必须利用资源来解决一个困难的数学难题。
第三，他必须向网络公布交易的区块。
在我们进一步了解之前，请记住，如果David不想验证任何交易，他不必这样做，他可以让其他用户来做。

# v5

# Verifying transactions



# v5

## Why the math?

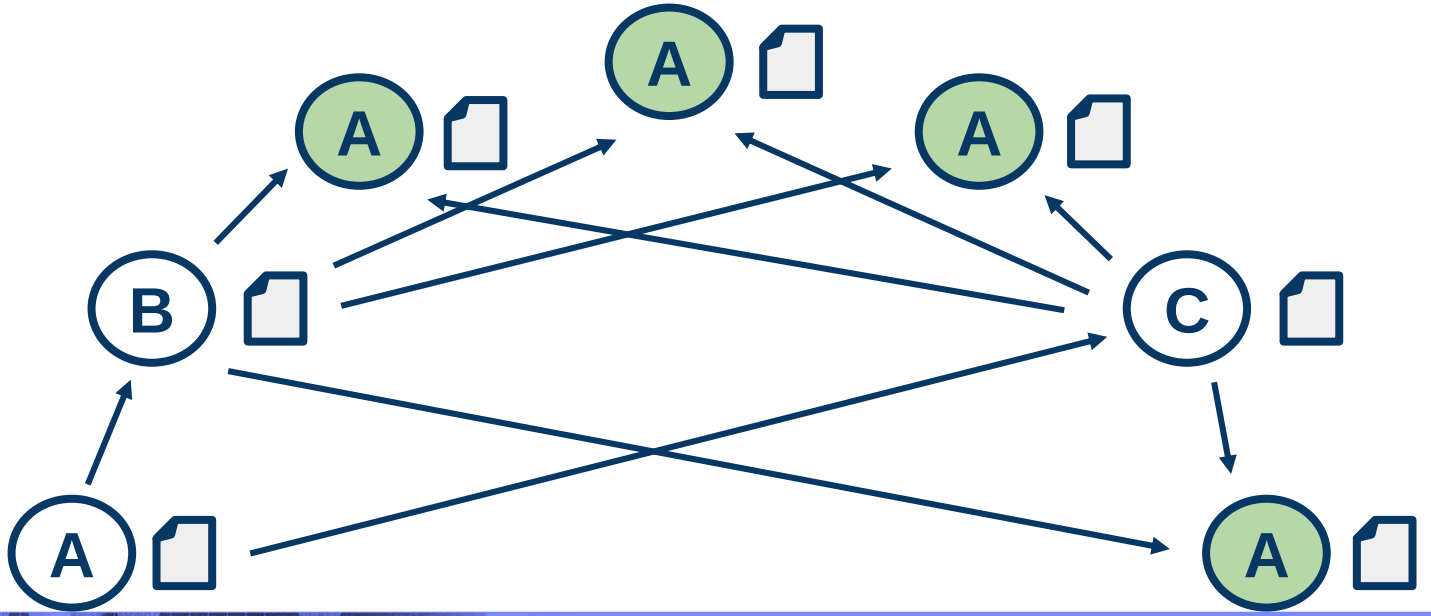
这是一个重要的问题。通过要求大卫的电脑解决一个数学难题，我们实际上是在解决重复消费的问题。让我们再看一遍那个问题

This is an important question. By requiring David’s computer to solve a math puzzle we are actually solving the problem of double spending. Let’s take a look at that problem again.

# v4

## Alice double spends with her multiple identities

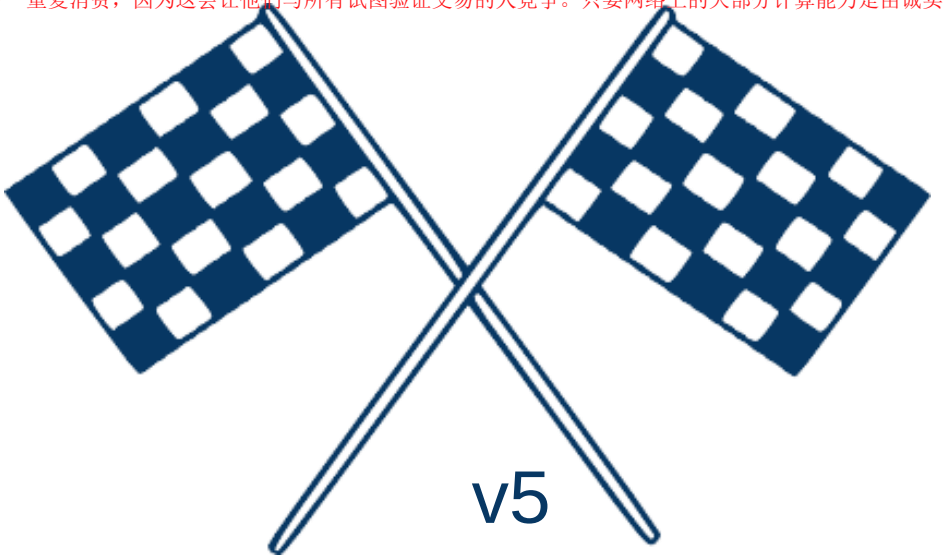
爱丽丝用她的多重身份重复消费



You can think of proof-of-work as a competition to verify transactions. In bitcoin, people call this **mining**.
If your computer solves the math puzzle before the other computers on the network, you will verify the pending transactions and receive some bitcoin as a reward.

Proof-of-work prevents bad actors like Alice from double spending because it puts them in competition with everyone else trying to verify transactions. So long as most of the computing power on the network is controlled by honest people, malicious actors like Alice will have a hard time doing dishonest things, like double spending.

您可以将工作证明视为验证交易的竞争。在比特币中，人们称之为采矿。
如果您的计算机在网络上的其他计算机之前解决了数学难题，您将验证未决的交易并获得一些比特币作为奖励。
工作证明可以防止像爱丽丝这样的“坏行为者”重复消费，因为这会让他们与所有试图验证交易的人竞争。只要网络上的大部分计算能力是由诚实的人控制的，像爱丽丝这样的恶意行为者就很难做不诚实的事情，比如重复消费



v5

# Proof-of-work as a competition

# Summary

| Version | Major feature | Value added |
|---|---|---|
| 1 | 发布到网络的签名消息<br>Signed messages announced to the network | Basis of entire system |
| 2 | Serial numbers | 交易的唯一标识<br>Uniquely identifiable transactions |
| 3 | The block chain | Shared record of transactions |
| 4 | Everyone verifies transactions | Increased security |
| 5 | Proof-of-work | Prevents double spending |

# Overview

- Bitcoin Concepts
- Consensus Build-up
- Mining Overview
- Cryptocurrency Mining

简述的比特币挖掘-工作证明

# Sketch of Bitcoin Mining - Proof of Work

- **Solution to the Byzantine Generals Problem: Proof-of-Work (PoW)**
  - “Miners” continuously compete to solve a very computationally difficult problem
  - Proof of work is an example of a "Byzantine consensus algorithm"
- **Proof of work criteria:**
  - Easy to verify
  - Hard to compute
- **SHA-256 Hash function satisfies these**
  - One-way hash function; can hash any arbitrary data
  - Pretty much random (very useful property)
- **Example**
  - SHA256("Donald Trump") == "e4f2e1f0e2ae4d3ce7018cf3b4f3577c99714bdc9f5a4ac28e3e7cb2c505db6c"
  - SHA256("Donald trump") == "6ad2fa6a5caaee9143578931456322c4433a92ae2af8f0d5c9b4f9bb080f49d6"

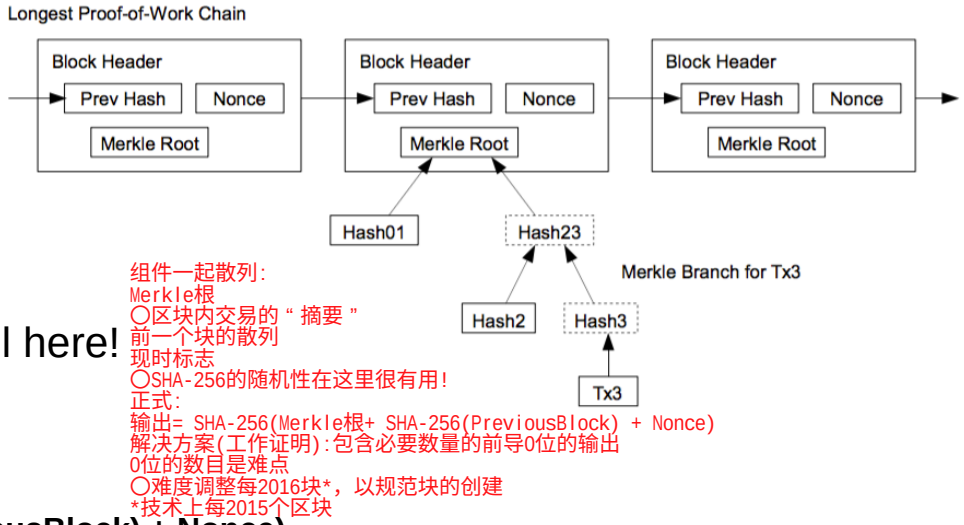
拜占庭将军问题的解决方案:工作证明(PoW)
○矿工们不断地争着解决一个非常难计算的问题
○工作证明是“拜占庭共识算法”的一个例子
工作证明标准:
○容易验证
○难以计算
SHA-256哈希函数满足这些条件
○单向哈希函数;可以散列任意数据
○非常随机(非常有用的性质)

# Sketch of Bitcoin Mining - Finding blocks

- **Finding the PoW => 'found' a block; can add block to blockchain**
  - Miner who found block adds "**coinbase transaction**"
    - contains mining reward (currently 12.5 BTC)
  - Miner broadcasts block
  - Other nodes verify, then add to their own copy of the blockchain
- **Timeline + stats**
  - This happens roughly every 10 minutes
    - Difficulty of the problem adjusted every 2 weeks
  - Block reward halving every 4 years (recently halved on July 9th)
    - Bitcoin is in limited supply - 21 million bitcoins by 2140
      - Deflationary
  - 15.2 million bitcoins currently in circulation today
  - ~$9.6 billion market cap
  - Price is currently ~$600 per bitcoin


求PoW =>“发现”一个块;可以添加块到区块链
○发现块的矿工增加了“coinbase交易”
包含采矿奖励(当前为12.5 BTC)
○矿工广播块
○其他节点验证，然后添加到自己的区块链副本
时间轴+数据
○这种情况大约每10分钟发生一次
问题难度每2周调整一次
○每四年奖励减半(最近在7月9日减半)
比特币的供应量有限，到2140年将达到2100万比特币
通货紧缩的
○目前流通的比特币为1520万
○96亿美元的市值
○当前比特币价格为~ 600美元/比特币

# Sketch of Bitcoin Mining - The Mining Problem




## Components hashed together:

- **Merkle Root**
  - 'summary' of the transactions in the block
- **Hash of previous block**
- **Nonce**
  - Randomness of SHA-256 is useful here!

## Formally:

- **Output = SHA-256(Merkle Root + SHA-256(PreviousBlock) + Nonce)**
- **Solution (Proof-of-work): an output that contains a requisite number of leading 0 bits**
  - The number of 0 bits is the **difficulty**
  - Difficulty adjusts every 2016 blocks* to regulate block creation
    - *technically every 2015 blocks

# Sketch of Bitcoin Mining - 51% Attacks

比特币的主要假设:
网络上不超过51%的人是不诚实的
诚实的大多数人总是会形成最长的工作证明链
51%攻击:试图压制网络的采矿能力

## Major assumption of Bitcoin:

- No more than 51% percent of the network is dishonest
- An honest majority will always form the longest proof-of-work chain

**51% Attack: Attempt to overwhelm the mining power of the network**


Source: Martin Koppelmann presenting at SF Bitcoin Devs

# Sketch of Bitcoin Mining - Summary

功能为：
一种挖掘机制，确保以公平的方式分配硬币
鼓励人们帮助维护网络安全
在分散化货币中，使您达成共识的关键组成部分

## Functions as:

- A mining mechanism that ensures coins are distributed in a fair way
- An incentive for people to help secure the network
- Key component that enables you reach consensus in a decentralized currency

挖掘与彩票类似，只不过不是购买彩票，而是贡献计算能力

**Mining is similar to a lottery, except instead of buying lottery tickets, you contribute computational power**

# Overview

- Bitcoin Concepts
- Consensus Build-up
- Mining Overview
- Cryptocurrency Mining

# Cryptocurrency Mining

--Proof-of-Work Consensus

# What is profit

**If**

**reward > cost**
**Then $$$$**

**profit = reward - cost**

# Cost of Mining

**If**

**mining reward > mining cost**

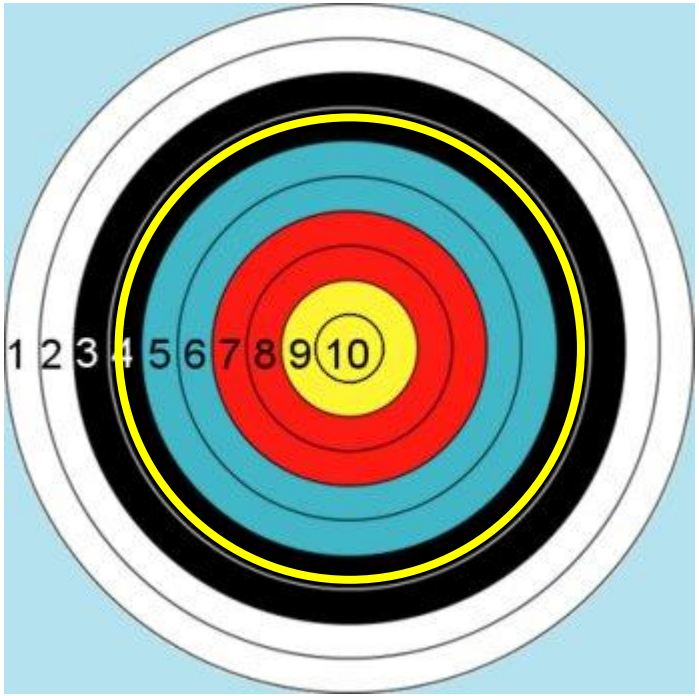
**then miner profits**

**where**

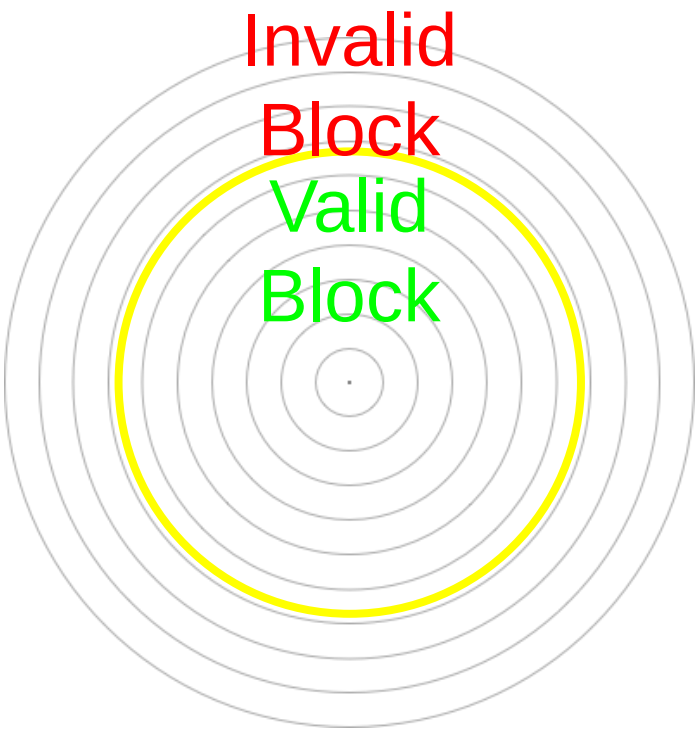
**mining reward = block reward + tx fees**

**mining cost = hardware cost + operating costs (electricity, cooling, etc.)**

# Block Reward :: Difficulty Adjustment 调整困难



# Block Reward :: Difficulty Adjustment



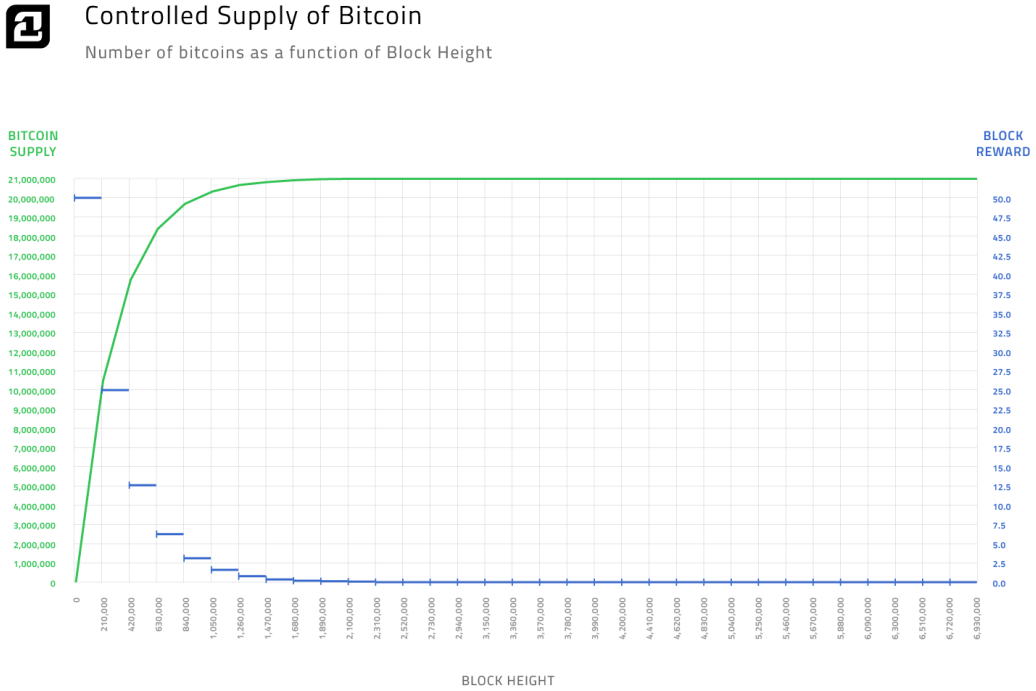

*H*(*nonce* || *prev_hash* || *tx* || *tx* || ... || *tx*) < *target*

等可能击中1 2 3环，
更快的矿工=更多的命中/秒
目标:在黄圈里面
不断减小黄圈的大小
开采难度调整每2016块

- Equally likely to hit ring 1, 2, 3, …
- Faster miners = more hits / second
- Target: inside the yellow ring
- Keep decreasing the size of the yellow ring…
- Mining difficulty adjustment every 2016 blocks
- Difficulty adjusted to

next_difficulty = previous_difficulty * (2 weeks) / (time to mine last 2016 blocks)

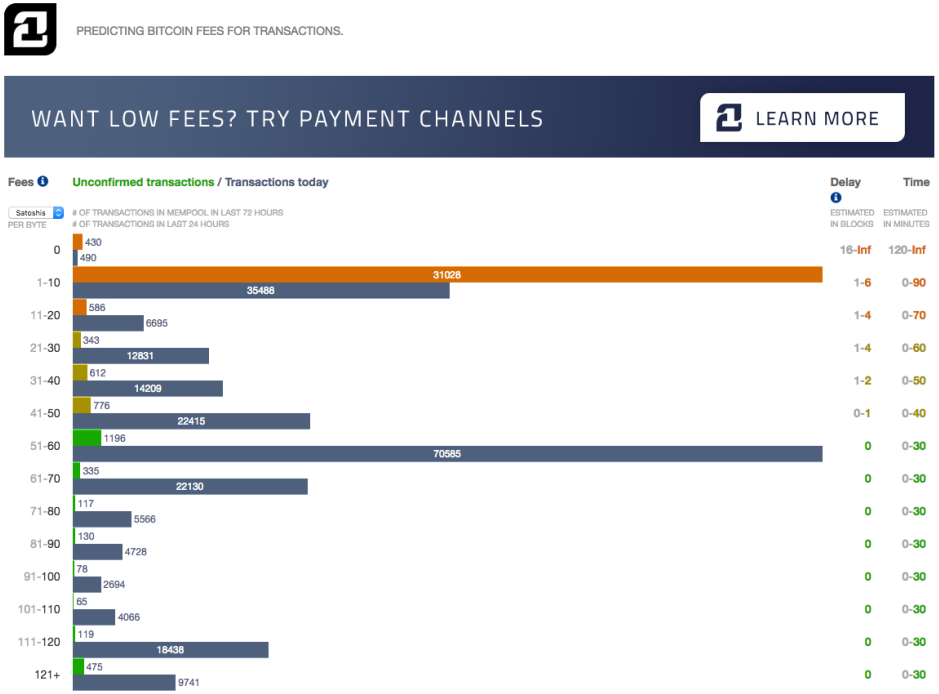
# Block Reward :: Bitcoin Halving 比特币减半




- Node creating block includes a special tx to self
- Current block reward: 12.5 BTC
- Monetary incentive for honest behavior
- Halves every 210,000 blocks
  - Deflationary currency!
- Geometric sum: ends at 21e6
- So what next?

节点创建块包括一个特殊的tx到自身
当前块奖励:12.5 BTC
诚实行为的金钱激励
每210,000块分成两半
○通货紧缩的货币!
几何和:以21e6结束
那么下一个

# Transaction Fees

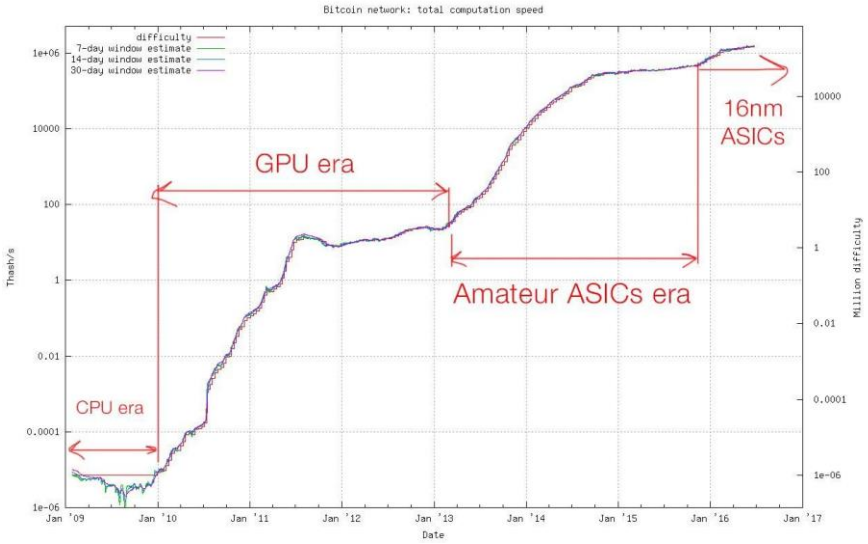



交易费用可选
矿业公司倾向于交易费用较高的交易
激励你把你的交易包括在他们正在挖掘的下一个区块
21e6后矿商收入的主要来源

- Transaction fees are optional
- Miners tend to favor transactions with larger transaction fees
  - Incentive to include your transaction in the next block they’re mining
- Primary source of revenue for miners after 21e6

# Hardware Cost

| | hashes / second | time to block |
|---|---|---|
| CPU | 20 million | 300,000 years |
| GPU | 200 million | 30,000 years |
| FPGA | 1 billion | 600 years |
| ASIC | 10 trillion | 22 days |



# CPU Mining

| | hashes / second | time to block |
|---|---|---|
| CPU | 20 million | 300,000 years |
| GPU | 200 million | 30,000 years |
| FPGA | 1 billion | 600 years |
| ASIC | 10 trillion | 22 days |

- **For each nonce**
  - Run SHA256
  - Check if result was a valid block
- **Slowest**
- **What Satoshi used**
- **Your comupter!**

对于每一个nonce
○SHA256运行
○检查结果是否为有效块
最慢的
什么Satoshi使用
你的电脑

# GPU Mining

| | hashes / second | time to block |
|---|---|---|
| CPU | 20 million | 300,000 years |
| GPU | 200 million | 30,000 years |
| FPGA | 1 billion | 600 years |
| ASIC | 10 trillion | 22 days |

- Designed for parallel computation
- Order of magnitude faster than CPUs
- Consumes a lot of energy, produces a lot of heat
- $446.66 for the R9 290 back in the day

为并行计算而设计
比cpu快一个数量级
消耗大量的能量，产生大量的热量
当时的r9290售价是446.66美元

显卡芯片：Radeon **R9 290**; 核心频率：947MHz;显存频率：5000MHz; 显存容量：4GB

# FPGA Mining

| | hashes / second | time to block |
|---|---|---|
| CPU | 20 million | 300,000 years |
| GPU | 200 million | 30,000 years |
| FPGA | 1 billion | 600 years |
| ASIC | 10 trillion | 22 days |

- Field Programmable Gate Arrays
  - Getting more application specific
- A trade-off between ASIC and general purpose

现场可编程门阵列
〇应用更加具体
ASIC和通用目的之间的权衡

# ASIC Mining

| | hashes / second | time to block |
|---|---|---|
| CPU | 20 million | 300,000 years |
| GPU | 200 million | 30,000 years |
| FPGA | 1 billion | 600 years |
| ASIC | 10 trillion | 22 days |

- **Application-Specific Integrated Circuits.**
  - Circuits specifically designed to do Bitcoin mining (SHA256)
  - Extremely expensive
- **Fastest miners around**
- **~$1600**

特定于应用程序的集成电路。
〇专门设计用于比特币挖掘的电路(SHA256)
〇非常昂贵
最快的矿商大约在1600美元左右

# Operating Costs 生产费用

- **Energy in Bitcoin**
  - Embodied Energy
  - Electricity
  - Cooling
- **Thermodynamic Limit of Landauer's principle**
  - kT ln 2 (k = Boltzmann constant) per bit
- **Bitcoin network energy usage (current)**
  - 14,000 GH/s : 1,375W
  - 1,820,429,066 GH/s : 178,792,140 W
  - ~10% of energy of large coal fire plant
- **Electric heater that makes you money**

能量比特币
◯蕴藏能量
◯电
◯冷却
朗道原理的热力学极限
kT ln 2(k =玻尔兹曼常数)每比特
比特币网络能源使用量(当前)
◯14000 GH/s: 1375 W
◯1820,429,066 GH/s: 178,792,140 W
◯大型火电厂~10%的能源
让你赚钱的电加热器

# Innovative Proof-of-Work Ideas

创新Proof-of-Work想法

# Proof of Work Puzzle Requirements

**Basic Puzzle Requirements:**

基本难题要求：
1。快速验证
2。赢得谜题的机会应该与计算能力成比例
3。无记忆或“无进步”
a。解决难题的几率必须独立于你已经花了多少功夫去解决它
b。一般来说，以试错为基础
比特币谜题是一个“部分哈希-原象谜题”
为部分指定的散列输出查找原图像

1. **Quick to verify**
2. **Chance of winning puzzle should be proportional to amount of computational power**
3. **Memoryless or "Progress free"**
   a. Odds of solving puzzle must be independent of how much work you have already spent trying to solve it
   b. In general, trial-and-error based

**Bitcoin's puzzle is a "partial hash-preimage puzzle"**

- **Find preimage for a partially-specified hash output**

# ASIC-Resistance

支持:
如果没有对抗asic，生态系统中的大多数个体在采矿过程中就没有任何作用
更民主、更分散-“一CPU一票”
反对:
SHA-256专用集成电路只对挖掘比特币有用
〇因此，矿工持有hashpower 51%的股权投资于比特币的安全性
破坏汇率=>攻击者在无用的硬件上浪费了很多钱
〇攻击者可以租用通用计算资源，如Amazon EC2，攻击后不承担任何后果

**Arguments for:**

- Without ASIC-resistance, most individuals in the ecosystem don't have any role in the mining process
- More democratic and decentralized - "One CPU one vote"

**Arguments against:**

- **SHA-256 ASICs are only useful for mining Bitcoin**
  - Therefore, miner with 51% of hashpower is invested in the security of Bitcoin
- **Crashing the exchange rate => attacker wasted a bunch of money on useless hardware**
  - Otherwise, attacker can rent general computing resources, such as Amazon EC2 and bear no consequence after their attack

# ASIC-Resistance: Memory-hard Algorithms

**Memory-hard: Requires a large amount of memory to compute, instead of computation time**

**Memory-bound: The amount of memory dominates the total computation time**

**Why do these help ASIC-resistance?**

- **Logical operations required to compute modern hash functions are only a small proportion of CPU capabilities**
  - CPUs are generalized; allows ASICs to optimize based off of this fact
- **Memory performance increase much slower than computational power**
  - The cost of solving this puzzle would decrease much slower

内存困难:需要大量内存来计算，而不是计算时间
内存限制:内存的数量支配总计算时间
为什么这些有助于对抗asic ？
计算现代哈希函数所需的逻辑操作只占CPU能力的一小部分
cpu是广义的;允许ASICs基于这个事实进行优化
内存性能的提高比计算能力慢得多
解决这个难题的成本会降低得慢得多

# Example: Scrypt

Scrypt is a hash function. The mining puzzle is the same partial hash-preimage puzzle.

Design considerations:

- Used for hashing password
- Hard to brute-force

Used by Litecoin, Dogecoin

Scrypt是一个哈希函数。挖掘难题是相同的部分哈希-原像难题。
设计注意事项:
用于密码散列
很难蛮力
由莱特币，狗币使用



# ASIC-Resistance: Scrypt

在不使用内存的情况下，必须动态计算V[j]。只要选择N，就可以更快地使用内存
两个主要的步骤：
1。用相互依赖的数据填充缓冲区
2。以伪随机的方式访问此数据
缺点：(1)需要同样多的内存来验证，(2)已经ASIC'd

**Without using memory, V[j] must be computed on the fly. Simply choose N such that using memory is faster**

## Two main steps:

1. Fill buffer with interdependent data
2. Access this data in a pseudorandom way

## Drawbacks: (1) Requires just as much memory to verify, (2) already ASIC'd

Figure 8.1: Scrypt pseudocode

```
1 def scrypt(N, seed):
2     V = [0] * N  // initialize memory buffer of length N

  // Fill up memory buffer with pseudorandom data
3   V[0] = seed
4   for i = 1 to N:
5       V[i] = SHA-256(V[i-1])

  // Access memory buffer in a pseudorandom order
6   X = SHA-256(V[N-1])
7   for i = 1 to N:
8       j = X % N  // Choose a random index based on X
9       X = SHA-256(X ^ V[j]) // Update X based on this index

10  return X
```

# ASIC-Resistance: Other Approaches

x11 or x13: Chain 11/13 different hash functions together (used by DASH)

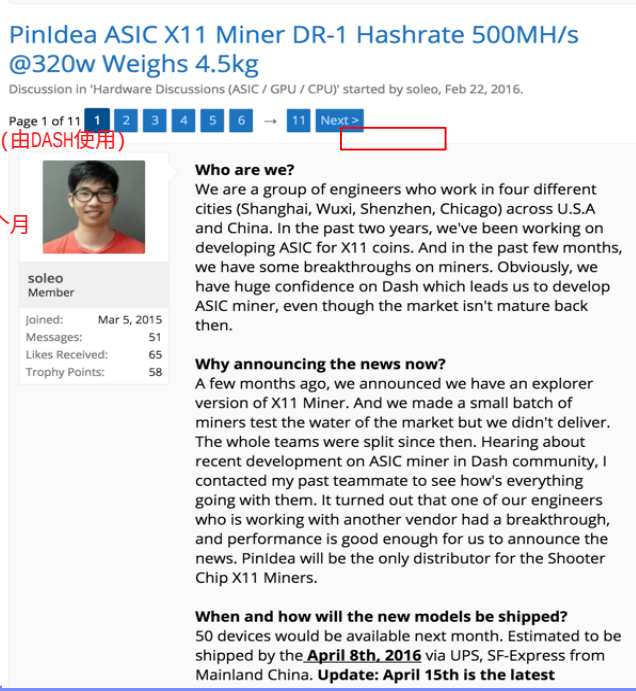
- Makes it significantly harder to design an ASIC
- ...But it's been done

Periodically changing mining puzzle

- E.g. switch between SHA-1, SHA-3, Scrypt for 6 months each
- Easy to work around; Not implemented

Mike Hearn: "There's really no such thing as an ASIC-resistant algorithm."

x11或x13:链11/13不同的哈希函数在一起(由DASH使用)
使得ASIC的设计更加困难
...但它已经完成了
定期改变采矿谜题
如。在SHA-1，SHA-3，Scrypt之间切换6个月
易于操作;没有实现
Mike Hearn:“真的没有防asic的算法。



PinIdea ASIC X11 Miner DR-1 Hashrate 500MH/s @320w Weighs 4.5kg

Discussion in 'Hardware Discussions (ASIC / GPU / CPU)' started by soleo, Feb 22, 2016.

Page 1 of 11 1 2 3 4 5 6 → 11 Next >

soleo
Member

Joined: Mar 5, 2015
Messages: 51
Likes Received: 65
Trophy Points: 58

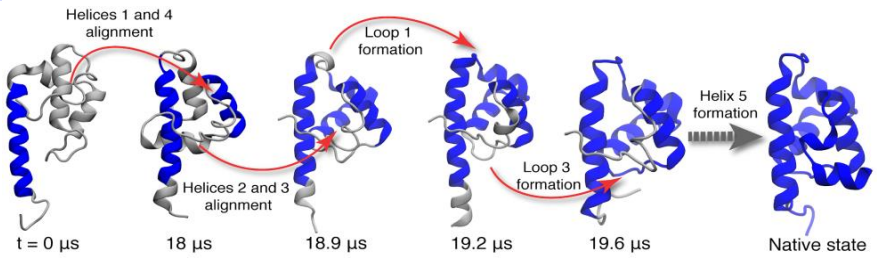
**Who are we?**
We are a group of engineers who work in four different cities (Shanghai, Wuxi, Shenzhen, Chicago) across U.S.A and China. In the past two years, we've been working on developing ASIC for X11 coins. And in the past few months, we have some breakthroughs on miners. Obviously, we have huge confidence on Dash which leads us to develop ASIC miner, even though the market isn't mature back then.

**Why announcing the news now?**
A few months ago, we announced we have an explorer version of X11 Miner. And we made a small batch of miners test the water of the market but we didn't deliver. The whole teams were split since then. Hearing about recent development on ASIC miner in Dash community, I contacted my past teammate to see how's everything going with them. It turned out that one of our engineers who is working with another vendor had a breakthrough, and performance is good enough for us to announce the news. PinIdea will be the only distributor for the Shooter Chip X11 Miners.

**When and how will the new models be shipped?**
50 devices would be available next month. Estimated to be shipped by the **April 8th, 2016** via UPS, SF-Express from Mainland China. **Update: April 15th is the latest**

# Proof of Useful Work

总体思路:“回收”计算能力;用它做一些有用的事情
例子:
搜索较大的素数
发现外星人
用于研究疾病的蛋白质折叠的原子水平模拟
创建预测气候模型
太阳币:分发给生产太阳能的人




General idea: "Recycle" computing power; repurpose it for something useful

Examples:

- Searching for large prime #'s
- Finding aliens
- Atomic-level simulations of protein folding to research disease
- Creating predictive climate models
- SolarCoin: Distributed to people who generate solar power

| Project | Founded | Goal | Impact |
|---|---|---|---|
| Great Internet Mersenne Prime Search | 1996 | Finding large Mersenne primes | Found the new “largest prime number” twelve straight times, including $2^{57885161} - 1$ |
| distributed.net | 1997 | Cryptographic brute-force demos | First successful public brute-force of a 64-bit cryptographic key |
| SETI@home | 1999 | Identifying signs of extraterrestrial life | Largest project to date with over 5 million participants |
| Folding@home | 2000 | Atomic-level simulations of protein folding | Greatest computing capacity of any volunteer computing project. More than 118 scientific papers. |

Princeton Textbook Table 8.3

# Proof of Useful Work: Challenges

**Most of the distributed computing problems are unsuitable for proof of work**

- **Fixed amount of data**
  - SETI@Home could run out of raw data to compute on
  - Missing an **inexhaustible puzzle space**
- **Potential solutions are not all equally likely**
  - Missing an **equiprobable solution space**
- **Cannot rely on a central entity to delegate tasks**
  - Puzzle must be able to be **algorithmically generated**

大多数分布式计算问题不适合工作证明
固定数据量
〇SETI@Home可能会耗尽原始数据来进行计算
〇缺少一个无穷无尽的谜题空间
可能的解决办法并不都是一样的
〇缺少一个等可能解空间
不能依赖一个中央实体来委托任务
〇难题必须能够被算法产生


Example: SETI@Home (searching for aliens)

- **Some segments may have higher likelihood of containing anomalies**
  - All miners would search those areas first
  - Faster miners have higher likelihood of solving the puzzle
  - Therefore it would not be memoryless, large miners have advantage

某些部分可能有较高的可能性包含异常
· 所有矿工将首先搜索这些区域
· 速度快的矿工解决问题的可能性更高
· 因此不会没有记忆，大型矿商有优势

SETI@home is a scientific experiment, based at UC Berkeley, that uses Internet-connected computers in the Search for Extraterrestrial Intelligence (SETI). You can participate by running a free program that downloads and analyzes radio telescope data.

SETI@home是加州大学伯克利分校的一项科学实验，它使用联网的计算机搜索地外文明(SETI)。你可以通过运行一个免费的程序来下载和分析射电望远镜的数据

# Proof of Storage

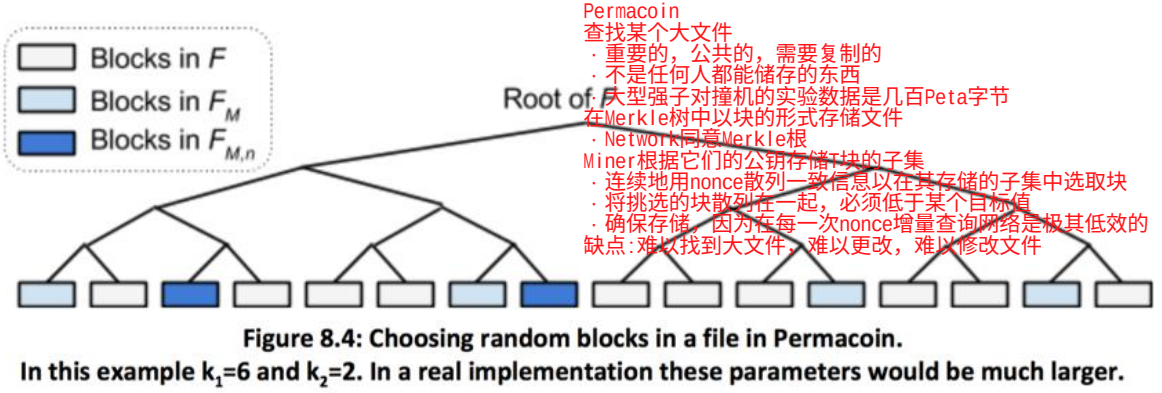



Figure 8.4: Choosing random blocks in a file in Permacoin.
In this example $k_1$=6 and $k_2$=2. In a real implementation these parameters would be much larger.

Princeton Textbook, Permacoin

## Permacoin

- **Find some large file**
  - Important, public, and in need of replication
  - Something that not any individual can store
  - Ex. Experimental data from Large Hadron Collider is several hundred Petabytes
- **Store file in blocks, in a Merkle tree**
  - Network agrees on the Merkle Root
- **Miner stores a subset of blocks of T, based off of their public key**
  - Continuously hash consensus information with nonce to pick blocks in their stored subset
  - Hash the picked blocks together, must be below some target value
  - Ensures storage, since querying network at every nonce increment is extremely inefficient
- **Drawbacks: Hard to find large file, to change difficulty, to modify file**

# Merge Mining

合并挖掘是指在不牺牲整体挖掘性能的情况下同时挖掘两种或更多加密货币的行为。从本质上讲，矿工可以使用他们的计算能力，通过所谓的辅助工作证明(AuxPoW)，在多个链上并发地开采区块。

Merged mining refers to the act of mining two or more cryptocurrencies at the same time, without sacrificing overall mining performance. Essentially, a miner can use their computational power to mine blocks on multiple chains concurrently through the use of what is known as Auxiliary Proof of Work (AuxPoW).

When launching an altcoin, you need hashpower to secure your network

- Mining is exclusive by default; can't work on two problems at once
- Mining altcoin => losing profits on other chains
  - Competition for hashpower
- Vulnerable to attacks from larger coins

"Altcoin infanticide"

- Individual Bitcoin miner/pool hashrate >> entire altcoin hashrate
- 2012: Eligius mining pool operator attacked CoiledCoin
  - Reversed multiple days' worth of transactions
  - Mined long chain of empty blocks

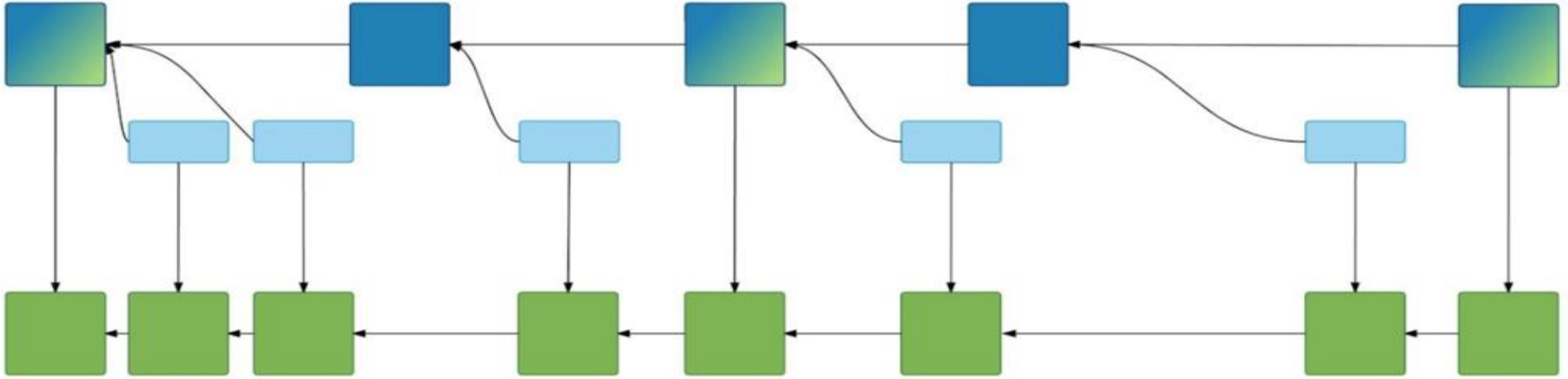
当启动altcoin时，您需要hashpower来保护您的网络
· 默认情况下，挖掘是独占的;不能同时解决两个问题
· 挖掘altcoin =>在其他链损失利润
  · hashpower竞争
· 容易受到较大硬币的攻击
“Altcoin杀婴 ”
· 比特币矿工/池hashrate>>整个altcoin hashrate
· 2012年:Eligius矿池操作员攻击CoiledCoin
  · 逆转了数天的交易
  · 开采出空块的长链

# Merge Mining

合并矿业
· 创建区块，该区块有从比特币和altcoin的交易
· 分享hashpower
· 实现
· 容易为altcoin -免费构建您的硬币，无论您喜欢
· 但是如何将altcoin交易包含在比特币中呢?
· 解决方案:在比特币的coinbase参数中包含altcoin交易摘要
· 摘要可以是altcoin交易的Merkle根
· 其他比特币客户并不在意
· Altcoin将尝试开采比特币区块
· 两个平行链

**Merge mining**

- **Create blocks that have transactions from both Bitcoin & the altcoin**
  - Share the hashpower
- **Implementation**
  - Easy for the altcoin - free to build your coin however you like
  - But how to include altcoin transactions in Bitcoin?
- **Solution: Include summary of altcoin transactions in Bitcoin's coinbase parameter**
  - Summary can be Merkle root of altcoin transactions
  - Other Bitcoin clients don't care
- **Altcoin will attempt to mine Bitcoin blocks**
  - Two parallel chains



Altcoin blocks

Bitcoin blocks mined by altcoin merge-miners

Bitcoin blocks mined by non-altcoin miners

Attempted Bitcoin blocks found by altcoin merge-miners that met the altcoin's difficulty target but not Bitcoin's target

让一切都回到交易上来

# Bringing it all together - Back to a transaction

- **I want to send money to Sunny**
  - Sign transaction
  - Broadcast to network
- **Miners receives transaction, adds to “zero-conf pool”**
  - Verify transaction: i.e. signature matches, enough money,
- **Miner finds PoW, broadcasts block**
  - Block propagates; others verify
- **Miners work on the next problem**

我想寄钱给桑尼
◯签署交易
◯广播到网络
矿机接收事务，添加到零conf池
◯验证交易:即签名匹配，足够的钱，
矿工发现PoW，广播区块
◯块传播;其他验证
矿工工作下一题

v5

Pending transactions

1. I, Tom, am giving Sue one bitcoin, with serial number 3920.
2. I, Sydney, am giving Cynthia one bitcoin, with serial number 1325.
3. I, Alice, am giving Bob one bitcoin, with serial number 1234.

# Overview

- ✓ **Bitcoin Concepts**
- ✓ **Consensus Build-up**
- ✓ **Mining Overview**
- ✓ **Cryptocurrency Mining**

# END !


धन्यवाद
Hindi
多謝
Traditional Chinese
ขอบคุณ
Thai
Спасибо
Russian
Gracias
Spanish
Thank You
English
شكراً
Arabic
Obrigado
Brazilian Portuguese
多谢
Simplified Chinese
Danke
German
Grazie
Italian
Merci
French
நன்றி
Tamil
Tamil
ありがとうございました
Japanese
감사합니다
Korean

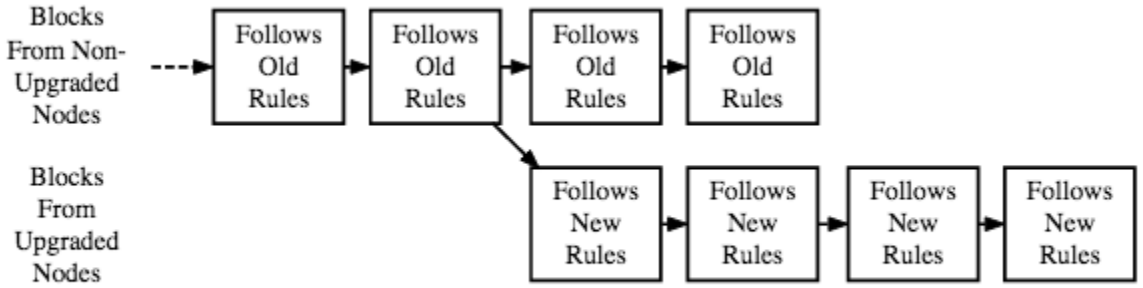
# Readings

- **"Bitcoin Wallets Explained"**
  - http://cryptorials.io/bitcoin-wallets-explained-how-to-choose-the-best-wallet-for-you/
- **"Bitcoin Multisig Wallet: The Future of Bitcoin"**
  - https://bitcoinmagazine.com/articles/multisig-future-bitcoin-1394686504
- **"What is a Bitcoin Hardware Wallet?"**
  - https://www.cryptocompare.com/wallets/guides/what-is-a-bitcoin-hardware-wallet/
- **(Optional) Vanity Bitcoin Addresses:**
  - https://www.cryptocoinsnews.com/get-custom-bitcoin-address/

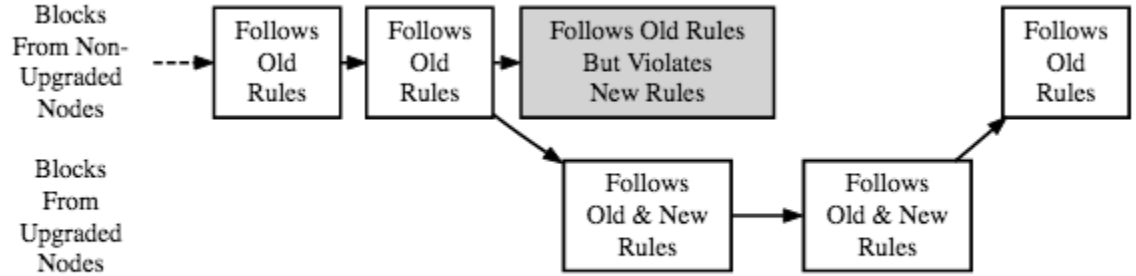
Source: Bitcoin.org Developer Guide

奖励:分叉+共识更新

# Bonus: Forking + Consensus Updates



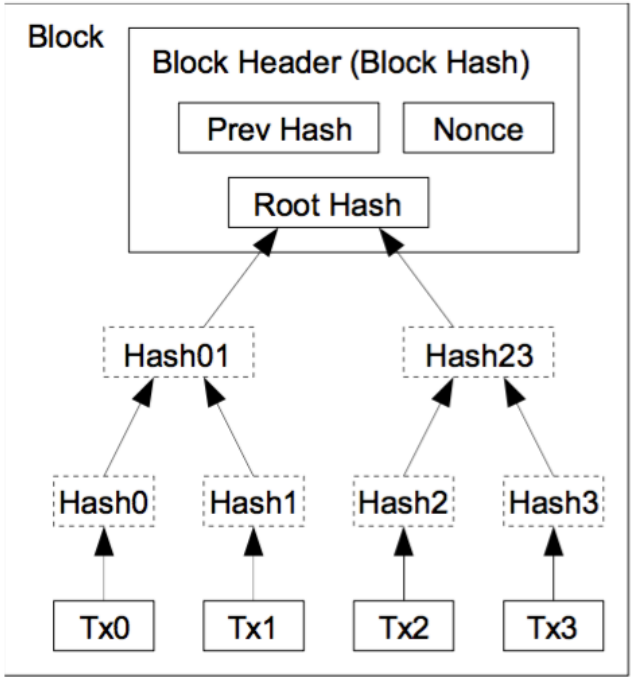

A Hard Fork: Non-Upgraded Nodes Reject The New Rules, Diverging The Chain



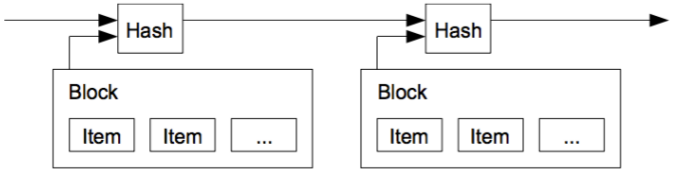

A Soft Fork: Blocks Violating New Rules Are Made Stale By The Upgraded Mining Majority

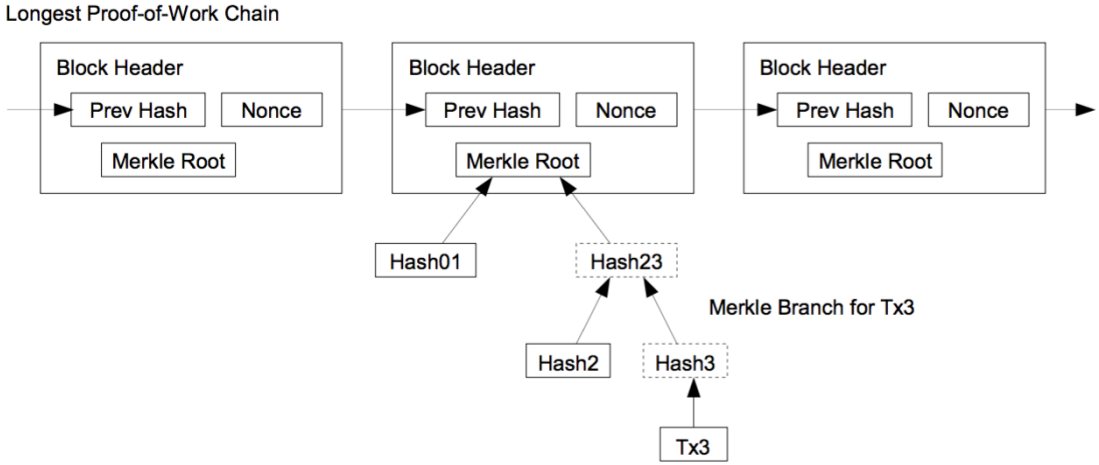
过期

# Bonus: Merkle Tree




Transactions Hashed in a Merkle Tree

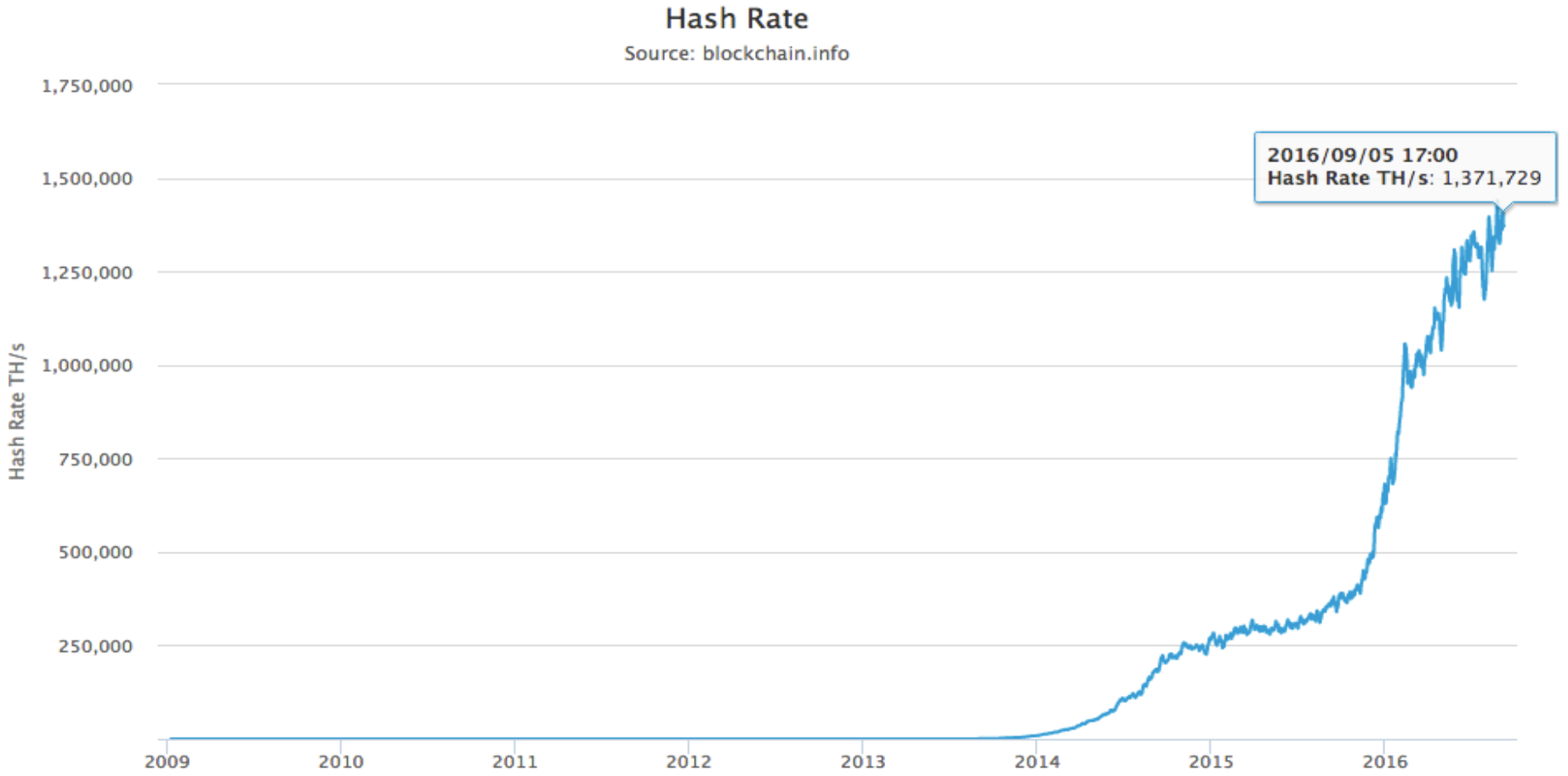



使交易历史不可变

- Makes transaction history immutable
- PoW to add chains

Source: Nakamoto 2009


Hash Rate
Source: blockchain.info
1,750,000
1,500,000
1,250,000
1,000,000
750,000
500,000
250,000
Hash Rate TH/s
2016/09/05 17:00
Hash Rate TH/s: 1,371,729
2009
2010
2011
2012
2013
2014
2015
2016