Verification of blockchains and smart contracts

Formal Methods Update, 2018
BITS Pilani Goa

Madhavan Mukund
Chennai Mathematical Institute
http://www.cmi.ac.in/~madhavan

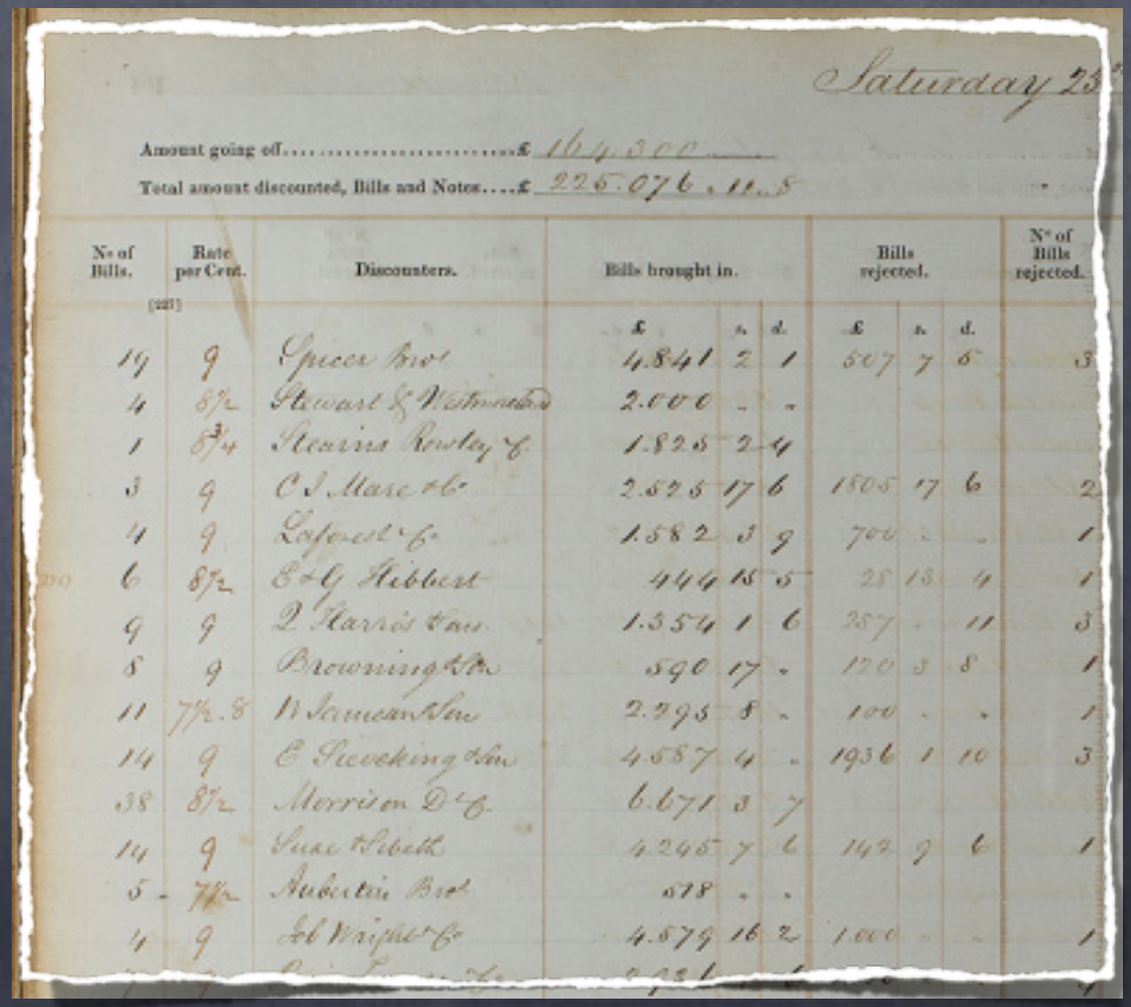
Outline

- Introduction to blockchains
- Smart contracts
- Verification issues

Introduction to blockchains

Banks and ledgers

- Record of all transactions
- Maintained by a trusted authority
- Each entry is validated
- Compute net balance etc



Saturday 25

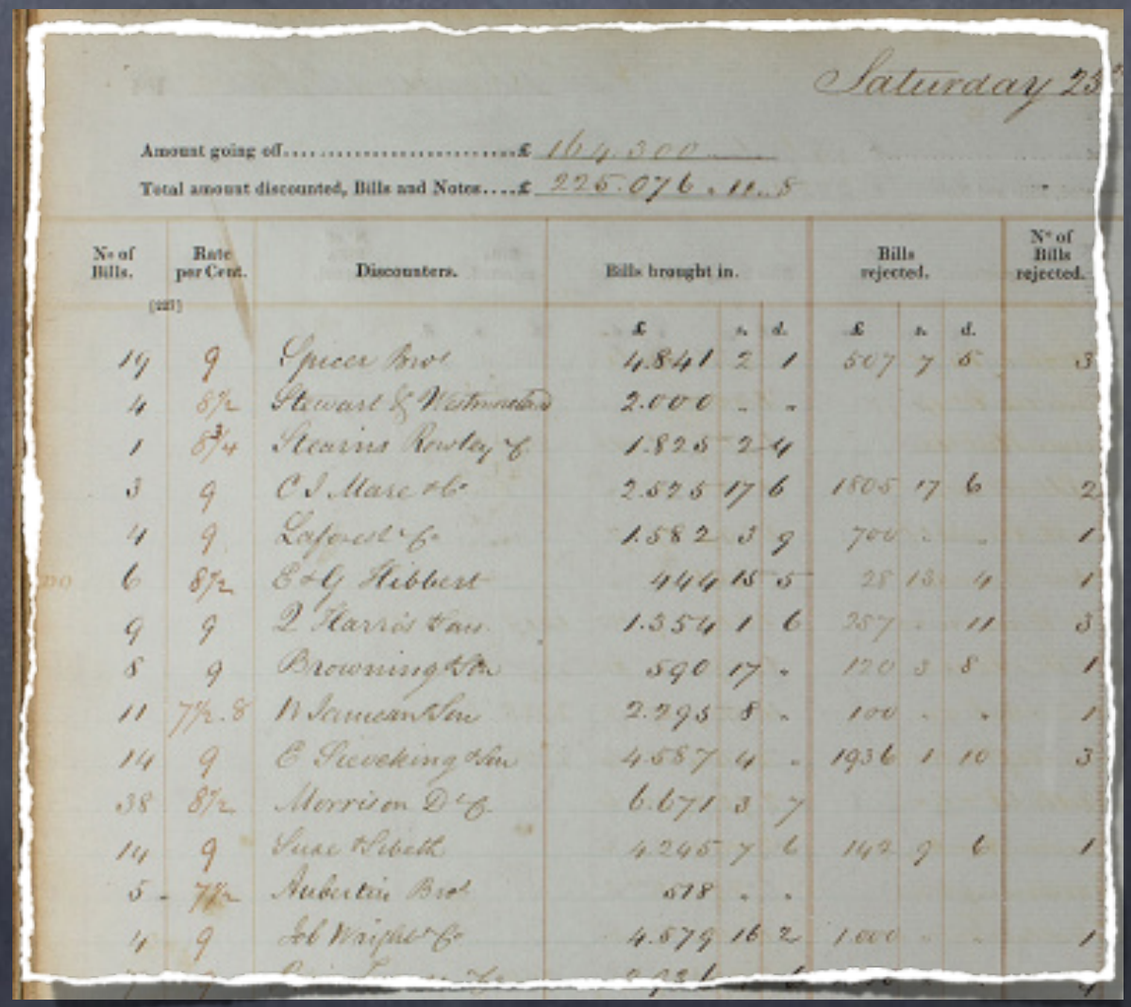
Amount going off..... £ 164 300

Total amount discounted, Bills and Notes.... £ 226 076 . 11 . 8

N° of Bills.	Rate per Cent.	Discounters.	Bills brought in. £	s.	d.	Bills rejected. £	s.	d.	N° of Bills rejected.
(221)									
14	9	Spicer Bro'	4841	2	1	507	7	6	3
4	8½	Stewart & Watmough	2.000	-	-				
1	8¾	Stearns Rowley &	1825	2	4				
3	9	C J Mare & C°	2.575	17	6	1805	17	6	2
4	9	Lafond & C°	1582	3	9	700	-	-	1
6	8½	B & G Hibbert	446	18	5	28	13	4	1
9	9	J Harris & Son	1.354	1	6	287	-	11	3
8	9	Browning & C°	590	17	-	120	3	8	1
11	7½.8	W Jameson & Son	2.295	8	-	100	-	-	1
14	9	E Steeking & Son	4587	4	-	1936	1	10	3
38	8½	Morrison D° & C°	6.671	3	7				
14	9	Isaac Hibbert	4.295	7	6	142	9	6	1
5	7½	Auberlin Bro'	578	-	-				
4	9	Jos Wright & C°	4.879	16	2	1000	-	-	1

Public ledgers

- Ledgers are private
- Can we maintain a public ledger?
- Eliminate trusted authority



Saturday 23

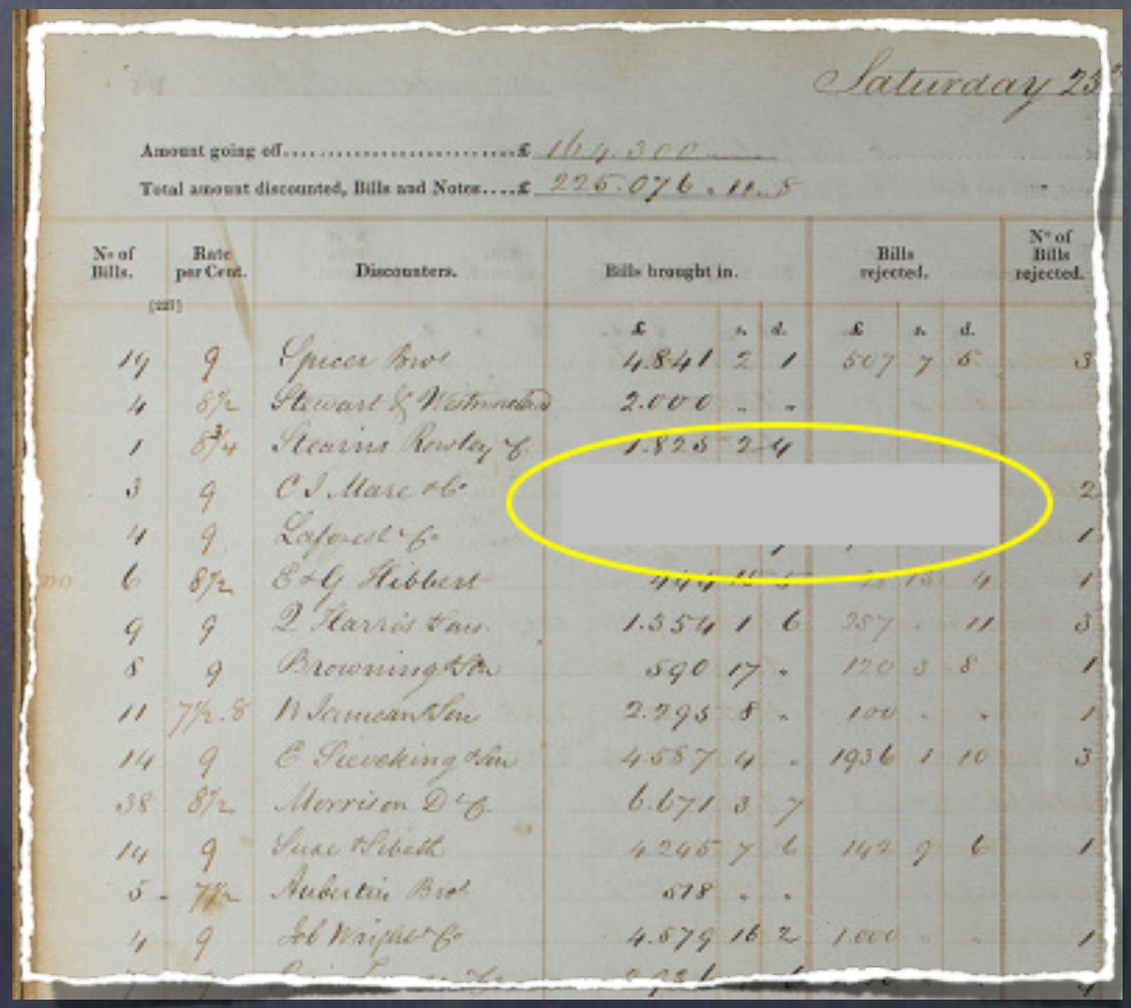
Amount going off..... £ 164 300

Total amount discounted, Bills and Notes.... £ 226 076 . 11 . 8

N° of Bills.	Rate per Cent.	Discounters.	Bills brought in. £	s.	d.	Bills rejected. £	s.	d.	N° of Bills rejected.
(221)									
14	9	Spicer Bro'	4.841	2	1	507	7	6	3
4	8½	Stewart & Westmacott	2.000	-	-				
1	8¾	Stearns Rowley &	1.825	2	4				
3	9	C J Marc & C°	2.575	17	6	1505	17	6	2
1	9	Lafond & C°	1.582	3	9	700	-	-	1
6	8½	E & G Hibbert	444	18	5	28	13	4	1
9	9	J Harris & Son	1.354	1	6	257	-	11	3
8	9	Browning & Co	590	17	-	120	3	8	1
11	7¾.8	N Sanuel & Co	2.295	8	-	100	-	-	1
14	9	E Sieveking & Co	4.587	4	-	1936	1	10	3
38	8½	Morrison D° & C°	6.671	3	7				
14	9	Isaac Hibbert	4.245	7	6	142	9	6	1
5	7¾	Aubertin Bro'	578	-	-				
1	9	Jd Waipler & C°	4.579	16	2	1000	-	-	1

Challenges

- Integrity of individual transactions
- Consensus on overall set of transactions



Saturday 25

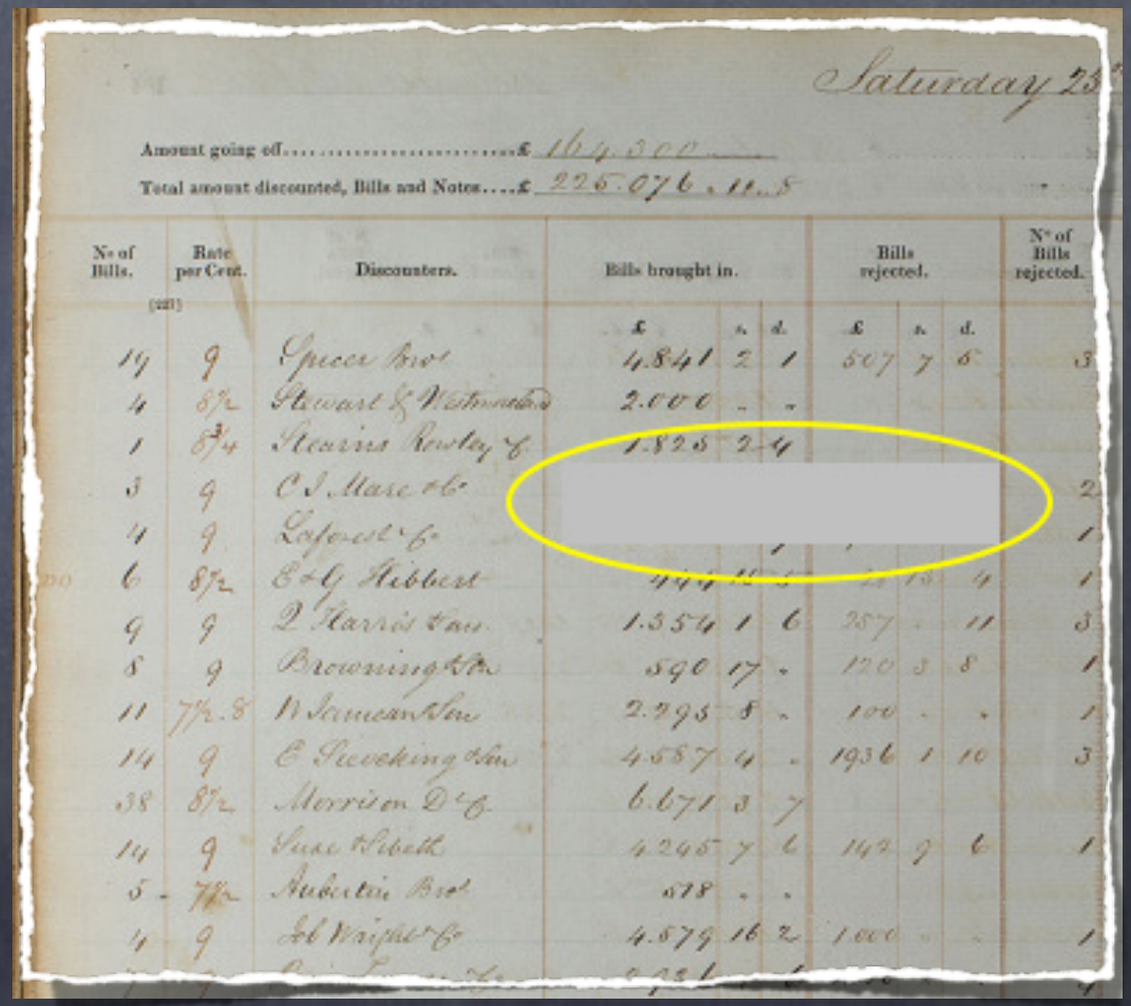
Amount going off..... £ 164 300

Total amount discounted, Bills and Notes.... £ 226.076.11.8

N° of Bills.	Rate per Cent.	Discounters.	Bills brought in. £	s.	d.	Bills rejected. £	s.	d.	N° of Bills rejected.
(221)									
14	9	Spicer Bro'	4.841	2	1	507	7	6	3
4	8½	Stewart & Vetmore's	2.000	-	-				
1	8¾	Stearns Rowley &	1.825	2	4				
3	9	C J Marc &							2
1	9	Lafoest &							1
6	8½	B & Hibbert							1
9	9	D Harris & Son	1.354	1	6	287	-	11	3
8	9	Browning & Co	590	17	-	120	3	8	1
11	7½.8	N Duncan & Co	2.795	8	-	100	-	-	1
14	9	C Siecking & Co	4.587	4	-	936	1	10	3
38	8½	Morrison & C	6.671	3	7				
14	9	Isaac Hibbert	4.245	7	6	149	9	6	1
5	7½	Aubertin Bro'	578	-	-				
4	9	Jd Wright & C	4.879	16	2	1000	-	-	1

A solution

- Maintain a distributed ledger
- Duplication prevents tampering
- Cryptography for authentication



Saturday 25

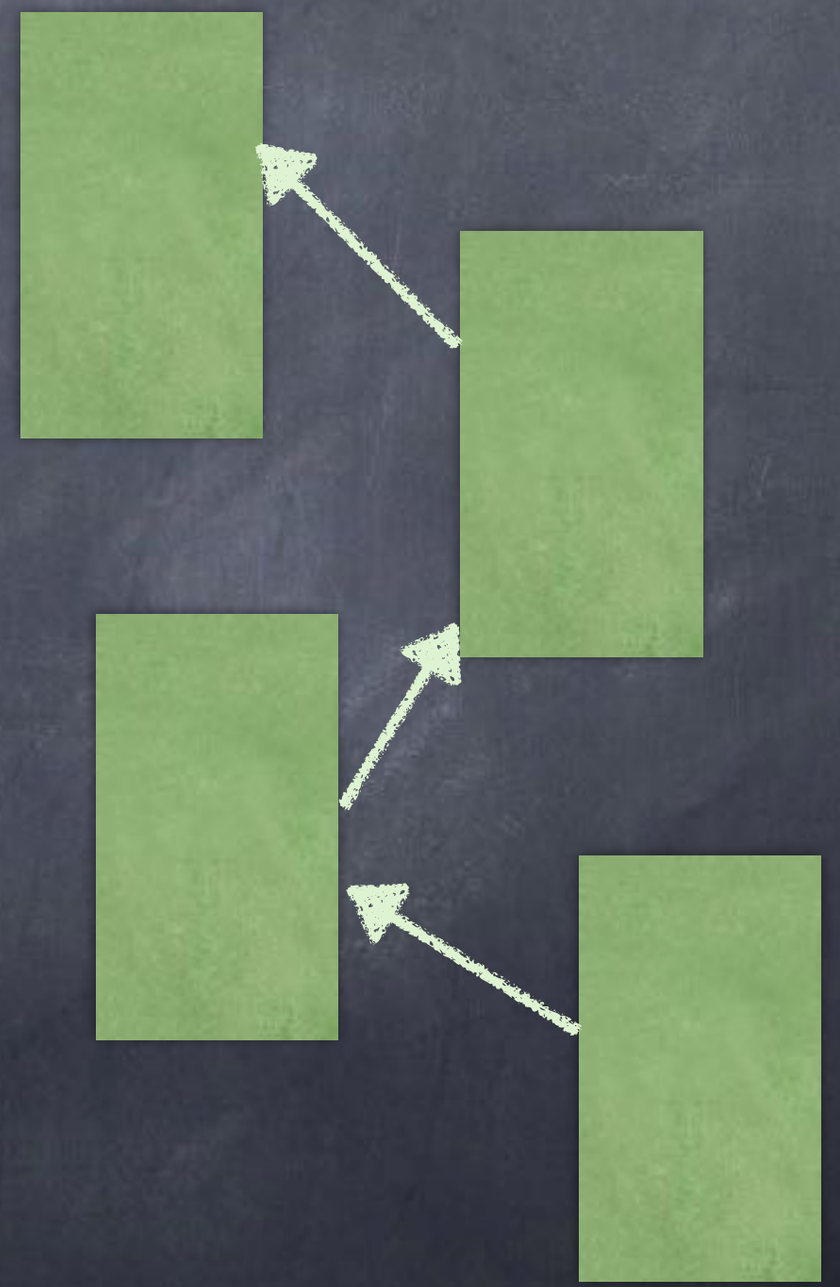
Amount going off..... £ 164,300

Total amount discounted, Bills and Notes.... £ 226,076. 11. 8

N° of Bills.	Rate per Cent.	Discounters.	Bills brought in. £	s.	d.	Bills rejected. £	s.	d.	N° of Bills rejected.
(221)									
14	9	Spicer Bro^s	4.841	2	1	507	7	6	3
4	8½	Stewart & Nethmudas	2.000	-	-				
1	8¾	Stearns Rowley &	1.825	2	4				
3	9	C J Marc & C^o							2
4	9	Lafond & C^o							1
6	8½	E & G Hibbert							1
9	9	D Harris & Son	1.554	1	6	257	-	11	3
8	9	Browning & C^o	590	17	-	120	3	8	1
11	7½.8	N Scruncton & C^o	2.795	8	-	100	-	-	1
14	9	C Steeking & Son	4.587	4	-	1936	1	10	3
38	8½	Morrison D^o & C^o	6.671	3	7				
14	9	Isaac Hibbert	4.245	7	6	149	9	6	1
5	7½	Asbertis Bro^s	578	-	-				
4	9	Ed Wright & C^o	4.879	16	2	1000	-	-	1

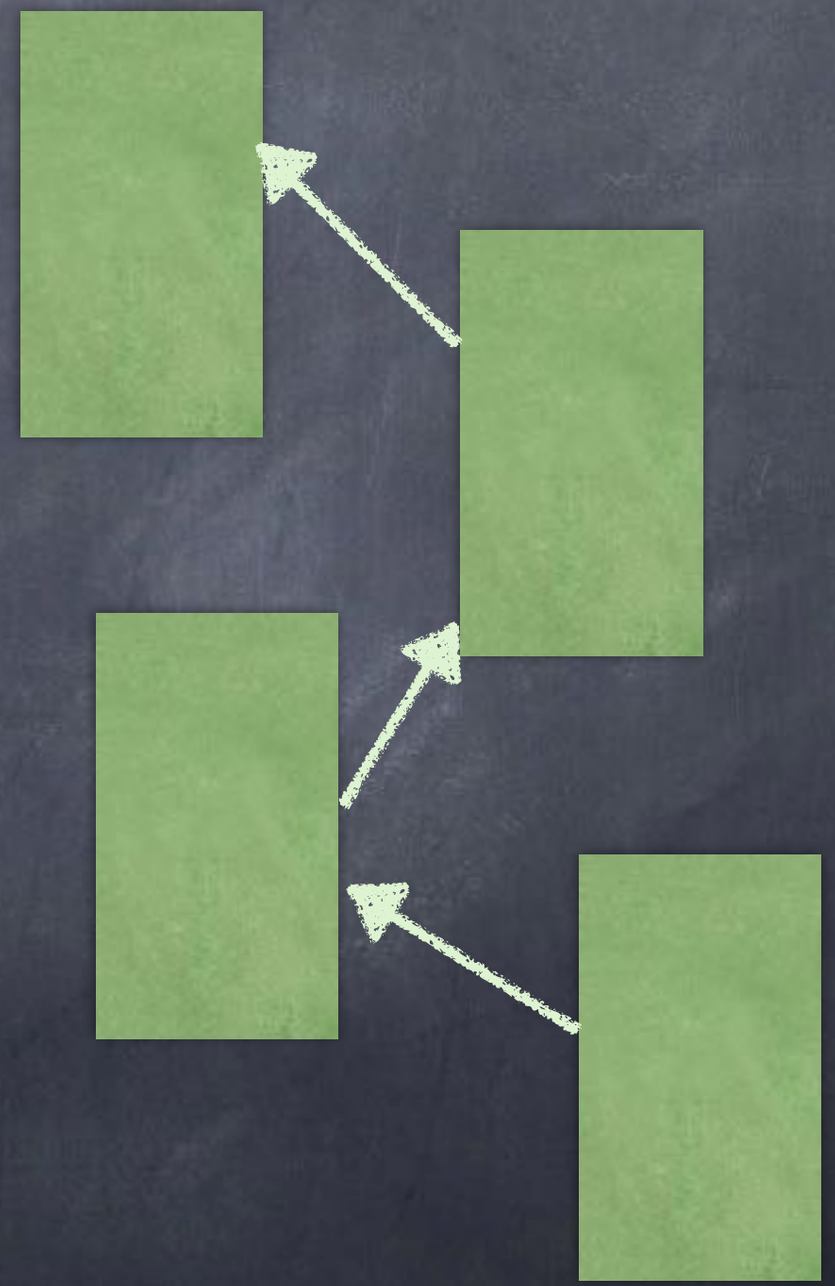
A solution

- A physical ledger has pages
- Distributed version has blocks of data
- These blocks are linked together
- Blockchain!



Blocks

- Each block is a collection of transactions
- Each block points to parent block



Hash function

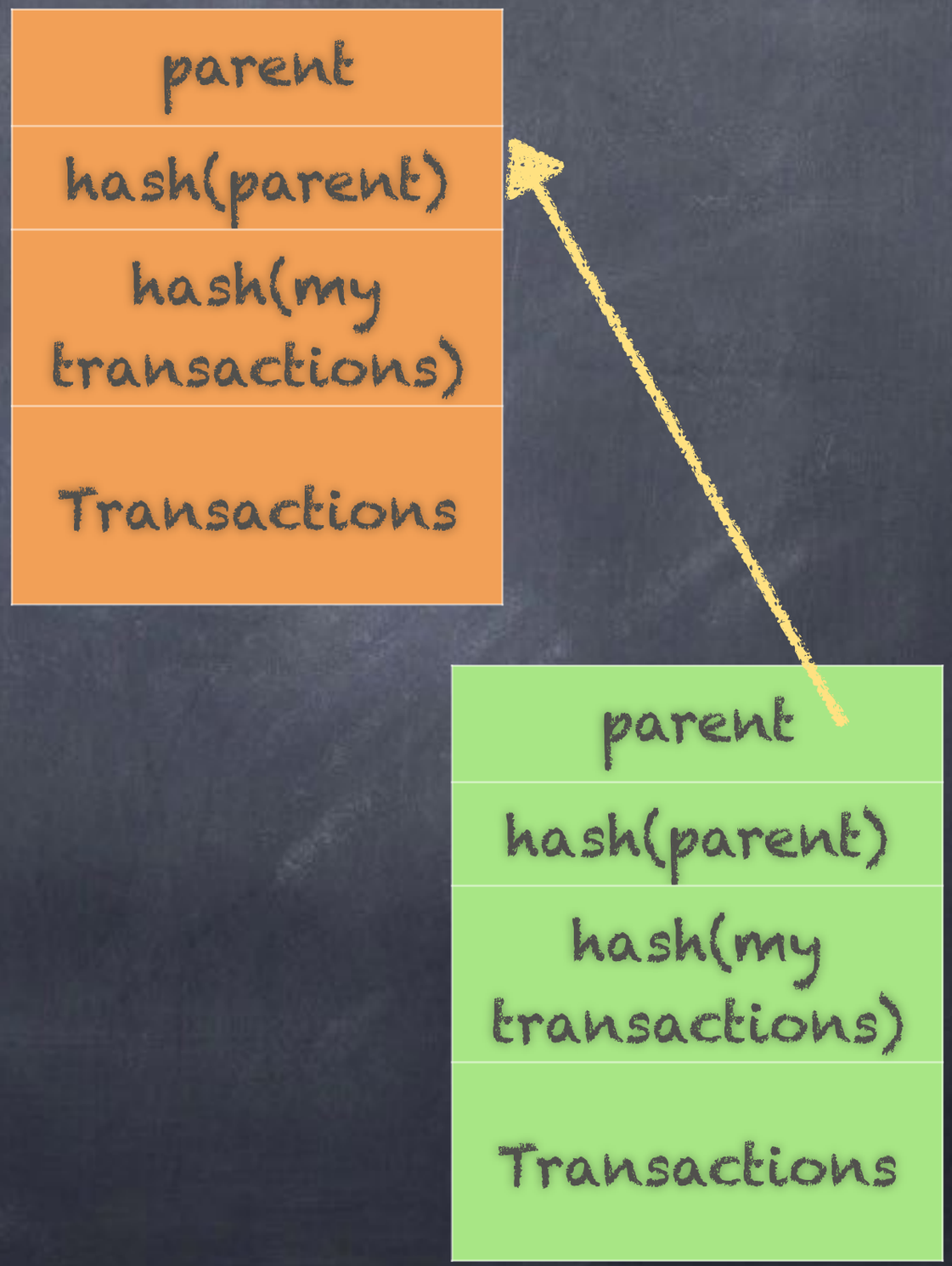
- Compute random summary of input
  - "Impossible" to invert
- Collisions rare
  - Different inputs produce different outputs

The quick brown fox jumps over the lazy dog.

0d7006cd055e94cf
614587e1d2ae0c8e

Blockchain integrity

- Each block has a hash of the transactions it contains
- Each block includes a hash of parent block



Public key cryptography

- Each person P has a public key U and a private key R
  - U and R are inverses

- To encrypt text t for P to read, send U(t)
- R(U(t)) = t

The quick brown fox jumps over the lazy dog.

↓ U

0d7006cd055e94cf
614587e1d2ae0c8e

↓ R

The quick brown fox jumps over the lazy dog.

Digital signatures

- U and R are inverses
  - R(U(t)) = t
    - Also, U(R(t)) = t !!
- Sign using R
- Recipient can verify using U

Madhavan Mukund

R

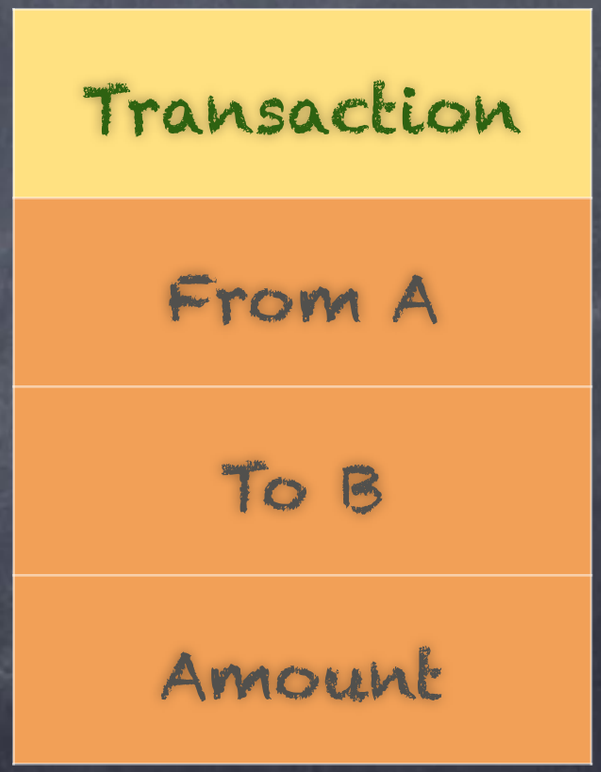
0d7006cd055e94cf
614587e1d2ae0c8e

U

Madhavan Mukund

Transactions

- Who writes the transactions in the blockchain?
- No centralised authority
- Transactions are created by originator



Transactions

- A digitally signs
  - Cannot repudiate later
- A uses B's public key to create a challenge only B can solve
  - Only B can claim this amount

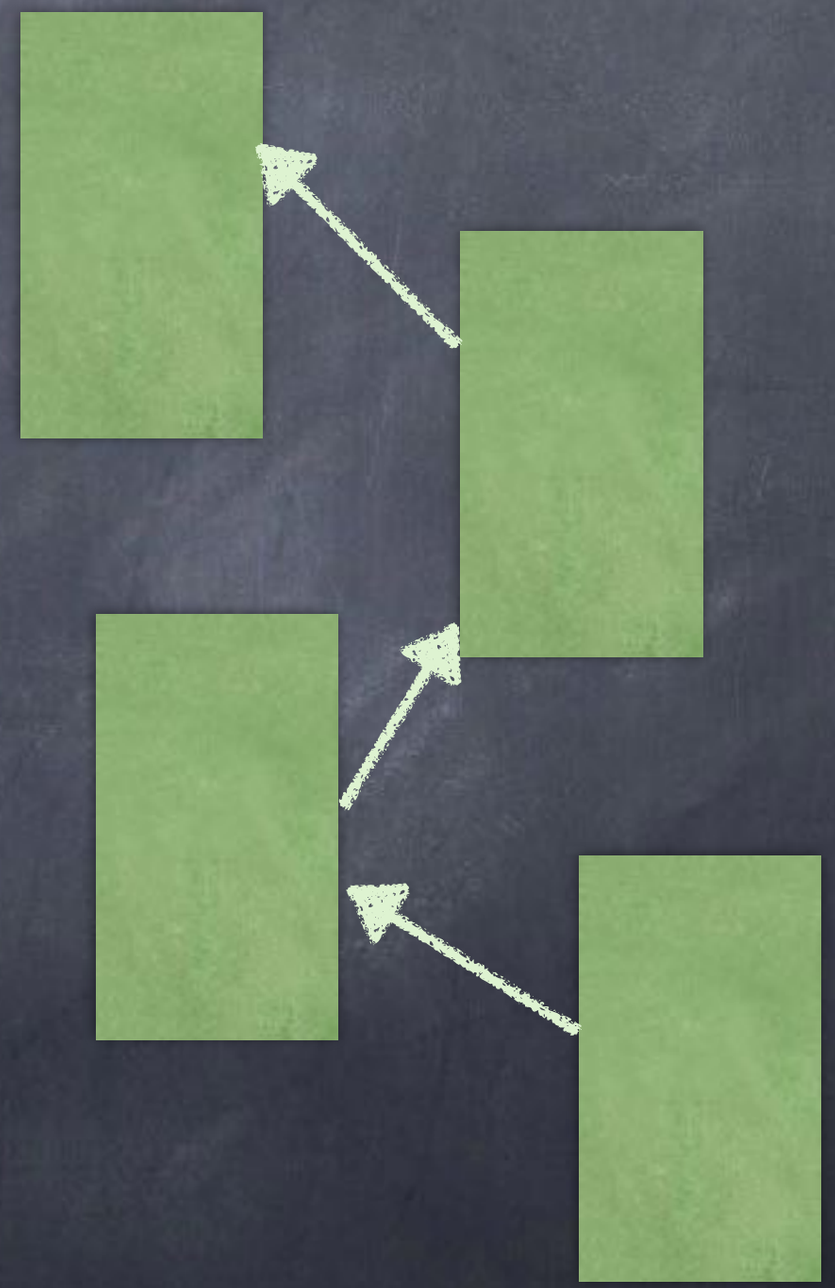
Transaction
From A
Dig Sig of A
To B
Challenge
Amount

Transactions

- Where's the money?
- No centralised authority to certify the money A holds
- Must refer to previous transactions where A acquired the money

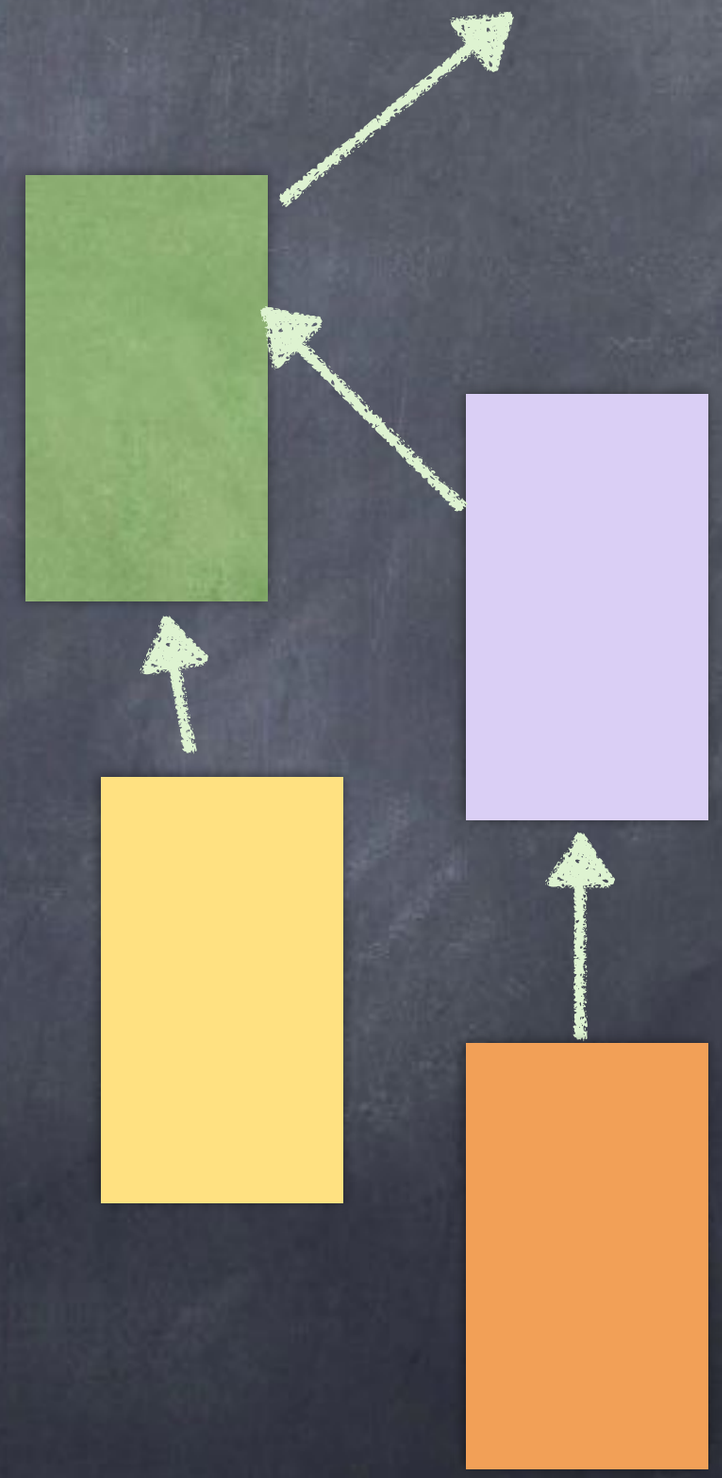
Adding blocks

- Peer to peer network
- Transactions broadcast to all nodes
- Periodically, collect transactions into a block and add to chain



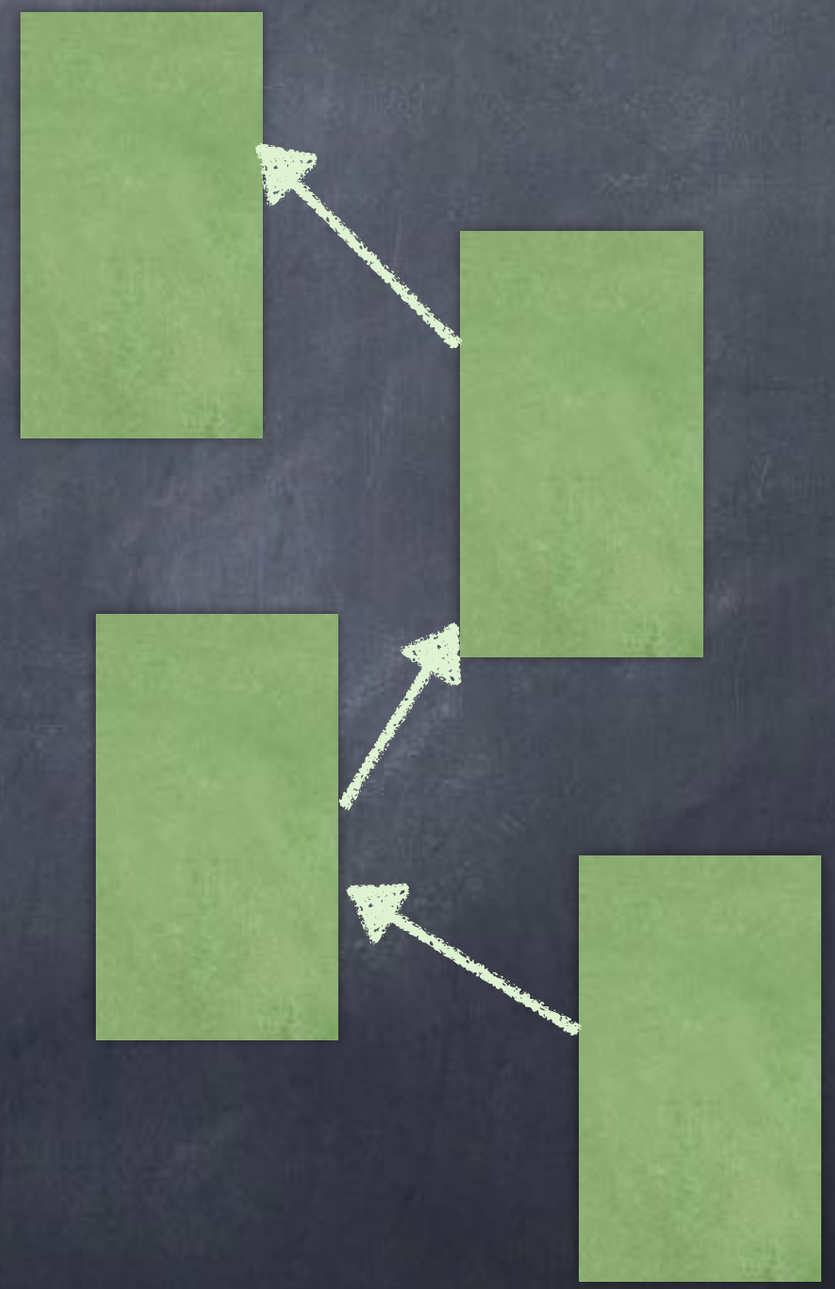
Mining blocks

- Process of adding a block is called **mining**
- Mining is decentralised
- Blockchain may **fork**
- Integrity of the ledger is lost!



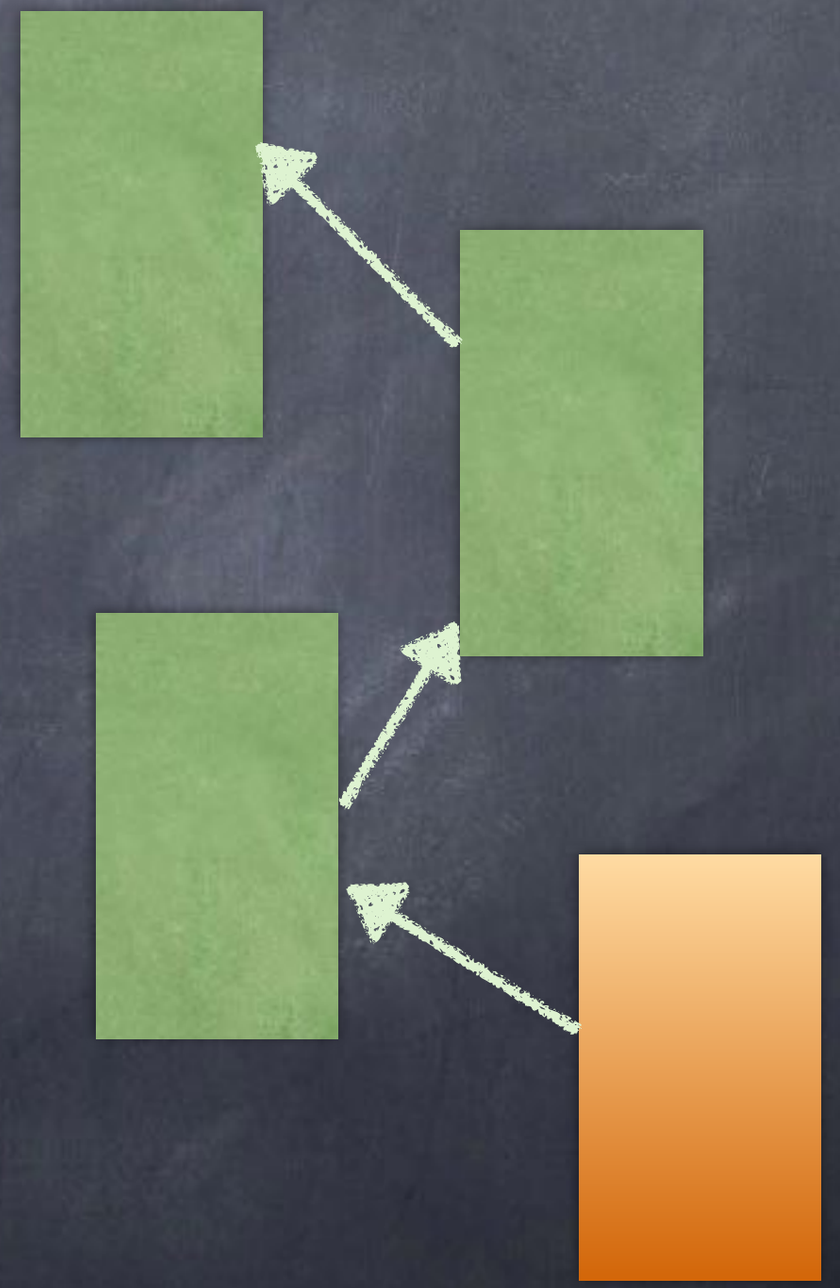
Distributed consensus

- All nodes should agree on blocks
- Elegant solution due to Satoshi Nakomoto
- Emerging distributed consensus



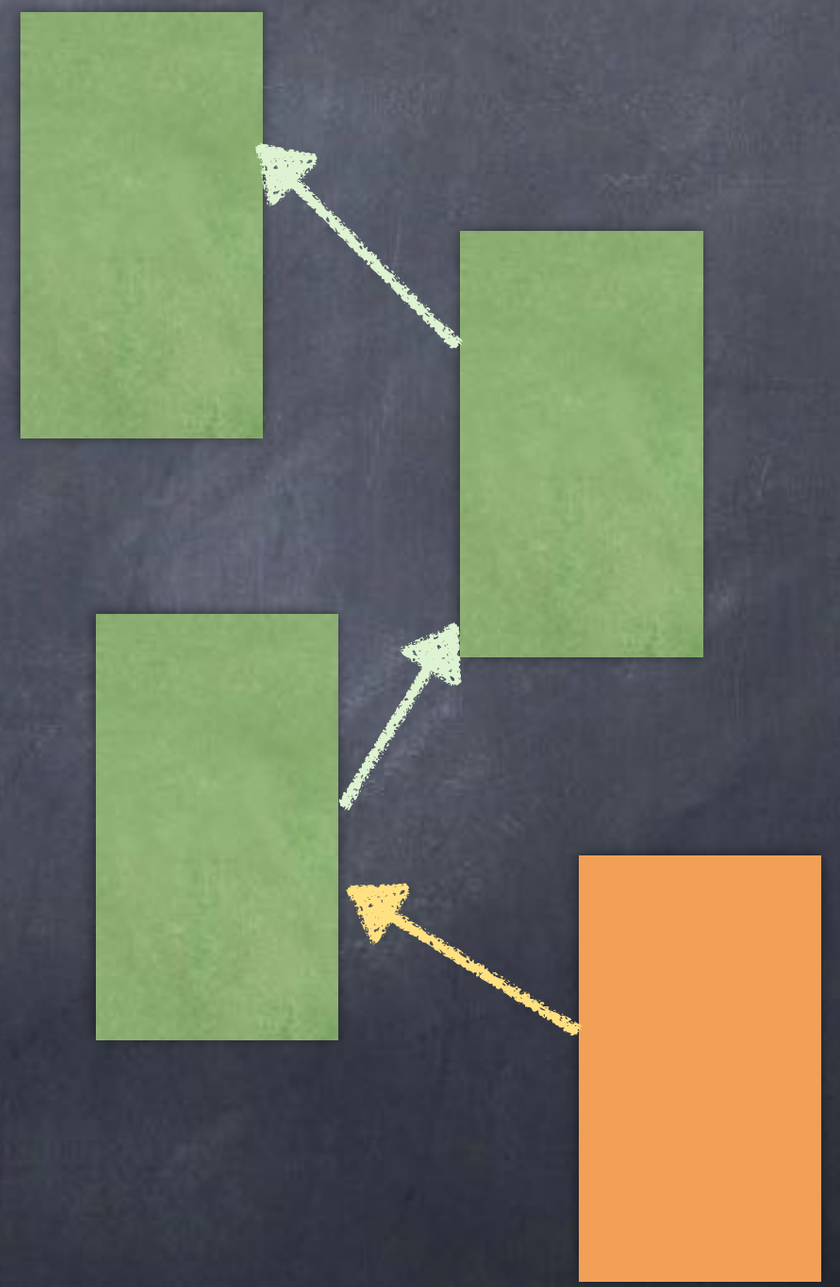
Proof of work

- Adding a node requires solving a hashing problem
  - Brute force search
- Calibrated so that it takes about 10 minutes to solve on current hardware



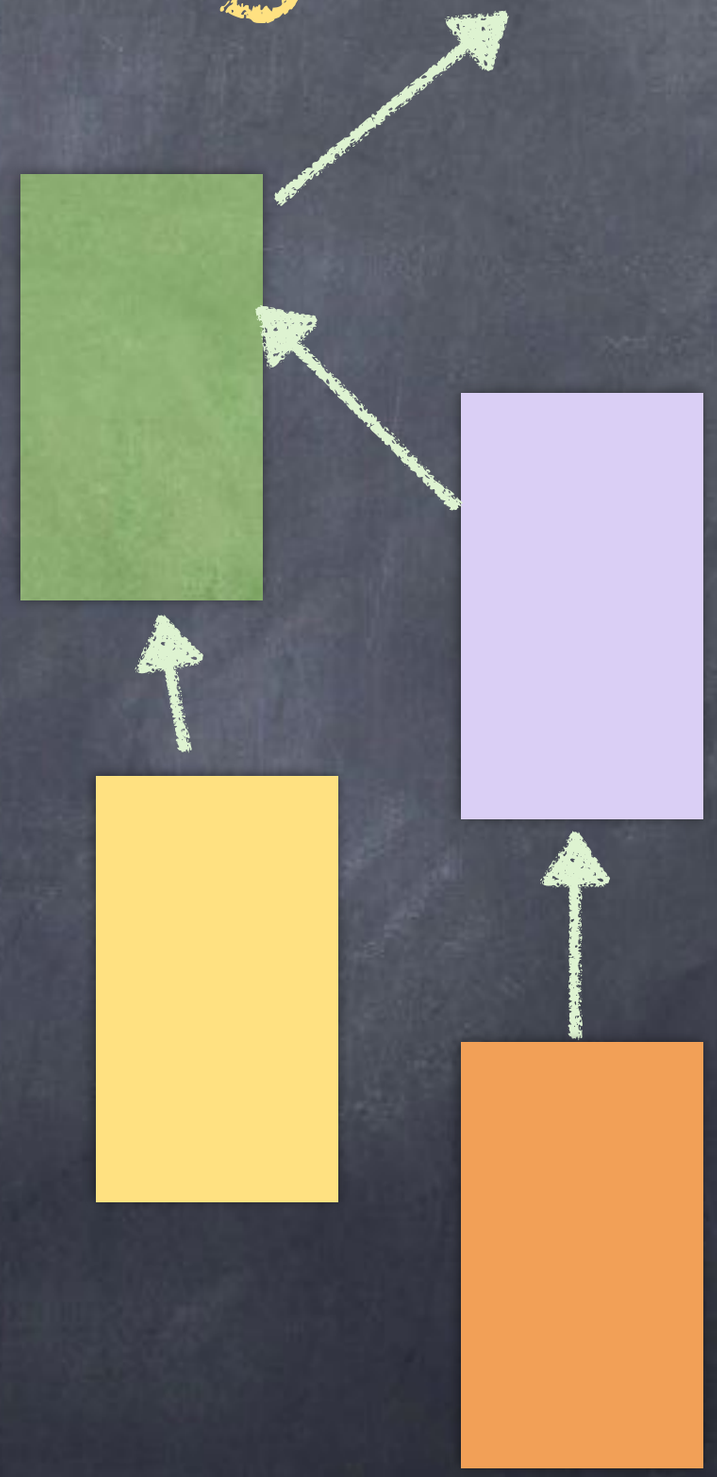
Proof of work

- After mining a block, miner broadcasts
- Other miners abandon efforts, accept this block, move to next block
  - Serial numbers



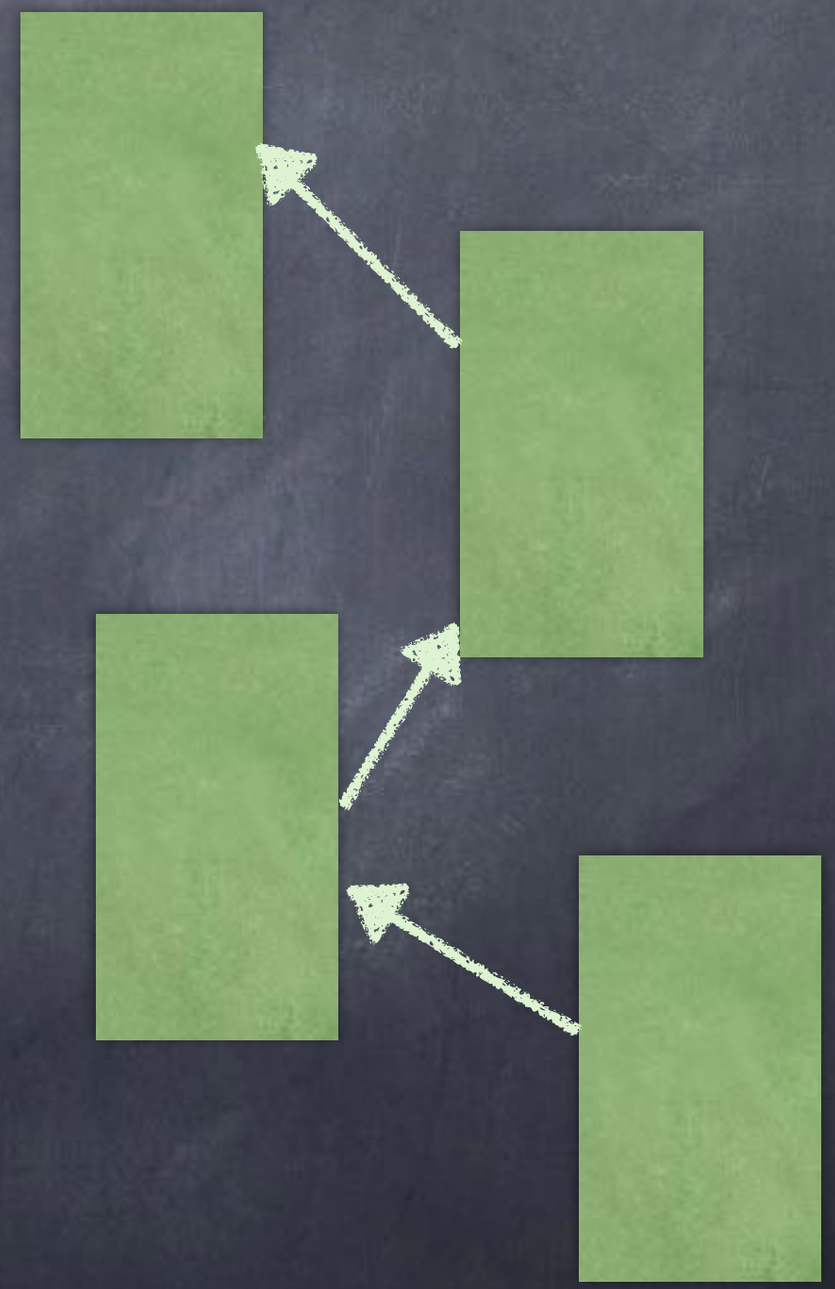
Blockchain forking

- Two miners may succeed in parallel
- Variants of chain may propagate
- Mismatch between your chain and new block – keep longer chain
- Eventually converges



Incentive for mining

- Why spend computational effort to mine?
- Transaction fees and other incentives
- Bitcoin!



Smart contracts

Transactions

- A uses B's public key to create a challenge only B can solve
  - Only B can claim this amount
- How is this done?

Challenge scripts

- Simple stack based programming language
- Locking script

```
DUP HASH160 <PubKHash> EQUALVERIFY CHECKSIG
```

  - <PubKHash> – hash of B's public key
- Unlocking script

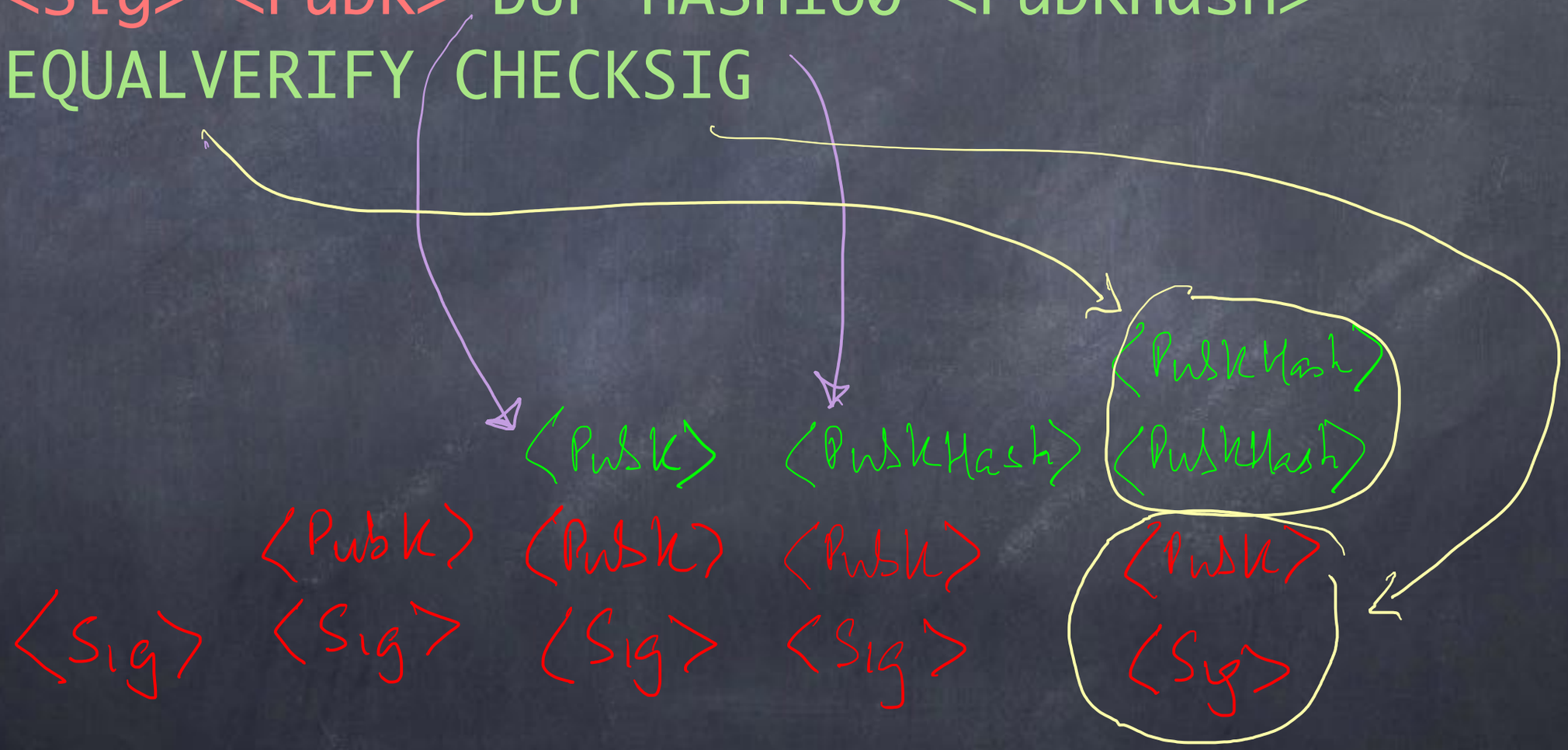
```
<Sig> <PubK>
```

  - <Sig> <PubK> – signature, public key of B

Challenge scripts ...

- Concatenate and execute on stack VM

```
<Sig> <PubK> DUP HASH160 <PubKHash>
EQUALVERIFY CHECKSIG
```



More general scripts

- Multisignature
  - N public keys recorded in the script
    - M must provide signatures to unlock
- Conditional
  - Three partners, majority must sign
  - Lawyer can access with one partner

Scripting Language

- Bitcoin
  - Scripting language is intentionally Turing incomplete
    - Conditionals, but no loops
- Ethereum
  - Richer language, Turing complete
  - High level language **Solidity** that compiles down to stack language

Smart contracts

- A script that executes when a transaction is invoked
- Ethereum contracts can express objects with encapsulated state
- Example: DAO
  - Decentralized Autonomous Organisation

Verification

Blockchain convergence

- Proof of work – eventually convergent solution to distributed consensus
  - Ensures blockchain does not fork
  - Need majority collusion to fabricate alternate chain
    - Would allow double spending

Vulnerability

Hijacking Bitcoin: routing attacks on cryptocurrencies, Apostolaki et al, IEEE Security and Privacy 2017

- Structure of Internet is not uniform
- Concentration of switches, routers make partitioning possible
- Can also delay packets

Model checking

Modeling and Verification of the Bitcoin Protocol, Chaudhury et al, MARS Workshop 2015

- UPPAAL model of Bitcoin network
- Investigate forking, double spending
- Model checking of a very small scale model, 4 nodes, 1 malicious

Smart contract verification

Online Detection of Effectively Callback Free Objects with Applications to Smart Contracts, Grossman et al, POPL 2018

- Decentralized Autonomous Organisation
- DAO bug stole $150 million dollars
- Reentrant code (callbacks)
- Automatic verification of effectively callback free objects

DAO

```
Object Dao

  Map <Object,int> credit
  int balance

  Invariant
  (sum o: credit[o]) = balance
```

```
Method
withdrawAll(Object o)

  if (credit[o] > 0)
    this.balance -=
        credit[o]
    o.pay(credit[o])
    credit[o] = 0
```

```
Method
deposit(Object o,
        int amount)

  credit[o] += amount
  balance += amount
```

DAO attack

```
Object Attacker

  Object Dao
  bool stop = false
  int balance

Method pay(int profit)

  this.balance +=
    profit

  if (!stop)

    stop = true
    Dao.
     withdrawAll(this)
    stop = false
```

Call back

```
Method
withdrawAll(Object o)

  if (credit[o] > 0)
    this.balance -=
        credit[o]
    o.pay(credit[o])
    credit[o] = 0
```

```
Method
deposit(Object o,
        int amount)

 credit[o] += amount
 balance += amount
```