#### Blockchains & Cryptocurrencies

#### **Mechanics of Bitcoin**



Instructor: Matthew Green
Johns Hopkins University - Fall 2020

# Housekeeping

- Pace of material?
- Office hours
- Al is due 9/24 | 1:59pm Baltimore time
- Project groups and proposal is due 10/8 end of day
  - Project ideas up on main Github Wiki page

## News?

# Today

- Finish talking about Bitcoin at a high level
- Talk about the low-level mechanics of real Bitcoin

#### Bitcoin: A Peer-to-Peer Electronic Cash System

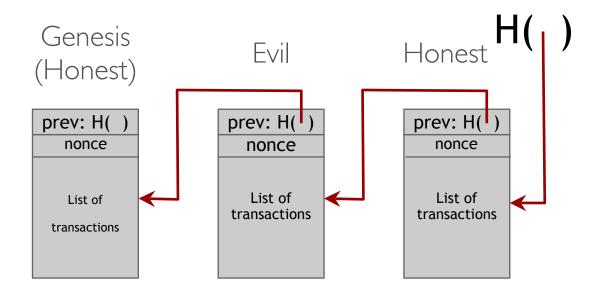
Satoshi Nakamoto smochin@graz.com www.bicein.org

Abstract. A purely pecenta-near version of electronic cosh would allow poline payments to be sent discully from one party to enables without going flavough a financial institution. Digital signatures provide part of the solution, but the main limities are less if a functed third party is will required to pursual disclosurations. We propose a solution to the double-speculing problem using a poer to peer network. The network timescence transactions by lasting these into an engoing than of hadrituned genoti-strate, familing a record that cames to changed without redesing the proof-of-work. The longest chain not only serves as proof of the sequence of wrints witnessed, but proof that is came from the largest paid of CPU power. As long as a majority of CPU power is controlled by node: that are not congentring to stack the arrowsk, they'll generate the languar thair and compact arackers. The retresh itself requires minimal structure. Messages are boundard on a best effort basic, and notice can leave and rejoin the network at will, accepting the longer proof-of-work thair as proof-of-when happened while they were gene.

#### 1. Introduction



Review



## Selecting T (Bitcoin)

Every block contains a (packed) encoding of T (target) which corresponds to "difficulty" (d)

```
d = ((2^{16} - 1) * 2^{8*26}) / T < relationship between d, T
```

#### Goals:

- Bitcoin block time should average 10 minutes
- 2016 blocks \* 10 minutes == 2 weeks
- Everyone in the network agrees on T
- Hence must be a function of chain data

This brings back a notion of time

Sometimes two separate nodes find a valid solution simultaneously

This can result in network partition



Image CC-BY-3 Theymos taken from the Bitcoin wiki

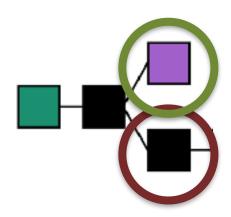


Image CC-BY-3 Theymos taken from the Bitcoin wiki

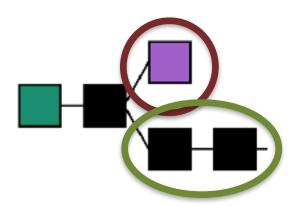
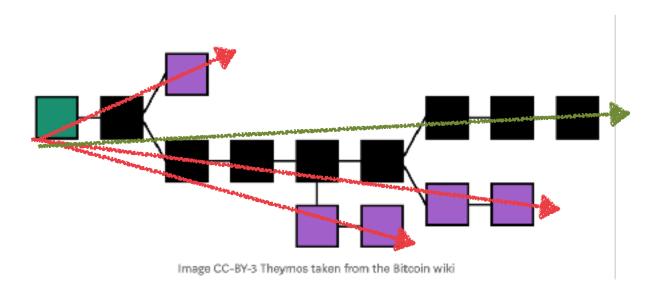
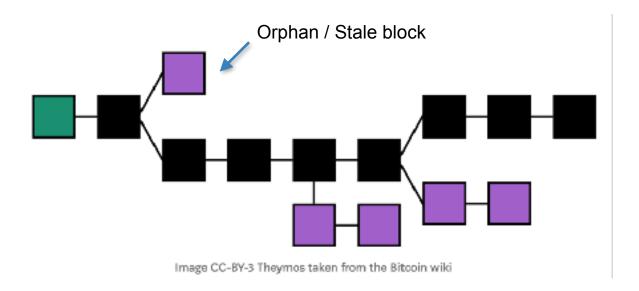


Image CC-BY-3 Theymos taken from the Bitcoin wiki

# "Longest chain rule"



# "Longest chain rule"



#### This is good and bad

Good: if we experience a "chain fork" and the network is connected (I.e., not totally partitioned), then eventually we will learn about both forks

Good: if the "hash power" behind the two chains is unequal, we will <u>probably</u> end up with one chain getting longer

Even if the hash power is equal, the inherent randomness of the puzzle (PoW) will likely cause one chain to advance

As one chain grows longer, other nodes will adopt it, and start adding to it

#### What's the bad?

#### What's the bad?

When a chain becomes longer than the "current chain" a node thinks is the longest chain, they must abandon that older chain

### **Finality**

"Finality is the assurance or guarantee that cryptocurrency transactions cannot be altered, reversed, or canceled after they are completed."

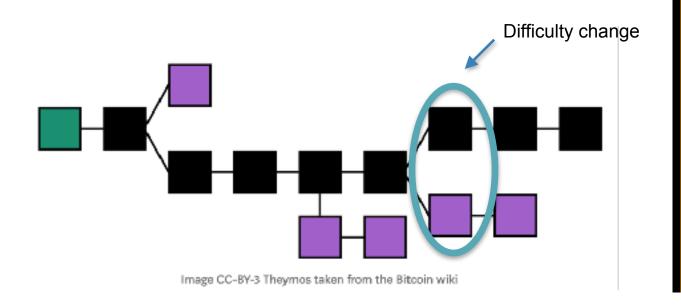
## **Finality**

"Finality is the assurance or guarantee that cryptocurrency transactions cannot be altered, reversed, or canceled after they are completed."

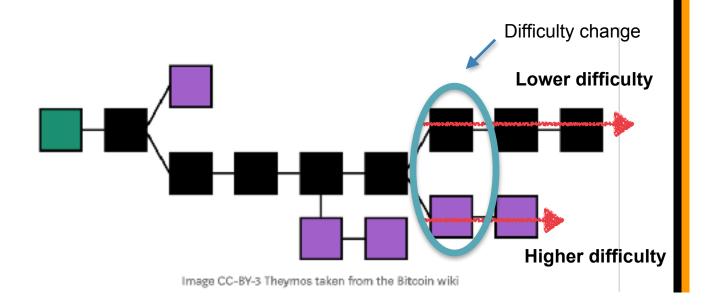
Bitcoin's finality is <u>probabilistic</u> (and computational)

Reorganizations become less probable (and more expensive) over time, but they never become impossible\*

# Q: What if difficulty changes?



# Q: What if difficulty changes?



# A: "Chain with most hashwork"

Bitcoin doesn't exactly use the longest chain rule

Instead, it employs a calculation that takes <u>block</u> <u>difficulty</u> into account

Each block has a difficulty. Convert to expected # of hashes to find block. Total these values. Chain with largest total is "longest".

Most of the time, this is equivalent to longest chain

#### How many blocks can the adversary make?

Consider an adversary that controls a r-fraction of the hash power

How many blocks in expectation can they build in a t-block window?

What is the probability that they dominate that t-block window?

We will see some attacks that leverage these simplified calculations later....

# How do we incentivize mining?

# How do we incentivize mining?

Two answers to this question in Bitcoin:

- 1. "Transaction fees" Each transaction has a "tip" that can be collected by the node who mines it into a block (incentivizes inclusion of transactions)
- 2. "Block reward" 50/25/12.5/... BTC made from scratch (in a special Coinbase transaction) and given to the miner

Bitcoin transactions

# An account-based ledger (not Bitcoin)

time

Create 25 coins and credit to Alice ASSERTED BY MINERS

Transfer 17 coins from Alice to Bob<sub>SIGNED(Alice)</sub>

Transfer 8 coins from Bob to Carol<sub>SIGNED(Bob)</sub>

Transfer 5 coins from Carol to Alice<sub>SIGNED(Carol)</sub>

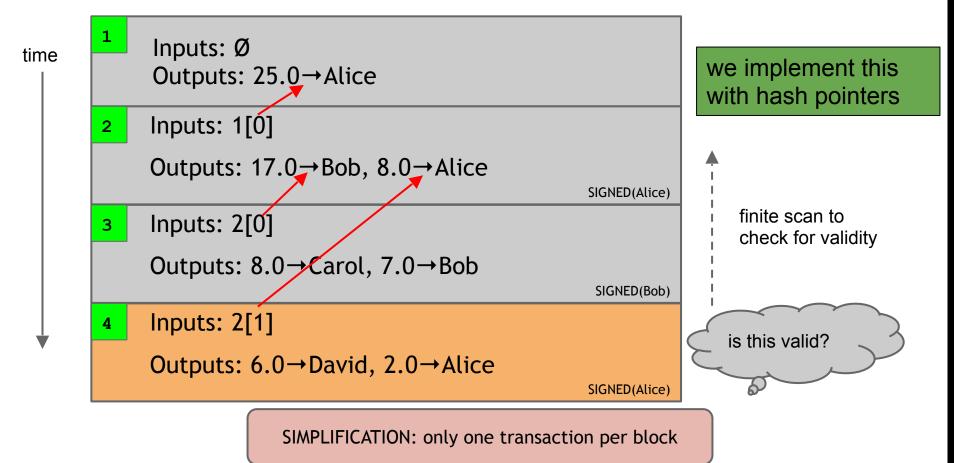
Transfer 15 coins from Alice to David<sub>SIGNED(Alice)</sub>

might need to scan backwards until genesis!

is this valid?

SIMPLIFICATION: only one transaction per block

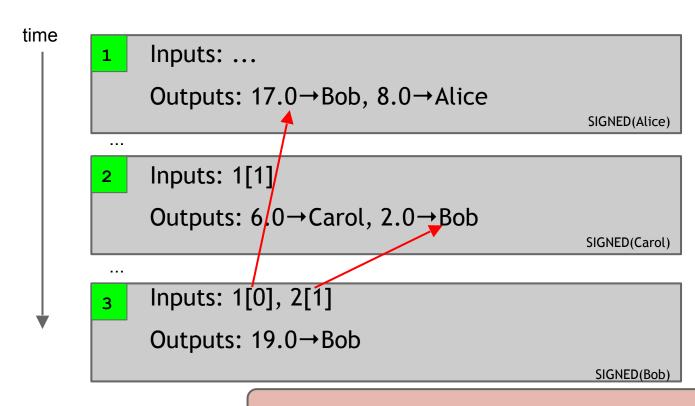
# A transaction-based ledger (Bitcoin)



## **Referencing Transactions**

- Hash pointers for transactions
- Within a transaction, refer to a particular output via serial numbers

## Merging value



SIMPLIFICATION: only one transaction per block

## Joint payments

```
time
                 Inputs: ...
                Outputs: 17.0 \rightarrow Bob, 8.0 \rightarrow Alice
                                                                          SIGNED(Alice)
                 Inputs: 1[1]
                Outputs: 6.0 \rightarrow Carol, 2.0 \rightarrow Bob
                                                                          SIGNED(Carol)
                 Inputs: 2[0], 2[1]
          3
                                                                two signatures!
                Outputs: 8.0→David
                                                               SIGNED(Carol), SIGNED(Bob)
```

SIMPLIFICATION: only one transaction per block

#### The real deal: a classical Bitcoin transaction

```
"hash":"5a42590fbe0a90ee8e8747244d6c84f0db1a3a24e8f1b95b10c9e050990b8b6b",
                                     "ver":1,
                                     "vin_sz":2,
                                     "vout sz":1.
metadata
                                     "lock time":0,
                                     "size":404.
                                     "in":[
                                        "prev out":{
                                         "hash": "3be4ac9728a0823cf5e2deb2e86fc0bd2aa503a91d307b42ba76117d79280260",
                                         "n":0
                                          "scriptSig":"30440..."
input(s)
                                        "prev_out":{
                                         "hash":"7508e6ab259b4df0fd5147bab0c949d81473db4518f81afc5c3f52f91ff6b34e",
                                         "n":0
                                        "scriptSig":"3f3a4ce81...."
                                     "out":[
                                        "value": "10.12287097",
                                        scriptPubKey":"OP_DUP OP_HASH160 69e02e18b5705a05dd6b28ed517716c894b3d42e OP_EQUALVERIFY OP_CHECKSIG"
output(s)
```

#### The real deal: transaction metadata

```
"hash": "5a42590...b8b6b",
transaction hash
                       "ver":1,
                       "vin_sz":2,
housekeeping
                       "vout_sz":1,
                       "lock_time":0,
"not valid before"
                                            more on this later...
                       "size":404,
housekeeping
```

# The real deal: transaction inputs

```
"in":[
                         "prev_out":{
                           "hash": "3be4...80260",
previous
transaction
                           "n":0
                      "scriptSig":"30440....3f3a4ce81"
signature
(more inputs)
```

# The real deal: transaction outputs

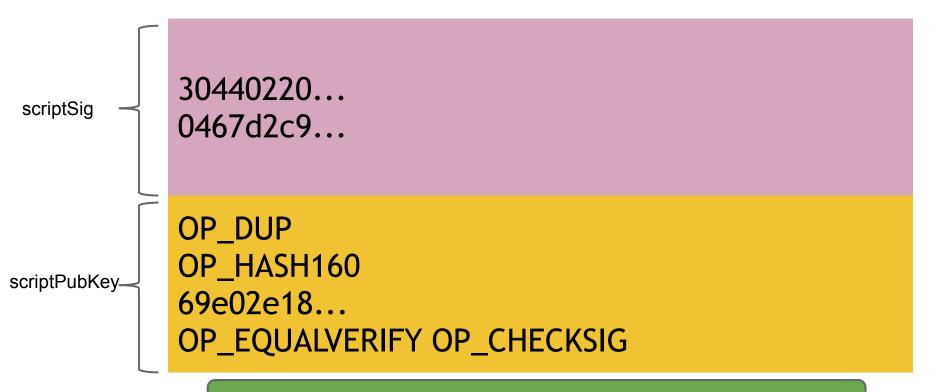
```
"out":[
                       "value":"10.12287097",
output value
                       "scriptPubKey": "OP_DUP OP_HASH160 69e...3d42e
                OP FQUALVERIFY OP_CHECKSIG"
recipient
address??
                                         more on this soon...
(more outputs)
```

# Bitcoin scripts

#### Output "addresses" are really scripts

```
OP_DUP
OP_HASH160
69e02e18...
OP_EQUALVERIFY OP_CHECKSIG
```

## Input "addresses" are also scripts



TO VERIFY: Concatenated script must execute completely with no errors

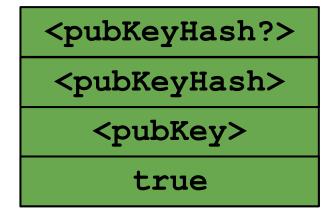
### Bitcoin scripting language ("Script")

#### Design goals

- Built for Bitcoin
- Simple, compact | lam not impressed
- Support for cryptography
- Stack-based
- Limits on time/memory
- No looping



#### Bitcoin script execution example

















#### Bitcoin script instructions

256 opcodes total (15 disabled, 75 reserved)

- Arithmetic
- If/then
- Logic/data handling
- Crypto!
  - Hashes
  - Signature verification
  - Multi-signature verification

#### OP\_CHECKMULTISIG

- Built-in support for joint signatures
- Specify *n* public keys
- Specify *t*
- Verification requires t signatures



BUG ALERT: Extra data value popped from the stack and ignored

#### Bitcoin scripts in practice ("original")

- Most nodes whitelist known scripts
- 99.9% are simple signature checks
- ~0.01% are MULTISIG
- ~0.01% are Pay-to-Script-Ha More on this soon
- Remainder are errors, proof-of-burn

\* numbers from NBFMG and slightly out of date

#### Proof-of-burn

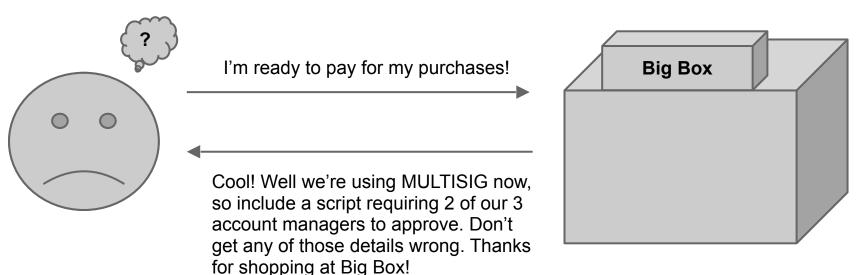
nothing's going to redeem that

OP\_RETURN <arbitrary data>

## Proof-of-burn: Applications

- Can be used to publish arbitrary data on the blockchain (e.g., timestamping a document)
- Bootstrap Altcoins by requiring people to destroy bitcoins in order to get new altcoins

## Should senders specify scripts?



## Idea: use the hash of redemption script

```
<signature>
<puble>
<puble>
OP_CHECKSIG
```

## Pay to script hash



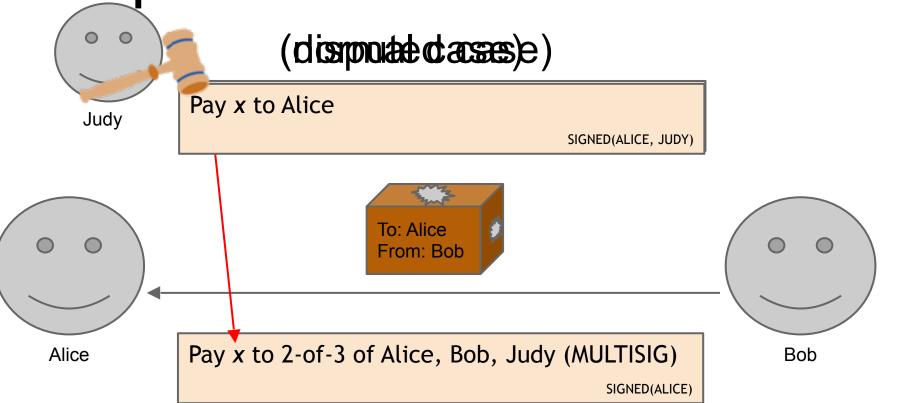
## Block size limits and Segwit

## Applications of Bitcoin scripts

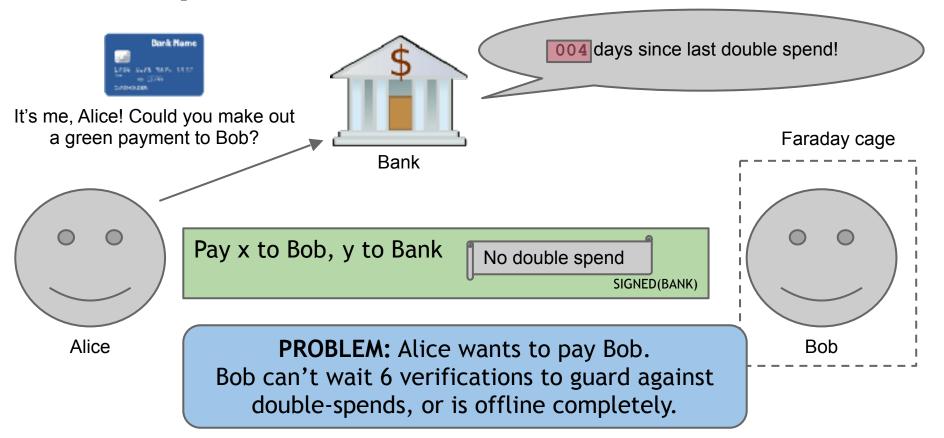
#### Example 1: "Fair" transactions

- <u>Problem</u>: Alice wants to buy a product from an online vendor Bob
- Alice doesn't want to pay until after Bob ships
- Bob doesn't want to ship until after Alice pays

#### Example 1: Fair transactions via Escrow



## Example 2: Green addresses



#### Example 2: Micro-payments

- Pay-as-you-go WIFI: Alice wants to pay WIFI provider (Bob) for each minute of WIFI service. But she doesn't want to incur a transaction fee for every minute
- Similarly, pay-as-you-go online subscriptions
- Ad-free websites

### Example 2: Micro-payments with Bitcoin

- <u>Main Idea</u>: Instead of doing several transactions, do a single transaction for total payment (and thus incur only a single transaction fee)
- How to implement it?

#### Example 2: Micro-payments with Bitcoin

What if Bob never signs?? Input: x; Pay 42 to Bob, 58 to Alice all of these could SIGNED(ALICE) SIGNED(BOB) be doublespends! Alice demands a timed refund transaction before starting Input: x; Pay 100 to Alice, LOCK until time t SIGNED(ALICE) SIGNED(BOB) TI publish! Pay US to BOD, 9/ to Alice I'm done! SIGNED(ALICE) Input: k; Pay 02 to Bob, 98 to Alice SIGNED(ALICE) Pay 01 to Bob, 99 to Alice Input: SIGNED(ALICE) Bob Input: 7; Pay 100 to Bob/Alice (MULTISIG) Alice SIGNED(ALICE)

```
lock_time
```

```
"hash": "5a42590...b8b6b",
 "ver":1,
 "vin_sz":2,
 "vout_sz":1,
 "lock_time":315415,
 "size":404,
                    Block index or real-world timestamp before
                    which this transaction can't be published
```

#### Micro-payments from Cryptocurrencies

More recent constructions, that achieve better properties

- Pass, shelat [CCS'16]
- Chiesa, Green, Liu, Miao, Miers, Mishra [EUROCRYPT'17]

#### More advanced scripts

- Fair multiplayer lotteries and fair multiparty computation [Andrychowichz-Dziembowski-Malinowski-Mazurek, S&P'14; Bentov-Kumaresan, CRYPTO'14]
- Hash pre-image challenges

## "Smart contracts"

Later: More powerful smart contracts with Ethereum (Turing-complete scripting language)

# Bitcoin blocks

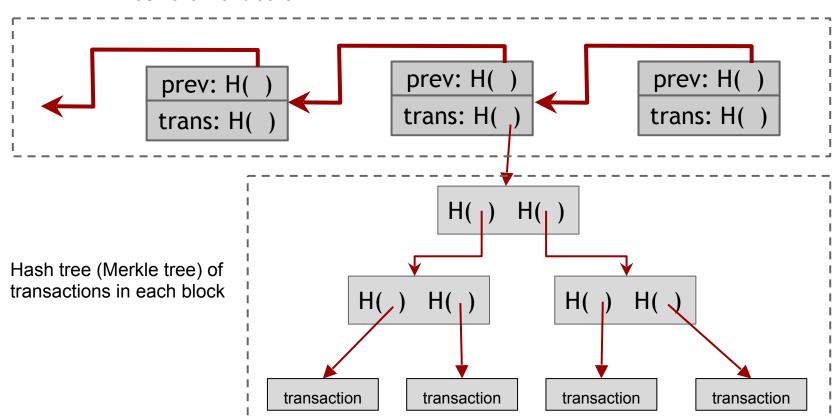
#### Bitcoin blocks

Why bundle transactions together?

- Single unit of work for miners
- Limit length of hash-chain of blocks
  - Faster to verify history

#### Bitcoin block structure

Hash chain of blocks

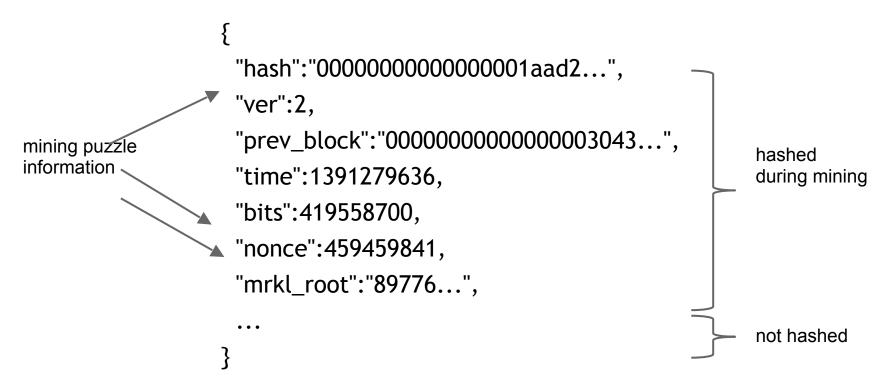


#### The real deal: a Bitcoin block

```
"hash":"0000000000000001aad2...",
                                   "ver":2,
                                   "prev_block":"00000000000000003043...",
                                   "time":1391279636,
 block header
                                   "bits":419558700,
                                   "nonce":459459841,
                                   "mrkl_root":"89776...",
                                   "n_tx":354,
                                   "size":181520,
                                   "tx":[
                                   "mrkl_tree":[
transaction
                                    "6bd5eb25...",
                                    "89776cdb..."
```

data

#### The real deal: a Bitcoin block header



#### The real deal: coinbase transaction

```
"in":[
                        "prev_out":{
                                              Null hash pointer
                          "hash":"000000.....
                                               0000000",
redeeming
                          "n":4294967295
nothing
                                         First ever coinbase parameter:
arbitrary
                                          The Times 03/Jan/2009 Chancellor
                                         on brink of second bailout for banks"
                              block reward
                                  transaction fees
                     "value":"12.53371419",
                     "scriptPubKey": "OPDUP OPHASH160 ... "
```

#### See for yourself!

Transaction View information about a bitcoin transaction

1515758d193a76d84a82534512888814523a8a3139a3cba454810958846484 1KvrdrQ3oGqMAIDTMEYCodDSnVaGNW2YZh 1.0194 BTC 1KryFUt9tXHyaoCYTNPbqpWPJKQ717YmL5 1KryFUI9DXHvaoCYTNPbqpWPJKQ717YmL5 3,458 BTC 9 Confirmations Inputs and Outputs Summary 257 (bytes). Total Input 4.4775 BTC: Size Total Oulput Received Time 2014-08-05 01:55:25 4.4774 BTC Included In Blocks 314018 (2014-08-05 02:00:40 +5 minutes). 0.0001 BTC Fees Continuations: 9 Continuations Estimated BTC Transacted 1.0194 BTC Relayed by IP 9 Scripts: Show scripts & combase Blockchain.info Visualize View Tree Chart

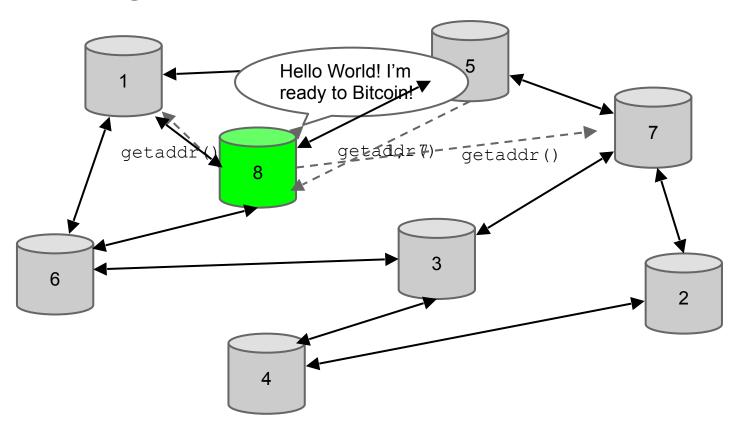
#### blockchain.info (and many other sites)

## The Bitcoin network

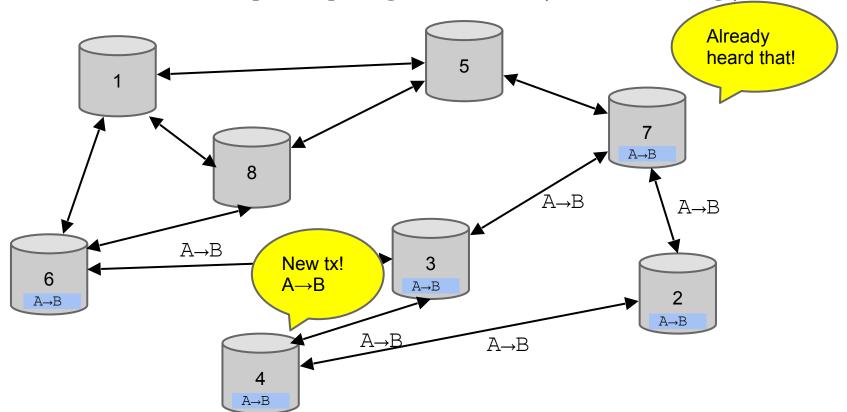
#### Bitcoin P2P network

- Ad-hoc protocol (runs on TCP port 8333)
- Ad-hoc network with random topology
- All nodes are equal
- New nodes can join at any time
- Forget non-responding nodes after 3 hr

### Joining the Bitcoin P2P network



Transaction propagation (flooding)



#### Should I relay a proposed transaction?

- Transaction valid with current block chain
- (default) script matches a whitelist
  - Avoid unusual scripts
- Haven't seen before
  - Avoid infinite loops

Sanity checks only...
Some nodes may ignore them!

- Doesn't conflict with others I've relayed
  - Avoid double-spends

#### Nodes r New tx! liffer on transaction pool A→C $A \rightarrow C$ 5 А→В A→C $A \rightarrow C$ $A \rightarrow B$ $A \rightarrow C$ $A \rightarrow B$ А→В А→В

#### Race conditions

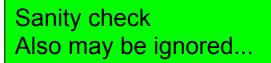
Transactions or blocks may conflict

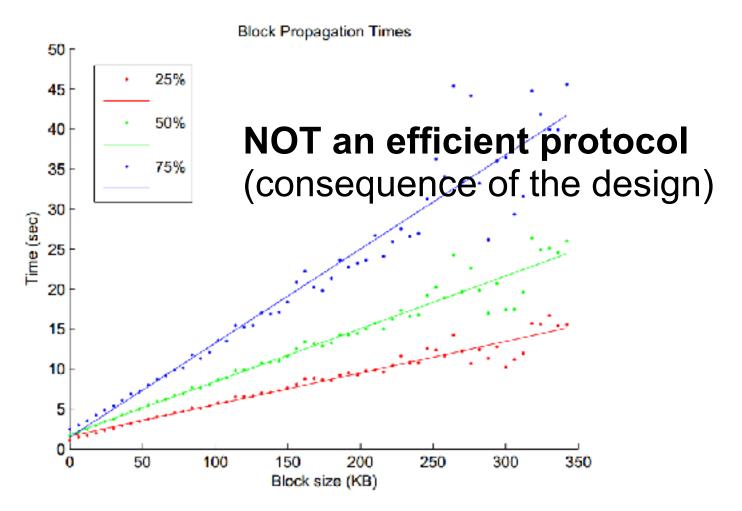
- Default behavior: accept what you hear first
- Network position matters
- Miners may implement other logic!

### Block propagation nearly identical

Relay a new block when you hear it if:

- Block meets the hash target
- Block has all valid transactions
  - Run all scripts, even if you wouldn't relay
- Block builds on current longest chain
  - Avoid forks





Source: Yonatan Sompolinsky and Aviv Zohar: "Accelerating Bitcoin's Transaction Processing" 2014

#### How big is the network?

- Unclear how to measure exactly
- Estimates-up to 1M IP addresses/month
- Only about 5-10k "full nodes"
  - Permanently connected
  - Fully-validate
- This number may be dropping!

#### Fully-validating nodes

- Permanently connected
- Store entire block chain
- Hear and forward every node/transaction

# Thin/SPV clients (not fully-validating)

Idea: don't store everything

- Store block headers only
- Request transactions as needed
  - To verify incoming payment
- Trust fully-validating nodes

Limitations & improvements

#### Hard-coded limits in Bitcoin

- 10 min. average creation time per block
- 1 M bytes in a block
- 20,000 signature operations per block
- 23M total bitcoins maximum
- 50,25,12.5... bitcoin mining reward

These affect economic balance of power too much to change now

#### Throughput limits in Bitcoin

- 1 M bytes/block (10 min)
- >250 bytes/transaction
- 7 trai

Improving throughput: A strong motivation for Altcoins

Compar

- VISA: z,000-10,000 transactions/sec
- PayPal: 50-100 transaction/sec

# Cryptographic limits in Bitcoin

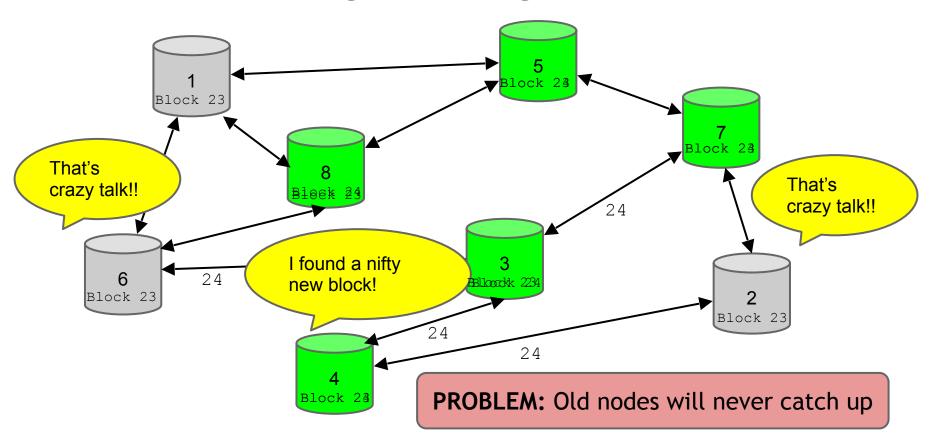
- Only 1 signature algorithm (ECDSA/P256)
- Hard-coded hash functions

Some of these crypto primitives used here might break by 2040 (e.g., collision-found in hash function, or powerful quantum computer breaks ECDSA)...

# Why not update Bitcoin software to overcome these limitations?

 Many of these changes require "hard forks", which are currently considered unacceptable

# "Hard-forking" changes to Bitcoin



#### Soft forks

Observation: we can add new features which only *limit* the set of valid transactions

Need majority of nodes to enforce new rules

Old nodes will approve

RISK: Old nodes might mine now-invalid blocks

# Soft fork example: pay to script hash

```
<signature>
<<pub/>
<<pre>c<pub/>
pubkey> OP_CHECKSIG>
```

OP\_HASH160 <hash of redemption script> OP\_EQUAL

Old nodes will just approve the hash, not run the embedded script

### Soft fork possibilities

- New signature schemes
- Extra per-block metadata
  - Shove in the coinbase parameter
  - Commit to unspent transaction tree in each block

#### Hard forks

- New op codes
- Changes to size limits
- Changes to mining rate
- Many small bug fixes

Currently seem unlikely to happen

Many of these issues addressed by Altcoins