Bitcoin

Tom Anderson
Admin

Course evals

My office hours next week are cancelled
Bitcoin Goal

Electronic money without trust

$34B market value

Created out of thin air, from a paper + some code
Pros/cons of Cash

+ portable
+ cannot spend twice
+ cannot repudiate after payment
+ no need for trusted 3rd party
+ anonymous (serial #s?)
- doesn't work online
- easy to steal (in moderate amounts)
+/- hard for government to monitor/tax/control
+/- government can print more as economy expands
Pros/cons of Credit Cards/PayPal?

+ works online
+ somewhat hard to steal
+/- can repudiate
- requires trusted 3rd party
- tracks all your purchases
- can prohibit some transactions (e.g. wikileaks donations)
+/- easy for government to monitor/tax/control

Q: gift cards? Paid for in cash?
Bitcoin

Suppose we had a system where a penny was just a string of bits

What's hard technically?

– Forgery: what's to keep someone creating many copies?
– Double spending: what's to keep someone from using the bits twice?
– Theft: what's to keep someone from learning the bits and then spending them?
Bitcoin

What's hard socially/economically?

– Why does the string of bits have value?
– How do you convert it to cash?
– How to pay for infrastructure that manages/assigns strings of bits?
– Monetary policy (intentional inflation, ...)
– Laws (taxes, money laundering, drugs, terrorists)
Crossing the Chasm

Theory of technology adoption (Geoffrey Moore)

Early adopters (hype)
  – Tech that solves a compelling problem
  – Worth hassle of a partially working system

Early majority (graveyard of hype)
  – Pragmatists: need whole product solution

Late majority/laggards
  – Tech needs to be cheap, reliable, widely used
Examples

• Cellphones
  – Early users: drug dealers, intl business travel
• Email/web
  – Early users: scientists, pornographers
• Cloud computing
  – Early users: Internet search, high-speed traders
• Bitcoin
  – Early users: drug dealers, money laundering, ransomware, export control avoidance, ...
• Driverless cars, MOOCs, space tourism, ...
Encryption

- Cryptographer chooses functions $E$, $D$ and keys $K^E$, $K^D$
  - Suppose everything is known ($E$, $D$, $M$ and $C$), should not be able to determine keys $K^E$, $K^D$ and/or modify msg
  - provides basis for authentication, privacy and integrity
Public Key Encryption (RSA, PGP)

Keys come in pairs: public (K-public) and private (K-priv)
- Each principal gets its own pair
- Public key published; private is secret to entity
- can’t derive K-priv from K-public, M, (M)^K-priv
- Sign with private key to authenticate
Public Key: Authentication

Keys come in pairs: public and private

- $M = ((M)^{K\text{-private}})^{K\text{-public}}$
- Ensures authentication: can only be sent by sender
Public Key: Secrecy

Keys come in pairs: public and private

- $M = ((M)^K_{-public})^K_{-private}$
- Ensures secrecy: can only be read by receiver
Message Digests (MD5, SHA)

- Cryptographic checksum: message integrity
  - Typically small compared to message (MD5 128 bits)
  - “One-way”: infeasible to find two messages with same digest

```
<table>
<thead>
<tr>
<th>Initial digest</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transform</td>
</tr>
<tr>
<td>Transform</td>
</tr>
<tr>
<td>...</td>
</tr>
<tr>
<td>Transform</td>
</tr>
<tr>
<td>Message digest</td>
</tr>
<tr>
<td>Message (padded)</td>
</tr>
<tr>
<td>512 bits</td>
</tr>
</tbody>
</table>
```
Infocoin Straw Proposal

Transfer is statement: ”Ellis gives Jialin infocoin #57” signed in Ellis's private key

Issues?

– Who assigned the serial #? Can Ellis mint money?
– Easy for Jialin to copy Ellis’s statement; why can't he use it twice?
– Easy for Ellis to sign statement; why can’t he do that twice?
With a Trusted Intermediary (Bank)

• Ellis withdraws a coin from the bank; gets a unique serial # (signed with Bank's private key)
• Ellis signs certificate (with his private key)
• Jialin checks certificate with bank to see that serial # is valid (belongs to Ellis) and not double spent
Do we have to trust the bank?

Suppose bank keeps a visible log of operations

- Replicated public ledger (block chain) with all transfers in sequence
- Replicas could be run by volunteers!

To transfer coin, Ellis signs block and adds it to chain:

- Hash of previous chain, Jialin's public key, coin #

To transfer coin, Jialin signs block and adds it to chain:

- Hash of previous chain, Doug's public key, coin #

Jialin/Doug can read any (up to date!) replica to ensure transfer is a valid coin owned by Ellis/Jialin
Managing the Public Log

• Need updates to be applied in the same order at each replica
• Different replicas receive updates at different times
  – How do readers know replica is up to date?
• Use Paxos?
  – What if replicas aren’t trusted?
• Use Byzantine Paxos?
  – Still need to trust 2f + 1 replicas
Bitcoin

Protocol for managing replicated log
  Replicas run by (greedy) volunteers
  Allow double spending to be detected
  Provided a majority of replicas don’t collude
  Make it hard for anyone to control a majority

Limitations:
  Few transactions/second
  No backsies
Log Management Straw Proposal

- Assume large number of replicas
- Every new op sent to one replica, rebroadcast to all
- Slow system down to reduce the chance of a conflicting updates
  - Every node picks a random delay before applying update
  - For 1M nodes, 1/600M => 1 update every 10 minutes
  - Might still conflict
  - For higher throughput, batch transactions
- Still requires some trust (e.g., to pick random #)
Sybil Attack

- If anyone can be a replica, then:
  - Ellis runs a billion replicas
  - Jialin will only be able to check a subset
  - How does Jialin know the subset isn’t run by Ellis?

- Proof of work: force replicas to do work
- Will discourage volunteers!
  - Easier for Ellis to acquire a majority of replicas!
- Instead: reward replicas for doing work
Proof of Work

• Replicas perform a puzzle
  – Puzzle is public: whoever completes the puzzle first determines the next (batch of) ops in log
  – and gets a reward (currently 12.5 bitcoins)

• Bitcoin find a nonce such that:
  – SHA256(msg!nonce) = 0...

• SHA is a cryptographic hash: no easier way to find a match except to guess
Proof of Work

Match on first zero?
  – Too easy; two tries on average

Match on first two zeroes?
  – Too easy; four tries on average

Bitcoin requires 71 leading zeroes
  – 4M tera-hash/sec (liquid cooled ASICs)
  – $25K reward per solution, 10 minutes
  – Difficulty adjusted to keep solutions at fixed rate
How Long Is a Good Password?

• Entropy in computer-selected passwords
  – $2^6$ bits/character

• Entropy in human-selected passwords
  – 2 bits/character (measured)

• Bitcoin gives price of password cracking
  – Most websites store passwords as SHA hashes
  – $10$ to crack a 30 character (human) password
Some Bitcoin Details

Hash difficulty is not binary
  • SHA256(msg|nonce) < value
  • Allows fine-grained adjustment of proof of work

Prevent solving ahead
  • SHA256(previous hash|msg|nonce) < target

Block contains multiple transactions
  • Current rate ~ 5/second
  • Money laundering vs. buying coffee
Reward

• Solution is broadcast to every replica; what keeps replicas from stealing the solution?
• Every replica works on a slightly different puzzle
• Ellis works on:
  – SHA(previous hash, mint coin and give it to Ellis, set of transactions, nonce) < target
• Jialin works on:
  – SHA(previous hash, mint coin and give it to Jialin, set of transactions, nonce) < target
When Nonce is Found

Replicas have a choice:

- Ignore the answer and continue to try to find another one
- Take the answer as a given and work on the next puzzle.

Which should it choose?

- If more than half of the computational power chooses (b), replica should choose (b)
Who Wins?

• If two nodes find the nonce at about the same time, who wins?
• Depends on solution to the next puzzle!
• Everyone has an incentive to work on chain that others will work on
  – If next solution uses A’s solution, A wins
  – If next solution uses B’s solution, B wins
Who Wins?

• Replicas have an incentive to prevent others from announcing their solutions

• DoS attacks
  – flood replica with traffic so TCP connections fail

• BGP prefix hijacking
  – Internet is shortest path routing, without security
  – Announce your network has shorter path to target replica
  – Traffic sent to a blackhole
Mining Groups

• Reward is (very) sporadic: if 1M replicas search for hash, each will win once every few decades.
• Pool resources: pay nodes to look for solutions
• Where Doug is a coordinator, ask replicas to:
  – SHA(previous hash, mint coin for Doug, msg, nonce)
• Why would anyone do this for Doug?
  – Ex: hand out 0.001 bitcoin for 60 leading zeros
Serial Numbers Revisited

• Proof of work solves how we create new coins
  – Every 10 minutes, another reward

• What about inflation?
  – Reward decreases by 2x every few years
  – Increasing number of coins in circulation
  – Fixed total number of coins (93% of total already mined)

• Do miners stop working when reward stops?
Theory of Money

• Why do bitcoins have value?

• Why does gold?

• Why does cash?

• Why does Facebook or Google stock?
Who Wins?

• Bitcoin founder(s) performed early mining
  – Reserved a fraction of bitcoins for themselves
  – But haven’t spent them (bitcoin log is public)
  – Is it possible for them to sell without tanking Bitcoin?
Double Spending

• Suppose Y creates two transactions: Y->Z, Y->Q

• Z and Q probably don't check all the peers
  – Y has a chance to tell diff peers diff transactions

• Maybe some peers are corrupt and cooperating with Y
  – hide Y->Q from Z, hide Y->Z from Q

• Only need to play tricks briefly
  – just until Z gives the hamburger to Y
Double Spending

How long should Z wait before giving Y the hamburger?

Until Z sees Y flood the transaction to many peers?
Double Spending

How long should Z wait before giving Y the hamburger?

Until Z sees Y flood the transaction to many peers?

– not in the chain, Y might flood conflicting xaction

Until Z sees one peer with chain ...<-BZ (containing Y->Z)?
Double Spending

How long should Z wait before giving Y the hamburger?

Until Z sees Y flood the transaction to many peers?

- not in the chain, Y might flood conflicting transaction

Until Z sees one peer with chain ...<-BZ (containing Y->Z)?

- maybe that peer is corrupt, in league with Y

Until Z sees lots of peers with chain ...<-BZ?
Double Spending

How long should Z wait before giving Y the hamburger? Until Z sees Y flood the transaction to many peers?
   – not in the chain, Y might flood conflicting xaction
Until Z sees one peer with chain ...<-BZ (containing Y->Z)?
   – maybe that peer is corrupt, in league with Y
Until Z sees lots of peers with chain ...<-BZ?
   – risky -- some other chain may win
   – perhaps that chain won't have Y->Z
Until Z sees chain with multiple blocks after BZ?
Double Spending

How long should Z wait before giving Y the hamburger?

Until Z sees Y flood the transaction to many peers?
  – not in the chain, Y might flood conflicting transaction

Until Z sees one peer with chain ...<-BZ (containing Y->Z)?
  – maybe that peer is corrupt, in league with Y

Until Z sees lots of peers with chain ...<-BZ?
  – risky -- some other chain may win
  – perhaps that chain won't have Y->Z

Until Z sees chain with multiple blocks after BZ?
  – slim chance attacker can catch up
Transaction Reward

• When a replica receives a request what should it do?
  – Ignore it?
  – Add it to the next batch?
  – Forward it?
Transaction Reward

• When a replica receives a request what should it do?
  – Ignore it?
  – Add it to the next batch?
  – Forward it?

• Transactions can have multiple outputs
  – Main payment to recipient
  – Side payment to the winning miner
Private Exchanges

• Bitcoin
  – can only perform a few operations per second (worldwide)
  – performs operations slowly (minutes to confirm)
  – No accountability if seller reneges

• Private exchanges/escrow
  – Both parties trust exchange
  – Execute operations on internal account record
  – Exports internal account to cash or public bitcoin

• How is this different from a bank?
Bitcoin and Other Cryptocurrencies

• Bitcoin is not the only electronic cash standard
• Zerocoin
  – Better anonymity (better money laundering!)
• Ethereum
  – Better scripting (better for creating new coins!)
• Ripple
  – Stable price (better for commercial banking!)
Bitcoin Discussion

• Where does value of a Bitcoin come from?
  – Why is there a limit on # of bitcoins?
• How long will SHA-256 last?
• How do we make changes to the protocol?
• Is Bitcoin anonymous? Linkability, zerocoin
• Is Bitcoin ethical? Ransomware, money laundering
• Private exchanges and security of wallets
• Non-reversible (vs. credit cards)
• Attacks: mining monopolies, BGP route hijacks, …