6.824 2018 Lecture 20: Bitcoin

Bitcoin: A Peer-to-Peer Electronic Cash System, by Satoshi Nakamoto, 2008

bitcoin:
  a digital currency
  a public ledger to prevent double-spending
  no centralized trust or mechanism <-- this is hard!
  malicious users ("Byzantine faults")

why might people want a digital currency?
  might make online payments easier
  credit cards have worked well but aren't perfect
    insecure -> fraud -> fees, restrictions, reversals
    record of all your purchases

what's hard technically?
  forgery
  double spending
  theft

what's hard socially/economically?
  why do Bitcoins have value?
  how to pay for infrastructure?
  monetary policy (intentional inflation &c)
  laws (taxes, laundering, drugs, terrorists)

idea: signed sequence of transactions
  (this is the straightforward part of Bitcoin)
  there are a bunch of coins, each owned by someone
  every coin has a sequence of transaction records
    one for each time this coin was transferred as payment
  a coin's latest transaction indicates who owns it now

what's in a transaction record?
  pub(user1): public key of new owner
  hash(prev): hash of this coin's previous transaction record
  sig(user2): signature over transaction by previous owner's private key
  (BitCoin is much more complex: amount (fractional), multiple in/out, ...)

transaction example:
  Y owns a coin, previously given to it by X:
    T7: pub(Y), hash(T6), sig(X)
  Y buys a hamburger from Z and pays with this coin
    Z sends public key to Y
    Y creates a new transaction and signs it
    T8: pub(Z), hash(T7), sig(Y)
  Y sends transaction record to Z
  Z verifies:
    T8's sig() corresponds to T7's pub()
  Z gives hamburger to Y

Z's "balance" is set of unspent transactions for which Z knows private key
  the "identity" of a coin is the (hash of) its most recent xaction
  Z "owns" a coin = Z knows private key for "new owner" public key in latest xaction

can anyone other than the owner spend a coin?
  current owner's private key needed to sign next transaction
  danger: attacker can steal Z's private key, e.g. from PC or smartphone

can a coin's owner spend it twice in this scheme?
  Y creates two transactions for same coin: Y->Z, Y->Q
    both with hash(T7)
  Y shows different transactions to Z and Q
  both transactions look good, including signatures and hash
  now both Z and Q will give hamburgers to Y

why was double-spending possible?
  b/c Z and Q didn't know complete set of transactions

what do we need?
  publish log of all transactions to everyone, in same order
    so Q knows about Y->Z, and will reject Y->Q
    a "public ledger"
  ensure Y can't un-publish a transaction

why not rely on CitiBank, or Federal Reserve, to publish transactions?
  not everyone trusts them
  they might be tempted to reverse or restrict

why not publish transactions like this:
  1000s of peers, run by anybody, no trust required in any one peer
  peers flood new transactions over "overlay"
  transaction Y->Z only acceptable if majority of peers think it is valid
    i.e. they don't know of any Y->Q
    hopefully majority overlap ensures double-spend is detected
  how to count votes?
    how to even count peers so you know what a majority is?
    perhaps distinct IP addresses?
  problem: "sybil attack"
    IP addresses are not secure -- easy to forge
    attacker pretends to have 10,000 computers -- majority
    when Z asks, attacker's majority says "we only know of Y->Z"
    when Q asks, attacker's majority says "we only know of Y->Q"
  voting is hard in "open" p2p schemes

the BitCoin block chain
  the goal: agreement on transaction log to prevent double-spending
  the block chain contains transactions on all coins
  many peers
    each with a complete copy of the whole chain
    proposed transactions flooded to all peers
    new blocks flooded to all peers
  each block:
    hash(prevblock)
    set of transactions
    "nonce" (not quite a nonce in the usual cryptographic sense)
    current time (wall clock timestamp)
  new block every 10 minutes containing xactions since prev block
  payee doesn't believe transaction until it's in the block chain

who creates each new block?
  this is "mining"
  all peers try
  requirement: hash(block) has N leading zeros
  each peer tries nonce values until this works out
  trying one nonce is fast, but most nonces won't work
    it's like flipping a zillion-sided coin until it comes up heads
    each flip has an independent (small) chance of success
    mining a block *not* a specific fixed amount of work
  it would likely take one CPU months to create one block
  but thousands of peers are working on it
  such that expected time to first to find is about 10 minutes
  the winner floods the new block to all peers

how does a Y->Z transaction work w/ block chain?
  start: all peers know ...<-B5
    and are working on B6 (trying different nonces)
  Y sends Y->Z transaction to peers, which flood it
  peers buffer the transaction until B6 computed
  peers that heard Y->Z include it in next block
  so eventually ...<-B5<-B6<-B7, where B7 includes Y->Z

Q: could there be *two* different successors to B6?
A: yes, in (at least) two situations:
   1) two peers both get lucky (unlikely, given variance of block time)
   2) network partition
  in both cases, the blockchain temporarily forks
    peers work on whichever block they heard about before
    but switch to longer chain if they become aware of one

how is a forked chain resolved?
  each peer initially believes whichever of BZ/BQ it saw first
  tries to create a successor
  if many more saw BZ than BQ, more will mine for BZ,
    so BZ successor likely to be created first
  if exactly half-and-half, one fork likely to be extended first
    since significant variance in mining success time
  peers always switch to mining the longest fork, re-inforcing agreement

what if Y sends out Y->Z and Y->Q at the same time?
  i.e. Y attempts to double-spend
  no correct peer will accept both, so a block will have one but not both

what happens if Y tells some peers about Y->Z, others about Y->Q?
  perhaps use network DoS to prevent full flooding of either
  perhaps there will be a fork: B6<-BZ and B6<-BQ

thus:
  temporary double spending is possible, due to forks
  but one side or the other of the fork highly likely to disappear
  thus if Z sees Y->Z with a few blocks after it,
    it's very unlikely that it could be overtaken by a
    different fork containing Y->Q
  if Z is selling a high-value item, Z should wait for a few
    blocks before shipping it
  if Z is selling something cheap, maybe OK to wait just for some peers
    to see Y->Z and validate it (but not in block)

can an attacker modify a block in the middle of the block chain?
  not directly, since subsequent block holds block's hash

could attacker start a fork from an old block, with Y->Q instead of Y->Z?
  yes -- but fork must be longer in order for peers to accept it
  so if attacker starts N blocks behind, it must generate N+M+1
    blocks on its fork before main fork is extended by M
  i.e. attacker must mine blocks *faster* than the other peers
  with just one CPU, will take months to create even a few blocks
    by that time the main chain will be much longer
    no peer will switch to the attacker's shorter chain
  if the attacker has 1000s of CPUs -- more than all the honest
    bitcoin peers -- then the attacker can create the longest fork,
    everyone will switch to it, allowing the attacker to double-spend

there's a majority voting system hiding here
  peers cast votes by mining to extend the longest chain

summary:
  if attacker controls majority of CPU power, can force honest
    peers to switch from real chain to one created by the attacker
  otherwise not

validation checks:
  peer, new xaction:
    no other transaction spends the same previous transaction
    signature is by private key of pub key in previous transaction
    then will add transaction to txn list for next block to mine
  peer, new block:
    hash value has enough leading zeroes (i.e. nonce is right, proves work)
    previous block hash exists
    all transactions in block are valid
    peer switches to new chain if longer than current longest
  Z:
    (some clients rely on peers to do above checks, some don't)
    Y->Z is in a block
    Z's public key / address is in the transaction
    there's several more blocks in the chain
  (other stuff has to be checked as well, lots of details)

where does each bitcoin originally come from?
  each time a peer mines a block, it gets 12.5 bitcoins (currently)
  it puts its public key in a special transaction in the block
  this is incentive for people to operate bitcoin peers

Q: what if lots of miners join, so blocks are created faster?

Q: 10 minutes is annoying; could it be made much shorter?

Q: are transactions anonymous?

Q: if I steal bitcoins, is it safe to spend them?

Q: can bitcoins be forged, i.e. a totally fake coin created?

Q: what can adversary do with a majority of CPU power in the world?
   can double-spend and un-spend, by forking
   cannot steal others' bitcoins
   can prevent xaction from entering chain

Q: what if the block format needs to be changed?

esp if new format wouldn't be acceptable to previous s/w version?

Q: how do peers find each other?

Q: what if a peer has been tricked into only talking to corrupt peers?
how about if it talks to one good peer and many colluding bad peers?

Q: could a brand-new peer be tricked into using the wrong chain entirely?
what if a peer rejoins after a few years disconnection?
a few days of disconnection?

Q: how rich are you likely to get with one machine mining?

Q: why does it make sense for the mining reward to decrease with time?

Q: is it a problem that there will be a fixed number of coins?
what if the real economy grows (or shrinks)?

Q: why do bitcoins have value?
e.g., 1 BTC appears to be around $8700 on may 14 2018.

Q: will bitcoin scale well?
as transaction rate increases?
claim CPU limits to 4,000 tps (signature checks)
more than Visa but less than cash
as block chain length increases?
do you ever need to look at very old blocks?
do you ever need to xfer the whole block chain?
merkle tree: block headers vs txn data.
sadly, the maximum block size is limited to one megabyte

Q: could Bitcoin have been just a ledger w/o a new currency?
e.g. have dollars be the currency?
since the currency part is pretty awkward.
(settlement... mining incentive...)

key idea: block chain
public ledger is a great idea
decentralization might be good
mining is a clever way to avoid sybil attacks
tieing ledger to new currency seems awkward, maybe necessary