Primecoin

CS1699: Blockchain Technology and Cryptocurrency

16. More Variations on a Theme: Non-SHA256 Puzzles

Bill Laboon

Is Bitcoin Mining Wasteful?

- **Lots** of energy used to… solve partial hash-preimage puzzles
- Couldn't we use all of this computation to cure cancer or something?

Distributed Computing

- Volunteer projects have found great success in the past
- Cheating has occurred, but not to a great extent and is generally detectable
- Examples: Great Internet Mersenne Prime Search, SETI@home, Folding@home
- However, none of these are good fits for a cryptocurrency!

Challenges

- Need an *equiprobable solution space* (i.e., no way to "cherry-pick" data, should be puzzle-friendly)
- Need an *inexhaustible puzzle space* (i.e., not bounded by available data)
- Needs to be *algorithmically generated* to avoid centralized administrators determining problems and veracity of solutions

Why Wouldn't These Work?

- **SETI@home** - Lack of equiprobable solution space, lack of inexhaustible puzzle space, data not algorithmically generated
- **Folding@home** - Lack of equiprobable solution space, inexhaustible puzzle space, data not algorithmically generated
- **Great Internet Mersenne Prime Search** - Equiprobable solution space (minus some trivialities), puzzle space proven to be inexhaustible, data can be algorithmically generated… *hmmmm*.. but on average, one Mersenne Prime found per year

Primecoin

- The first (and one of the only) proof-of-useful work coins out there
- Looks for Cunningham chains of primes
- How “useful” is this work?

Cunningham Chain of Prime Numbers

- A sequence of k prime numbers $p_1, p_2, p_3, \ldots p_n$ such that $p_i = 2p_{i-1} + 1$ for each number
- That is, double a prime number and add one, and repeat this for k iterations. However many iterations you can do this with the number still prime is a Cunningham chain of length k.
- Example: 2, 5, 11, 23, 47, ~~95~~ Chain of length $k = 5$
- Conjectured but not proven that there are Cunningham chains of length k for any k (although longest known is $k = 19$)

Parameters

- For a given challenge x (hash of previous block)
  - Take first m bits of x
  - Consider valid a chain of length k or greater in which:
    - the first prime in the chain is an n-bit prime AND
    - the first prime in the chain has the same m leading bits as x
  - Can adjust n and k to adjust puzzle difficulty (k modifies it exponentially, n polynomially)

Variation - Golem

- Golem - Runs on Ethereum, allows you to “rent out” computing power to perform large-scale computing tasks
- Currently used mostly for 3-D rendering
- How to verify?

Proof of Storage

- Also known as “proof of retrievability”
- Represent a large file or collection of data as a very large Merkel tree, and all participants agree on Merkel root
- Miners each store a subset $F_m \subseteq F$ of data, associated with their key
- Miners must find a nonce along with the data that when SHA-256 hashed, is less than the target difficulty
- PermaCoin (never seriously implemented past white paper, as far as I can tell) , Chia (in beta) *(notice a trend?)*

Proof of Useful Work - A Mirage?

- Many technical drawbacks and few benefits
- What computations that meet all of the requirements are actually useful?
- From an economics standpoint, one could argue that you need to have something which is generally NOT useful otherwise in order to have sound money! (See Saifedean Ammous's *The Bitcoin Standard*)

The Problems of Pools

- Pools perhaps tend to lead to centralization
- Pools are also targets for malicious actors
- Pool miners who are not admins aren't running full nodes
- Pools not part of "Satoshi's Vision"
- Pools only possible because:
    - Possibility of shares - can easily prove that they're working to find valid blocks
    - Imposible to modify the recipient (would modify nonce)

Block-Discarding Attacks

- Assume you want to harm a pool and don't care about the damage to yourself
- You could submit shares, but then when you find a valid block, ignore it
- Not seen much (or at all) in reality, but theoretically a valid attack if you have a few large pools

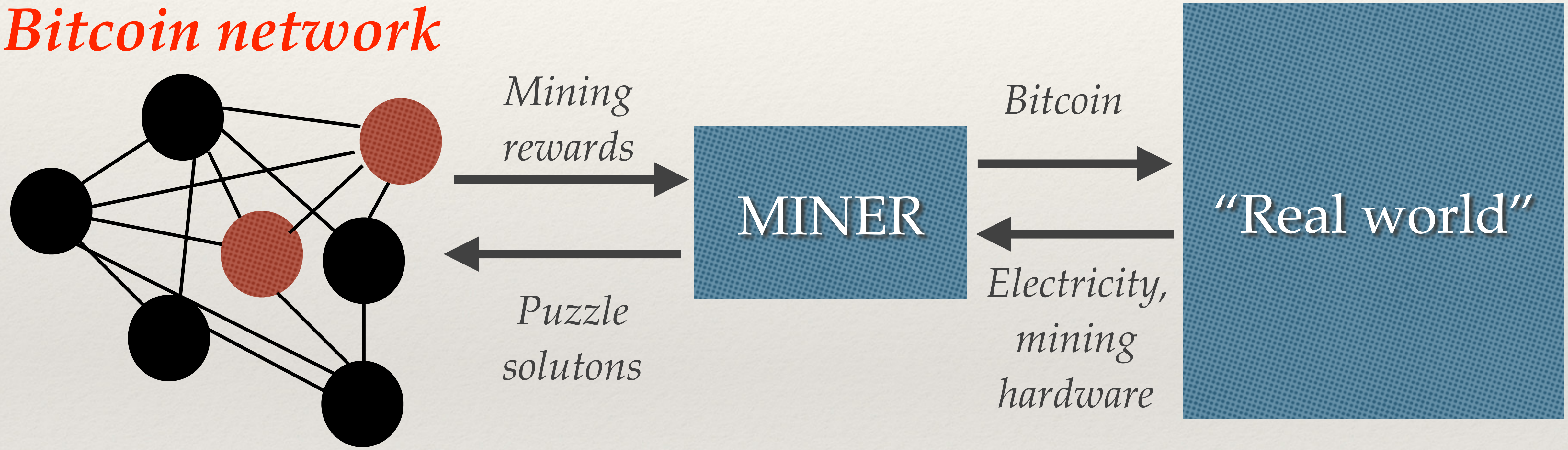
Nonoutsourceable Puzzles

- Puzzles which disincentivize pools or collusion in general
- Easiest solution: require all miners to know a private key
- Example: find a block whose hash of a signature is below a certain threshold, with signature computed using the public key of the recipient address
- As far as I can tell, no real implementation of this

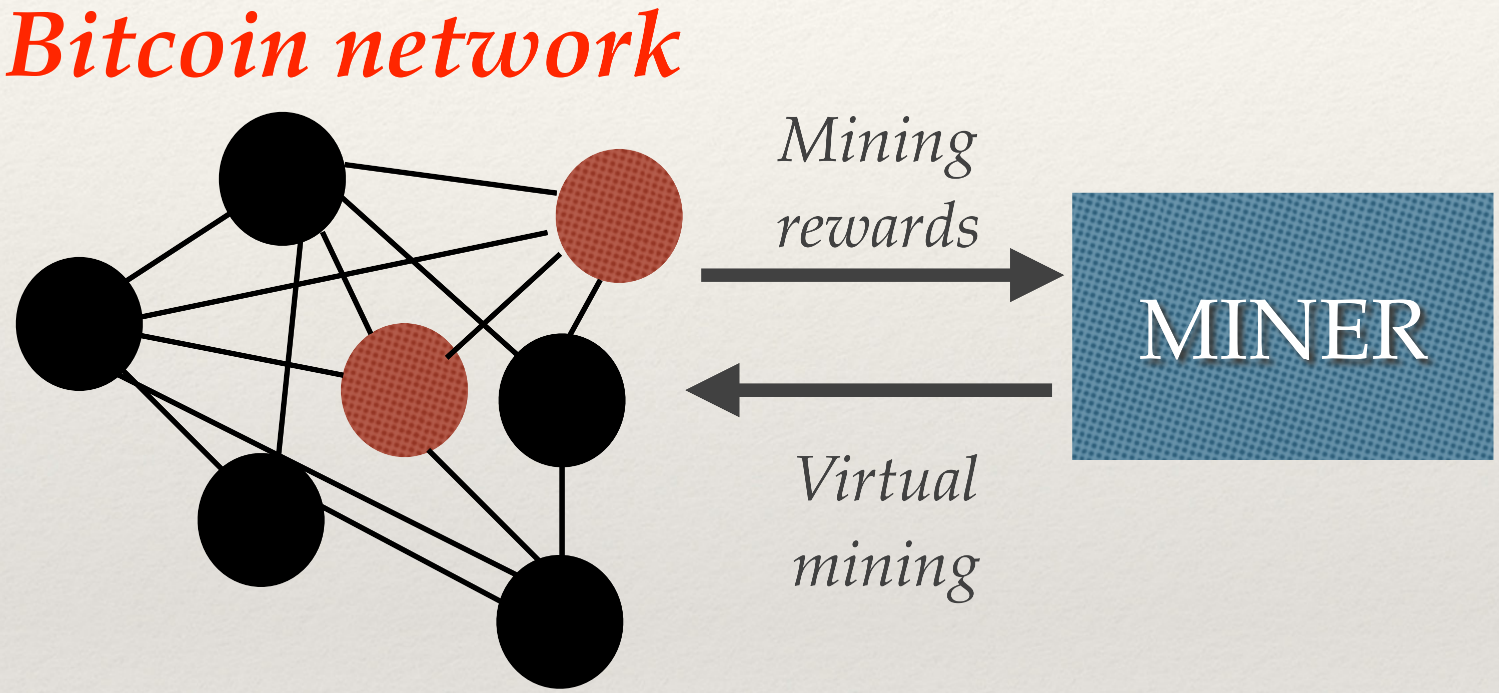
Proof of Stake

- The idea behind mining is that we have “skin in the game” - we have to do something hard to “put our money where our mouth is” in terms of valid blocks
- But if we already have a valid currency that can be consumed, can’t we use that instead of “wasting” electricity?
- “Virtual mining” - invest the currency itself instead of electricity and mining hardware

Proof of Work



Virtual Mining



Bitcoin network

Mining rewards

MINER

Virtual mining

Benefits

- Removes all of the mining - much more efficient!
- Creates a simpler, closed system - easier to analyze from a game-theory perspective
- Decentralization - No mining hardware, so anyone could buy Bitcoin and mine on the network
- Physical miners have slightly different incentives than currency holders - if you had to use Bitcoin to mine, incentives are almost 100% in alignment

Drawbacks

- Security - with no real-world implications, can people game the system?
- What if someone gets 51% of the currency? They can keep getting more forever and hard for someone else to get on top
- Not entirely theoretically understood - “here be monsters”
- Can it even work on a large scale?

Kinds of Stake

- **Pure proof of stake** - How much currency you own is your share of the mining power *(Example: NavCoin)*
- **Delegated proof of stake** - Vote for a "representative" to vote for you *(Example: Nano)*
- **Proof of deposit** - Deposit coins as a "security deposit" to "stake" them for mining *(Example: Ethereum in the far future)*
- **Hybrid** - Use both a kind of proof-of-stake and proof-of-work, perhaps by making the work easier based on stake *(Example: Ethereum in the near future)*

Nothing-At-Stake Problem

- Also known as a “stake-grinding attack”
- Suppose an attacker with $\alpha < 0.5$ is trying to create a fork (for a double-spend attack, say)
- This would have a high cost in the real world for a proof-of-work currency

Nothing-At-Stake Problem

- However, this has NO cost to the attacker in a proof-of-stake currency! No opportunity cost!
- Miner can use stake on longest chain and simultaneously try to create a fork
- If fork doesn't come into fruition, all of the currency they used to stake will revert right back to them since only longest chain is valid

Solving Proof of Stake

- Need to have some way to punish bad actors
- Ethereum "Slasher" - miner has to sign blocks, and if a miner tries to use the same stake to sign two inconsistent chains, they are punished on the longest chain
- Should work in theory, but in practice…
- Ethereum has been promising proof-of-stake Real Soon Now since 2015 or so… and there was just a stall of their proof-of-stake system on the Ropsten testnet - https://xbt.net/blog/ethereum-blog/ethereums-constantinople-fails-to-activate-on-testnet/