Blockchain and Cryptocurrencies

Week 6 - Chapter 8: Alternative Mining Puzzles

Dr. Thi Thu Ha Doan   Prof. Dr. Peter Thiemann

Albert-Ludwigs-Universität Freiburg, Germany

SS 2020

Contents

Bitcoin Mining Puzzles

Mining puzzles are at the core of Bitcoin:

- determine the incentive system
- limit the ability to control the consensus process
- play an important role in steering and guiding participation

Essential Puzzle Requirements

Provide non trivial problem, miners have to invest to reach the next block, but It has to not be that difficult

- reasonable difficulty: puzzles are difficult, but not too hard
- fast verification: puzzle solutions need to be quick to verify
- adjustable difficulty: the difficulty can be changed over time
- progress free: probability of winning a puzzle solution should be roughly proportional to the hash power used

The Bitcoin Puzzle

Example (Bitcoin): SHA-256 hash below threshold

- checking solution is trivial
- simple to adjust mining difficulty
- puzzle solutions are found at a fairly predictable rate (roughly10 minutes)

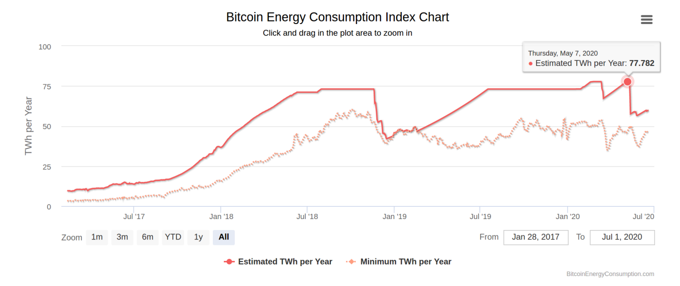
Digiconomist, "Bitcoin Energy Consumption Index": https://digiconomist.net/bitcoin-energy-consumtion

The Bitcoin Puzzle

Example (Bitcoin): SHA-256 hash below threshold

- checking solution is trivial
- simple to adjust mining difficulty
- puzzle solutions are found at a fairly predictable rate (roughly10 minutes)

Problem: Energy Consumption of Mining



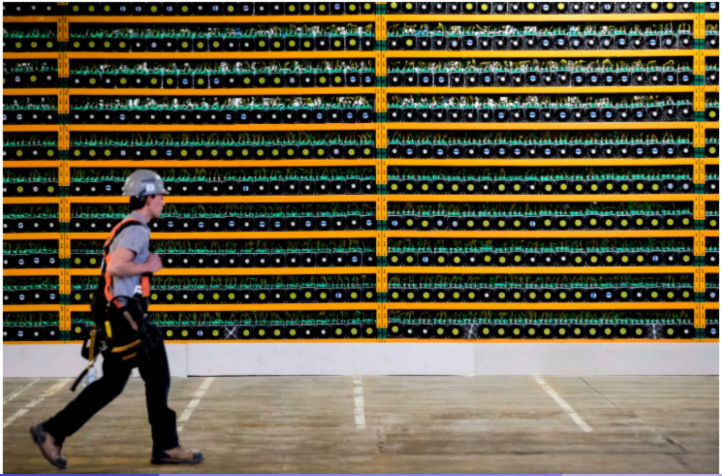
Digiconomist, "Bitcoin Energy Consumption Index": https://digiconomist.net/bitcoin-energy-consumtion

Bitcoin consumes more energy than Switzerland, according to new estima

Though researchers acknowledge that reliable estimates are 'rare'

By James Vincent | Jul 4, 2019, 8:33am EDT

SHARE



Energy Usage Ranking

The average yearly energy consumption of the Bitcoin network exceeds that of the entire nation of Switzerland. | Source: CBECI

Source: Cambridge Bitcoin Electricity Consumption Index https://www.cbeci.org/

Alternative Mining Puzzles

Are there different ways to design the puzzles?

Approaches

- ASIC resistance: some computers (ordinary) are less efficient than others (ASIC supported) at mining
- pool resistance: users delegate their participation to large centralized mining pools
- ...

Contents

ASIC Resistant Puzzles

Mining with specialized ASICs is more efficient than general-purpose computing equipment $\rightarrow$ ASIC-mining results in a major performance gain over normal hardware mining

ASIC Resistant

- Goal:
  - allow ordinary computers to mine!
  - prevent the large ASIC manufacturers from dominating the bitcoin mining
- Approach: reduce the "gap" between customized hardware and general purpose computers

Memory-Hard Puzzles

Most widely used puzzles designed to be ASIC Resistant

Fact: the performance of memory is more stable than for processors

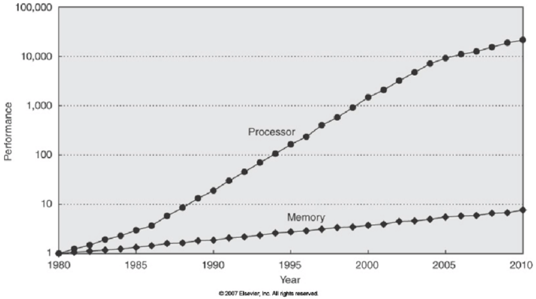


Figure: CPU-memory performances gap

Main idea: design mining puzzles that require a large amount of memory to solve

Figure from Nowak, A. (2014). Opportunities and choice in a new vector era. Journal of Physics: Conference Series. 523. 012002. 10.1088/1742-6596/523/1/012002

Scrypt

Most famous memory-hard mining puzzle used in many cryptocurrencies, such as Litecoin

- same puzzle condition (find hash below threshold)
- uses the memory-hard hash function scrypt instead of SHA-2
- requires fixed (large) amount of memory to be computed
- used for other security purposes (i.e., password hashing)

Scrypt Pseudocode

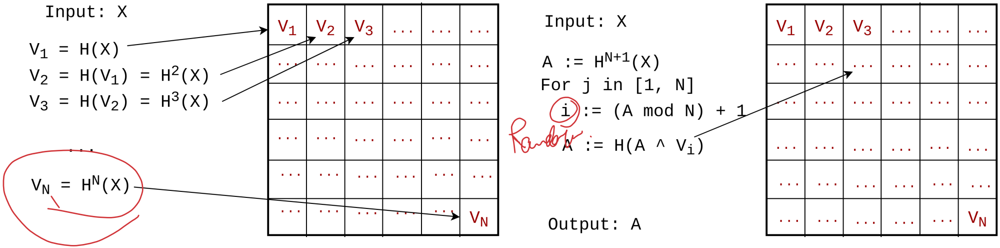
```
def scrypt(H, N, X):
    // initialize memory buffer of length N
    V = [0] * N

    // Fill up memory buffer with pseudorandom data
    V[0] = H(X)
    for i in range(1, N):
        V[i] = H(V[i-1])

    // Access memory buffer in a pseudorandom order
    A = H(V[N-1])
    for i in range(N):
  // choose a random index based on A
        j = A % N
  // update A based on this index
        A = Hy
        (A ^ V[j])
    return A
```

Scrypt computation

1. fill a large buffer of random access memory with random values
2. read from this memory in a pseudorandom order



(1) Write

(2) Read

Scrypt disadvantages

- Scrypt trades memory for computation speed
- Scrypt verification requires the same amount of memory/computation: N steps and N memory to check the correctness of the proof
- some scrypt ASICs are already available

Contents

Proof of Useful Work

Bitcoin:

- consensus protocol not designed for doing useful computations
- solving puzzles has wasted electric power

Could we design a puzzle that does not only waste energy, but is useful for the solution of practical computational problems?

Popular Volunteer Distributed Computing Projects

Project	**Founded**	**Goal**	**Impact**
Great Internet Mersenne Prime Search	1996	Finding large Mersenne primes	Found the new "largest prime number" twelve straight times
distributed.net	1997	Cryptographic brute-force demos	First successful public brute force of a 64-bit cryptographic key
SETI@home	1999	Identifying signs of extraterrestrial life	Largest project to date with more than 5 million participants
Folding@home	2000	Atomic-level simulations of protein folding	Greatest computing capacity of any volunteer computing project

These projects may be suitable for using as computation puzzles but what is the measure of success?

Primecoin

Puzzle based on finding chains of prime numbers

Cunningham chain

- A Cunningham chain of length k is a sequence $p_1, p_2, p_3, \ldots, p_k$ where each p_i is a large prime number and $p_i = 2p_{i-1} + 1$.
- The longest known Cunningham chain has length 19 and starts at 79.
- Conjecture: Cunningham chains exist for each length k.

Primecoin

Puzzle based on finding chains of prime numbers

Cunningham chain

- A Cunningham chain of length k is a sequence $p_1, p_2, p_3, \ldots, p_k$ where each p_i is a large prime number and $p_i = 2p_{i-1} + 1$.
- The longest known Cunningham chain has length 19 and starts at 79.
- Conjecture: Cunningham chains exist for each length k.

Computation puzzle with parameters m, n, and k

- Given a challenge x (the hash of the previous block), take the first m bits of x
- valid solution: any chain of length $\geq k$ where the first element is an n-bit integer and has the same m leading bits as x

Primecoin

Puzzle based on finding chains of prime numbers

Cunningham chain

- A Cunningham chain of length k is a sequence $p_1, p_2, p_3, \ldots, p_k$ where each p_i is a large prime number and $p_i = 2p_{i-1} + 1$.
- The longest known Cunningham chain has length 19 and starts at 79.
- Conjecture: Cunningham chains exist for each length k.

Computation puzzle with parameters m, n, and k

- Given a challenge x (the hash of the previous block), take the first m bits of x
- valid solution: any chain of length $\geq k$ where the first element is an n-bit integer and has the same m leading bits as x

Puzzle gets harder

- increasing n polynomial
- increasing m (large enough to avoid precomputation)
- increasing k (exponentially harder, perhaps infeasible)

Contents

Primary Disadvantages to Proof of Work Systems

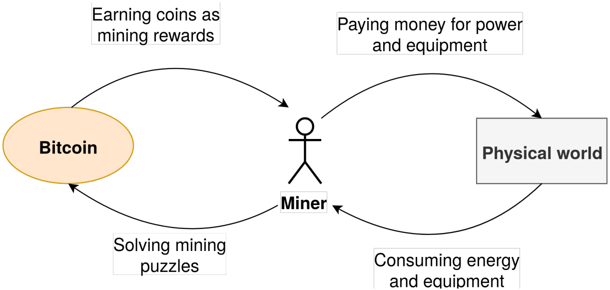
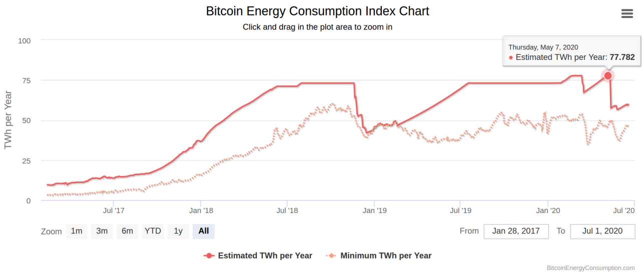


Figure: Loop on mining in Proof of work algorithms

Primary Disadvantages to Proof of Work Systems

Mining process burns energy and raw materials

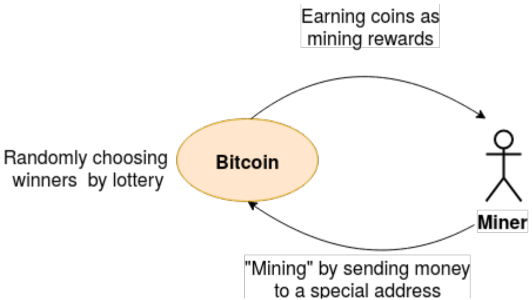


- Energy usage: enormous amounts of energy to secure the blockchain
- Mining pools: large make the blockchain centralized

Digiconomist, "Bitcoin Energy Consumption Index": https://digiconomist.net/bitcoin-energy-consumption

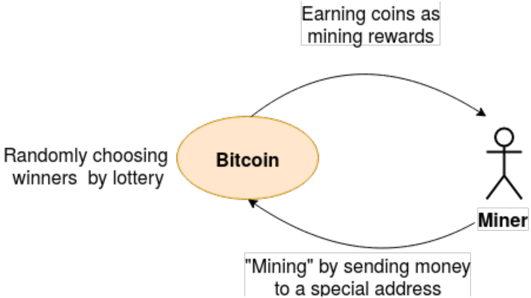
Virtual Mining

Allocate mining "power" to currency holders in proportion to amount of coins held



Virtual Mining

Allocate mining “power” to currency holders in proportion to amount of coins held



Why virtual mining?

- remove the wasteful half of the proof of work mining cycle
- may also remove the problem of trending toward centralization (large mining pool)

Proof of stake

- each coin holder has “stake” (i.e., account balance) in the coin system
- the miner of the next block is chosen randomly
- size of stake determines the probability to be chosen for mining

Contents

Peercoin: Coin Age-Based Selection

Stake based on coin age

- age of a coin = time since last use of the coin
- stake is determined by age

Miners must solve a SHA-256-based computational puzzle, but the difficulty of this puzzle is lowered for older coins

Contents

Tezos: Liquid Proof of Stake

Tezos

Tezos is a 3rd generation blockchain featuring

1. blockchain and cryptocurrency
2. expressive contract language
3. live-upgrade of protocol and governance

Tezos: Liquid Proof of Stake

Tezos

Tezos is a 3rd generation blockchain featuring

1. blockchain and cryptocurrency
2. expressive contract language
3. live-upgrade of protocol and governance

PoS in Tezos

- Proof of Stake (PoS) assigns minting power based on the proportion of coins held
- Besides minting there is validation also based on stake
- "Liquid": coin holders can delegate their coins to others for minting or validating.

Tezos: Liquid Proof of Stake

Tezos

Tezos is a 3rd generation blockchain featuring

1. blockchain and cryptocurrency
2. expressive contract language
3. live-upgrade of protocol and governance

PoS in Tezos

- Proof of Stake (PoS) assigns minting power based on the proportion of coins held
- Besides minting there is validation also based on stake
- "Liquid": coin holders can delegate their coins to others for minting or validating.

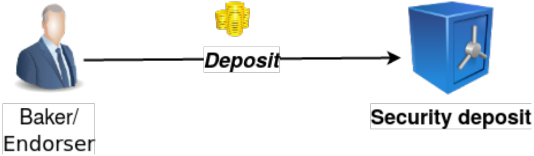
Remark

Peers that mint new coins are called **bakers** in Tezos

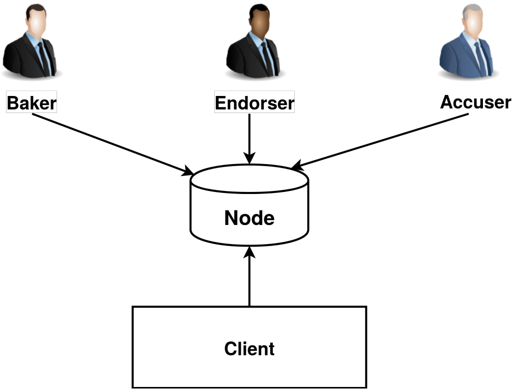
Tezos: Liquid Proof of Stake

Trust Model

- bakers deposit a certain amount of coins as stake (security deposit)
- bakers lose their stake if they approve invalid transactions



Tezos Architecture Overview



Tezos Architecture Overview

Node

The local component of the system

- manage the context as the local knowledge of the Tezos blockchain state and sync with the Tezos network
- communicate with other nodes via P2P network
- connect local endorser, baker, accuser, and client to the network

Tezos Architecture Overview

Node

The local component of the system

- manage the context as the local knowledge of the Tezos blockchain state and sync with the Tezos network
- communicate with other nodes via P2P network
- connect local endorser, baker, accuser, and client to the network

Client

Interface to the node

Tezos Architecture Overview

Job Descriptions

Baker

Baking (creating) new blocks

Tezos Architecture Overview

Job Descriptions

Baker

Baking (creating) new blocks

Endorser

Verifying the validity of a block and agreeing on a block by endorsing that block.

Tezos Architecture Overview

Job Descriptions

Baker

Baking (creating) new blocks

Endorser

Verifying the validity of a block and agreeing on a block by endorsing that block.

Accuser

Monitoring all blocks and looking for invalid transactions

Tezos Architecture Overview

- Everything's organized in cycles
- Baking rights and endorsement rights: determined at the beginning of a cycle.
- Incentivize: rewarded for baking and endorsing.
- Double-baking or double-endorsement: a security deposit is frozen and could be released or burnt.

Tezos Basic Concepts

Cycles

Blocks are group into cycles of BLOCKS_PER_CYCLE = 4,096 blocks.

- TIME_BETWEEN_BLOCKS = one minute: (1 cycle = 2 days, 20 hours, and 16 minutes)
- the current cycle: n, the nth cycle from the beginning of the chain
- PRESERVED_CYCLES = 5 cycles (14 days, 5 hours, and 20 minutes).

https://tezos.gitlab.io/

Tezos Basic concepts

Rolls

A roll represents a set of coins delegated to a given key: TOKENS_PER_ROLL = 10,000 (8,000 currently) tokens.

- each delegate has a stack of roll ids.
- roll snapshots are taken every BLOCKS_PER_ROLL_SNAPSHOT = 256 blocks (16 times per cycle).
- rolls are used to determine baking and endorsement rights

https://tezos.gitlab.io/

Tezos Basic concepts

Delegations

Active and passive delegates

- a passive delegate cannot be selected for baking or endorsing.
- a baker becomes passive for cycle n: if it failed to create any blocks or endorsements in the past PRESERVED_CYCLES cycles

https://tezos.gitlab.io/

Baking in Tezos

- A baker has to hold at least one roll (owned or delegated).
  - holding 2/10 of those rolls

  $\Rightarrow$ 20% probability of being given the rights to create the next block.
  - holding 10,000 XTZ or 19,999 XTZ: same probability to earn baking rights in the system.

https://tezos.gitlab.io/

Baking in Tezos

Baking rights are priorities

- For example: the net randomly selects a priority list as follows.
  Priority0 = Roll 6
  Priority1 = Roll 9
  Priority2 = Roll 4
  Priority3 = Roll 5
  ...
  Priority9 = Roll 7
- the holder of Roll 6 will have first priority in proposing the block.
- If "Roll 6" does not create and broadcast a block within 1 minute, the owner of Roll 9 may take over, and so on

https://tezos.gitlab.io/

Baking in Tezos

Minimal block delays

A block is valid only if its timestamp has a minimal delay with respect to the previous block's timestamp.

TIME_BETWEEN_BLOCKS[0] + TIME_BETWEEN_BLOCKS[1] * p + DELAY_PER_MISSING_ENDORSEMENT * MAX (0, INITIAL_ENDORSERS - e), where:

- TIME_BETWEEN_BLOCKS[0] = 60 seconds
- TIME_BETWEEN_BLOCKS[1] = 40 seconds,
- DELAY_PER_MISSING_ENDORSEMENT = 8 seconds,
- INITIAL_ENDORSERS = 24,
- p is the block's priority ,
- e is the number of endorsements the block contains.

https://tezos.gitlab.io/

Baking in Tezos

Baking reward

$$\text{Reward} = \text{block reward} + \text{all fees paid by transactions}$$
$$\text{Block reward} = e \cdot \text{BAKING\_REWARD\_PER\_ENDORSEMENT}[p']$$

- BAKING_REWARD_PER_ENDORSEMENT = [1.250, 0.1875]
- e is the number of endorsements the block contains
- p' depends on p

https://tezos.gitlab.io/

Baking in Tezos

Baking reward

$$\text{Reward} = \text{block reward} + \text{all fees paid by transactions}$$
$$\text{Block reward} = e \cdot \text{BAKING\_REWARD\_PER\_ENDORSEMENT}[p']$$

- BAKING_REWARD_PER_ENDORSEMENT = [1.250, 0.1875]
- e is the number of endorsements the block contains
- p' depends on p

Security deposit

- BLOCK_SECURITY_DEPOSIT = 512 XTZ per block created
- frozen for PRESERVED_CYCLES = 5 cycles.

https://tezos.gitlab.io/

Tezos Endorsements

ENDORSERS_PER_BLOCK $=32$ endorsers by randomly selecting active rolls.

- verify the last block baked (at level n) and emits an endorsement operation baked in block $n+1$.
- once block $n+1$ is baked $\Rightarrow$ no other endorsement for block n will be considered valid.
- an endorser may have more than one endorsement slot.

https://tezos.gitlab.io/

Tezos Endorsements

Endorsement reward

$$e * \text{ENDORSEMENT\_REWARD}[p']$$

- ENDORSEMENT_REWARD = [1.250, 0.833333]
- e is the number of endorsement slots

Security deposit

ENDORSEMENT_SECURITY_DEPOSIT = 64 XTZ per endorsement slot.

https://tezos.gitlab.io/

Tezos Delegation

Nodes can delegate coins to a baker

- not enough XTZ
- do not want to set up computing infrastructure to bake blocks

- delegation lets coin owners “lend” their coins to a baker: no transfer of ownership; baker cannot spend delegated coins.
- the baker has a higher probability of being selected.
- the baker shares the additional revenue with the coin holder.

https://tezos.gitlab.io/

Tezos Fork Choice Rule

The canonical chain based on the number of bakers that endorsed the block

- at every block height, 32 random rolls are selected to endorse a block.
- the block with the most endorsements is treated as the canonical one.

https://tezos.gitlab.io/

Contents

Algorand: Pure Proof of Stake

Algorand (https://algorand.com)

- new method to implement a public ledger
- convenience and efficiency of a centralized system, without the inefficiencies and weaknesses of current decentralized implementations.
- Based on ALGOrithmic RANDomness to select verifiers in charge of constructing the next block
- selections are provably immune from manipulations and unpredictable
- no different classes of users (as “miners” and “ordinary users” in Bitcoin)
- consensus based on fast algorithm for Byzantine agreement: probability of forks very small ($\approx 10^{-18}$)

from Algorand white paper https://arxiv.org/pdf/1607.01341.pdf

Algorand Proof of Stake Algorithm

Stake determined by tokens (coins) held by user

Phase 1: Proposal

- a token is randomly chosen among all tokens
- the owner of this token proposes, signs, and broadcast a new block

Phase 2: Agreement

- 1,000 tokens are randomly chosen among all tokens
- the owners of these tokens agree on (sign) the proposed block (from Phase 1)

Remark

all choices are public and consensual

https://www.algorand.com/resources/white-papers

Algorand Proof of Stake Algorithm

How to choose proposers and verifiers?

Self-selecting

- based on the last block . . .
- each user runs his/her own lottery, cannot cheat, but can prove the winning
- winners broadcast their winning tickets and their agreements about the proposed block

Remark

- a new community (of 1,000) is chosen for each proposed block
- not predictable $\Rightarrow$ very hard to subvert

https://www.algorand.com/resources/white-papers

Algorand's Techniques

A new block is generated via a message-passing Byzantine agreement protocol (BA*)

A new Byzantine agreement protocol

- its binary-input version consists of 3-step loop, in which a player i sends a single message m_i to all other players
- each loop of the protocol ends in agreement

The members of a selected community "quickly" agree on a new block via the this BA* protocol

https://www.algorand.com/resources/white-papers

Algorand's Techniques

Cryptographic sortition

- the players of BA* are selected to be a much smaller subset of the set of all users
- each new block B^r will be constructed and agreed upon, via a new execution of BA* by a separate set of selected verifiers, SV^r

https://www.algorand.com/resources/white-papers

Algorand's Techniques

The quantity (seed) Q_r

The last block B_{r-1} is used to determine the next verifier set as well as the leader in charge of constructing the new block B_r

- Q_r is unpredictable (therefore not influenceable) by powerful adversary
- users play a special role in the generation of the rth block

https://www.algorand.com/resources/white-papers

Algorand's Techniques

Q_{r-1} contained in B_{r-1} $\Rightarrow$ adversary might immediately corrupt all verifier and the leader

Secret cryptographic sortition and secret credentials

- leaders and verifiers secretly and independently learn of their role
- they can compute a proper credential to prove possession of the role
- the leader for the next block, secretly assembles the proposed new block and then disseminates it for certification with proof-of-leadership

https://www.algorand.com/resources/white-papers

Algorand's Properties

- amount of computation is minimal
- new block generated in less than 10 minutes
- will never leave the blockchain (no fork, almost certainly)
- all power resides with the users themselves

https://www.algorand.com/resources/white-papers

Algorand: Pure Proof of Stake

Decentralized ✓

Scalable ✓

Secure ✓

https://www.algorand.com/resources/white-papers

Thanks!