actually: Cryptocurrencies without Depletion of Physical Scarce Resources

Cryptocurrencies without Proof of Work

Iddo Bentov
Technion

Ariel Gabizon
Technion

Alex Mizrahi
chromaway.com

Financial Cryptography 2016 – 3rd Workshop on Bitcoin and Blockchain Research

Proof of Stake

Definition of *Proof of Stake*

If Alice has 10 coins and Bob has 50 coins in the system, then Bob has ≈ 5 times more decision-making power, and collects $\approx 5x$ fees.

Benefits of *Proof of Stake* in comparison to *Proof of Work*

- The operating costs of stakeholders are minuscule compared to those of miners $\Rightarrow$ it is more likely that the market can bear the cost of funding the security maintenance of the network.
- Stakeholders have a vested interest to keep the system secure, comparable to PoW miners with ASIC that's useless with any other cryptocurrency and useless for anything else.
- Centralization hazard of PoW: large data centers can obtain new ASIC equipment in bulk for a cheap price, compared to hobbyist miners who wish to the run the equipment at home.

Pure Proof of Stake

Can we have a decentralized cryptocurrency that relies ***only*** on *Proof of Stake*, without any PoW at all?

Problem #1: fair issuance of the money supply.

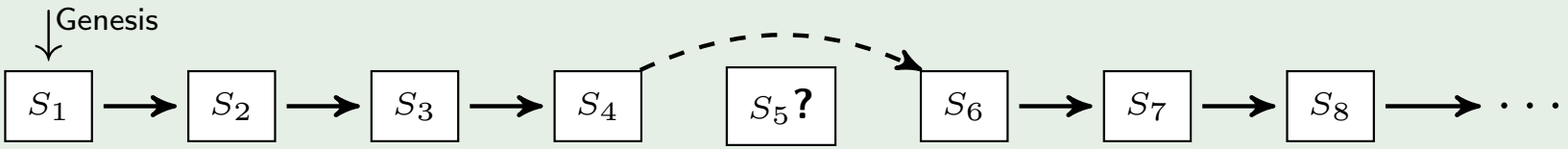
- Methods such as the voucher privatization in Russia (1992-1994) come to mind?
- Let's assume first that our initial state is a decentralized network of many small stakeholders, and then try reduce the full problem to this setting.

Problem #2: can the protocol be robust, or is it too fragile?

- Bribe attacks: is it easy for an attacker to double-spend by soliciting stakeholders to sign a hostile chain?
  - If the attack fails then the colluding stakeholders didn't lose much, because they don't need to deplete resources while participating in the attack (as opposed to PoW miners).
- Is it rational for stakeholders to sign multiple competing forks?

Toy Protocol

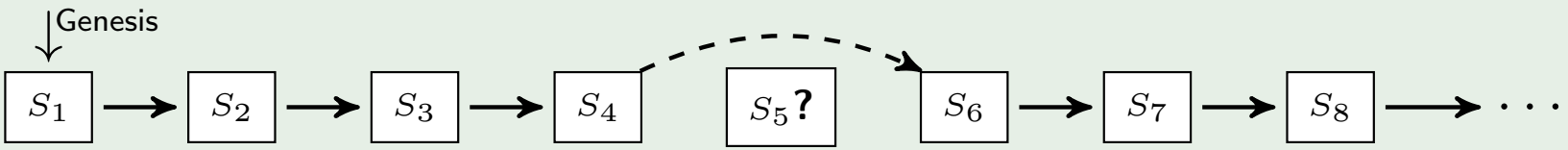
Stakeholders play (i.e., sign their block) according to a fixed order



- Naive assumption: $>50\%$ of the stakeholders are honest.
- #blocks to wait so that a payment won't be reversed w.h.p.?

Toy Protocol

Stakeholders play (i.e., sign their block) according to a fixed order



- Naive assumption: $>50\%$ of the stakeholders are honest.
- #blocks to wait so that a payment won't be reversed w.h.p.?

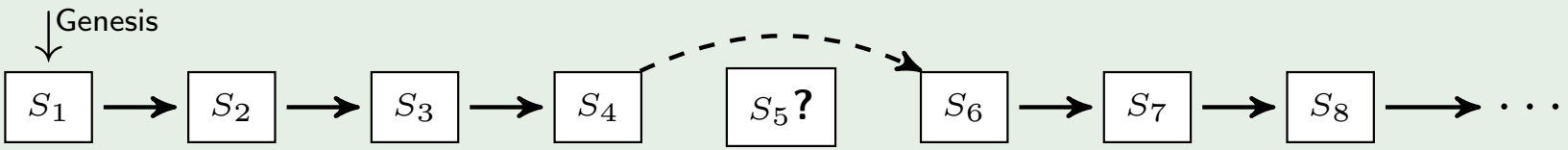
Monotone sequence: $\overset{1}{\square}, \overset{2}{\square}, \overset{3}{\square} \cdots \overset{N}{\square}, \overset{1}{\square}, \overset{2}{\square} \cdots$ (e.g., $N = 21{\cdot}10^6{\cdot}10^8$)

Digits of π: $\overbrace{3\,1\,4\cdots\bigcirc}^{51\text{ bits}}, \overbrace{\bigcirc\bigcirc\cdots\bigcirc}^{51\text{ bits}}, \overbrace{\bigcirc\bigcirc\cdots\bigcirc}^{51\text{ bits}}, \cdots$

PRNG(seed): $\overbrace{\bigcirc\bigcirc\cdots\bigcirc}^{51\text{ bits}}, \overbrace{\bigcirc\bigcirc\cdots\bigcirc}^{51\text{ bits}}, \overbrace{\bigcirc\bigcirc\cdots\bigcirc}^{51\text{ bits}}, \cdots$

Toy Protocol

Stakeholders play (i.e., sign their block) according to a fixed order



- Naive assumption: $>50\%$ of the stakeholders are honest.
- #blocks to wait so that a payment won't be reversed w.h.p.?

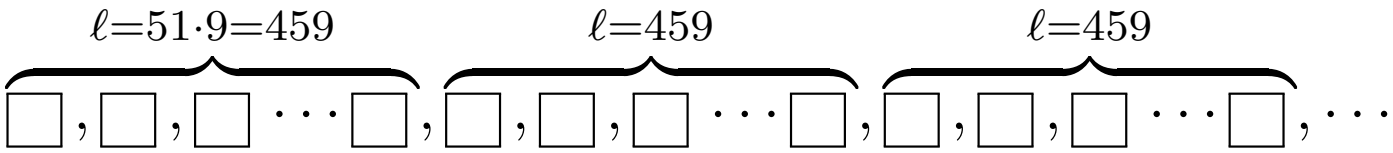
Monotone sequence: $\overset{1}{\square}, \overset{2}{\square}, \overset{3}{\square} \cdots \overset{N}{\square}, \overset{1}{\square}, \overset{2}{\square} \cdots$ (e.g., $N = 21{\cdot}10^6{\cdot}10^8$)

Digits of π: $\overbrace{3\,1\,4\cdots\bigcirc}^{51\text{ bits}}, \overbrace{\bigcirc\bigcirc\cdots\bigcirc}^{51\text{ bits}}, \overbrace{\bigcirc\bigcirc\cdots\bigcirc}^{51\text{ bits}}, \cdots$

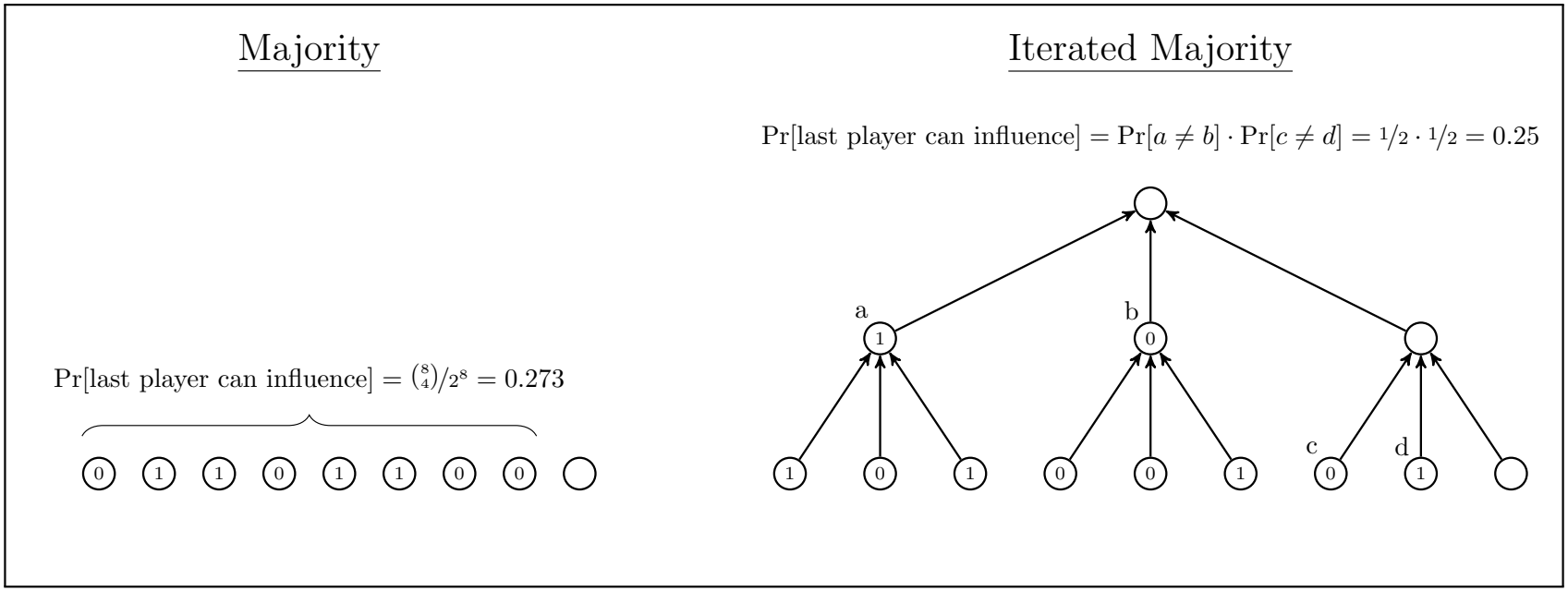
PRNG(seed): $\overbrace{\bigcirc\bigcirc\cdots\bigcirc}^{51\text{ bits}}, \overbrace{\bigcirc\bigcirc\cdots\bigcirc}^{51\text{ bits}}, \overbrace{\bigcirc\bigcirc\cdots\bigcirc}^{51\text{ bits}}, \cdots$

Problem: Attacker can see into the future and buy consecutive coins.

Construction 1: "Chains of Activity" (CoA)

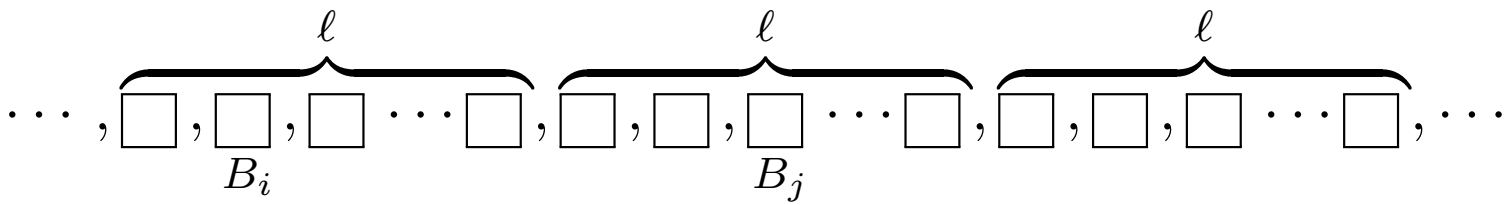


- Each block in a segment of ℓ blocks contributes a bit.
- The ℓ-bits seed picks the identities in the (after-) next round.
- Translate string to coin by *follow-the-satoshi* $\Rightarrow$ linear rewards.



Reward probability for a stakeholder who plays last in a segment.

Construction 1: "Chains of Activity" (CoA) - few of the rules

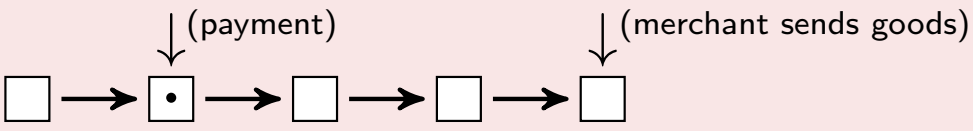


- The distance between the timestamps of B_i and B_j must be at least $|j - i - 1| \cdot T_0$, for example with $T_0 = 5$ minutes.
- Honest nodes in the network will consider a newly created block to be invalid if its timestamp is too far into the future relative to their local time.
- This accommodates offline stakeholders while avoiding a hostile competing chain with large gaps between blocks.
- If the stakeholder's unspent output is less than C_0 coins, she must sign an extra "security deposit" that shows that she controls at least C_0 coins.
- These C_0 coins become locked for a T_1 safety duration, and will be confiscated if she double-signs in a competing chain.

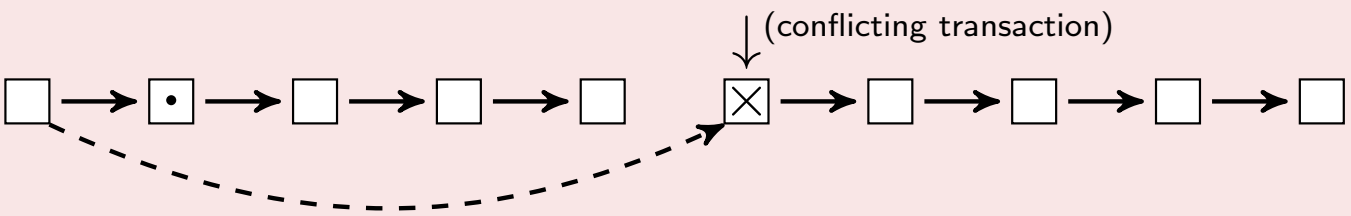
Construction 1: "Chains of Activity" (CoA) - collusion attack

- Protocol rule: if the network nodes see multiple competing blockchains, they consider the blockchain that consists of the largest number of blocks to be the winning blockchain.

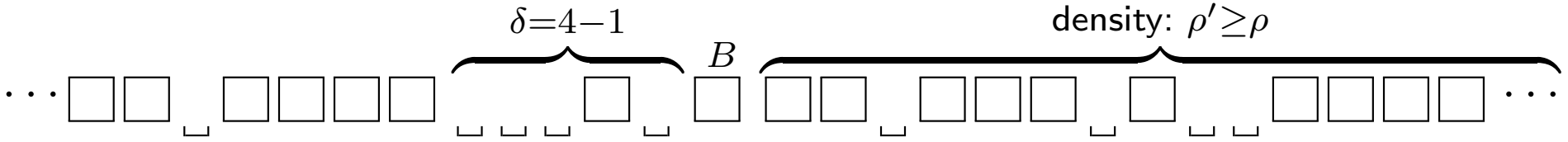
How does a double-spending attack in CoA look like:



For a double-spending attack to succeed, an alternative history of 5 blocks needs to be created, by extending the previous block with a chain that includes a conflicting transaction:



Construction 1: "Chains of Activity" (CoA) - flavor of a security proof



Density assumption. Let $\rho > 1/2$. In the longest blockchain, for every segment of K or more potential blocks, at least ρK of those blocks were created.

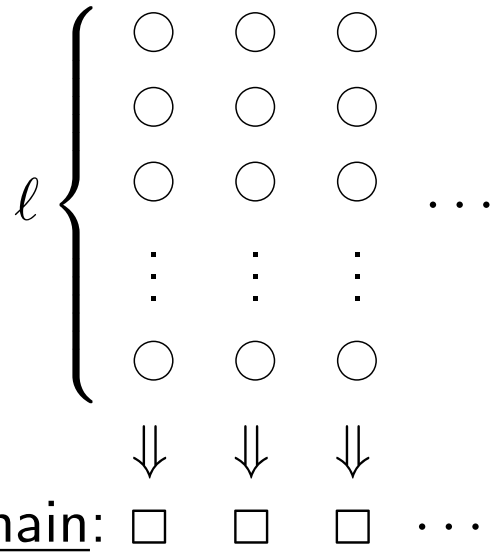
- ε: average reward that a stakeholder earns for creating a block.
- V: value that the attacker assigns for reversing the block B.
- δ: #blocks missing in the largest segment with participation rate $\leq 1/2$ prior to B, so the density assumption $\Rightarrow \delta < K$.

$\Rightarrow$ the merchant is safe by waiting for S confirmations, for S that satisfies $V < \varepsilon(\rho S - K + 1)$.

- Example: $\rho = 7/10$, $K = 20$, $\varepsilon = 10$ coins, $V = 100$ coins.
- $10 \cdot (7/10 \cdot S - 19) > 100$, so $S = 42$ blocks are sufficient.

Construction 2: Dense-CoA

Each block of the chain is created by a group of ℓ stakeholders:



Each ℓ stakeholders engage in a 2-round protocol to create a block

- Round 1: for every $j \in \{1, 2, \ldots, \ell\}$, the stakeholder S_j picks a random secret $R_j \in \{0,1\}^n$, and broadcasts $h(R_j)$.
- Round 2: for every $j \in \{1, 2, \ldots, \ell - 1\}$, the stakeholder S_j signs the message $M \triangleq h(R_1) \circ h(R_2) \circ \cdots \circ h(R_\ell)$, and broadcasts her signature $\mathrm{sign}_{sk_j}(M)$ and her preimage R_j.

Construction 2: Dense-CoA - properties

- $h : \{0,1\}^n \to \{0,1\}^n$ is a one-way permutation.
- The seed that derives ℓ stakeholders for the next block is $hash(R_1, R_2, \ldots, R_\ell)$.
- $\Rightarrow$ this seed is computationally indistinguishable from random even if only a single stakeholder S_j picked a random R_j.
- $\Rightarrow$ the identities of stakeholders who should create the next blocks are not known in advance.
- This makes collusions and bribe attacks more difficult.

Construction 2: Dense-CoA - properties (contd.)

- We use a signature scheme with *multisignature* (defined e.g. at ePrint 2002/118) support, so S_ℓ can aggregate the signatures $\{\mathsf{sign}_{sk_j}(M)\}_{j=1}^{\ell}$ into a single signature $\hat{\mathsf{s}}(M)$.
- The size of $\hat{\mathsf{s}}(M)$ depends only on the security parameter of the signature scheme (not on ℓ), and verifying $\hat{\mathsf{s}}(M)$ is faster than verifying ℓ ordinary (ECDSA) signatures separately.

Construction 2: Dense-CoA - details

- S_ℓ signs and broadcasts a finalized block that includes the ℓ preimages $R_1, R_2, \ldots, R_\ell$, and $\hat{\mathsf{s}}(M)$.
- To verify that the block B_i is valid, the network nodes invoke h to compute the images $h(R_1), h(R_2), \ldots, h(R_\ell)$, then concatenate these images to form M, and then check that $\hat{\mathsf{s}}(M)$ is a valid signature of M with respect to the public keys $pk_1, pk_2, \ldots, pk_\ell$ that control the winning coins of the stakeholders $S_1, S_2, \ldots, S_\ell$.
- If some of the ℓ stakeholders are offline or otherwise withhold their signatures, then after timeout (e.g., 5 minutes) the nodes who follow the protocol will derive alternative ℓ identities from the previous block.

Construction 2: Dense-CoA - susceptibility to DoS by large stakeholders

- The parameter ℓ should be big enough in order to prevent large stakeholders from controlling consecutive seeds and re-deriving themselves.
- For example, to force a stakeholder who holds 10% or 20% of the total stake into making $\approx 2^{100}$ $hash$ invocations on average until re-deriving herself as all of the ℓ identities of the next block, we need $\ell = 30$ or $\ell = 43$, respectively.
- $\Rightarrow$ with 5 minutes timeout and $\ell = 23$, a malicious stakeholder with 10% of the total stake will have $1 - (90/100)^{23} \approx 91\%$ probability to be one of the derived stakeholders $S_1, S_2, \ldots, S_\ell$ and then refuse to participate in creating the next block, so it will take $5 \cdot (1 - 91\%)^{-1} \approx 56$ minutes on average to create each next valid block.
- Large stakeholders probably wouldn't wish to diminish the value of their stake...

The costless simulation attack

Costless simulation

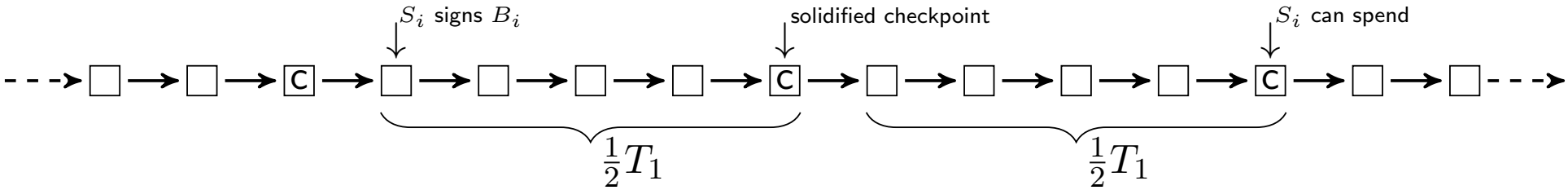
- In any decentralized system where extending the ledger history requires no effort $\Rightarrow$ there exists an attacker who can prepare an alternative history of the ledger without a cost.
- In pure *Proof of Stake* systems, stakeholders who held a large majority of coins a long time ago and have since traded those coins for other goods can collude to extend the ledger from the point at which they had control over the system.
- It is rational for them to mount this attack since it is costless and they have no stake in the current system.

The costless simulation attack (contd.)

Costless simulation

- In (Dense-)CoA, even a single stakeholder with few coins can create a fork with large enough time gaps between blocks, but other nodes will reject this forks because the timestamps will be too far ahead in the future relative to their local time.
- $\Rightarrow$ if the average participation level among current stakeholders is $p\%$, and the stakeholders who collude to carry out this attack have had control at the earlier history over $q\%$ of the coins, then $q > p$ implies that the attack will succeed.
- Because $p\% = 1$ is highly unlikely, and collusion among participants who held $q\% > p\%$ stake at an earlier point is costless and rational, this attack vector is quite dangerous.

The costless simulation attack - mitigation



- To mitigate the costless simulation attack, we propose checkpointing at constant intervals as a rigid protocol rule.
- Each node that receives a candidate checkpoint block B_j that extends the earlier candidate checkpoint block B_i (i.e., $j \geq i + \frac{1}{2}T_1$), solidifies the history from genesis until B_i.
- B_j can still be discarded as a result of a competing fork.

The costless simulation attack - mitigation (contd.)

Problems with the checkpointing mechanism

1. New nodes who enter the decentralized network for the first time cannot tell whether the checkpoint blocks that they receive are trustworthy.
   - The new node cannot simply download all blocks from genesis, because a costless simulation attack may cause the node to download a hostile fork.
2. Adversarial stakeholders can prepare a fork of length $\frac{1}{2}T_1 + 1$ and broadcast it at the same time as the honest branch, to create an irreversible split among the network nodes.

- $\Rightarrow$ Rely on an external "Web of Trust" to fetch old blockchain data from reputable sources.
- With our proposed parameters, a fork of $\frac{1}{2}T_1 + 1$ blocks represents more than one week of ledger history.

Initial distribution of the money supply

How to conduct fair issuance of the money supply?

- Have an auction or an IPO of some sort? This goes against decentralization, as all the coins are initially controlled by a trusted party.
- Use PoW ***only*** for minting new coins into circulation?
- With this option, we can peg the value of each new coin to the cost of production, because there is no need for predictable 10-minute gaps between blocks.
- $\Rightarrow$ **stable exchange rate during the issuance phase.**
- Difficulty re-adjustment is needed only if blocks get created very fast, to avoid unfairness due to propagation latency.
- For example, the difficulty can re-adjust so that the interval between blocks is at least 1 minute.

Initial distribution of the money supply (contd.)

Issuance of the money supply by using PoW

- Assume that the cost of producing a coin in terms of electricity and erosion of the equipment will be approximately fixed throughout the issuance process.
- Then, we are effectively pegging the value of the newly minted coins to the cost of producing these coins, because:
  1. If the value of each coin is less than the cost of producing a coin, then more mining equipment will be brought online to produce larger amounts of coins at the fixed cost, and then larger amounts of newly minted coins will come into existence - which implies that the value of each coin decreases.
  2. If the value of each coin is more than the cost of producing a coin, then some of the mining equipment that participates in the minting process will quit, and then smaller amounts of coins will come into existence - which implies that the value of each coin increases.

Thank you.

version 1.0