Proof of Stake:

Relying on computational resources $\Rightarrow$ Relying on stake

"a majority of stake* is honest". ——

PoW:

$$H(\text{prev\_hash}, \text{merkle\_root}, \text{nonce}) < D$$

nonce ↑ keeps changing. D ↖ difficulty parameter

PoS: Attempt 1:

$$H(\text{prev\_hash}, \text{merkle\_root}_i, pk_i) < D \cdot \text{stake}_i$$

merkle_root_i ↳ "set of txns."

Problem: Grinding (on a^{100} transactions).

Attempt 2:

$$H(\text{prev\_hash}, pk_i) < D \cdot \text{stake}_i$$

Problem: How to set D?

either too many blocks, & no block deadlocked.

Attempt 3:

$H(prev\_hash, ts, pk_i) < D \cdot stake_i$

You know you are in the committee KES Adv.

Adv. can adaptively choose the bad guys.

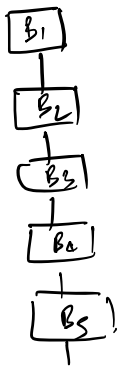
Attempt 4:

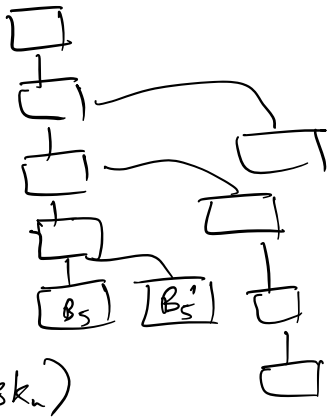
$VRF(prev\_hash, ts, sk_n) < D \cdot stake_i$

KES.

* No predictability but you lose security

Nothing at Stake:





$VRF(H(B_5), ts, sk_n)$

$VRF(H(B_4), ts, sk_n)$.

$VRF(H(B_5'), ts, sk_n)$

$\frac{1}{2}$ Byzantine faults $\Rightarrow < \frac{1}{1+e}$ Byzantine faults

$\simeq 27\%$.

Goal: Avoid grinding on blocks.

Instead of ~~be~~ computing on previous block, if we compute on a much earlier block,

Attempt 5:

$VRF(\text{genesis}, ts, sk_n) < D \cdot stake_i$ ← Ouroboros Praos (Cardano).

stops NaS attack.

Problem: You know far down the line, you are the leader.

D...

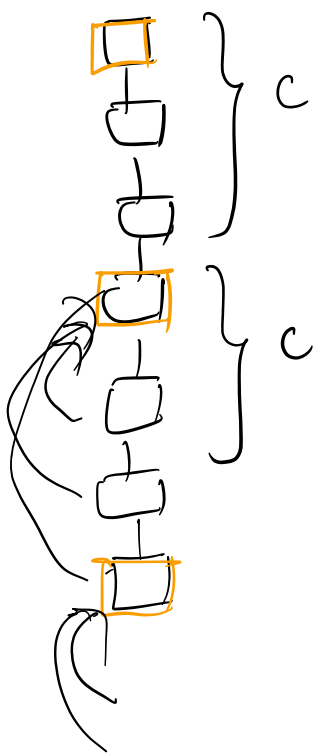
Bribing issues.

Security and predictability

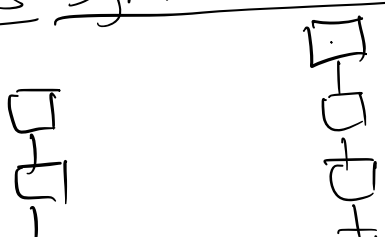
Security & predictability:

Attempt 6:

$VRF(\text{prev\_blk}, \text{ts}, sk_i) < D \cdot stake_i$ (paper.)

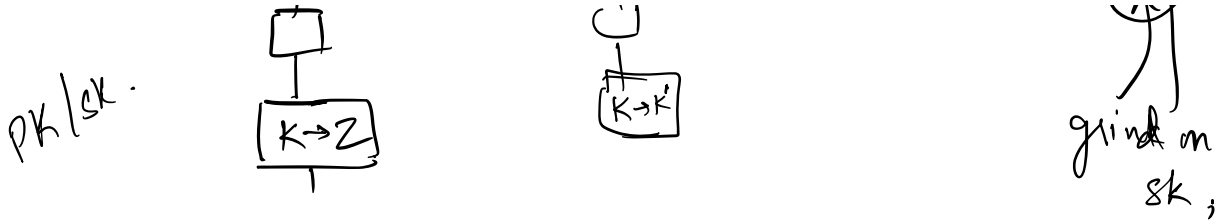


Static vs Dynamic stake:



$IM : K : (pk, sk)

$K' : (pk, sk')$



Soln: Use stake from some k blocks earlier (Ouroboros Genesis)