Proof of Stake:

Relying on computational resources

⇒ Relying on stake

"a majority of stake is honest".

POW:

\[ H(\text{prev-hash, mekle-root, nonce}) < D \]

[difficulty parameter]

keeps changing.

PoS: Attempt 1:

⇒ \[ H(\text{prev-hash, mekle-root, } pk_i) < D \cdot \text{Stake} \]

set of term \[^3\]

\[ a^{10b} \]

Problem: Grinding (on transachine).

Attempt 2:

\[ H(\text{prev-hash, } pk_i) < D \cdot \text{Stake} \]
Problem: How to set $D$?

either too many blocks, or no block dead locked.

\underline{Attempt 3:}

$H(\text{prev-hash}, ts, pk_i) < D \cdot \text{stake}$

You know you're in the committee $KES$.

Adv. can adaptively choose the bad guys.

\underline{Attempt 4:}

$\text{VRf} (\text{prev-hash}, ts, sk_i) < D \cdot \text{stake}$

$KES$.

\& No predictability but you lose security.

Nothing at stake.
Goal: Avoid grinding on blocks.

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

\[ \text{Attempt}^{5'}: \quad VRF(\text{\underline{genesis}}, ts, sk_n) < D \cdot \text{Stake} \]

stops NaS attack.

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

Security and predictability

Security & predictability:

Attempt 6:

\[ \text{VRF} \left( \text{prev-block, ts, sk_i} \right) < D \cdot \text{stake, \textit{(paper.)}} \]

Static vs Dynamic Stake:

\[ \text{IM}: K \left( Pk, sk \right) \]

\[ K' \left( Pk', sk' \right) \]
Soln: Use stake from some f blocks.

earlier (Octobor Genesis)