Hi guys, any comments on my VDF&VRF based PoS?

- Winner-takes-all is avoid by exponential VDF steps that adjust using average block time.
- VRF (ie. bijective signing) and current stake is used to generate pseudo-random seeds for each miner to calculate their current block VDF steps to compute. Basically is a Synthetic Proof-of-Work, because we are simulating Random Clocks.

Thanks!

Draft:

github.com

manfr3d/vixify/blob/master/README.md

Vixify Blockchain

A modern pure Proof-of-Stake blockchain based on a verifiable delay functions (VDF) and a verifiable random function (VRF). Implements a synthetic Proof-of-Work using the VDF and VRF based on coin stakes and non-parallelizable mining.

build unknown

Summary

Vixify is a blockchain adopting a pure Proof-of-Stake consensus protocol based on a verifiable random function (VRF) and a verifiable delay function (VDF) that has the following properties: a) all addresses with a positive stake can participate in consensus; b) is fair regarding the stake and the distribution of rewards; b) is tolerant to several classic attacks such as Sybil attacks, "Nothing-at-stake" attacks and "Winner-takes-all" attacks.

Blockchain Features

Vixify Blockchain has the following features:

- Proof-of-Stake - only stakeholders can participate in consensus and recieve rewards.
- Energy-efficient Single-thread Mining - Using a VDF allow the blockchain with blocks mined on a single-thread by each stakeholder. Under certain chip technologies the design if secure (for example, no miner has a chip technology that is x3 or x4 faster than the current state of the art in commercial chips).
- Secure - Using a verifiable random function (VRF) allows next-block miner to be unpredictable, discouraging attacks on stakeholders nodes.
- Catastrophic Failure-tolerant - supports catastrophic >50% stake failure or network fragmentation, unlike PBFT Proof-of-Stake blockchains that stop working under catastrophic conditions.

This file has been truncated. show original

Very basic Proof of Concept (with pseudo-VDF):

GitHub

manfr3d/vixify

A modern pure Proof-of-Stake blockchain based on VDFs and VRFs, design to be a "plug-n-play" replacement of Proof-of-Work protocols. - manfr3d/vixify

Consensus:

[

vixify-README-md at master · manfr3d-vixify 13-04-2020 17-38-38

1894×1582 545 KB

](https://ethresear.ch/uploads/default/original/2X/7/74c26969d2dc98b4ec0d7dc581cc676bdb4b410c.png)

Winner-takes-all Protection draft:

difficulty = slow-moving variable self-regulated by average block time (for example, miners can move this deterministically by 1% each block up or down).

minerStake = current block # of coins stake of the miner address holding the coins

stake = minerStake / totalCoins

slot = int(round(1/stake))

miner_vrf_seed = vrf_sign(prev_block_hash, miner_private_key). # VRF is just a deterministic signature, bijective.

random.set_seed(miner_vrf_seed)

slotRange = [1:slot+1] # the range of possible integer slots for a given miner holding stake on a given address.

slotNumber = random.random_integer_on_range(slotRange) # slotNumber = a deterministic slot number based for address or miner holding stake on a given blocknumber.

vdfSteps = 2 ^ (difficulty * slotNumber) # this number is like the average mining time of traditional PoW.