I've seen a few talks about decentralised, trustless pools including one made by Carl Beekhuizen & Dankrad Feist (Devon 5). I wanted to describe a more detailed system of how this might look like in a real-world application.

This is also my first post on ethresear.ch so that's exciting!

introduction

The need behind staking pools is simple, as ETH price goes up so does the cost of becoming a validator. Many won't be able to put 32 ETH when ETH is $500 or $1000.

This will encourage centralized behaviour as they will deposit their ETH in an exchange that will stake for them.

A decentralized staking pool should maintain the same level of engagement that is required from a validator but at a lower ETH price point. That is, a participant in a staking pool should still be online, sign his own attestations/ proposals and get penalized if he doesn't.

This design takes advantage of BLS signatures, distributed key generation (DKG), proactive secret sharing and a consensus layer to manage signing operations of a pool of validators, all maintaining a single validator.

A group of participants in a pool that operates validator V_i

is denominated as P_i

.

In order to form P_i

there is a setup phase happening between the participants of P_i

and a contract.

Setup phase

1. A participant wishing to join some pool should start by depositing a pre-defined constant value of ETH. All pool participants participate with the same amount.
2. a list of active participants is kept within the contract.
3. A pool initiator calls an assemble function on the contract to randomly assemble P_i

out of the active participants list.

1. participants of P_i

go through a distributed-key-generation (DKG) process where they collectively generate V_i

's public key pk_i

. sk_i

is never re-constructed by the participants.

1. Each participant of P_i

should verify the shares he got from the other participants. Otherwise he should exit the process.

After P_i

was constructed and all participants verified pk_i

they need to initiate a deposit to the beacon chain deposit contract.

When V_i

becomes active, all participants of P_i

have the following responsibilities

:

1. Be online, up to date with the latest beacon-chain block and act upon pool tasks.
2. Randomly get selected, as a coordinator, to prepare a task for the rest of the committee depending on V_i

's duties (beacon chain duties)

1. Should not propose a task which could get V_i

slashed

Consensus layer

Has the responsibility of coordinating between P_i

's participants, select a coordinator every epoch that has the responsibility of proposing a duty (attestation, block proposal) for the pool to sign on.

Once 2/3+1 of P_i

sign the task, the coordinator will broadcast the signed task.

The coordinator will simply aggregate individual participant's signatures thanks to BLS and its awesome features.

If the coordinator fails to create a valid task (for example a valid future block to attest to) or doesn't create any task he could get penalised.

Proposing a task that could get V_i

slashed could cause the coordinator to get slashed.

Penalties

Failing to meet any of the above responsibilities will cause the participant to get penalised with an interest bearing fine.

Unpaid fines could, eventually, get the participant slashed out of the pool and lose his stake.

Fines are paid on eth 1.0 to the contract.

Interest on unpaid fines should add up super-linearly until they reach some pre-defined MAX_PENALTY which then gives the participant some time to comply or get slashed

Replacing a slashed participant

To maintain a full quorum of participants for V_i

that can successfully carry out it's duties, slashed participants need to be replaced. Otherwise, if more than 1/3 of participants get slashed the pool will get halted.

To replace a participant:

1. The slashed participant's stake will be auctioned off to the highest bidder. A mechanism is TBD as there are some risk factors associated with existing participants buying out control of the pool and taking over it.
2. A bidder must calculate his own risk when bidding as the risk of the pool gets higher as more previous participants were replaced (see below Slashed participants collusion risk)
3. Once a winning bidder was chosen, P_i

will start a key rotation between all active members. Including the new bidder and excluding the slashed participant. Taken from here or here

1. The original participants list of P_i

were all active at round 0 (R_0

) of V_i

. Every key rotation, R

's index increases. We call the current round R_j

and the current active P_i

as P_{i_j}

Slashed participants collusion risk

A slashed participant still has a share of the secret that can re-construct V_i

's private key or sign on its behalf. A slashed participant can try and collude with 2/3-1 of the other participants to try and get V_i

slashed or (when it’s possible) transfer assets to himself.

If the number of such slashed participants is low, the risk of them colluding is low. The more slashed participants there are the higher the overall risk of the pool is.

bidder that buy-out slashed participants should take such risk into their considerations and the final bid they submit.