Stake House Withdrawal Sweep Properties

Contract $ShanghaiSweepReporting$

For convenience define $\mathcal{T} = universe.transactionManager()$.

Invariants

1. $totalSweepsReportedAgainstUnknownTopUpsForBlsPublicKey[k] \leq totalETHSentToBlsPublicKey[k] - 32\, ether$
2. $totalETHSentToBlsPublicKey(k)$ % $1\, gwei = 0$

The $validatorSetSizeBeforeShanghai$ is the active validators at time shanghai, supposed to be the number of registered validators at time shanghai minus the validator activation delay.

Property of function $previewTotalMintableDETH$

Assuming non-reverting input, this function satisfies

$$max(0,\, R.sumOfAllSweeps \cdot 1\, gwei - b - t) = previewTotalMintableDETH(R),$$

where

- $b_0 = hasBLSPublicKeyBeenReportedInShanghai[R.blsPublicKey]$
- $t_0 = totalSweepsReportedAgainstUnknownTopUpsForBlsPublicKey[R.blsPublicKey]$
- $b = universe.saveETHRegistry()$
  $.dETHRewardsMintedForKnot(R.blsPublicKey),$
  if $b_0 = false$ and $R.sweeps[0].validatorIndex < validatorSetSizeBeforeShanghai$
- $b = 0,$
  if $b_0 = true$ or $R.sweeps[0].validatorIndex \geq validatorSetSizeBeforeShanghai$
- $t = R.totalETHSentToBLSKey - 32\, ether - t_0.$

Properties of function $reportSweeps$

First grab some variables from the pre-state:

- $b_0 = hasBLSPublicKeyBeenReportedInShanghai[R.blsPublicKey]$
- $t_0 = totalSweepsReportedAgainstUnknownTopUpsForBlsPublicKey[R.blsPublicKey]$
- $b = universe.saveETHRegistry()$
  $.dETHRewardsMintedForKnot(R.blsPublicKey),$
  if $b_0 = false$ and $R.sweeps[0].validatorIndex < validatorSetSizeBeforeShanghai$
  - $b = 0,$
    if $b_0 = true$ or $R.sweeps[0].validatorIndex \geq validatorSetSizeBeforeShanghai$

Now apply $x = reportSweeps(R,\, S,\, V)$ and define the following variables in the post-state:

- $t = R.totalETHSentToBLSKey - 32\, ether - t_0.$

The properties of the post state are

1. For each sweep s in $R.sweeps$,
   $isSweepReported[s.withdrawalIndex] = true$
2. If $R.sweeps[0].validatorIndex < validatorSetSizeBeforeShanghai$ then
   $hasBLSPublicKeyBeenReportedInShanghai[R.blsPublicKey] = true$
3. If $V.activeBalance \geq 32\ ether\ /\ 1\ gwei$ then
   $universe.accountManager().getLastKnownActiveBalance(k) = 32\ ether/\ 1\ gwei$
4. $totalSweepsReportedAgainstUnknownTopUpsForBlsPublicKey[R.blsPublicKey] =$
   $t_0 + min(max(0,\ R.sumOfAllSweeps \cdot 1\ gwei - b),\ t)$
5. $x = previewTotalMintableDETH(R)$
6. The $savETHRegistry$ mints x dETH for the validator with BLS key $R.blsPublicKey$.

Properties of function $reportAndWithdrawETH$

Properties 1-5 are the same as those for $reportSweeps$. And then property 6 of $reportAndWithdrawETH$ is that the amount x is withdrawn as ETH (rather than minted as dETH) so that the sender's ETH balance will increase by x and account manager's ETH balance decrease by x. But note if x is too small then $reportAndWithdrawETH$ will not succeed when $\_optimisticWithdrawal = false$.

Properties of function $reportSweepsForMultipleBLSPublicKeys$

Performs n calls to $reportSweeps$.

Properties of function $function\ unwrapDETH$

Burn the amount of dETH and transfer the amount from account manager to sender.

Contract $FullWithdrawals$

Properties of function $reportFinalSweepAndWithdraw$

Given a successful invocation of $reportFinalSweepAndWithdraw(T,\ K,\ W,\ E,\ V,\ S)$, the action on the state is

1. $isFullSweepReported[E.sweep.withdrawalIndex] = true$
2. $universe.slotRegistry().userWithdrawn(msg.sender,\ K) = true$
3. $finalSweepAmountReportedForBlsPublicKey(K) = E.sweep.amount * 1\ gwei$
4. $sweepReportingContract.totalETHSentToBlsPublicKey(K) = T \cdot 1\ gwei$
5. x ETH are transferred from account manager to $msg.sender$

for $x = sweepReportingContract.totalETHSentToBlsPublicKey(K) -$
$universe.saveETHRegistry().dETHToken().balanceOf(msg.sender) -$
$sweepReportingContract.totalReportedETHNotWithdrawn(K)$.