It doesn't matter if it is costless to simulate, provided that the simulation is sufficiently different.

Andrew Poelstra's paper about how POS is impossible:
https://download.wpsoftware.net/bitcoin/pos.pdf

His arguments rest upon a false assumption: that consensus can only be achieved if the longest blockchain is also the correct one.

From this false assumption, it is easy to prove that long-range attack is always possible in a POS blockchain, because it does not cost anything to build a longer blockchain. This is known as "costless-simulation argument".

In my proposal, long-range attack forks work the same way as altcoins. Bitcoin and Litecoin don't care who is longer, they would never download blocks from each other. Similarly, in my scheme, forks that happened over 3000 blocks ago would never download blocks from each other.

implementation details below:

I start with this https://blog.ethereum.org/2014/01/15/slasher-a-punitive-proof-of-stake-algorithm/ and make a couple changes.

1) The reward for mining a block should be negative. No one makes blocks until there are enough transaction fees to pay for a new block.

2) The POW for each block should be far lower. 5 seconds on a core 2 DUO doing CPU mining.

I deviate from Vitalik's Slasher in a couple more ways, but to explain them, I need to explain how Vitalik's Slasher is broken. This is called a "long-range attack". There is no incentive to protect old private keys after you spend the money. It is possible that someone could collect a lot of old private keys, and that person would own more than 50% of money at a time in history. With this they can build a fork which is longer than the real fork. When new nodes boot up and join the network, they will download the wrong fork, and will be unable to receive funds.

To make this type of attack impossible, I make the following changes.

1) Everyone must periodically pledge to a fork, your money gets deleted in every other fork.

2) If anyone simultaneously pledges to completing forks, their money is erased in both forks.

3) Each node needs a list of public keys that have money. When a node boots up, it downloads the longest blockchain where at least one of the public keys from the list still has money.