Oracle manipulation under POS

The transition of Ethereum mainnet from PoW to PoS resulted in significant changes in block building. One of the differences under PoS is that validators/pools with a large market share have a high probability of producing consecutive blocks, statistical analysis by Alvaro Revuelta. Consecutive blocks by the same validator/pool increases the risk of TWAP (Time-Weighted Average Price) oracle manipulation, research by Uniswap.

Because of the increased risk of TWAP oracle manipulation the current design seems insufficient for use in DeFi (Decentralized Finance). For example Euler changed the oracle for multiple markets from TWAP to ChainLink, Proposal 30. To increase the safety of the TWAP oracle under PoS I propose an oracle extension that acts as a guard to only allow price updates for a predefined max delta between price observations. The guard needs to be gas efficient.

Oracle extension guard design

The guard extension defines

- OBSERVATIONS

number of observations

- SKIP

step between two observations

- MAX_OUTLIERS

max number of observations out of range

- MAX_SINGLE_TICK_DELTA

max delta between observations

- MAX_TOTAL_TICK_DELTA

max delta in array with most observations

The guard checks a set number of OBSERVATIONS

within the range OBSERVATIONS * SKIP

on every i * SKIP

slot. For every observation the delta with previous observations is calculated. When delta > MAX_SINGLE_TICK_DELTA

it stores the observation in a new array. For the array with most observations the guard checks array length > OBSERVATIONS - MAX_OUTLIERS

and total_array_delta < MAX_TOTAL_TICK_DELTA

. The guard halts the price update if one of these is not met.

Check the GitHub repository for the POC implementation.

Safety and liveness

In this design the guard halts price updates when outside of safety assumptions, favoring safety over liveness. To manipulate the price more than the predefined delta a manipulator needs manipulated_observations > OBSERVATIONS - MAX_OUTLIERS

and to halt price updates the manipulator needs manipulated_observations > MAX_OUTLIERS

.

Observations

The manipulated observations need to be within the observed range OBSERVATIONS * SKIP

. The observed range used a SKIP

variable to increase the range without needing to check every observation. A large observed range can be used so validators do not know all the blocks they will propose. More OBSERVATIONS

increases safety and more MAX_OUTLIERS

increases liveness at the cost of safety (OBSERVATIONS

).

Example

The example uses the following boundaries. For simplicity low boundaries are used.

OBSERVATIONS = 8 MAX_OUTLIERS = 4 SKIP = 2

[

Example_guard_safety_attack

1420×269 10.7 KB

](https://ethresear.ch/uploads/default/original/2X/6/6d1beecee2364ce632a1927e5a52aec36fb36e7a.png)

In this example the guard makes 8 observations in a range of 16. The manipulator needs 4 observations to adjust the price outside of the predefined max delta. For these observations a minimal of 1.5 * OBSERVATIONS - MAX_OUTLIERS = 1.5 * 4 = 6

blocks need to be controlled. Notice that an attacker can manipulate the guard observation in block 3 by manipulating the observation in block 2 or block 3 (in the example block 2). This is because prices in the TWAP oracle are stored as a cumulative of all previous price observations and the guard calculates the price based on the difference between two cumulative observations.

Gascost

The guard is an extension that checks for outliers. This check has to be efficient with gas for it to be used in DeFi. The current POC estimates the example above at ~100k gas. More observations result in more gas usage, for OBSERVATIONS=28

, MAX_OUTLIERS=4

and SKIP=4

estimated at ~200k gas. Check theGitHub repository to run gas estimates with adjusted settings.

Conclusion

The oracle guard uses the current TWAP oracle observations allowing it to be an extension without the need for lower level adjustments. It favors safety over liveness by only updating the price if there are a sufficient number of observed prices within a preset delta. These boundaries can be custom set by a protocol for specific needs between safety, liveness and gas usage.