Thanks to @djrtwo for collaborating on this idea.

Problem

In order to prove a fraudulent transition of an optimistic rollup, the challenger typically needs to first prove that the input to the fraud proof is the same input the sequencer originally attested to. This avoids challengers fabricating the input to the state transition.

To prove the input, the transactions in the sequencer's calldata must be authenticated. This can be done by creating a commitment to the input within the contract during a block submissions. This would give the rollup a trusted root to authenticate against in future fraud proofs. The downside is that this incurs costs, such as memory expansion for loading the calldata, calculating the root in EVM, and finally, storing the root in storage.

This is unfortunate because the calldata is already

merkleized in the the transaction trie as part of normal consensus operations. It would be preferable to utilize the transaction trie root instead. This is accessible via the BLOCKHASH

opcode, however it can only provide hashes of the 256 most recent blocks. Proofs beyond those blocks would need to be recursively proved against the oldest available block hash. This is problematic because the proof size would increase linear overtime, and would eventually no longer be able to processed in a single block.

Solution

To avoid this, we propose that the rollup contract support the retroactive creation of "trusted checkpoints". Instead of recursively proving the block in question all in one go, the challenger(s) can iteratively build trusted checkpoints in the rollup contract until root of the block in question is trusted by the contract. At that point, the fraud proof may be submitted as normal.

With a long challenge period, it is infeasible for an adversary to censor the root chain long enough to stop challengers from recursively proving the root of the invalid block and submitting the fraud proof to roll the chain back. In most cases, the fraud proof will be able to be submitted immediately and immediately use the output of BLOCKHASH

.

Yes, this will cause fraud proofs to be much more costly in the worst case. However, it greatly optimizes for the optimistic case.

Feasibility

To show the feasibility of this approach, we'll sketch out a rough estimate for the number of headers that can be authenticated in a block.

The formula will generally follow the form of intrinsic_cost + calldata_cost + mem_expansion_cost + hashing_cost + overhead_cost

.

A header is approximately 500 bytes. At 16

gas per non-zero byte, providing a single header will cost 500 * 16 = 8000

gas. The cost to hash the header is 126

gas1

. The memory will need to be large enough to fit the parent header's hash, the current header being hashed, and it's hash. That works out to 32 * 2 + 500 = 564

bytes. To expand the memory to this size is 54

gas2

. Each header will need to be loaded from calldata into memory. This costs 51

gas3

.

Filling in the formula from above, we should have something like:

21000 + X * (8000 + 126) + 51

where X

is the number of headers. Solving for a gas limit of 12.5mm

, we find X

to be 1535 blocks. To account for the many jumps, comparisons, and shuffling of data, we should safely assume that we can authenticate at least

1400

headers per block.

Example

Suppose there exists an ORU with a 1 week finalization delay. At block N, an invalid state transition is committed to. The sequencer who submitted the invalid transition is able to censor fraud proof attempts for 23,000 blocks (~3.5 days). Roughly 1400 blocks can be authenticated at a time on L1, so at block N + 23000 + ceil(23000 / 1400) = N + 23017

, the header for block N will be authenticated and the fraud proof can proceed.

_

1: https://github.com/wolflo/evm-opcodes/blob/aec21e512e0f303513c3cd9d306f8661afd2fc65/gas.md#a4-sha3

2: https://github.com/wolflo/evm-opcodes/blob/aec21e512e0f303513c3cd9d306f8661afd2fc65/gas.md#a2-memory-expansion

3: https://github.com/wolflo/evm-opcodes/blob/aec21e512e0f303513c3cd9d306f8661afd2fc65/gas.md#a5-copy-operations