COMS 4995-001 (Science of Blockchains): Homework #7

Due by 11:59 PM on Wednesday, April 9th, 2025

Instructions:

(1) Solutions are to be completed and submitted in pairs.

(2) We are using Gradescope for homework submissions. See the course home page for instructions, the late day policy, and the School of Engineering honor code.

(3) Please type your solutions if possible and we encourage you to use the LaTeX template provided on the Courseworks page.

(4) Write convincingly but not excessively. (We reserve the right to deduct points for egregiously bad or excessive writing.)

(5) Except where otherwise noted, you may refer to your lecture notes and the specific supplementary readings listed on the course Web page *only*.

(6) You are not permitted to look up solutions to these problems on the Web. You should cite any outside sources that you used. All words should be your own. Submissions that violate these guidelines will (at best) be given zero credit, and may be treated as honor code violations.

(7) You can discuss the problems verbally at a high level with other pairs. And of course, you are encouraged to contact the course staff (via the discussion forum or office hours) for additional help.

(8) If you discuss solution approaches with anyone outside of your pair, you must list their names on the front page of your write-up.

Problem 1

(10 points) Recall that in many blockchains, e.g. Bitcoin and Ethereum, a valid transaction must include a set of valid signatures of all parties concerned with the transaction. E.g., a money transfer from Alice to Bob must include a valid signature by Alice. In the context of rollups, we saw in the lecture that along with an alleged state root r, the sequencer provides Data Availability (DA) by also posting the set of transactions it executed in the current batch. Does the sequencer also need to provide the corresponding set of signatures for each of these transactions if:

1. We are in the optimistic rollup case? Explain your answer.
2. We are in the validity rollup case? Explain your answer.

Problem 2

(16 points) Recall from lecture that in the "classic" rollup architecture, the sequencer periodically posts batches of rollup transactions along with a new state commitment/root (reflecting any consequences of executing those transactions). Similarly to the last problem set, we continue to explore deviations from this architecture and the consequences for security against various types of faults, this time in the context of validity rollups. (Below, whenever a sequencer posts a SNARK alleging the existence of a witness, you can assume that the underlying L1 verifies and reports to everyone whether the SNARK is valid.) In all cases, explain your answers.

1. Suppose that the sequencer posts the state root r, along with a SNARK proof π for the statement "I know a set of *valid* transactions that yield a post-execution state with state root r". Beyond that, it provides no additional information about the transactions or the state. Consider a sequencer that behaves honestly for some time and then crashes. Can another sequencer pick up where they left off and continue to post correct state roots to the L1 rollup contract in the future?

2. Suppose that the sequencer posts the state root r, along with a SNARK proof π for the statement "I know a set of *valid* transactions that yield a post-execution state with state root r". Beyond that, it provides no additional information about the transactions or the state. If the sequencer is byzantine and posts a fraudulent state root, can a challenger detect and prove to others that such a fault took place?

3. Suppose that along with the state root r, the sequencer publishes "state diffs," meaning the alleged changes to the rollup state caused by executing the current batch of transactions. (E.g., notifications of the form "the 7th word of persistent memory associated with address a now has the value x.") Suppose further that the sequencer, along with the state root and the state diffs, posts a SNARK proof π for the statement "I know a set of valid transactions whose execution causes the posted state diffs and has state root r". Consider a sequencer that behaves honestly for some time and then crashes. Can another sequencer pick up where they left off and continue to post correct state roots to the L1 rollup contract in the future?

4. Suppose that along with the state root r, the sequencer publishes "state diffs," as above, along with a SNARK proof π, as above. If the sequencer is byzantine and posts a fraudulent state root, can a challenger detect and prove to others that such a fault took place?

Problem 3

(10 points) In this question we consider a hybrid approach between the optimistic and validity approaches to rollups. Recall the bisection game in the context of optimistic rollups (Lecture 16). Recall that at the final stage of the bisection game, the L1 takes as input an EVM state μ_i, re-executes a step of the execution trace (i.e., a single line of EVM bytecode generated by the submitted batch of transactions), and checks whether the resulting EVM state is equal to a purported μ_{i+1} state. Suppose we wanted to further optimize the bisection game, as follows (for simplicity, we'll consider the optimization only in the event that the sequencer's state commitment is incorrect and the challenger's state commitment is correct):

1. During each step of the bisection game, we would like to avoid having the challenger send Merkle proofs for each of the claimed intermediate states (e.g., in the first iteration, for $\mu_{N/2}$).

2. At the final stage of the bisection game, we would like the L1 to perform a SNARK verification rather than the re-execution of one line of EVM bytecode.

The price we are willing to pay to implement the above optimizations is a single SNARK proof sent by the challenger during the final stage of the bisection game. Describe the precise and exact statement that the challenger must send a SNARK proof for during the final stage of the bisection game to implement the above optimizations, while maintaining the correctness (i.e., soundness and completeness) of the bisection game.