COMS 4995-001 (Science of Blockchains): Homework #6

Due by 11:59 PM on Wednesday, April 2nd, 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) In the context of optimistic rollups, we've seen in lecture that consistency and liveness of the L1 doesn't fully rule out the possibility of a byzantine sequencer finalizing a false state root on the L1. The reason for this was bribery attacks, in which the byzantine sequencer pays the L1 validators a sufficiently high amount to exclude the transactions necessary to challenge the sequencer's purported state root and complete the ensuing dispute resolution protocol. The goal of this question is to obtain a better understanding of the budget a challenger needs in order to successfully complete the dispute resolution protocol without being censored by the byzantine sequencer. We use the same abstraction of the problem that was mentioned in lecture: over the course of T rounds (the game duration), the challenger must be able to successfully include a transaction on chain at least N times, where N is the number of rounds in the bisection game. For the purposes of this question, let us simplify transaction inclusion in the L1, and assume that in every round $i \in [T]$, each block consists of *exactly one* transaction. In each round, Alice and Charlie bid for the right to include a transaction in the next block in the following way. Denote by A Alice's total budget, and by C Charlie's total budget.

1. Charlie bids b, where $b \leq C$.

2. Alice observes b, after which she bids b', where $b' \leq A$.

3. If $b' \geq b$, Alice wins the round. Otherwise, Charlie wins the round. (We imagine that the validator in charge of transaction inclusion for this round always goes with the higher bid, and collects the winning bid as payment for inclusion.)

4. The winner's budget is deducted by an amount equal to their bid.

We say that Charlie *wins* the game if he won at least N of the rounds.

1. Assume that $\frac{T-N+1}{N} \cdot C > A$. Consider the following strategy by Charlie: Each round, bid $b = \frac{C}{N}$. Prove that this is a winning strategy for Charlie. That is, no matter how Alice bids, Charlie is guaranteed to win the game.
2. Assume that $\frac{T-N+1}{N} \cdot C \leq A$. Prove that Alice has a winning strategy (and explicitly give such a strategy). That is, when Alice follows this strategy, she is guaranteed to win the game, no matter what Charlie does.

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). In this question, we explore deviations from this architecture and the consequences for security against various types of faults. In all cases, explain your answers.

1. Suppose that the sequencer only posts the state root, and 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 only posts the state root, and 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, 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.") 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, the sequencer publishes "state diffs," 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?