COMS 6998-006 (Foundations of Blockchains): Homework #4

Due by 11:59 PM on Tuesday, October 26, 2021

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.

(9) Some of these problems are difficult, so your group may not solve them all to completion. In this case, you can write up what you've got (subject to (4), above): partial proofs, lemmas, high-level ideas, counterexamples, and so on.

Problem 1

(15 points) Suppose we turned off Bitcoin's difficulty adjustment algorithm (i.e., always use the same difficulty level). Does the selfish mining attack from Lecture 10 (in the super-synchronous model, with worst-case (i.e., arbitrary) tie-breaking by honest nodes) still benefit the attacker? Either way, support your answer with a proof.

Problem 2

(20 points) For this question, cite the source(s) you used in your answers (just the URLs are fine).

(a) (10 points) Describe the types of cryptopuzzles used in Ethereum (Ethash). Your description should include an explanation of the ideas behind "Dagger" and "Hashimoto," and how these ideas are combined in Ethash.

(b) (5 points) Explain why ASICs are less helpful to miners in Ethereum than in Bitcoin.

(c) (5 points) Discuss the pros and cons of using ASIC-resistant hash functions.

Problem 3

(20 points) Recall that the difficulty adjustment algorithm in Bitcoin uses the timestamps that miners place in blocks to measure the amount of real-world time used to add a batch of 2016 blocks to the longest chain, and then adjust accordingly the proof-of-work difficulty for the next 2016 blocks.

(a) (7 points) Suppose miners were free to put whatever timestamps they wanted into blocks. How could miners use this power to manipulate Bitcoin's difficulty adjustment algorithm and boost their rewards? Give a concrete example of an attack.

(b) (5 points) Explain the rules that govern block timestamps in the Bitcoin protocol. (Cite the source(s) you used to answer this question; just the URLs are fine.)

(c) (8 points) Discuss to what extent the rules in (b) mitigate the attack(s) that you described in part (a).

Problem 4

(20 points) For this problem, assume that all block timestamps are accurate. Different blockchains have different difficulty adjustment algorithms.

(a) (5 points) Explain how the difficulty adjustment algorithm currently works in Bitcoin Cash, and explicitly compare and contrast it with the difficulty adjustment algorithm in Bitcoin. (Cite the source(s) you used to answer this question; just the URLs are fine.)

(b) (8 points) Would the selfish mining attacks in Lecture 10 and in Problem 5 below be more or less effective in Bitcoin Cash than in Bitcoin? Or are there arguments in both directions? In any case, justify your answer with examples and/or mathematical analysis.

(c) (7 points) Discuss any additional pros and cons that you can think of between the Bitcoin Cash difficulty adjustment algorithm and that of Bitcoin.

Problem 5

(45 points) This problem studies selfish mining (in the super-synchronous model) in the case of best-case tie-breaking. Specifically, whenever there is a tie for the longest publicly known chain and there is such a chain that ends in an honest block, an honest leader will extend that block.[1] Adversarial nodes can continue to extend whatever blocks they want, and also have the option of delaying the announcement of any blocks they produce.

We consider the following strategy for Alice, the attacker.

Case 1: Suppose there is currently a unique longest chain (among all blocks, including any unannounced ones), and it ends with an honest block. Then:

- Alice mines (i.e., tries to extend through a proof-of-work solution) on the end of the longest chain;
- if Alice finds the next block b, she keeps it a secret and privately starts trying to find a another block b' that extends b;
- if the next block b is found by an honest miner, Alice switches to trying to extend b, the new end of the (unique) longest chain.

Case 2: Suppose that Alice has produced (and kept secret) exactly one block at a height larger than those of all the publicly known blocks. That is, Alice has at her disposal a "secret longest chain," which she can announce at any time. Then:

[1] Recall from Lecture 8 that, in the super-synchronous model, there is at most one honest block at each height. So there can't be two longest chains that both end with honest blocks.

- Alice mines on the end of the secret chain;
- if Alice finds the next block b (extending the secret chain), she keeps it a secret and privately starts trying to find a another block b' that extends b;
- if the next block b is found by an honest miner (extending one of the known longest chains), Alice continues to mine on the end of the secret chain (which is now tied for the longest overall).

Case 3: Suppose that Alice's secret chain is tied for longest with a publicly known chain. Then:

- Alice mines on the end of the secret chain;
- if Alice finds the next block b, she immediately publishes all of her secret blocks (which now constitute the unique longest chain);
- if the next block b is found by an honest miner (extending one of the known longest chains), Alice gives up and returns to Case 1.

Case 4: Suppose that Alice's secret chain is at least two blocks longer than the longest publicly known chain. Then:

- Alice mines on the end of the secret chain;
- if the next block is found by an honest miner (extending one of the known longest chains) *and* Alice's private chain is then only one block longer than the new longest public known chain, she immediately publishes all of her secret blocks (which now constitute the unique longest chain);
- otherwise, Alice continues to mine on the end of her secret chain.

Whether or not this strategy is effective (i.e., boosts Alice's share of blocks on the longest chain higher than it would be otherwise with honest mining) depends on the fraction $\alpha \in (0, \frac{1}{2})$ of the overall hashrate that Alice controls. (The other $1 - \alpha$ fraction of the hashrate is assumed to be controlled by honest nodes.)

To analyze this attack, we'll use a (discrete-time) Markov chain.[2] The first ingredient is a collection of states, labeled by the nonnegative integers $\{0, 1, 2, 3, \ldots, \}$ and also an additional state f (for "fork"). State f corresponds to case 3 above, meaning that Alice's secret private chain is currently tied in length with the longest publicly known chain. State 0 corresponds to case 1, state 1 to case 2, and states $\{2, 3, 4, \ldots\}$ to case 4, according to how much longer Alice's secret chain is than the longest publicly known chain.

The second ingredient is a collection of transition probabilities; for states i and j, the corresponding transition probability p_{ij} indicates the probability of a state transition $i \mapsto j$ (given that the process is currently at state i).

(a) (10 points) Argue that the appropriate transition probabilities for modeling the selfish mining attack above are:

- From state 0, go back to state 0 with probability $1 - \alpha$ and otherwise go to state 1.
- From state 1, go to state f with probability $1 - \alpha$ and to state 2 otherwise.
- From state 2, go to state 0 with probability $1 - \alpha$ and to state 3 otherwise.
- From state $i \geq 3$, go to state $i - 1$ with probability $1 - \alpha$ and state $i + 1$ otherwise.
- From state f, go to state 0 with probability 1.

(b) (15 points) For a state i, define π_i as the limit (as $T \to \infty$) of

$$\frac{\text{expected number of visits to state } i \text{ over } T \text{ Markov chain steps}}{T}.$$

You can assume that the π_i's are uniquely defined, independent of the starting state, and constitute a probability distribution (with $\pi_f + \sum_{i \geq 0}^{\infty} \pi_i = 1$).[3] Similarly, you can also assume that the Markov chain visits state 0 infinitely often with probability 1.

[2] For a review, Wikipedia works fine; perhaps also look up the closely related idea of random walks in (directed) graphs.

[3] This follows from the basic of Markov chain theory and the assumption that $\alpha < \frac{1}{2}$.

Assume that

$$\pi_i \cdot \alpha = \pi_{i+1} \cdot (1 - \alpha)$$

for all $i \geq 1$. (Intuitively, for $i \geq 2$, this is because for each of the infinitely many transitions from i to $i+1$, there is a matching transition later on from $i+1$ to i. For $i = 1$, the correspondence is between transitions $1 \mapsto 2$ and transitions $2 \mapsto 0$.) Similarly, assume that $\pi_1 = \alpha\pi_0$ (every visit to 0 has an α chance of turning into a visit to 1, and there's no other way to reach state 1) and $\pi_f = (1 - \alpha)\pi_1$.

Using these assumptions above, derive closed-form formulas for all the π_i's.

[Hint: You should get $\pi_0 = \frac{1-2\alpha}{2\alpha^3-4\alpha^2+1}$.]

(c) (6 points) Prove that the expected rate at which honest blocks are added to the eventual longest chain is

$$(1 - \alpha) \cdot \pi_0 + 2 \cdot (1 - \alpha) \cdot \pi_f. \tag{1}$$

(d) (7 points) Prove that the expected rate at which Alice's blocks are added to the eventual longest chain is

$$2 \cdot \alpha \cdot \pi_f + 2 \cdot \alpha \cdot \pi_1 + \sum_{i=2}^{\infty} \pi_i \cdot \alpha. \tag{2}$$

(e) (7 points) Complete the proof of the Eyal-Sirer theorem by deriving[4]

$$\frac{(2)}{(1) + (2)} = \alpha \cdot \frac{4\alpha^3 - 9\alpha^2 + 4\alpha}{\alpha^3 - 2\alpha^2 - \alpha + 1}.$$

[4] A quick plot shows that the second factor is greater than 1 (indicating a profitable attack) when $\alpha > 1/3$.