EE374 Scaling Blockchains **Stanford, Spring 2020**

Homework #2*

Due: Thurs, 30-April-2020, 11:59pm – Gradescope entry code: `92J6Y3`

Please upload your answers timely to Gradescope. Start a new page for every problem. We strongly suggest LaTeX to type your answers. For the programming/simulation questions you can use any reasonable programming language (please no assembly, brainfuck, etc. ☺). Comment your source code and include the code and a brief overall explanation with your answers. A tentative point distribution (in % of the total) is provided in brackets. For most problems there is more than one valid way of solving them!

1. (30%) Nakamoto proposed the k-deep confirmation rule, which we analyzed in the lectures. Here we study a different confirmation rule and compare its performance with the k-deep rule under the private attack. In this confirmation rule, which we will call a t-wait rule, we confirm a block b t seconds after the block has been mined. t is a parameter the user can choose. The adversary starts mining a private chain from the parent of b immediately after b is mined. Let λ be the total mining rate, of which β fraction belongs to the adversary.

   a) Give an exact expression for the probability p_t that the adversary has an equal or longer chain than the honest chain at confirmation. Your expression can involve an infinite summation. For the rest of the question, we will take p_t as a proxy for the probability of confirmation error of the t-wait rule.[1]

   b) Using Chernoff bound or otherwise, give an upper bound to p_t to show that it decreases exponentially with t. Explicitly give the exponent in your bound.

   c) The confirmation latency of this rule is obviously t seconds, a deterministic quantity. The latency of the k-deep confirmation rule is however random. What is the expected value of the latency? By matching this expected value to t, compare the performance of the two confirmation rules, in terms of the rate of exponential decrease in confirmation error probability. You can compare analytically (preferred) or numerically. Can you draw any definitive conclusion which rule is better? Can you give an intuitive explanation for your conclusion?

2. (30%) There is a flaw in the proof of the worst attack on the longest chain protocol in Lecture 4! Can you help fix it?

*Version: 1 – Last update: 23-April-2020

[1] p_t is not exactly the confirmation error probability because the adversary can catch up with the honest chain strictly after time t, but for large t, these events will add negligibly small probability.

a) The proof starts by assuming that the attack π succeeds so that there are two chains of length at least k at the time T when the attack succeeds. However, it further makes the (unjustified) assumption that all the mined blocks up to that time are contained in these two chains. Can you figure out where that assumption is used in the proof?

b) Fix the flaw in the proof by removing this assumption.

(**Hint:** Define A and H differently.)

3. (40%) Assume the Ethereum chain where difficulty is adjusted such that on average a new block is created every 15 seconds ($\lambda = \frac{1}{15}$, unit $\frac{1}{\text{s}}$). Suppose it takes Δ seconds to communicate a newly found block of size 20 KBytes to the remaining miners, during which the remaining miners continue to try to mine a new block off of the previous (now old) block.

a) By either analysis or computer simulation (choose one), examine the fraction of blockchain heights which have more than one block (i.e., a fork has occurred) as a function of Δ. Assume that honest miners follow the longest-chain rule and break ties by mining on top of the block with the oldest timestamp. You can also assume that there is an infinite number of honest miners each with infinitesimal mining power.

b) What is the effect of forking on the blockchain as a consensus system? What happens if additionally a fraction of the mining power tries to attack the system?

c) A common ad-hoc proposal to scale the transaction throughput of a blockchain system is to either increase blocksize or to increase block creation rate. Explain how these two proposals are captured by your analysis in (a) and use the intuition gained in (b) to argue why neither represents a feasible scaling proposal.

d) Now suppose that there is a finite number of honest miners n and each miner has $1/n$-th of the hashing power. As n decreases, there is an increased amount of *centralization* in the network. Discuss what is the effect of increased centralization on your answer to part (c). Discuss whether this effect is consistent or inconsistent with the Blockchain Trilemma.