CS251: Cryptocurrencies and Blockchain Technologies **Fall 2019**

Assignment #2

Due: 11:59pm on Mon., **Oct. 21, 2019**
Submit via Gradescope (each answer on a separate page) code: **MG7EP3**

Problem 1. Suppose two groups independently implement the Bitcoin protocol. Some miners run implementation A and other miners run implementation B. At some point an attacker finds a vulnerability in implementation A that causes miners running that implementation to accept transactions that double spend a UTXO. Implementation B treats such transactions as invalid.

a. Suppose 80% of the mining power runs the buggy implementation and 20% runs the non-buggy one. What will happen to the blockchain once a block containing a double-spending transaction is submitted to the network?

b. What will happen to the blockchain in the reverse situation where 20% of the mining power runs the buggy implementation and 80% runs the non-buggy one?

Problem 2. In this exercise we look at two estimates for the amount of energy consumed by the Bitcoin network. Assume in your answer that the current exchange rate is 1BTC = US$6000 and that there are no transaction fees (only the block reward of 12.5BTC per block). Recall that energy is measured in killoWatt-hours (kWH). You may assume that one bitcoin block is generated every 10 minutes exactly.

a. Estimate the network's hourly energy consumption assuming the entire block reward is spent on electricity for mining. Use US$0.05/kWH as the price of energy and express your answer in kWH.

b. Next, estimate the network's hourly energy consumption assuming all mining is done on an Antminer S9 Hydro that has a hash rate of 18 terra-hash/sec and consumes 1.7 kW of power (running the device for an hour consumes 1.7 kWH of energy). Assume the current difficulty of generating a bitcoin block is $D = 2^{75}$.

c. Explain why there is such a large gap between the two estimates.

Problem 3. Recall that a selfish miner temporarily refrains from publishing mined blocks in an effort to get several blocks ahead of other miners, thereby causing other miners to waste effort mining orphan blocks. When a selfish miner is only one block S ahead of the public chain, if another miner mines and publishes a block O at the same height as S, the selfish miner immediately publishes S. Let γ be the probability that, when this happens, an honest miner will try to mine the next block on S instead of on O. What is a backwards-compatible change in honest miners' behavior that would result in $\gamma \approx 0.5$?

Problem 4. Mining pool sabotage. Recall that in section we discussed now mining pools enable individual miners to lower the variance of their earnings, while keeping the same expected returns. Participants repeatedly submit shares (blocks that are valid at a lower difficulty) to prove how

much work they are doing. Whenever the pool finds a block, the coinbase from that block is split among the participants in proportion to the number of shares each submitted. One risk of this is sabotage, in which a participant submits shares, but withholds full solutions if they are found (no coinbase is awarded for these withheld solutions).

a. Consider a participant with mining power $\beta \in [0, 1]$ (as a fraction of global mining power) in a pool with total mining power $\alpha \in [0, 1]$ (as a fraction of global mining power), where $\beta < \alpha$. What is the expected fraction of the overall mining rewards (the rewards collectively earned by the entire network) that this individual will earn if he or she mine honestly? Assume rewards are distributed proportionally to the number of shares submitted by each participant. **Hint:** there is no need for complicated expectation calculations throughout this entire question.

b. What is the expected fraction of the overall mining rewards (the rewards collectively earned by the entire network) that this individual will earn if it devotes all of its power to sabotage? **Hint:** Because β power is no longer used to find blocks, the total network mining power is now only $(1 - \beta)$ times its full power. Therefore, P's useful mining power, as a fraction of the entire network, is now $(\alpha - \beta)/(1 - \beta)$.

c. Now consider two pools, P_1 and P_2 with mining power α_1 and α_2, respectively. What will P_2's expected share of the total earnings be if it dedicates $\beta < \alpha_2$ power towards sabotaging P_1? Note that when P_2 finds a block, it gets the entire coinbase. When P_1 finds a block, P_2 receives a fraction of the coinbase proportional to the number of shares P_2 generated while mining for P_1.
Hint: P_1's total mining power is now $\alpha_1 + \beta$, but only α_1 is used for finding a new block.

d. Provide concrete values for $\alpha_1, \alpha_2, \beta \in [0, 1]$ in which this attack is profitable for P_2 over honest behavior.

Problem 5. In lectures 5 and 6 we looked at Nakamoto consensus, as originally designed, in the Proof-of-Work permission model. This homework problem explores how the permission model can be replaced with Proof-of-Stake, while keeping the underlying consensus protocol the same.

Consider the following adaptation of Nakamoto's protocol to a permissioned setting with fixed set of N players identified by public keys $\{PK_1, ..., PK_N\}$. Assume the players have synchronized clocks. Players start their clocks at time t_0 and measure timesteps at fixed intervals denoted by timestamps $\{t_0, t_1, t_2, ...\}$, i.e. the distance between t_i and t_{i+1} is a fixed parameter δ. Let $\mathcal{H} : \mathcal{K} \times \mathcal{T} \to [0, 2^\lambda)$ be a random oracle (i.e. a collision-resistant hash function modeled as a random function) where $\mathcal{K}$ is the domain of public keys, $\mathcal{T}$ is the domain of timestamps, and λ is a security parameter (e.g., $\lambda = 256$). At timestep t, a player with key PK_i is *eligible for* t if and only if $\mathcal{H}(PK_i, t) < 2^\lambda/N$. (There may be multiple elgible players at any given timestep). Eligible players at timestep t broadcast a "block", which is a tuple $(h_{-1}, \mathsf{txs}, t, PK)$ where txs is a list of new transactions and h_{-1} is a hash of a previous block with an earlier timestamp. Similarly to Nakamoto's protocol, honest players set h_{-1} to be the hash of the last block in the longest contiguous chain of blocks they have seen so far (i.e., at the start of timestep t). Honest players never extend chains with out of order timestamps, or timestamps "in the future" (i.e., a block with $t' > t$ would be rejected at timestep t). Between blocks at the end of two chains of equal length, honest players choose to extend the one with a more recent timestamp, or otherwise with the lower eligibility hash $\mathcal{H}(PK, t)$ if the timestamps are equal.

a. Argue that if all players are honest and the distance between timesteps is longer than the maximum message delay Δ in the network, then at the beginning of any timestep t all players have a perfectly consistent view of the longest chain, except with very small probability.

b. What would go wrong if the protocol were modified so that the entire block tuple $(h_{-1}, \mathsf{txs}, t, PK)$ was included in the input to the hash function $\mathcal{H}$ to determine player eligibility to submit this block at timestep t? (**Hint:** Describe an attack on consensus that prevents consistency or liveness and requires controlling only one key).

c. What would go wrong if honest nodes accepted chains with out of order timestamps or timestamps "in the future"? (**Hint:** Describe an attack on consensus that requires controlling only one key).

d. For this problem, we use the following fact that bounds the length of forks the adverary can build, assuming it controls $p < 1/2$ of players. If no honest player contributes to the adversary's chain spanning m timestamps, its length is at most the number of timestamps for which the adversary was eligible. This length can be approximated by a Poisson variable X_m with expected value pm. Similarly, an honest chain has length given by the Poisson variable Y_m with expected value $(1-p)m$. Setting $K = X_m - Y_m$ to be the difference, it is a fact that $P(K \geq k) < e^{-m}\frac{1-p}{1-2p}(\frac{p}{1-p})^k$.

Suppose that at time t the honest players have a consistent view of a chain C_t. Suppose C_t includes some block hash h at some depth k such that the longest fork in each honest player's view that omits h is at least d blocks behind C_t (*i.e.*, the length of any fork C_t^* omitting h is at least d blocks shorter than C_t). Note that $d \geq k$. Argue that if the adversary controls a $p < 1/2$ fraction of players, the probability that at time $t' > t$ the longest valid chain in some honest player's view omits h is small (i.e., decaying exponentially in k) .

e. Consider a more powerful adversary with the ability to selectively shut down honest players for the duration of a time period (e.g., by a DOS attack). The adversary chooses the players to shut down at time step t, and we assume the total number of players it can shut down or corrupt is less than $1/2$. How can the adversary take advantage of the current protocol to prevent liveness?

f. Sketch a solution that avoids the problem in (e) by using a VRF. (Refer to the lecture notes on committee election with a VRF for help).

g. Explain in a few words how you would adapt the protocol to the weighted PKI model (i.e. proof-of-stake model). Refer to the lecture notes on committee election with a VRF for help.

Problem 6. Bob posts the following wallet contract to Ethereum to manage his personal finances:

```
contract BobWallet {
    function pay(address dest, uint amount) {
       if (tx.origin == HardcodedBobAddress) dest.send(amount);
} }
```

Suppose Mallory can trick Bob into calling a method on a contract she controls. Explain how Mallory can transfer all of Bob's money to her own account.