Blockchain Foundations **NTUA, Spring 2026**

Theory Exercise 4

Due: 14:00, Friday, June 19, 2026

Problem 1

Recall the longest chain PoS protocol, Ouroboros Praos, we studied in class. Ouroboros Praos is *available* but not *accountable*. We modify Ouroboros Praos to include a slashing mechanism: if the same key is used to create two blocks for the same height, we slash the key. Is the modified protocol t-accountably safe? If yes, prove it and find t. If not, is there another slashing mechanism that could make the protocol t-accountably safe?

Problem 2

In this problem, we will consider the Simplex protocol run by n permissioned nodes, with a tunable quorum size q for notarization.

1. Suppose we want to maximize the resilience of the protocol, i.e., maximize the number of adversarial nodes f that can be tolerated such that the protocol is both safe and live. Derive the optimal quorum size q and show that the resulting optimal resilience is $f = n/3$.

2. Suppose you believe that an attack against safety is more likely than an attack against liveness (since a double-spend can provide significant rewards to the attacker). Hence, you want to tune Simplex to increase the resilience against safety attacks, even at the expense of decreasing the resilience against a liveness attack. Can this be done? If so, exhibit and plot the tradeoff between the two resiliences. If not, explain why not.

Problem 3

Consider the Simplex protocol we presented in class. Consider the following variations of the protocol and in each case explain if the security (safety and liveness) of the protocol is preserved. In each case, provide a proof that the protocol retains each virtue or a show a counterexample in which it fails.

1. We completely remove the timer of the protocol.

2. We change the timer from 3Δ to 2Δ.

3. We change the timer from 3Δ to 4Δ.

Problem 4

We modify the Simplex protocol to include a timestamp r in each non-dummy block. When an honest party creates a block, they put their clock's time as the timestamp r. In a valid chain, consecutive blocks have non-decreasing timestamps and no timestamp is in the future. We assume that honest parties have sychronized clocks and the network is synchronous. We are interested in how much the timestamp of the finalized tip of an honest party can deviate from their clock's time. Compute an upper bound v such that any honest party's tip will not deviate from their clock more than v except with negligible probability. Your bound does not need to be tight.