Homework7

Part 1 (DAG: GHOST & Conflux)

Read the following materials about DAG, GHOST, and Conflux:

1. GHOST protocol paper: https://eprint.iacr.org/2013/881.pdf
2. Conflux paper: https://arxiv.org/abs/1805.03870

Then answer the following questions.

Now suppose a blockchain using the Bitcoin protocol runs at a very fast block generation rate, so that only 40% of the generated blocks are on the main chain on average when the blockchain runs normally without attackers. Now there is an attacker controlling 33% of the total network hash rate. Is it possible for this attacker to deterministically launch double-spending attacks? Why?

For the previous question, if the chain uses GHOST protocol instead. All the rest conditions stay the same. Is it possible for the attacker to deterministically launch double-spending attacks? Why?

One claims that by using GHOST protocol, we can now simply lower the difficulty to speed up the block generation to achieve arbitrarily high transaction throughput, because fast block generation no longer harms the security. Is this claim true or not? Why?

In Conflux, is it possible for an attacker to create a malicious block choosing a very early block as its parent block to disrupt the transaction total order? Why?

How much hash power an attacker must control for him/her to deterministically launch double spending attacks in Conflux? Why?

Part 2 (Algorand)

Read the Algorand paper (https://people.csail.mit.edu/nickolai/papers/gilad-algorand-eprint.pdf). Answer the following questions:

Is it possible that the Algorand sortition algorithm selects a committee that contains a majority of malicious nodes? If so, is this going to be a problem?

The random seed of the sortition algorithm is critical for the security of the Algorand. Suppose an attacker now can control the starting seed of the Algorand sortition algorithm. Explain an attack strategy for the attacker to subvert the Algorand blockchain.

In Algorithm 8, why the algorithm only considers the consensus “final” when it reaches the consensus at the very first step (step == 1)?

In Algorithm 8, why the algorithm needs a CommonCoin mechanism?

In Algorithm 8, why the algorithm terminates after MAXSTEPS? What would happen if the algorithm keeps trying without a MAXSTEP limit? What’s the security consequence here?

Why Algorand uses a more complicated BFT algorithm that may not always reach final consensus? Why not Algorand runs the standard PBFT algorithm among selected committee members? (Hint: Read the last part of the introduction of the paper)