HW1

Junhong Chen

Section 4 argues that the longest chain rule represents the majority vote of the transaction history. This argument relies on the assumption that block propagation is fast enough, so that blocks generated by honest participants will form a chain. What would happen if this assumption is violated?

It would increase the likelihood of different nodes in the network receiving difference blocks at approximately the same time. This could result in a fork situation, which could amplify the risk of double spending and increase the uncertainty and confusion within the network. Slow block propagation can also cause delays in the system.

Can you think of any other assumption the above claim relies on? Write down these assumptions and discuss them.

More than 50% of computation power belongs to honest nodes.

If more than 50% of computation power belongs to a malicious entity, the risk of 51% attacks could increase. This is because the malicious entity can be able to create a longer chain and then release it to the network and the network will adopt chain created by malicious entities because it is the longest.

Block generation should be comparably slow.

If block generation is fast, the blockchain size will increase significantly, which might cause storage and bandwidth problems. In addition, it could increase the likelihood of the fork situation, as it increases the likelihood of multiple nodes solving the proof-of-work puzzle simultaneously.

What would happen if Bitcoin removes the Proof-of-Work puzzle from block generation? Imagine a hypothetical system in which all transactions form a chain directly instead of being packed into blocks first. Anyone who wants to submit a transaction can append to the end of the transaction chain without PoW. In this hypothetical system, the longest transaction chain is considered the correct transaction history. What could go wrong for the system?

Every node has the potential to forge fake transactions and spread those fake transactions over the network easily, which could cause Sybil attacks. This is because malicious nodes can create many fake identities easily to create a longer chain. Additionally, those nodes can also create a large number of transactions to flood the network, causing performance problems. Furthermore, the system might struggle to prevent the double spending problem because it’s hard to determine which transaction is valid without the PoW algorithms.

Section 5 mentions that each transaction and each block are broadcasted to everyone else in the network. How is this actually implemented in Bitcoin? You may want to search additional materials on the web to answer this question. Based on the discussion in Section 11, how many blocks does the recipient of a transaction need to wait if 1) he or she assumes that the attacker controls up to 20% of computation powers and he or she can only tolerate 0.1% of double-spending risk? 2) How many blocks to wait, if the attacker controls 30% of power and he or she can only tolerate 0.01% of double-spending risk?

The sender broadcast the transaction to the mining nodes through a peer-to-peer network. The miners check the validity of the transactions by checking the digital signature, transaction structure, and whether the inputs are unspent. Valid transactions are then put into the mempool, waiting to be collected into a block. Once a miner collects valid transactions from their mempool and creates a block, it then competes to solve the math puzzles and propagates the block through the peer-to-peer network.
(1) He or she should wait for 11 blocks. This is because when the number of blocks is 10, the probability of successful attack is 0.1067%. When the number of blocks is 11, the probability of successful attack is 0.0560%, which is less than 0.1%.
(2) He or she should wait for 32 blocks. This is because when the number of blocks is 31, the probability of successful attack is 0.0115%. When the number of blocks is 32, the probability of successful attack is 0.0087%, which is less than 0.01%.