CS251

Leon Kloker

October 2023

Exercise 1

a)

The miners running implementation A will accept the transaction and post it to the blockchain, whereas the miners running implementation B will reject the posted block. Therefore, the blockchain will fork into two chains, one faulty chain containing the double-spending transaction and one correct chain. The miners A will continue mining on top of the faulty chain and miners B will mine on top of the correct chain. As miners A constitute 80% of the mining power, however, the faulty chain will grow faster until the difference in length (or in proof-of-work) is big enough such that the correct chain will be discarded and the faulty chain is accepted by all miners unless some external action is taken.

b)

As the miners running implementation B realize the transaction is invalid, it won't be added to a new block as long as one of them is the leader. When a miner A is elected as leader, he might add the transaction to the block and when the block is posted the blockchain forks again into two chains. In this case, however, the correct chain will outgrow the faulty one until the faulty chain is discarded and all miners start mining on the correct chain again.

Exercise 2

a)

As the attacker has exactly halve of the mining power, the private chain will most likely not outgrow the public chain. Hence, when the private chain is publicized after one year, both chains will have very similar lengths. When the attacker continues to mine on his chain with 50% of the computing power, he can hope that for some reason other miners decide to continue mining on his chain which would give him the advantage. As all transaction are empty in this chain, honest miners should recognize this as an attack and continue mining on the true chain. In this case, the attacker chain will most likely not overtake the true chain and unless he gives up mining, both forked chains will coexist until one chain outgrows the other one at random due to the random mining process.

b)

When the attacker only controls 20% of the mining power, the privately mined chain will (on average) grow slower than the public chain. Hence, when the chain is released after one year, most likely it will only be about 0.25% of the size of the publicly mined chain and, thus, immediately discarded by all other miners.

Exercise 3

a)

In this case, there would be at least $2 \cdot \frac{3n}{4}$ votes overall. There should be only maximally n votes. Thus, $2 \cdot \frac{3n}{4} - n = \frac{n}{2}$ additional votes are cast, meaning there would need to be at least $\frac{n}{2}$ malicious validators.

b)

If the malicious validators stop voting completely, no block will receive enough votes to be finalized. Hence, the liveliness of the protocol can't be guaranteed anymore.

Exercise 4

a)

If the sender is honest, then all honest nodes will output the same bit meaning safety is guaranteed. If the sender is dishonest and sends different bits to different non-senders, they will exchange the different messages, realize that the sender is dishonest and all output 0. Finally, if the sender doesn't send anything the non-sender won't output anything. Thus, safety is guaranteed.

b)

Again, if the sender is honest, all non-senders receive the same bit, sign and exchange their messages. Even when a dishonest non-sender sends out a different bit to other non-senders, the message won't be signed by the sender, hence, the honest non-sender discard it and will all output the senders original bit. Thus, validity is guaranteed.

c)

Let the sender and one more non-sender be dishonest. Then, the sender sends a signed message with bit 1 to all honest senders and a signed message with 0 to the dishonest non-sender. Then, the dishonest non-sender sends this 0 message to only one of the honest non-senders. This node will output 0 as he received conflicting bits signed by the sender, all the other honest nodes will output 1 as there is no conflict and they received a 1 by the sender. Hence, there is no safety.

d)

All non-senders receive the same bit. Two dishonest nodes can now send message with changed bits to other non-senders but as these messages won't be signed by the sender, they will be ignored. Thus, all the honest non-sender will output the bit that was sent by the sender.

e)

If the sender doesn't send a message to the honest non-sender but he sends one to the dishonest non-sender, the dishonest non-sender can send this message to only one of the honest non-senders in the second step. Thus, this honest non-sender will output something as the message was signed by the sender but all other honest non-senders didn't receive a message and will stay quiet. Thus, totality isn't guaranteed.

Exercise 5

tx.origin doesn't refer to the immediate sender of the message but to the externally-owned address that started the chain reaction of transactions. Thus, Alice could build a contract that looks like a legitimate service and convince Bob to interact with this contract. Then, from within this contract, Alice can call Bobs pay function with an arbitrary amount and her own address as destination. As Bob was the initiator of the original transaction calling Alice's contract, tx.origin will be equal to Bob's address and the pay function will execute, sending the specified amount (potentially all of Bob's ETH) to Alice's address.