Selfish Mining: A 25% Attack Against the Bitcoin Network

One of Bitcoin's core security guarantees is that, for an attacker to be able to successfully interfere with the Bitcoin network and block and reverse transactions, they need to have more computing power than the rest of the Bitcoin network combined. The reason for this is that the Bitcoin network builds up its transaction history in the form of a "blockchain", with a random node adding a new block on top of the previous block every ten minutes. To reverse a transaction, an attacker would need to make a transaction and then "fork" the blockchain one block behind the block the transaction was included in – from which point the attacker would be in a computation race against all of the other miners combined as he attempts to catch up. A new paper by Cornell University researchers Ittay Eyal and Emin Gun Sirer, however, significantly reduces Bitcoin's security guarantee by introducing another type of attack – an economic attack. The economic attack does not allow hostile miners to mount successful attacks against Bitcoin unilaterally, but it does change the incentives such that normally honest, profit-maximizing nodes would want to join the attacker's coalition, potentially allowing for 51% attacks as a second stage.

The high-level overview of the attack is this: rather than acting as a normal miner and publishing blocks to the network immediately upon finding them, the attacker selectively publishes blocks, sometimes sacrificing his own revenue but also often publishing many blocks all at once and thus forcing the rest of the network to discard blocks and lose revenue. This does reduce the attacker's revenue in the short term, but it reduces everyone else's revenue even more, so neutral nodes now have the incentive to join the attacker's coalition to increase their own revenue. Eventually, the attacker's coalition would expand to above 50% in size, potentially giving the attacker a high degree of control over the network.

The attacker's precise strategy is as follows. The attacker keeps track of its own "private chain", which is separate from the "public chain" that the rest of the network works on. At first, the private chain and the public chain start out the same. The attacker always mines on the private chain and keeps any blocks that he finds private. The strategy dictates exactly when the attacker should publish blocks. Suppose the attacker's portion of the network hashpower is X, and when there are two competing public chains the portion of the network that picks up on the attacker's chain is Z.

- **State 0**: If the attacker's private chain is the same as the public chain, mine on the private chain. With probability X, the attacker discovers a block and advances to state 1 (private chain 1 block ahead). With probability 1-X, the public network discovers a block, and the attacker resets his private chain to the public chain.

- **State 1**: If the attacker’s private chain is 1 longer than the public chain, mine on the private chain. With probability X, the attacker advances to state 2 (private chain 2 blocks ahead). With probability 1-X, the public network discovers a block, setting the system to state 0’.
- **State 0’**: The attacker publishes his block. There are now two competing chains, both one block long. With probability X, the attacker will discover another block, causing the network to switch over to the private chain. The attacker gains a revenue of 2, and the system resets to state 0. With probability (1-X)*Z, the network finds a block on top of the attacker’s block. The attacker and the network gain a revenue of 1, and the system resets to state 0. With probability (1-X)*(1-Z), the network finds a block on top of its own block, the network gains a revenue of 2 and the system resets to state 0.
- **State 2**: With probability X, the attacker advances to state 3 and earns a revenue of 1 (technically, the attacker will earn the revenue later, but it’s easier to account for it here). With probability 1-X, the network finds a block, so the attacker publishes his 2-block private chain, which is still one block longer than the public chain, so the network will switch to the attacker’s chain. The attacker earns a revenue of 2.
- **State n** (n > 2): with probability X, the attacker advances to state n+1 and earns a revenue of 1. With probability 1-X, the attacker falls back to state n-1.

To see why this strategy works, suppose that Z is close to one. In this case, there is never any chance that the attacker has to discard a block; the only time that might happen is from state 0’, and if Z ~= 1 almost all of the network, attacker and other nodes included, is mining on the attacker’s block so the attacker’s block will not be discarded. Thus, the attacker is mining at full efficiency. However, the public network might see blocks discarded at state 0’ and state 2, so the public network is mining at partial efficiency. Thus, neutral (profit-maximizing) nodes have the incentive to join the attacker’s coalition to increase their revenue. As Z decreases, the attacker’s advantage goes down; at Z = 0.5, Eyal and Sirer showed that the attacker becomes more efficient than the public network at X > 1/4, and if X > 1/3 the attacker is more efficient than the public network at any Z.

So how do we calculate Z? Currently, the Bitcoin network is set to follow a simple rule: every node only mines and propagates the first block that it sees. Against attackers with only mining power, this is a successful defense; because the attacker’s strategy is reactive, publishing blocks only after the public network does, Z is close to zero. However, well-funded attackers (or attackers controlling botnets) can mount a “Sybil attack”, creating millions of nodes and inserting them into the network in as many places as possible. During an attack, the Sybil nodes would propagate only the attacker’s blocks. In this case, the Eyal and Sirer conjecture, Z can potentially get very close to one. In reality, that is not quite true; at the minimum, every mining pool will be the first to hear about its own blocks, so Z <= 0.8 is essentially guaranteed, but it is a matter of debate just how much a Sybil attack can do. To make Bitcoin secure against Sybil attacks, Eyal and Sirer argue, honest miners should switch to the strategy of propagating all blocks and if they receive multiple competing chains of the same length mining on a random one. If all miners implement this, we would have Z = 0.5, creating a reasonably threshold of X >= 1/4 for this attack to work.

Is this a fatal threat to Bitcoin? Not really. The idea behind the attack is not new; very similar attacks were theorized about on the Bitcoin forums as early as 2010, and lead Bitcoin developer Gavin Andresen himself participated in the discussion. However, at the time no action was taken – largely because everyone at the time considered the attack to be not worth worrying about compared to the other threats that Bitcoin has to face. In practice, most Bitcoin miners act altruistically to support the network, both out of ideological considerations and because they do not want to destabilize the source of their own revenue. Such higher-level economic concerns are beyond the scope of Eyal and Sirer's paper, but they seriously reduce the chance that this economic attack will work in practice.

Furthermore, unlike a standard 51% attack, which only becomes obvious after the fact, this economic attack would need to be announced in advance to let neutral miners know that they have the opportunity to join the attacking coalition for their own benefit. Thus, mining pools cannot practically pull this off; as soon as one announces its intention to cheat the network, its users will leave out of ideological considerations, and even if they do not other mining pools will likely offer heavy discounts on fees to that mining pool's users to convince even profit-maximizing participants to switch away. But nevertheless, Eyal and Sirer's result, as well as the work of others in the years before, are an important and under-appreciated part of the game-theoretic research around Bitcoin, showing us that Bitcoin's network security is slightly less infallible than we at first might think it is.