Casper attempts to solve the problem of “catestrophic crashes”, where validators with more than 1/3rd of the voting weight go offline and thereby stop any more blocks from being validated, by using an “inactivity leak” where funds from the inactive validator’s stake are drained until active validators regain 2/3rds of the validator set’s voting weight.

This does solve the problem of “catestrophic crashes”, but introduces what I think might be a major attack vector where an attacking validator could capture the validator set for themselves by pretending all other validators aren’t voting, draining their funds, and then capturing the validator set (and all future validator sets) for themselves. The attack would happen as follows:

1. An attacker with more than 5/12ths of the validator set’s voting power lets 4/5ths of their stake (ie 4/12ths of the total validator set’s voting weight) go inactive with respect to the honest validators

2. The attacker starts building a secret chain with proof they’re voting and no proof that the honest validators are voting. The honest validators do the same.

3. While the honest validators are building those proofs, the attacker continues to use 1/5th of their stake to vote, ensuring that 1/5th doesn’t get drained for inactivity (the attacker’s other 4/5ths would be drained tho) while also ensuring there aren’t enough votes for a block to be validated.

4. Once the attacker’s secret chain builds enough proof to bring their side to 2/3rds of the voting weight again, the attacker continues the epoch and releases the secret chain publicly. The attacker will likely achieve this first and will then become the longest validated chain and eventually choose a new validator set they control entirely.

Because it would take much longer for someone with just over 1/3rd the validator stake to drain enough funds from the other 2/3rds to reach 2/3rds of the voting weight, than it would for the honest side that started out at almost 2/3rds, the attacker’s chain wouldn’t succeed unless it had substantially more than 2/3rds of the voting weight. However, while you might think the attacker would need a full 50% of the voting weight, this isn’t the case because the attacker doesn’t have to go fully inactive - only 1/3rd of the total stake must be inactive and the attacker can keep any leftover stake active without risking a block being validated. The critical point where this allows the attacker to succeed is halfway betwen 1/3 and 1/2 (ie 5/12). So the attacker only needs 5/12ths of a validator set to successfully capture the system.

But an attacker doesn’t need 5/12ths of the active stake to achieive 5/12ths (ie 41.5%) of the validator set. In fact, if the attacker has 30% of the active stake, using 1-day epochs there is a 90% change they’ll achieve 5/12ths of the validator set within a year and a half (math: Math.log(1 -.9)/Math.log(1-probabilityOfQuorumCapture.valueOf())/365

where probabilityOfQuorumCapture = findAtLeastKInN(5*10, 12*10,.25)

using this function

). This number varies based on how many validators each validator set has and the length of an epoch. But this seems to imply, I think, that the cost of catastrophically attacking a system using Casper as it stands would be between about 25% and 35% of the active stake, which would be a significantly smaller fraction of the total coins - if only 10% of the coins are actively staked, this would be ~3% of the total coins. This is almost as low as the security that Bitcoin currently has.

My hypothesis is that it is impossible for a PoS system to achieve a minimum cost of attack higher than half of the active stake, and in fact I also guess that this is true for any distributed consensus mechanism with potentially malicious actors, that the maximum possible security is related to half of whatever weights go into deciding consensus (whether its PoS or PoW or something else). So I recommend that Casper reduce its threshold to an even 50% (instead of 2/3rds) in order to raise the minimum cost of attack to something like 40% or 45% of the active coins (up from 30% - it won’t make it to 50% because of the probability a validator with less than 50% of the stake can achieve 50% of a validator set).

Note also that the idea that “a formula [for the rate of inactivity leak] which increases the leak rate in teh event of a long streak of non-finalized blocks may be optimal” that is suggested in the Casper whitepaper would be helpful to an attacker, who could use their stake to vote just enough to prevent increased leak rates but not enough to cause 2/3rds consensus to be reached. That suggestion would make it easier for an attacker to execute this attack, so I don’t recommend a leak-rate function that increases over time in any way.

So what do you guys think? Is this attack doable or is there something about the protocol that would prevent this scenario from playing out?

Response to vbuterin

(since my account is on hold an no admin is paying attention): The attacker wouldn’t slash himself on his own chain that he fully controls. So I don’t think you’re correct that the attacker loses money on his own attacker-chain when the attacker participates in the honest chain. Also, the attacker won’t lose money on the honest chain while his chain is kept secret. The only way I can see the attacker being slashed in the honest chain is if, when the attacker releases his chain, the honest group decides to rewrite history and slash the attacker’s accounts after-the-fact. This seems likely to raise the cost of attack to near 50% of the active stake, but is that currently part of the protocol?