Reorg resilience and security in post-SSF LMD-GHOST

The consensus protocol of Ethereum is a hybrid protocol, combining an "available protocol", LMD-GHOST, with a finality gadget, Casper-FFG. The goal of this post is to introduce a few ideas about the security properties of LMD-GHOST with view-merge and without subsampling

, i.e. when we allow everyone to vote at once instead of cycling through multiple committees (the latter is what happens in the current protocol). Having everyone vote at once is a crucial component of single slot finality (SSF) research, and so we think it is worth exploring all the benefits it provides. We will see that everyone voting at once removes the fundamental issues

with LMD-GHOST, leaving a secure protocol with extremely desirable properties.

Introduction

LMD-GHOST has over the years revealed to have quite a few issues. Here we argue that the only truly fundamental issues are caused by subsampling. With subsampling, the possibility of ex ante reorgs is unavoidable, because it is possible to "stack up weight" from multiple committees, an attack akin to selish mining (see the background section in this post for more context and links to further material). Without subsampling, i.e. if everyone votes at once, those issues can be entirely avoided. In fact, we can have reorg resilience

and the provable security

which it directly implies, much as in the Goldfish protocol, another GHOST variant. Goldfish supports subsampling and makes weaker assumptions on participation, as we'll see in the next section, but it is extremely brittle to asynchrony, allowing for catastrophic failures such as arbitrarily long reorgs (see Section 6.3 in the paper). In this post, we will not focus on such issues and on why LMD-GHOST might be more resilient in that sense, and instead only focus on showing that the same security properties can be achieved. That said, keep in mind that this is the underlying reason why we even care about achieving those properties in LMD-GHOST, rather than just using Goldfish.

Assumptions on participation

We want to have the same security properties of Goldfish. For that, we make a somewhat stronger assumption on honest participation, not considering the dynamically available setting where fully variable participation is allowed. Goldfish can deal with this setting, only having to make the assumption that a majority of the online

validators is honest at any given time. On the other hand, we need that >50% of the validator set is honest and online

. We'll refer to this assumption simply as honest majority

, essentially considering offline validators to be faulty. Having the protocol be secure in settings of even lower honest participation requires a bit more work, for the reasons outlined in the paragraph "Stale votes in LMD-GHOST", in Section 6.1 of Goldfish, which we will not explore here.

Properties

Firstly, we want reorg resilience

, the property that honest proposals are always in the canonical chain if the network if synchronous.

In itself, this is a very strong and highly desirable property, because it guarantees that the adversary cannot waste honest proposals. Heaviest-chain protocols for example are very much not

reorg resilient, because of the possibility to withhold blocks (or more generally weight) and later reveal them to conclude a reorg. The same applies to GHOST. Similarly, as already mentioned, LMD-GHOST with subsampling is also not reorg resilient, because the same kind of withholding attack can be carried out to accumulate weight from multiple committees.

We also want the protocol to be secure

, i.e. have a safe and live confirmation rule. Luckily, this is trivial for a reorg resilient protocol! As in Goldfish, we can just consider a k

-deep confirmation rule, where k

is large enough to ensure that the every k

consecutive slots contain an honest one. Reorg resilience then ensures that the honest proposal of that slot stays in the canonical chain in all future slots, ensuring safety also of all blocks which come before it. Liveness and safety are easy

consequences. Therefore, we will only focus on reorg resilience. In the next section, we argue that LMD-GHOST with view-merge and without subsampling achieves this property.

Reorg resilience argument

Goldfish case study

The way Goldfish achieves reorg resilience even while allowing subsampling and fully variable participation

is through the use of ephemeral votes

, i.e. by only using votes from slot t

to run the fork-choice at slot t+1

, combined with view-merge, which allows proposers to synchronize honest views and always get honest attestations. The reason why this is sufficient is that ephemeral votes prevent the adversary from accumulating fork-choice weight, no matter how many slots they control. Once an honest slot (one in which the proposer is honest) comes, the block proposal in it receives all honest attestations, by view-merge, which are more than the adversarial attestations at this slot, because we assume honest majority of the online validators

. In the next slot, the only attestations which count are the ones from the honest slot, and so the honest proposal has a majority of the total fork-choice weight, and is thus canonical. All honest validators still attest to it, and by induction they keep doing that in future slots as well.

Summarizing, the argument is as follows:

- View-merge \implies

Honest proposals get all honest attestations in their slot

- Honest majority of online validators + ephemeral votes \implies

Honest attestations at a slot count for a majority of the eligible

fork-choice weight in the next slot \implies

a block getting all honest attestations (on its subtree) at a slot wins the fork-choice in the next slot, and by induction in all future slots

- View-merge + honest majority + ephemeral votes \implies

honest proposals are canonical in all future slots

Translating the argument to LMD-GHOST

Now, how does this argument relate to LMD-GHOST?

View-merge still gives the same guarantees to honest proposals, so we have to look at the second part of the argument. Is a block which receives all honest attestations in a slot guaranteed to win the fork-choice in the next slot? This is of course not true for LMD-GHOST with subsampling. Even if all validators in a slot vote for some block, there's no guarantee that their votes are not overpowered by votes from previous slots, from different validators, as in the ex ante reorg which was previously mentioned. The crux of the issue is that honest majority of the committee of a slot does not equal a majority of the eligible fork-choice weight

. In Goldfish on the other hand, this is true even with subsampling

, because all other weight is expired.

In LMD-GHOST, we actually achieve this property if we remove subsampling! This is due to the "L", the fact that we only consider latest

messages, which means that every validator only gets a single vote

(preventing the possibility of stacking up weight which exists in GHOST). If > \frac{1}{2}

of the entire

validator set is honest and votes at a slot, their votes then immediately constitute a majority of the eligible fork-choice weight. From this, we can immediately see why we cannot just assume that we have an honest majority of the online validators as in Goldfish: it does not correspond to a majority of the eligible fork-choice weight, because votes from offline validators do not

expire.

Summarizing as before:

- View-merge \implies

Honest proposals get all honest attestations in their slot

- Honest majority + LMD + no subsampling \implies

Honest attestations at a slot count for a majority of the total

fork-choice weight \implies

a block getting all honest attestations (on its subtree) at a slot wins the fork-choice in the next slot, and by induction in all future slots

- View-merge + Honest majority + LMD + no subsampling \implies

honest proposals are canonical in all future slots