Acknowledgement

We thank Mary Maller, Zhengfei Zhang, Luke Pearson and Ibrahim Yusufali for their insightful discussion.

TLDR

Utilizing zero-knowledge can enhance the BLS signature, leading to a significant reduction in communication size between validator nodes and full nodes. This enhancement paves the way for a more streamlined and expansive network, enabling every validator to sign each block. A low-hangling fruit in this area is the efficient light client protocol,“zkLightClient.” This protocol allows light clients to verify 512 BLS signatures with a single zk-proof verification, requiring a mere 200 to 300 bytes of communication.

Abstract

Single Slot Finality offers a promising approach to significantly reduce Ethereum’s latency. This document explores how this concept, combined with advanced zero-knowledge proofs (deVirgo) and improved signature schemes, can enhance Ethereum’s latency and user experience.

In this document, we introduce the concept of Single Slot Finality, its importance in blockchain consensus, and its benefits. We also discuss how to achieve this by using advanced zero-knowledge proofs and enhanced signature schemes.

Additionally, we introduce a low-hanging fruit in the form of a simple change to the Ethereum protocol that can significantly improve the light-client protocol. This change can be implemented in the short term and will have a positive impact on the Ethereum ecosystem. The idea comes from the zkBridge paper, which can be found here; it’s called “zkLightClient.”

If you are already familiar with all the concepts and importance of this problem, you can skip the introduction section.

Introduction

The Ethereum blockchain, a pioneering platform for decentralized applications and smart contracts, has continued to evolve in its pursuit of scalability and security. Among the many challenges blockchain networks face, achieving faster transaction finality while preserving decentralization and security remains a critical objective. This document explores the concept of Single Slot Finality by using a novel approach that holds the potential to revolutionize Ethereum’s latency and user experience.

The Importance of Single Slot Finality

In the realm of blockchain consensus mechanisms, finality signifies the point at which a transaction or block becomes irrevocable, ensuring that it cannot be tampered with or reversed. Achieving finality is fundamental for trust and security in decentralized systems, as it eliminates the risk of double-spending and other malicious activities. For example, bitcoin needs 6 blocks to finalize a transaction.

Single Slot Finality offers a compelling solution to expedite this process. It proposes that within a blockchain consensus mechanism, a single slot, or unit of time, can be considered “finalized”. It differs from the original Ethereum consensus because we can involve all validators in endorsing/signing the slot. This concept holds immense importance for Ethereum and other blockchain platforms, as it promises to drastically reduce transaction confirmation time and improves the overall user experience.

Achieving Single Slot Finality

The quest for Single Slot Finality involves innovative approaches to consensus and cryptographic techniques. Previous approaches can only involve 1/32 fractions of validators in one slot due to scalability constraints.

One of the central strategies for achieving this goal is the integration of advanced zero-knowledge proofs and using this tool to get an enhanced signature scheme. These technologies, often associated with privacy and security, have the potential to play a pivotal role in speeding up the finality of transactions on the Ethereum blockchain.

Additionally, we introduce a notable concept called “zkLightClient,” proposed by the zkBridge paper. This concept represents a practical, short-term solution to improve Ethereum’s light-client protocol, a critical component for lightweight, mobile, and resource-constrained Ethereum clients. By exploring the zkLightClient approach, we aim to present a tangible step toward achieving Single Slot Finality.

Understanding Single Slot Finality

In this section, we will introduce our way to achieve single slot finality. First we present Ethereum’s mechanism in simpler terms.

Committees and Blocks

Ethereum divides the job of verifying transactions among committees, with 32 committees groups in total. Each group is responsible for one block, and these blocks are organized into sets called “epochs.”

How Transactions Get Confirmed

When you initiate a transaction on Ethereum, it undergoes a validation process by committee members. To ensure the transaction’s security, it typically needs endorsements from at least two-thirds of the committee members. This confirmation process requires at least 2/3 of the epoch length and, in some cases, may take longer, especially when the block is near the end of the epoch.

Single Slot Finality

In response to the challenges posed by the existing system, we propose a transformative approach known as Single Slot Finality. This concept entails eliminating the division of committee members into groups. Instead, we advocate for all committee members to sign every slot. However, it’s essential to note that this approach, if implemented naively, can lead to significant scalability issues.

We propose to use zero-knowledge proofs to batch verify signatures, formally speaking:

Given the Merkle tree root of public keys of committee members M

, the size of committee N

, and a blockhash h

, the proof π

can prove that h

is signed by at least k

($k \leq N$

) of the committee members. This proof will be accepted if and only if $k \geq 2/3N$

and the proof itself passes the verification.

In the upcoming sections, we will delve deeper into the concept of Single Slot Finality and explore how deVirgo can be leveraged to achieve it effectively.

Advanced Zero-Knowledge Proofs

To efficiently prove the validity of a large number of signature verifications, we require an efficient proof system. In this context, we propose the use of deVirgo, a groundbreaking zero-knowledge proof system capable of verifying numerous signatures in a single proof. deVirgo is rooted in the Libra paper (link), which leverages the linear-time GKR algorithm. Remarkably, this algorithm demands just 6 field operations on each gate. Furthermore, we’ve implemented a recursive proof mechanism to reduce the proof size to a mere 200~300 bytes.

Additionally, we’ve developed a Rust-based prototype of deVirgo and conducted benchmarking tests on 64 Steam Decks. The results are impressive, demonstrating the capability to generate proof for 32,768 signatures within a mere 10 seconds. We’ve also conducted similar benchmarking on a 256-core server, which yielded comparable results. The benchmark results are illustrated below:

[

performance

1704×1108 102 KB

](https://ethresear.ch/uploads/default/original/2X/8/835c1a895869bdaee4ed34283649b1f26141b5be.png)

Notably, the total memory consumption remains below 100 GB and is evenly distributed across each machine. By utilizing 1,024 Steam Decks, it becomes feasible to verify the entire Ethereum validator’s signature set. However, it’s crucial to emphasize that this is a proof of concept, and further discussions and refinements are needed to fully define the algorithm and its implementation. We also compared with Groth16 (Gnark), two large data points are estimated. We observed that deVirgo is 100x faster than Groth16.

Enhanced Batched Signature Scheme

The aforementioned results naturally lead to the development of an efficient batch signature verification algorithm. Furthermore, this algorithm offers the capability to determine the number of valid signatures in a batch without disclosing the identities of the signers. This is particularly advantageous because revealing such information can be communication-intensive, especially when dealing with a large committee of, say, one million members. Traditionally, this might entail sending one million bits to the verifier, each bit represents the validator is signing / not signing, this is equivalent to 128KB of data. However, with the proposed algorithm, you only need to transmit a proof along with all public inputs, totaling less than 1KB in size.

Benefits and Challenges

- Benefits of the zk-based signature
- Significantly reduce the communication and computation cost of signature verification.
- Additional features, for example, counting the number of signers' stakes among all possible validators. In general, we can do arbitrary computation on the input data. It is crucial for PoS consensus to determine if the consensus has achieved supermajority.

- Significantly reduce the communication and computation cost of signature verification.
- Additional features, for example, counting the number of signers' stakes among all possible validators. In general, we can do arbitrary computation on the input data. It is crucial for PoS consensus to determine if the consensus has achieved supermajority.

- Challenges and Considerations
- Address challenges of gathering 1024

steam decks among all validators. (We estimate that it will take 32

to 64

high-end gaming PC to achieve the same performance.)

- This is a major upgrade. It will take a long time to implement and test.

- Address challenges of gathering 1024

steam decks among all validators. (We estimate that it will take 32

to 64

high-end gaming PC to achieve the same performance.)

- This is a major upgrade. It will take a long time to implement and test.

Low hanging fruit: zkLightClient

Integrating a new zero-knowledge proof system and redesigning the signature scheme are major upgrades that will take a long time to implement and test. However, we can still make some improvements in the short term. In this section, we will introduce a simple change to the Ethereum light-client protocol that can significantly improve performance. This change will not change the main consensus and the main-chain protocol, so it can be implemented in the short term. The idea is simple: proving all 512

validator signatures in one zk-proof. And the 200+ Byte proof can be easily propagated in the P2P network and verified by the light-client. This will significantly reduce the communication and computation cost of light-client. The zkLightClient is already developed by Polyhedra and deployed on LayerZero.

Conclusion

In the pursuit of making Ethereum more efficient, secure, and scalable, the concept of Single Slot Finality stands as a promising beacon of progress. By reimagining how transaction finality is achieved, we have the potential to significantly reduce latency, making Ethereum a more responsive and user-friendly blockchain platform.

Through this document, we have explored the key components of Single Slot Finality:

- Advanced Zero-Knowledge Proofs (deVirgo):

We introduced deVirgo, a groundbreaking zero-knowledge proof system capable of verifying large numbers of signatures in a single proof. With its roots in the Libra paper and the linear-time GKR algorithm, deVirgo showcases remarkable efficiency, reducing the proof size to just 200+ bytes. Benchmarks have shown its potential to validate 32,768 signatures within a mere 10 seconds.

- Enhanced Signature Schemes:

Our exploration naturally led to the development of an efficient batch signature verification algorithm. Beyond its efficiency, this algorithm offers the remarkable capability to determine the number of valid signatures in a batch without revealing the identities of the signers, significantly reducing communication and computation costs.

To smoothly integrate the proof system, we propose to integrate the “zkLightClient.” This lightweight, short-term solution, inspired by the zkBridge paper, significantly improves the Ethereum light-client protocol. The change to the system is minimal and has already been tested on bridges. By allowing the proof of all 512 validator signatures in a single zk-proof, zkLightClient reduces communication and computation costs, offering tangible benefits to Ethereum’s user base.

In conclusion, Single Slot Finality represents a profound shift in how we approach transaction finality in the Ethereum ecosystem. While challenges and complexities lie ahead, the potential for a faster, more scalable, and efficient Ethereum blockchain is within reach. As we move forward, it is essential to continue exploring, testing, and refining these concepts to pave the way for Ethereum’s continued growth and success.