Blockchain Focus

[ZK series - 1] ZK Rollups: Elephant In the Room

Presto Research

May 13th, 2024

Jaehyun Ha I Research Analyst
jaehyunha@prestolabs.io

Summary

- While Zero-Knowledge proofs (ZKPs) hold promise for a more private and scalable blockchain ecosystem, many aspects of ZK are misunderstood or implemented differently than commonly perceived.
- ZKPs have two main aspects: “Zero Knowledge” and “Succinctness”. While not incorrect, the majority of ZK rollups only utilizes the succinctness property; the transaction data and account information are not fully kept zero-knowledge nor private.
- ZK rollups may not be an optimal choice as a development stack for every kinds of DApps. For example, generating ZKPs may act as a bottleneck for fast finality, diminishing the performance of Web3 gaming, while state diff publish-based data availability assurance methods may detract from the service of DeFi lending protocols.

Figure 1: ZK is a good buzzword

The current state of the blockchain industry can be likened to the era of Zero-Knowledge (ZK). Everywhere you look, ZK is prominent… It's becoming increasingly rare to find next-generation blockchain projects that do not incorporate ZK into their names. From a technical perspective, there's no denying that ZK is a promising technology capable of contributing to a more scalable and private blockchain ecosystem. However, due to ZK's complex technical background, many investors, both retail and institutional, often find themselves investing in ZK projects based on the "belief" that it looks cool, new, and might solve the blockchain trilemma—without fully grasping how ZK technology benefits each project.

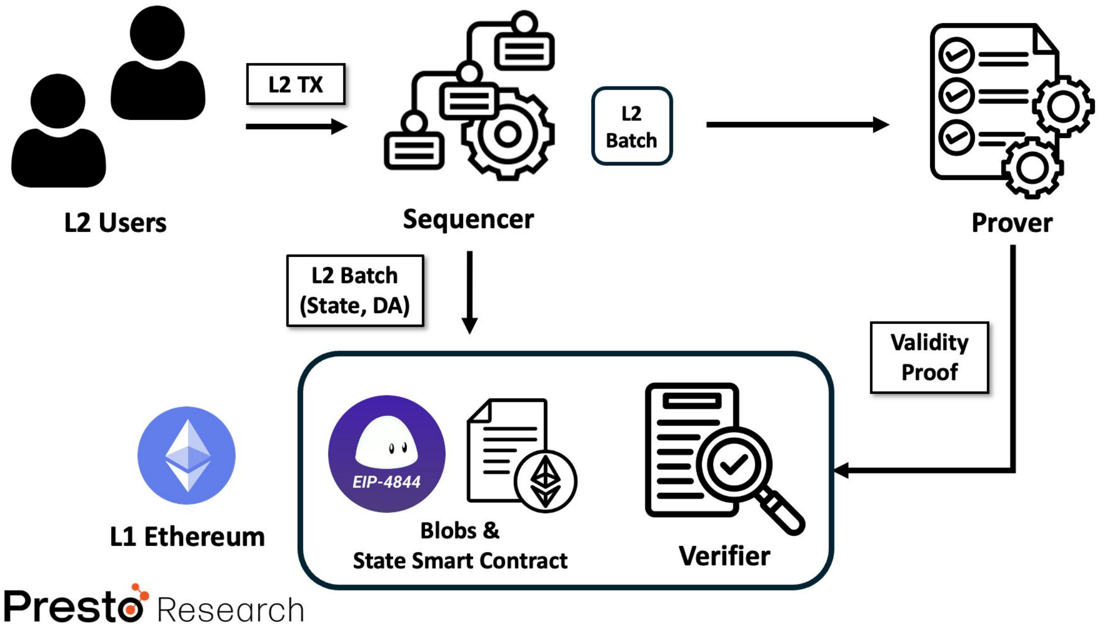
In this ZK series, we will explore both the inconvenient truths and the beneficial applications of ZK rollups. First, we will unpack the two core properties of ZK proofs (ZKPs) for blockchains: "zero-knowledge" and "succinctness"; then, we will discuss how a large number of ZK rollups currently in service don't actually utilize the "zero-knowledge" aspect. Next, we will examine the areas where applying ZK rollup is rather detrimental than beneficial, avoiding well-known issues like implementation complexity. Finally, we will highlight exemplary projects that effectively embody ZK principles and actually demonstrate tangible benefits from their use of ZK technology.

Recap: Transaction Lifecycle in ZK Rollups

Rollups are a scaling solution that addresses the throughput constraints of L1s by executing bundles of transactions off-chain and then storing summary data of the latest L2 state on the L1. Among them, ZK rollups stand out for its capability to promptly withdraw funds by submitting validity proofs for off-chain computation on-chain. Before we delve into issues with ZK rollups, let's briefly recap its transaction lifecycle.

Figure 2: Transaction lifecycle in ZK rollups

Source: Presto Research



1. Each L2 user generates and submits their transaction to the sequencer.
2. The sequencer aggregates and orders multiple transactions, then calculates the new rollup state by executing them off-chain. Subsequently, the sequencer commits this new rollup state to the on-chain state smart contract in the form of a “batch”, along with the corresponding L2 transaction data compressed into blobs to ensure data availability.
3. The batch is sent to the prover, and the prover creates a validity proof (or ZKP) of the batch’s execution. This validity proof is then sent to the L1’s verifier smart contract alongside the extra data (i.e., the previous state root) which helps the verifier recognize what it is verifying.
4. After the verifier contract checks that the proof is valid, the rollup’s state is updated and the L2 transactions in the committed batch are considered as finalized.

(Note that this explanation is a simplified version of the full ZK rollup process, and each of the implementations can vary depending on the protocol. There can be more entities in L2s if we separate the roles; such as aggregators, executors and proposers. Tiers of data chunk can also differ such as blocks, chunks, and batches depends on their usages. The above explanation assumes a situation where a centralized sequencer has strong authority that executes transactions and also produces a unified data chunk format as batches.)

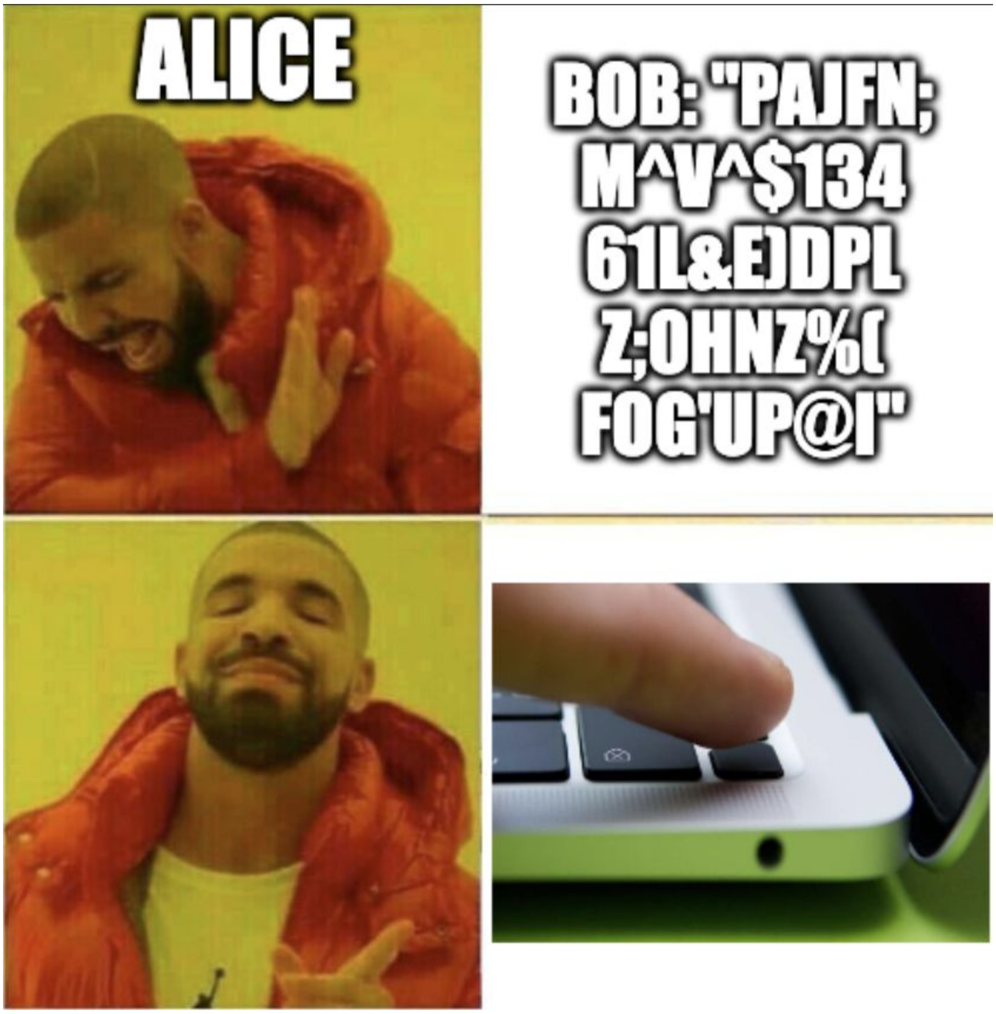
Unlike Optimistic rollups, thanks to ZKPs (e.g., ZK-SNARKs or ZK-STARKs), ZK rollups can verify the execution correctness of thousands of transactions just by verifying a simple proof, without replaying all of them. So, what is this ZKP, and what characteristics does it have?

Two Properties of ZKPs: Zero-Knowledge and Succinctness

As its name suggests, ZKP is basically a proof. A proof can be anything that can sufficiently backup the prover's claim. Let's say Bob (prover) wants to convince Alice (verifier) about the authority of his laptop computer. The easiest way to prove this is—Bob just tells Alice the password, and Alice types the password on the laptop and verifies that Bob has an authority. However, this verification process is unsatisfactory for both Alice and Bob. If Bob has set a really long and tangled password, it would be very challenging for Alice to type it correctly (assume that Alice cannot copy and paste). More realistically, Bob may be reluctant to disclose his password to Alice in order to prove his authority.

What if there is a verification process where Alice can swiftly verify the computer's authority, without Bob having to reveal his password? For instance, Bob can just tap his finger to unlock the laptop with touch ID in front of Alice as in Figure 3 (note that this is not a perfect example for ZKP). This is where both Alice and Bob can benefit from both key properties of ZKPs: the zero-knowledge property, and the succinctness property.

Figure 3: High-level intuition of Zero-Knowledge and Succinctness



Zero-Knowledge

The property "zero-knowledge" refers to a case where the proof generated by the prover reveals nothing about the secret witness (i.e., private data), leaving the verifier unaware of anything about the data except the validity of the proof. In blockchain, this property can be utilized for preserving privacy of individual users. If ZKPs are applied for each transactions, users can prove the legitimacy of their actions (i.e., proving that a user has enough funds to make a transaction) without exposing the details of their transactions (e.g., transfers, account balance updates, smart contract deployments, and smart contract executions) to the public.

Succinctness

The other property “succinctness” refers to ZK’s ability to generate a short and fast-to-verify proof from a big size claim. In other words, it is the consolidation of something big into something compact. In blockchain, this is especially utilized in rollups. With ZKPs, provers in L2s can claim the correct execution of transactions by submitting a succinct proof to the verifier in the L1 (validity of TBs of transactions can be represented with 10~100 KB of proof). Verifiers then can easily confirm the validity of executions in a short time (i.e., 10ms~1s) by verifying the succinct proof instead of replaying all the transactions.

ZK Rollup is Great, but Doesn’t Mean Privacy

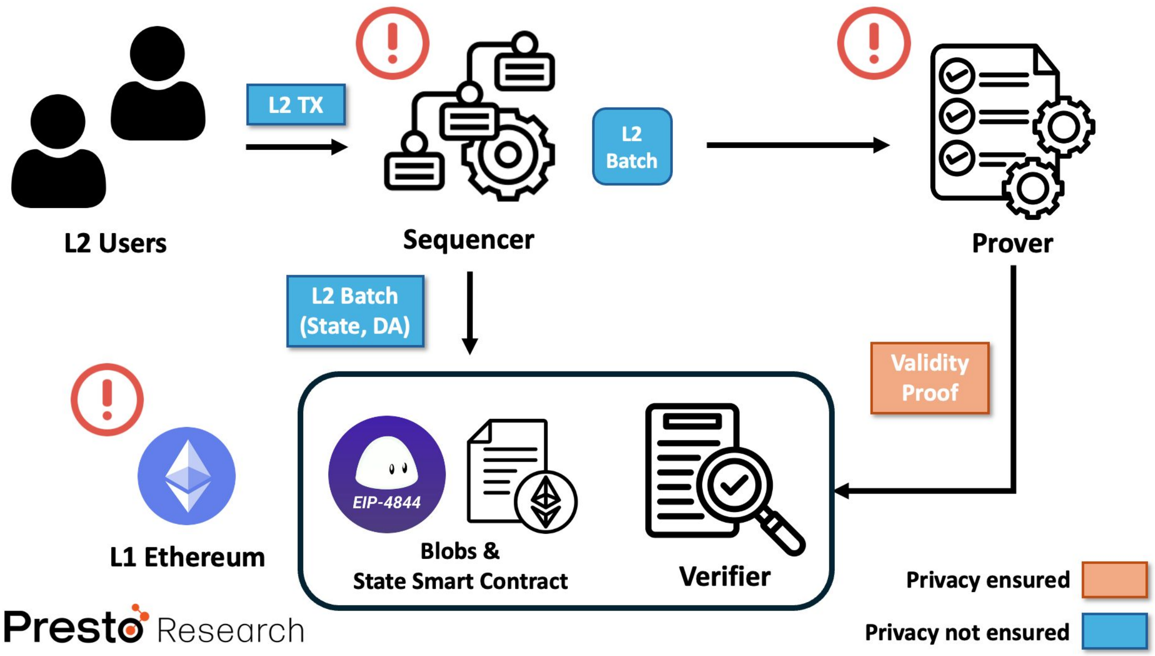
The aforementioned ZKP characteristics are well-utilized in ZK rollups. While verifiers cannot infer the original transaction data from the ZKPs received from the prover, verifying the succinct proof allows them to efficiently validate the prover’s claim (i.e., the new L2 state). That said, the assertion that the ZK rollups in their current iteration fully adheres to zero-knowledge and succinctness properties is misleading. This may be true when focused solely on the interaction between the prover and the verifier, but there also exists other components in ZK rollups, such as sequencer, prover and rollup nodes. Is the “zero-knowledge” principle assured for them as well?

The challenge in achieving full privacy with ZKPs in any ZK rollups arises from the potential compromise if other parts remain public while some are made private by ZK. Think of the transaction lifecycle in ZK rollups — is privacy maintained when the transaction is sent from a user to sequencers? How about for provers? Or is the privacy of an individual account information preserved when the L2 batch is submitted to the DA layer? None of the scenarios currently holds true.

In most of the mainstream ZK rollups, the sequencer or prover (or some other centralized entities with strong authority) has clear visibility of transaction details which include transfer amounts, account balance updates, contract deployments, and contract executions.

Figure 4: Privacy leakage in ZK rollups

Source: Presto Research



As an easy example, you can easily observe all the mentioned details by visiting any of the ZK rollup block explorers. Not only that, consider a situation where the centralized sequencer is somehow out of service and another rollup node tries to restore the rollup state. It will scoop up the publicly published L2 data from the DA layer (which is L1 Ethereum in most cases), and reconstruct the L2 state. In this process, any node capable of replaying the L2 transactions stored in the DA layer can recover the information about each users' account status.

Thus, the term "zero-knowledge" is implemented in a fragmented form in current ZK rollups. While this cannot be deemed as incorrect, it is evident that it differs from the commonly perceived notion that "ZK means zero-knowledge and equals to full privacy". The **novelty of current ZK rollups** is to leverage the "succinctness" property rather than "zero-knowledge", which is to execute the transactions off-chain, and **generate succinct proofs** for verifiers to verify the validity of the execution in a fast and scalable way without re-executing them.

For this reason, some ZK rollups such as Starknet refer to themselves as “Validity Rollups” to avoid confusion, while others that ensure true ZK privacy, like Aztec, label themselves as ZK-ZK rollups.

Think Through Practicality of ZK Rollups

As mentioned above, ZK privacy is not fully implemented in most ZK rollups. So, what should be our next goal? Achieving complete transaction privacy by fully deploying ZK in every part of the rollup? In fact, this is not a simple problem. Besides the need for significant technological advancements to further mature the technology, there remain contentious issues for ZK in terms of ideology (e.g., illegal usage of private transactions) and practicality (e.g., is it actually useful?). Given debating the morality of full transaction privacy is beyond the scope of this article, let’s focus our attention on the two practicality points of ZK rollups encountered by blockchain projects.

Point #1: Generating ZKPs can be a Bottleneck for Fast Finality

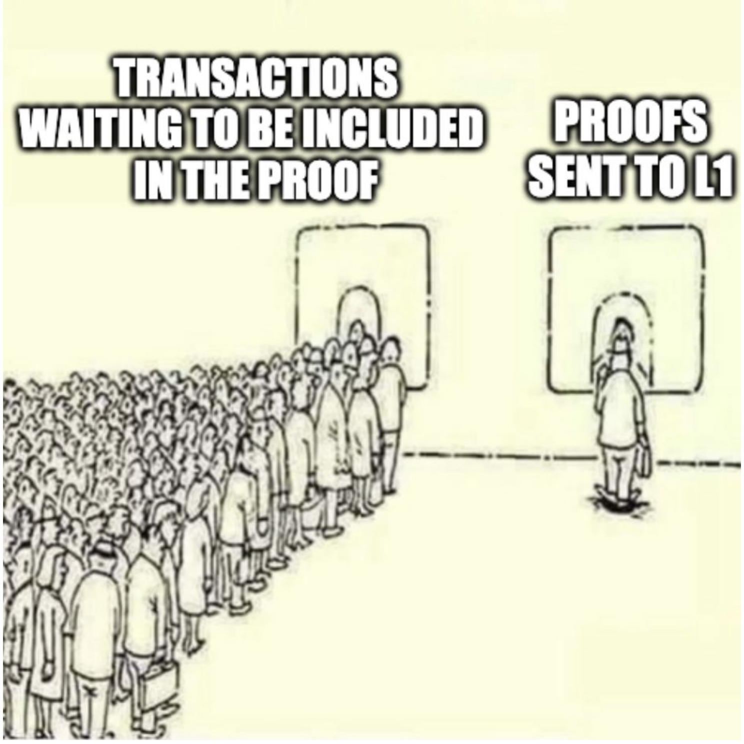
Let’s first discuss about the practicality of ZK rollups itself. The most compelling selling point of ZK rollups is the short asset withdrawal delay due to its “fast finality” of transactions thanks to ZKPs. Enhanced TPS and low transaction fees are a bonus. The sector that most effectively leverages the characteristics of ZK rollups is gaming, since deposits and withdrawals of in-game currency occur very frequently, and there is a high volume of in-game transactions generated every second.

But can ZK rollups truly be considered as the optimal stack for gaming? For this, we need to think a bit more about the concept of “fast finality” in ZK rollups. Imagine a situation where one user is enjoying a Web3 game running on a ZK rollup-based stack. The user trades an in-game item into an in-game currency, and attempts to withdraw that asset from the game.

To withdraw the asset, the in-game transaction has to be finalized; this means the transaction has to be included in the new rollup state commitment, the corresponding ZKP should be submitted to the L1, and there is a wait for the proof to be finalized in L1 Ethereum so it can guarantee that the transaction cannot be reverted. If all of these processes were to occur instantly, then yes, we could achieve the "instant transaction confirmation" for which ZK rollups are often touted, allowing the user to withdraw the asset right away.

However, the reality is far from that. According to statistics provided by L2beat on finality time for different ZK rollups, zkSync Era takes around 2 hours, Linea takes 3 hours, and Starknet averages around 8 hours. This is because it takes time to generate a ZKP from a prover, and it takes additional time to include more transactions into a single batch (i.e., single proof) to reduce the cost of transaction fees. In other words, the speed of generating and submitting the proofs is a potential bottleneck for achieving fast finality in ZK rollups, which can diminish the user experiences in Web3 gaming.

Figure 5: ZKP generation can be a potential bottleneck for fast finality in ZK rollups



On the other hand, gaming-optimized chains like Ronin (powers Web3 games such as Pixels and Axie Infinity) ensures super fast finality while sacrificing decentralization and security. Ronin is not a ZK or rollup-based chain: it is an EVM blockchain that runs under PoA (Proof of Authority) + DPoS (Delegated Proof of Stake) consensus algorithm. It selects 22 validators based on the amount of stake delegated, then these validators simply generate and validate blocks in a PoA manner (i.e., voting process only among the 22 validators). Hence, transactions are finalized swiftly on Ronin, as it has almost no delay for the transactions to be included in the block, and takes little time to be validated. After the Shillin hardfork, it takes an average of only 6 seconds to finalize each transaction. Ronin achieves all this without ZKP.

And yes, of course, Ronin has drawbacks too. Being governed by centralized validators makes it relatively more vulnerable to a 51% attack. Moreover, as it doesn't utilize Ethereum as a settlement layer, it cannot inherit Ethereum’s security. Security risks associated with the use of a cross-chain bridge also exists. But think in the users’ perspective: will they care about that? Current ZK rollups without decentralized sequencing also suffer from the the single point of failure (SPOF) problem. Ethereum offers them assurance as it reduces the likelihood of transactions reverting, but if the centralized sequencer or prover goes down, ZK rollups also freeze. Note again that “ZK” in ZK rollups is only utilized for verifying the validity of execution correctness. If there’s another project offering the same functionality as ZK rollups but faster and cheaper, ZK rollups may no longer be considered the top priority stack by Web3 game users & developers.

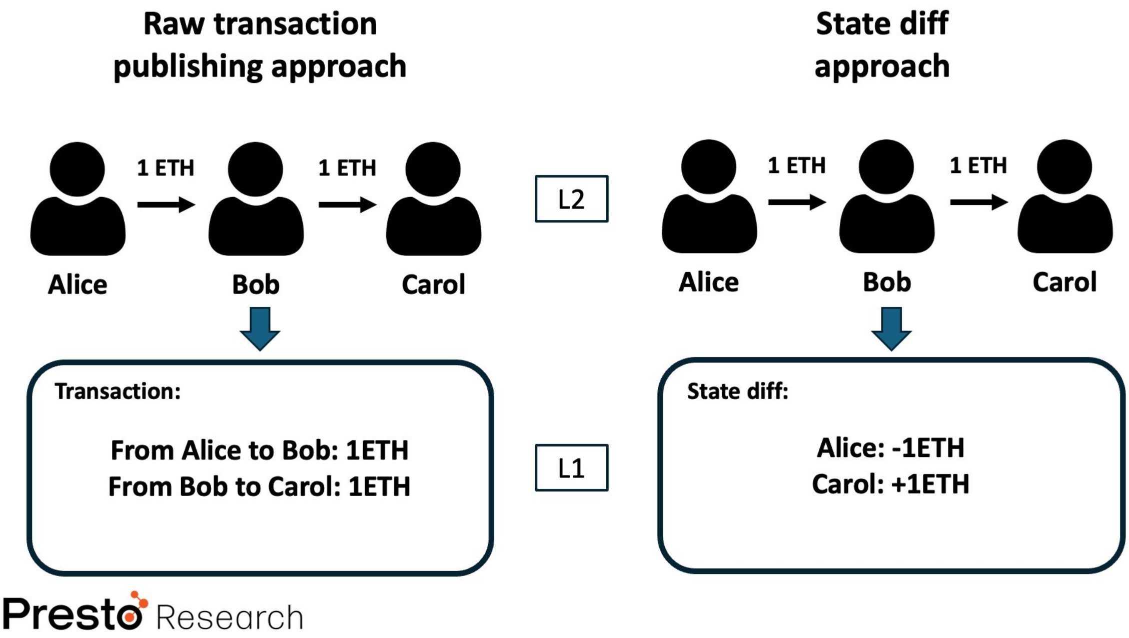
Point #2: Publishing State Diffs is a Double-edged Sword
Another point is the practicality of protocol implementations for ZK rollups. Among them, here we focus on state diff publishing, which is one of the ways to ensure data availability (See *Unlocking Dencun Upgrade: Unseen Truth of Scaling DA Layers*, Jaehyun Ha, 12Apr24) in ZK rollups.

An easy way to understand data availability in rollups is to think of a amateur climber certifying and recording his Mt. Everest climb. The simplest method is to record every step of the climb from the basecamp to the summit in a video. Though the video file may be large, anyone can verify the climber's ascent of Everest and perhaps replay the footage. This analogy can be likened to the **raw transaction data publishing method** for ensuring data availability. Optimistic rollups follow this approach in order to make the individual challengers replay and verify the correct execution, since there is nothing to trust about the sequencer's state commitment. Among ZK rollups, Polygon zkEVM and Scroll adopt this approach, storing raw L2 transaction data in a compressed form on the L1 so that anyone can replay L2 transactions to restore the rollup's state when needed.

Back to the example of the amateur climber, an alternative verification method might be a prominent mountaineer ascending Everest along with the amateur climber to verify to the world that the climb was indeed completed. Since the ascent has been certified by a trusted individual, the climber no longer needs to record every step for documentation. Simply taking a photo at the starting point and another at the summit would suffice, and others would just consider the climber to have reached the summit. This analogy reflects the **state diff method** used to ensure data availability. In ZK rollups, zkSync Era and StarkNet employ this approach, storing only the state difference before and after the L2 transactions are executed on the L1 so that anyone can calculate state differences from genesis to restore the rollup's state when necessary.

Figure 6: Raw transaction publishing vs. State diff publishing

Source: Presto Research



This state diff approach is undoubtedly beneficial in terms of cost compared to the raw transaction data publishing approach since it can skip out storing the intermediate transactions, reducing the storage cost in L1. However, although not commonly an issue, there is an underlying drawback here: this approach doesn’t allow for a restoration of the full L2 transaction history, which can be an issue for some DApps.

Let’s take Compound, the DeFi lending protocol as an example, and assume that it is built on top of a state-diff approach based ZK rollup stack. These protocols require the full transaction history in order to calculate the supply and borrow interest rates every second. But what will happen, if somehow the ZK rollup sequencer goes down, and other rollup nodes try to restore the latest state? It may restore the state, but the interest rate will be inaccurately restored since it can only track the snapshots between batches rather than every intermediary transactions.

Conclusion

This article mainly asserts that there is no “ZK” in most of today’s ZK rollups, and there are lot of areas in DApps that utilizing ZKP & ZK rollups may not be the most optimal choice. ZK technology might feel innocent for getting blamed; because there is nothing inherently wrong with itself—It’s just that in the process of leveraging its technical advancements, it may bring potential performance degradation in DApps. However, this is not to say that ZK technologies are useless for this industry. When ZKPs and ZK rollups eventually come through with technical maturity, they can certainly provide even better solutions to solve the blockchain trilemma. In fact, there exist ZK-based projects that maintains ZK privacy as well as many types of DApps that effectively leverage the benefits of ZKP and ZK rollups. We will explore this further in the next article - stay tuned!

About Presto

Presto is a Singapore-based algorithmic trading and financial services firm founded in 2014. Presto focuses on delivering exceptional value for clients through rigorous research-driven approach to investment and trade execution. With more than a 100 million trade executions in a day, Presto is a leading financial services firm in both digital assets and traditional finance markets.

Find out more at https://www.prestolabs.io.
Follow Presto for more content: X, LinkedIn

Authors

Jaehyun Ha, Research Analyst : X, LinkedIn

Required Disclosures

Any expression of opinion (which may be subject to change without notice) is personal to the author and the author makes no guarantee of any sort regarding accuracy or completeness of any information or analysis supplied. The views and opinions expressed herein are those of the author(s) and do not necessarily reflect the views of Presto Labs or its affiliates. This material by itself, is not and should not be construed as an offer or a solicitation to deal in any investment product or to enter into any legal relations. This material is for informational purposes only and is only intended for sophisticated investors, and is not intended to provide accounting, legal, or tax advice, or investment recommendations, or an official statement of Presto Labs or its affiliates. Presto Labs, its affiliates and its employees make no representation and assume no liability to the accuracy or completeness of the information provided. Presto Labs, its affiliates and its employees also do not warrant that such information and publications are accurate, up to date or applicable to the circumstances of any particular case. Certain statements in this document provide predictions and there is no guarantee that such predictions are currently accurate or will ultimately be realized. Prior results that are presented here are not guaranteed and prior results do not guarantee future performance. Recipients should consult their advisors before making any investment decision. Presto Labs or its affiliates may have financial interests in, or relationships with, some of the assets, entities and/or publications discussed or otherwise referenced in the materials. Certain links that may be provided in the materials are provided for convenience and do not imply Presto Labs' endorsement, or approval of any third-party websites or their content. Any use, review, retransmission, distribution, or reproduction of these materials, in whole or in part, is strictly prohibited in any form without the express written approval of Presto Labs. Presto Research and related logos are trademarks of Presto Labs, or its affiliates.