TL;DR

In a sharded blockchain protocol that utilizes recursive zk-SNARK proofs to enable scalable, private cross-shard transactions with constant-size proofs of validity. This architecture allows for horizontal scaling while maintaining strong privacy guarantees.

Background

Existing blockchain systems face significant challenges in terms of scalability and privacy. Sharding is a promising approach to improve transaction throughput by parallelizing computation across multiple chains. However, cross-shard communication remains a bottleneck, as verifying transactions across shards typically requires expensive cross-shard proofs.

Zero-knowledge proofs, particularly zk-SNARKs, offer a powerful tool for enhancing privacy by allowing users to prove knowledge of secret information without revealing it. Unfortunately, generating and verifying zk-SNARK proofs incurs high computational overhead, limiting their practicality for large-scale applications.

Prior solutions have attempted to combine zk-SNARKs with sharding, but fail to fully address the scalability challenges. For example, Zexe uses zk-SNARKs in a sharded setting but requires storing a linear-size “state proof” on-chain. Coda achieves constant-size proofs using recursive composition, but lacks the horizontal scaling benefits of sharding.

Proposal

We introduce a novel construction that synergistically combines sharding with recursive zk-SNARK proofs for unparalleled scalability and privacy.

At the core of this DLT is a hierarchy of zk-SNARK proofs that recursively attest to the validity of state transitions within and across shards. Each shard generates succinct proofs, called Zero-Knowledge Balance & Inclusion State Proofs (ZkBISPs), certifying the correctness of their local state updates. These ZkBISPs are then aggregated by a designated coordinator into a global proof, termed a Zero-Knowledge Succinct Nested Global-state Proof (ZkSNGP).

Crucially, the ZkSNGP is a constant-size proof that recursively verifies the validity of all shard-level ZkBISPs, thereby providing a succinct and efficient means to prove the integrity of the entire cross-shard state transition. Verifying the ZkSNGP requires only logarithmic time in the number of shards, enabling exponential savings compared to naively checking each shard’s proofs individually.

Formally, we define the intra-shard state transition language \mathcal{L}_{\mathsf{ST}}^{(t,i)}

for each shard i

at epoch t

as the set of tuples (x, w)

where:

- The statement x = (\mathsf{shardID}_i, \mathsf{root}_i^{(t-1)}, \mathsf{root}_i^{(t)}, B_i^{(t)})

includes the shard ID, starting and ending state roots, and final account balances.

- The witness w = (\mathsf{txs}_i^{(t)}, \mathcal{T}_i^{(t-1)}, \mathcal{T}_i^{(t)})

contains the list of transactions, along with the initial and final account state trees.

- (x, w) \in \mathcal{L}_{\mathsf{ST}}^{(t,i)} \Leftrightarrow \mathsf{root}_i^{(t-1)} = H(\mathcal{T}_i^{(t-1)}) \wedge \mathsf{root}_i^{(t)} = H(\mathcal{T}_i^{(t)}) \wedge \text{transition}(\mathcal{T}_i^{(t-1)}, \mathsf{txs}_i^{(t)}) \rightarrow \mathcal{T}_i^{(t)}

, i.e, the roots match the account trees and the final tree results from applying valid transactions to the initial tree.

Similarly, we define the cross-shard state transition language \mathcal{L}_{\mathsf{CST}}^{(t)}

for epoch t

as the set of tuples (x, w)

where:

- The statement x = (\mathsf{root}_G^{(t-1)}, \mathsf{root}_G^{(t)})

consists of the starting and ending global state roots.

- The witness w = \left(\left{\left(\mathsf{shardID}*i, \pi*{\mathsf{ST},i}^{(t)}, \mathsf{root}*i^{(t-1)},\mathsf{root}_i^{(t)},* B_i^{(t)}\right)\right}{i=1}^{\ell},\mathcal{T}_G^{(t-1)},\mathcal{T}_G^{(t)}\right)

includes the shard IDs, ZkBISPs, local roots and balances, and global account trees.

- (x, w) \in \mathcal{L}{\mathsf{CST}}^{(t)} \Leftrightarrow \forall i: \mathsf{Verify}{\mathsf{ST}}(\mathsf{vk}{\mathsf{ST}}, x_i, \pi{\mathsf{ST},i}^{(t)}) \wedge \mathsf{root}G^{(t-1)} = H(\mathcal{T}_G^{(t-1)}) \wedge \mathsf{root}_G^{(t)} = H(\mathcal{T}_G^{(t)}) \wedge \text{merge}(\mathcal{T}_G^{(t-1)}, {\mathsf{root}_i^{(t)}, B_i^{(t)}}{i=1}^{\ell}) \rightarrow \mathcal{T}_G^{(t)}

, i.e., the ZkBISPs verify w.r.t. their shards, the Merkle roots match, and the final global tree is the result of correctly merging the shards' final local trees and balances.

A shard's ZkBISP for epoch t

is generated as \pi_{\mathsf{ST},i}^{(t)} \leftarrow \mathsf{Prove}{\mathsf{ST}}(\mathsf{pk}{\mathsf{ST}}, x_i, w_i)

for (x_i, w_i) \in \mathcal{L}_{\mathsf{ST}}^{(t,i)}

, where \mathsf{pk}_{\mathsf{ST}}

is the proving key for the corresponding zk-SNARK scheme. The coordinator's ZkSNGP is computed analogously as \pi_{\mathsf{CST}}^{(t)} \leftarrow \mathsf{Prove}{\mathsf{CST}}(\mathsf{pk}{\mathsf{CST}}, x, w)

for (x, w) \in \mathcal{L}_{\mathsf{CST}}^{(t)}

The coordinator, randomly selected in each epoch, collects these ZkBISPs along with the shards' final state roots \mathsf{root}_i^{(t)}

and account balances B_i^{(t)}

. It then generates the ZkSNGP \pi_{\mathsf{CST}}^{(t)}

(green proof) certifying the validity of the overall state transition, including the correct application of all shard-level updates to the global state.

Advantages

The network simultaneously achieves exceptional horizontal scalability and privacy without sacrificing security or decentralization.

In terms of scalability, the concept supports an unprecedented number of shards and transactions per second while retaining a constant-size proof of the system's entire state. Concretely, if there are \ell

shards each processing N

transactions, the communication cost per epoch is only O(\ell)

for the coordinator to collect the ZkBISPs, and the ZkSNGP proof adds just O(1)

to the blockchain size. Crucially, verifying the ZkSNGP requires O(\log \ell)

time, an exponential speedup compared to naively verifying all \ell

shards.

For example, suppose the network is instantiated with \ell = 2^{10}

shards, each processing N = 2^{20}

transactions in 2-minute epochs. This configuration could support a peak throughput of roughly 1 billion transactions per epoch, or 500,000 transactions per second, with a ZkSNGP verification time of only 10\log \ell \approx 100

ms on ordinary hardware. The recursive proof would contribute a mere 1 KB to the blockchain per epoch, maintaining years of history in a highly compact format.

In terms of privacy, the ZkSNGPs inherit the zero-knowledge property of the underlying zk-SNARK scheme, revealing nothing about the shards' local transactions beyond the final state roots and balances. An adversary that compromises the coordinator cannot glean any additional information, as the shards' ZkBISPs are similarly zero-knowledge. Transactional privacy thus holds as long as at least one shard remains honest.

Compared to prior sharded blockchain designs, the network is the first to achieve sublinear proof sizes and verification times by recursively composing zk-SNARKs. Relative to Zexe, the network attains a qualitative improvement in scalability by eliminating the linear-size “state proof” in favor of constant-size ZkSNGPs. Compared to Coda, the network offers strictly stronger performance due to its sharded architecture, while still leveraging Coda’s core technique of recursive proof composition.

Applications

the network’s dual emphasis on scalability and privacy renders it a natural foundation for a variety of high-throughput, privacy-centric blockchain applications.

On the payments front, the network could serve as a backend for a globally-scalable digital currency with strong confidentiality guarantees, concealing both transaction amounts and participants. The subtransactions within each shard could clear near-instantaneously, while cross-shard payments would incur a maximum delay of one epoch (e.g., 2 minutes) before the ZkSNGP confirms finality. This would support a substantially higher payment volume than existing solutions like Zcash without leaking metadata.

More broadly, the network could function as a privacy-preserving platform for general smart contract execution. Shards would not only process token transfers but also arbitrary state transitions, with the ZkBISPs and ZkSNGP verifying the correctness of all contract logic and dependencies. This would enable complex applications such as private decentralized exchanges, automated market makers, and lending protocols to run at scale, without disclosing individual users’ balances or positions.

The network’s sharded architecture could also be adapted to specific domains to meet their unique performance requirements. For instance, a decentralized adtech ecosystem that handles billions of micropayments per day could utilize more granular sharding (e.g., \ell = 2^{20}

shards), with each shard perhaps corresponding to a particular geographic region or publisher. A secure messaging app that routes payments alongside packets could likewise tune its cross-shard spanning tree structure based on network topology.

Conclusion

the network introduces a powerful new paradigm for designing scalable and private blockchain protocols through recursive zk-SNARK proof composition. By strategically combining recursive proofs with sharding, the network enables a significant breakthrough in blockchain performance, supporting over a million transactions per second with sublinear proof sizes and verification times.