TL;DR

- For a ZKP/FHE task, we can decompose it into several subtasks.
- We introduce redundancy into subtasks such that the original task's result can be decoded from a subset of the subtask results, treating uncompleted subtasks as erasures. This is similar to the erasure code design in DA.
- For a (n,k)

coded ZKP/FHE system, we can decompose a ZKP/FHE task into n

subtasks, with k \leq n

subtask results, we can obtain the original task's result.

- With this coded design, we can design a decentralized, collaborative, robust ZKP/FHE system.

Background

ZKP/FHE systems play a pivotal role in blockchain ecosystems, ensuring privacy and enabling cost-effective verification. However, the resource-intensive nature of ZKP generation and FHE computation presents a significant challenge. To address this, numerous distributed algorithms have been devised to enhance scalability and are now integral to ZKP/FHE mining pools.

For instance, complex ZKP tasks can be subdivided into smaller subtasks, which are then distributed across multiple nodes for parallel processing. However, the efficacy of existing distributed algorithms falls short in ensuring robustness, hindering the realization of decentralized and collaborative systems.

Consider a scenario where a ZKP task is divided into k

subtasks and allocated to k

distinct nodes. Should one of these nodes fail to respond promptly, the entire computation process is stalled. While redundancy mechanisms, such as assigning each subtask to two nodes, may mitigate this risk, vulnerabilities persist. Even with this redundancy, if both nodes assigned to a task fail to respond in a timely manner, computational delays ensue.

In summary, while distributed algorithms offer scalability benefits, their current limitations impede the development of resilient decentralized and collaborative ZKP/FHE systems. Addressing these shortcomings is essential for advancing the efficacy and reliability of such systems within blockchain environments.

Proposal

In this proposal, we introduce redundancy into subtasks to enhance the robustness of Zero-Knowledge Proof/Fully Homomorphic Encryption (ZKP/FHE) systems, akin to the erasure code design in Distributed Algorithms (DA). Specifically, in a (n,k)

coded ZKP/FHE system, a ZKP/FHE task is decomposed into n

subtasks, which are then distributed across n

nodes. With a minimum of k

completed subtask results, where k \leq n

, the original task's result can be obtained.

To illustrate this concept, let's consider a toy model of matrix multiplication in zkML/fheML.

[

Snipaste_2024-05-05_15-58-22

960×294 36.2 KB

](https://ethresear.ch/uploads/default/original/3X/2/b/2b2ff2f906f369f930c8722e9adf55b9d01aab43.png)

Consider a system comprising three worker nodes and one master node. In this setup, a data matrix A

is divided into two submatrices, A_1

and A_2

. Specifically, node W_1

stores A_1

, node W_2

stores A_2

, and node W_3

stores the sum A_1 + A_2

. Upon receiving input X

, each node computes the product of X

with the respective stored matrix and transmits the result to the master node. Notably, the master node can reconstruct the product AX

upon receiving any two products, thus obviating the need to await the slowest response. For instance, consider a scenario where the master node receives A_1X

and (A_1 + A_2)X

. Through subtracting A_1X

from (A_1 + A_2)X

, the master node can deduce A_2X

and consequently reconstruct AX

.

We can further adopt an (n,k)

MDS code in this matrix multiplication example for generalization. For example, in zkML or fheML, we can adopt the (n,k)

coded approach to design a decentralized, collaborative, robust ZKP/FHE System. In zkML, we can decompose the task into n

subtasks, with the results and zkp of k

subtasks, we can aggravate the zkp of these subtasks with the decoded process. In fheML, we can decompose the task into n

subtasks, with the results of k

subtasks, we can apply the fhe computation on the decoded process.

The preceding discussion has focused on a particular use case, namely zkML/fheML. The coded design methodology explored can be extrapolated to the foundational elements of the ZKP/FHE framework, facilitating the creation of a comprehensive coded ZKP/FHE system capable of supporting applications such as zkRollup and fheEVM computation.

Specifically, this coded approach can be applied to various components of the ZKP system, enabling the development of a distributed coded system. For instance, the R1CS instance in ZKP involves numerous multi-scalar multiplications, which can seamlessly integrate with the coded design. With the distributed computation algorithms applied in current distributed ZKP systems, we can further enhance the efficiency and scalability.

I may design and implement a PoC version of the coded ZKP system in my free time

Advantages

- This coded design significantly accelerates computation within ZKP/FHE systems. Theoretically, assuming a node count of n

and subtask runtimes with exponential tails, the coded approach could be \theta(\log n)

times faster than conventional uncoded distributed algorithms.

- With this coded design, the ZKP/FHE system is more robust. For example, in a (n,k)

coded ZKP/FHE system, we can tolerate the downtime or delay of n-k

nodes.

Applications

Utilizing the coded design paradigm, a ZKP/FHE mining pool can be devised, where decentralized agents function as worker nodes engaged in the computation of subtasks. The managerial role within this context is assumed by the manager of the mining pool, serving as the master node responsible for aggregating and decoding the results submitted by the worker nodes. Furthermore, the conventional master node architecture can be supplanted by a smart contract, assuming the duties of result aggregation and decoding. This architectural transformation facilitates the establishment of a decentralized, collaborative, and resilient ZKP/FHE system, wherein cryptographic operations are conducted in a distributed manner, enhancing the system's robustness and scalability.

Conclusion

By breaking down ZKP/FHE tasks into subtasks and incorporating redundancy, a (n,k)

coded ZKP/FHE system emerges, enabling the decomposition of tasks into n

subtasks, with k \leq n

subtask results required for task reconstruction. This coded approach facilitates the creation of decentralized, collaborative, and resilient ZKP/FHE systems, promising enhanced efficiency and reliability in cryptographic operations.