ACCELERATING THE POSEIDON HASH FUNCTION

Prize Sponsor

Aleo

Prize Architect

panther.

Prize Description

Summary

An efficient Posiedon, in this instance, will be one which can effectively leverage a customisable proof system to construct a high degree hash within a low degree circuit. Utilizing the customisable PLONK proving system, one can define new operations which facilitate creative approaches to implementing an efficient and secure hash function.

The Poseidon hashing algorithm is deemed a zero-knowledge friendly cryptographic hash function; hashing is one of the most critical operations within state of the art zero-knowledge applications. The choice of SNARK in this case is PLONK instantiated with the BLS12-381 curve and the KZG polynomial commitment scheme. The target library is the ZK-Garage.

Optimization Objective

This prize consists of two distinct sub-tasks:

- Implement a custom gate for x^{α} and reflect the relevant changes in the proving and verifying algorithm
- Implement Poseidon with the lowest possible number of constraints per hash using the BLS12-381 scalar field. Using $\alpha = 5$, $T = 9$ and $arity = 8$
- Both tasks need to be completed to be eligible for the prize
- Participants do not need to derive their own parameters and should use the constants provided.
- The polynomial commitment scheme used must be KZG10
- The arithmetic gates should have $arity = 4$, meaning 3-input and 1-output witness wires
- The Implementation must be inside ZK-Garage
- Implementation of the hash function must be inside the hashing crate.
- Implementation of the custom gate must be inside the core proving crate.
- Submissions must include documentation, in English, to understand the optimization approach.
- Submissions must include tests using, at a minimum, the provided test vectors.

Timeline

- **June 10 -** Competition begins
- **July 25 -** Mid-competition submission due
- **September 10** - Final submission due

Judging

Correctness

We will provide a set of test input/output pairs so that the competitors can sanity check the correctness of their code.

Performance

Performance will be measured in constraint numbers. Given that the competitors adhere to the constraints above, specifically the target library and gate arity. The lowest concrete constraint count will be the winner (specifics defined in the next category). Only below a constraint count of 450 gates, per single hash, will submissions be considered eligible for the prize.

Prize Allocation

The prize amounts will each be divided among the top three finishers according to the following proportions: 75% to winning implementation, 15% to second place, and 10% to third place. In the event that there are only two qualifying submissions, first place will receive 80% of the prize pool and second place 20%. In the event there is only one qualifying submission, they will receive 100% of the prize pool.

In addition, all submissions will be manually reviewed by the prize committee consisting of representatives designated by Aleo Systems Inc., Polychain Capital, and Panther Protocol.

Notes

All submission code must be open-sourced at the time of submission. Code and documentation must be dual-licensed under both the MIT and Apache-2.0 licenses.

Although only the only an implementation of x custom gate is required, participants are encouraged to explore deep optimisations to improve the hashing algorithm.

Competitors are advised to clone the target repository and not fork it, as forking will make the work public.

All will be defined as detailed issues in the Github before competition start date.

This will include extra details, tips and and a framework for implementation.

Link where it will be described: https://github.com/ZK-Garage/plonk/issues

Questions

If any questions, please reach out via *TG, Discord or by email*:

Saif@pantherprotocol.io or luke@polychain.capital