HALO2 COMBINE STEP ACCELERATION

Prize Sponsor

MINA

Prize Architect

MINA

O(1) LABS

Prize Description

Summary

The Bulletproof inner-product argument used in Halo-type SNARKs is one of the more costly operations for provers. This prize focuses on improving the performance of this operation in browser-based provers.

Optimization Objective

Competitors should provide, using the arkworks library and its curves implementations, a function

```
fn combine(g1: &[ark_pallas::Affine], g2: &[ark_pallas::Affine], chal: &[bool]) ->
Vec<ark_pallas::Affine>;
```

The semantics of this function should be as in this gist.

Constraints

- Runtime must be WASM
- The MSM must be over the Pallas curve as described.
- Web workers may be used

NOTE: All submissions must include documentation (in English) sufficient to understand the approach being taken.

Timeline

- **June 17 -** Competition begins
- **Aug 2 -** Mid-competition submission due
- **September 17** - Final submission due

Judging

Submissions will be checked for correctness and ranked by performance. In addition, documentation (in English) must be provided along with the implementation. The documentation can be written in-line or as a separate document. It should be thorough and explanatory enough to provide an understanding of the techniques used in the submitted implementation without requiring an associated verbal explanation.

Correctness

Submissions will be tested against the sample implementation described in the above gist.

Performance

Random arrays g1 and g2 of size 2^16 and a random challenge of 128 bits will be selected to test the speed of the implementation. The score of the submission will be the average performance across 100 trials.

In addition, all submissions will be manually reviewed by the prize committee.

The fastest existing implementation is here, which makes use of Pallas's curve-endomorphism (x, y) -> (base_endo_coeff * x, y), where base_endo_coeff = 0x2D33357CB532458ED3552A23A8554E5005270D29D19FC7D27B7FD22F0201B547

It also makes use of multithreading and batched-affine operations.

The endomorphism-based scalar multiplication is defined in section 6.2 of the Halo paper.

Hardware & Benchmarks

Implementations will be compiled to WASM and benchmarked on a consumer-laptop with at least 8 cores in Google Chrome.

Prize Allocation

The prize amount will be divided among the top two finishers according to the following proportions: 75% to winning implementation and 25% to second place. Prizes will be given out in good faith and at the sole discretion of the prize committee, which in this case consists solely of representatives designated by O(1) Labs.

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.

Questions

If there are any questions about this prize, please contact Brett Carter (@o1brett) on the ZPrize Discord server.