Not to be published before MoneroKon

Full Chain Membership Proofs

Solving one of Monero’s last privacy hurdles

Who Am I

- Luke Parker, Kayaba, or kayabaNerve
- Developer and informal cryptographer
- Lead Developer of Serai DEX
- MAGIC Monero Fund Committee Member
- Servant of the Monero Community

What are full chain membership proofs?

- Full chain membership proofs are a replacement for ring signatures
- They'd remove statistical analysis possible on inputs, solving the EAE/EABE/Overseer/tainted dust/etc attacks
- They are being proposed as part of Seraphis, not an alternative to
- Would remove needing a decoy selection algorithm, of which Monero's has had bugs multiple times

Curve Trees

- Curve Trees uses a pair of Bulletproofs to prove an output exists in some tree of outputs
- While Monero currently uses Bulletproofs to prove output amounts are valid, Bulletproofs can also prove arbitrary programs were correctly executed
- It expects a curve cycle to be in use, something Monero doesn’t currently have

Moving to a curve cycle

- With Seraphis, we will have to recreate all addresses
- This makes it a perfect time to switch curves
- Thanks to Chase, Orrù, Perrin, Zaverucha’s proof (2022), we can now do so extremely cheaply
- While we can implement curve trees without a cycle, it’d roughly halve the performance
- We’d also have to maintain four different curve libraries for the entire lifetime of Seraphis, under current discussions
- If we don’t switch curves with Seraphis, we’ll have to later or will be permanently so handicapped

So how do we actually implement this?

- Over the past month and a half, I’ve implemented all of these proofs
- Cleanly written, intended for hardening and auditing
- Ready to start discussing building around today

Performance

- Verification is batched, with the larger the batch the lower the time per proof
- Currently, ~100ms per proof in a batch of 10 for a set of 777 million outputs
- With an academic progression, it'd be just ~33ms (again, batch size equals 10)
- Grootle proofs, currently proposed for a ring of 128, are 3.7ms in a batch of 10
- Further optimizations are still available
- Performance of the node overall will be impacted

Security

- Is this moon math?
- How much of this has been formally proven?
- What hasn’t been formally proven?
- How should we ensure the security of this moving forward?

Next steps

- Fund formalization and proofs of a vector commitment scheme, the prior mentioned “academic progression”
- Fund formalization and proofs of Seraphis itself
- Get more developers familiar with the codebase and its concepts
- Discuss how Seraphis should move forward
- Work the codebase into something ready for production
- Eventually, hire auditors

Questions?