FINAL REPORT

ZERO-KNOWLEDGE PROOF OF

INNOCENCE IN CARDANO

Project #1300084

Name of the project: Designing an API for ZK-Snark proof verification in Aiken - Eryx

Project number: #1300084

Name of project manager: Agustin Franchella

Date project started: Jan 20, 2025

Date project completed: Sep 19, 2025

List of challenge KPIs and how the project addressed them

1. **Integration of Zero-Knowledge Proofs with Cardano smart contracts:**
   Aiken-zk successfully demonstrated the ability to integrate ZK proofs (from the Groth16 proving system) into Aiken validators, allowing developers to offload computation off-chain and validate results on-chain using proofs.
2. **Developer accessibility and tooling usability:**
   The tool provides a simple CLI (*aiken-zk new*, *aiken-zk build*, *aiken-zk prove*) and a tutorial for developers, significantly reducing the complexity of combining Circom, Snark.js, and Aiken in one transparent workflow.
3. **Cross-compatibility with ecosystem components:**
   Aiken-zk works with Circom, Snark.js, MeshJS, and Aiken, integrating across common developer tools, supporting both testing and deployment scenarios.

List of project KPIs and how the project addressed them

1. **Functional prototype released:**
   The compiler tool was implemented in Rust and successfully generates valid Aiken code with embedded ZK verification scripts.
2. **End-to-end proof workflow validated:**
   The process (from offchain definition to proof generation and verification) was fully tested through automatic tests and QA validation.
3. **Documentation and community engagement:**
   A detailed README and a Medium tutorial were published, guiding users through installation, usage, and example scenarios.

Key achievements (in particular around collaboration and engagement)

- Successfully bridged two communities: the Aiken smart contract developers and the ZK-proof developers using Circom.
- Created a clear pedagogical example via the public tutorial to foster community experimentation and adoption.

Key learnings

- Simplifying the user experience for ZK integration requires strong tooling automation; hiding cryptographic complexity is key to adoption.
- Consistency in Circom and Snark.js versions is critical—version mismatches were a frequent source of build failures early in development.
- Providing explicit developer feedback through CLI output improved debugging and reduced onboarding problems.

Next steps for the product or service developed

- Integrate Tx3 for deployment
- Show public values during proof creation
- Improve scope and reach of the tool. Aiming it to be a project management tool rather than an extension of Aiken.
- Mock verification key creation to make tests run lightning-fast and make them independent of the CIRCOM version used
- Add more functions/circuits (e.g. Blake2)
- Improve usability and capability of the tool:
    - Make redeemer variable name be definable by programmer
    - Remove Groth16 verifier from user generated file
    - Make generated aiken proof code be a library, hide the proofs along the mZKRedeemer for Meshjs during deploy
    - Improve importing of Circom templates from other files
    - Take the circuit for proof generation instead of the contract_with_offchain.ak file
    - Make inputs on inputs.json be automatically generated

Final thoughts/comments

In this grant we created a tool that simplifies the use of ZK proofs in Aiken.
We expect this tool to be useful for programmers who need privacy capabilities in their contracts but they are just entering the ZK community and want to avoid the inherent complexities of these kinds of tools.

Links to other relevant project sources or documents

- **GitHub repository:** https://github.com/eryxcoop/aiken-zk
- **Tutorial article:** https://medium.com/eryxcoop/aiken-zk-tutorial-d11b440a7d1a
- **Documentation and examples** in *README.md* within the repository

Close-out video

https://youtu.be/smll90J6PJE