PEACE Protocol

Decentralized On-Chain Data Encryption

Final Close-Out Report

(Project Catalyst Fund 14)

Project: Decentralized On-Chain Data Encryption

Public Repository: https://github.com/logical-mechanism/peace-protocol

Catalyst Milestones: https://milestones.projectcatalyst.io/projects/1400046

Close Out Video: https://youtu.be/Eabw-j2hjrg

---

1. Executive summary

This project delivered the PEACE (PRE-ECIES-AES-GCM Encryption) protocol, an open-source proof-of-concept for decentralized data encryption on the Cardano blockchain. It uses Proxy Re-Encryption (PRE), Elliptic Curve Integrated Encryption Scheme (ECIES), and Advanced Encryption Standard in Galois/Counter Mode (AES-GCM).

Across three completed milestones, the work progressed from: 1) **technical research and security analysis**, to 2) an **MVP implementation** spanning **on-chain Aiken smart contracts** and **off-chain Python/Bash tooling**, and finally to 3) a **comprehensive testing and validation phase**, including **Cardano pre-production (preprod) evidence** and reproducible test execution.

This report gives a reviewer-friendly summary of technical and functional progress, validation, findings, lessons, and links to artifacts.

---

2. Development timeline (high-level)

- **Milestone 1:** Research, protocol specification, threat model, and feasibility analysis
- **Milestone 2:** MVP implementation (on-chain + off-chain) and "happy path" execution
- **Milestone 3:** Expanded test coverage, reproducibility documentation, and preprod validation evidence
- **Final Milestone:** This report summarizing the full development process and results

---

3. Milestone-by-milestone outcomes and evidence

Milestone 1: Technical research report + security assessment

Primary objective:

Produce a report defining the protocol's cryptographic design and security within Cardano's on-chain context.

Delivered outputs:

- A detailed technical research report (research-paper style) analyzing ECIES, proxy re-encryption (PRE), and AES-GCM in the context of on-chain data encryption.
- Clear protocol explanation and a breakdown of cryptographic primitives used (ECIES, AES-GCM, PRE).
- Explicit security model and assumptions (trust model, what is protected, what is not).
- Threat analysis (including metadata leakage and limitations/risks).
- Performance and feasibility considerations for on-chain execution.
- Methodology write-up for reviewer traceability.

Evidence links:

- Repo: https://github.com/logical-mechanism/peace-protocol
- Technical report (PDF): https://github.com/logical-mechanism/peace-protocol/blob/main/documentation/technical_report.pdf
- Technical report (Markdown): https://github.com/logical-mechanism/peace-protocol/blob/main/documentation/technical_report.md
- Methodology: https://github.com/logical-mechanism/peace-protocol/blob/main/documentation/methodol
- Completion PR: https://github.com/logical-mechanism/peace-protocol/pull/2

Milestone 2: MVP implementation (Aiken on-chain + Python/Bash off-chain)

Primary objective:

Deliver an MVP to demonstrate the protocol flow with scripts and clear documentation.

Delivered outputs:

- MVP implementation integrating **on-chain Aiken validators** with **off-chain Python/Bash tooling**.
- Offered guided happy path setup and usage.
- Documented build and test workflow for on-chain contracts.

- Developed off-chain tooling and artifact production.
- On-chain tests confirmed the correctness of the encrypted state.
- Enabled reproducibility with a single-command test runner.
- Produced logs confirming successful path execution.

Evidence links:

- Repo: https://github.com/logical-mechanism/peace-protocol
- MVP guide: https://github.com/logical-mechanism/peace-protocol/blob/main/app/README.md
- Happy path setup: https://github.com/logical-mechanism/peace-protocol/blob/main/app/README.md#happy-path-setup
- Happy path usage: https://github.com/logical-mechanism/peace-protocol/blob/main/app/README.md#happy-path-usage
- One-command tests: https://github.com/logical-mechanism/peace-protocol/blob/main/app/run_tests.sh
- Commands & reproduction notes: https://github.com/logical-mechanism/peace-protocol/blob/main/app/commands/README.md
- Completion PR: https://github.com/logical-mechanism/peace-protocol/pull/7

Milestone 3: Comprehensive testing + Cardano preprod validation

Primary objective:

Increase reliability and reviewer confidence by running all tests and providing end-to-end preprod results.

Delivered outputs:

- Comprehensive testing phase covering as many scenarios as possible.
- Maintained a reliable Aiken build throughout.
- Ran all tests with documented reproducible steps.
- Cardano pre-production (**preprod**) evidence demonstrating end-to-end operation.
- Documented reproduction, execution, and verification steps.

Evidence links:

- Test plan / coverage summary: https://github.com/logical-mechanism/peace-protocol/blob/main/app/coverage.txt
- Preprod evidence (transaction hashes + verification notes):
  - **Hashes are intentionally kept in the feasibility document due to length:** https://github.com/logical-mechanism/peace-protocol/blob/main/app/feasibility.md
- Contracts directory: https://github.com/logical-mechanism/peace-protocol/tree/main/app/contracts
- Contract compile instructions: https://github.com/logical-mechanism/peace-protocol/tree/main/app/contracts#compiling

- Contract testing instructions: https://github.com/logical-mechanism/peace-protocol/tree/main/app/contracts#testing
- Commands directory: https://github.com/logical-mechanism/peace-protocol/tree/main/app/commands
- One-command tests: https://github.com/logical-mechanism/peace-protocol/blob/main/app/run_tests.sh
- Completion PR: https://github.com/logical-mechanism/peace-protocol/pull/8

---

4. Key achievements

Research to implementation continuity

The project maintained alignment from research through a working MVP to verification. The result is a set of aligned protocols, implementations, and tests.

Security posture captured early

Assumptions, threats, and limitations, including metadata risks, were documented early. This supports future hardening and audits.

Reproducibility treated as a deliverable

Artifacts such as documentation, scripts, test runner, logs, and evidence were created for each milestone, supporting objective review and reducing reliance on narrative claims.

---

5. Challenges encountered and how they shaped the work

1) **Finding something that would actually work on Cardano**

A key challenge was finding an approach that fit Cardano's on-chain limits while preserving strong cryptography.

2) **Reworking pairing mathematics to fit on-chain constraints**

The paper-level constructions involving pairings needed adaptation for practical on-chain use, since cryptographic pairings cannot be stored as data objects on-chain.

3) **Identifying and mitigating CCA-style attack surfaces early**

Potential chosen-ciphertext-style risks (where an attacker can trick the system into decrypting modified ciphertexts) were identified early and addressed using a combination of sigma-protocol-style structures, pairing relations, and a SNARK (succinct non-interactive argument of knowledge) where applicable.

4) **Learning the weaknesses and practical boundaries of re-encryption**

This work clarified where re-encryption is strong or brittle, and which use cases fit the threat model, leakage, and on-chain constraints.

---

6. Cost and performance observations (real-world implications)

A key conclusion from implementation and testing is that the protocol is expensive in its current state. Elliptic curve pairing operations are computationally expensive, and verifying SNARK proofs requires even more computing power.

For production, the protocol would likely need to use staged workflows, off-chain work, or split transactions to manage computation and cost.

---

7. Tooling and environment notes

The repository is designed to be reproducible using **current stable versions** of: - `aiken` - `cardano-cli` / `cardano-node`. - Python 3 toolchain and dependencies referenced by the repository

The repository and scripts remain usable across stable releases, given toolchain versioning differences. Reviewers can use the repository as the source for commands and outputs.

---

8. Reviewer reproduction flow (quick path)

The repository provides full reproduction guidance. A reviewer can follow this general flow:

1) Read the protocol overview and security model:

- https://github.com/logical-mechanism/peace-protocol/blob/main/documentation/technical_report.md
- https://github.com/logical-mechanism/peace-protocol/blob/main/documentation/technical_report.pdf

2) Follow the MVP instructions and run the happy path:

- https://github.com/logical-mechanism/peace-protocol/blob/main/app/README.md#happy-path-setup
- https://github.com/logical-mechanism/peace-protocol/blob/main/app/README.md#happy-path-usage

3) Execute the full test suite via the single command runner:

- https://github.com/logical-mechanism/peace-protocol/blob/main/app/run_tests.sh

4) Verify preprod evidence (transaction hashes + notes):

- https://github.com/logical-mechanism/peace-protocol/blob/main/app/feasibility.md

5) Review the test plan / coverage summary:

- https://github.com/logical-mechanism/peace-protocol/blob/main/app/coverage.txt

---

9. Evidence index

Core

- Repo: https://github.com/logical-mechanism/peace-protocol

Milestone 1

- Technical report (PDF): https://github.com/logical-mechanism/peace-protocol/blob/main/documentation/technical_report.pdf
- Technical report (MD): https://github.com/logical-mechanism/peace-protocol/blob/main/documentation/technical_report.md
- Methodology: https://github.com/logical-mechanism/peace-protocol/blob/main/documentation/methodol
- PR: https://github.com/logical-mechanism/peace-protocol/pull/2

Milestone 2

- MVP: https://github.com/logical-mechanism/peace-protocol/blob/main/app/README.md
- Tests: https://github.com/logical-mechanism/peace-protocol/blob/main/app/run_tests.sh
- Commands: https://github.com/logical-mechanism/peace-protocol/blob/main/app/commands/README
- PR: https://github.com/logical-mechanism/peace-protocol/pull/7

Milestone 3

- Coverage / test plan: https://github.com/logical-mechanism/peace-protocol/blob/main/app/coverage.txt
- Preprod evidence (tx hashes): https://github.com/logical-mechanism/peace-protocol/blob/main/app/feasibility.md
- Contracts: https://github.com/logical-mechanism/peace-protocol/tree/main/app/contracts
- PR: https://github.com/logical-mechanism/peace-protocol/pull/8

---

10. Close-out statement

All milestone outputs are delivered in the public repository, with reproducible documentation and proof artifacts. The completed work demonstrates a full development arc from technical research and security analysis, through MVP implementation, to comprehensive testing and preprod validation evidence.

This close-out report consolidates that record into a single reviewer-friendly summary, with direct links to all supporting proof-of-achievement materials.