National College of Ireland

Configuration Manual

Balancing Privacy and Compliance in Financial Transactions Using Zero-Knowledge Proofs

M.Sc Financial Technology

Lohit Uchil

Student ID:x23267895

School of Computing

National College of Ireland

Supervisor:Dr. Noel Cosgrave

National College of Ireland

MSc Project Submission Sheet

School of Computing

StudentName: …Lohit Uchil……

Student ID: ……2367895……..……

Programme : M.sc Fintech……

Module: ……MSc (Research) Practicum/Internship……

Supervisor: ……Dr. Noel Cosgrave……..……

Submission Due Date: ……11-08-2025……

Project Title: ……Balancing Privacy and Compliance in Financial Transactions Using Zero-Knowledge Proofs……

Word Count: …… **Page Count:** ……..……..

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: ……Lohit Uchil……

Date: ……11-08-2025……

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only

Signature:	Lohit Uchil
Date:	11-08-2025
Penalty Applied (if applicable):	

Configuration Manual

Lohit Uchil
Student ID: 23267895

Section 1: Execution Environment

This section details the primary software and operating system used to conduct the research, ensuring the foundational environment can be replicated.

1.1. Execution Environment

- Operating System: Ubuntu 22.04 LTS (via Windows Subsystem for Linux 2)
- Programming Language: Python 3.12.x

Section 2: Required Python Packages

This section lists all essential Python packages and their versions, which are crucial for the proper execution of the project's scripts and analyses. A requirements.txt file is provided with the project for automated installation within a virtual environment.

2.1 Required Python Packages

- py_ecc==6.0.0: Utilized for elliptic curve operations (BN128 curve) in the conceptual ZKP benchmarks.
- hashlib: A standard library module (included with Python 3.12) used for SHA-256 hashing in the Merkle tree implementation.
- SimPy==4.0.1: The core discrete-event simulation framework used to model the financial network.
- pandas==2.0.3: Used for loading, processing, and analyzing the tabular data (.csv) generated by the project scripts.
- matplotlib==3.7.1 & seaborn==0.12.2: The visualization stack used to generate all statistical plots.
- Jupyterlab : The interactive environment for running the final analysis notebook.
- Psutil : Used to measure system resource usage during benchmarking.

Section 3 Comprehensive Replication Guide

This section provides a precise, step-by-step guide to fully reproduce the research, from initial environment setup to the generation of the final analysis. The process is divided into two main phases: A) Environment Setup (a one-time process) and B) Artifact Execution (the sequence of commands to generate results).

3.1. Prerequisites

- A host machine with Windows Subsystem for Linux (WSL 2) installed.
- The Ubuntu 22.04 LTS distribution installed from the Microsoft Store.
- Basic familiarity with the Linux command line.
- All project files (Python scripts, Jupyter Notebook, requirements.txt) must be located in a single root project directory.

3.2. Phase A: Environment Setup (One-Time Process)

This phase prepares your system with all the necessary software and libraries.
Install Python:
Open your Ubuntu terminal and ensure Python 3.12 and its package management tools are installed:

```
sudo apt update
sudo apt install python3.12 python3.12-venv python3-pip -y
```

Create and Prepare the Virtual Environment:
Navigate to your project's root directory. The following commands will create an isolated Python environment for this project to avoid conflicts.
Navigate to your project folder (e.g., py_zkp_simulation)

```
cd /path/to/your/project
```

Create the virtual environment named 'venv'

```
python3.12 -m venv venv
```

Install Required Packages:
Now, activate the environment and install all dependencies from the provided requirements.txt file.
Activate the environment - your prompt will now start with (venv)

```
source venv/bin/activate
```

Install all required packages into this environment

```
pip install -r requirements.txt
```

Your environment is now fully configured. For all subsequent steps, ensure your terminal prompt shows (venv) to indicate the virtual environment is active.

3.3. Phase B: Artifact Execution (Generating Results)

Run the following scripts in sequence from your project's root directory. Ensure your virtual environment is activated for every step.

Step 1: Generate Benchmark Data

- **Purpose:** This step creates the benchmark_data.csv file, which contains the empirical performance data that the simulation relies on. This data is gathered through conceptual prototyping and benchmarking of core ZKP functionalities, using a Python-native approach to ensure consistent and controlled performance parameters.
- **Command:**

```
python 1_zkp_prototypes.py
```

Step 2: Analyze Benchmarks

- **Purpose:** This step processes and analyzes the raw benchmark data generated in Step 1, likely preparing it for use in the subsequent simulation or for direct visualization.
- **Command:**

```
python 3_analyze_benchmarks.py
```

- **Expected Outcome:** This script will likely generate processed data files or intermediate analysis results that are then consumed by the simulation.

Step 3: Run the Scalability Simulation

- **Purpose:** This script uses the data from the previous steps to simulate the system under various loads and create the simulation_results.csv file. This discrete-event simulation, built with SimPy, models the financial network to analyze performance, scalability, and identify potential bottlenecks under various user loads and verifier capacities.
- **Command:**

```
python 4_zkp_financial_sim.py
```

Step 4: Perform Qualitative Validation

- **Purpose:** This script demonstrates the core privacy and compliance features of the architecture, serving as a qualitative, tangible proof-of-concept. It explicitly separates a "secret world" of private user data from a "public world" with a public compliance anchor and an anonymous transaction ledger.
- **Command:**

```
python 5_privacy_demo.py
```

- **Expected Outcome:** The terminal will display a detailed log showing:

    - **Privacy Preservation:** Successful transactions appearing on the public ledger with **concealed identities** of participating parties (e.g., tx_hash, amount, timestamp but no sender/receiver IDs).
    - **Compliance Enforcement:** Non-compliant transactions (e.g., from a user not on the whitelist) being **successfully identified and blocked** at the ZKP verification stage, before any business logic is executed. An example log entry would be [FAIL] Transaction blocked: Compliance check failed (Sender: False, Receiver: True)..

Step 5: Analyze and Visualize Results

- **Purpose:** This final step uses the Jupyter Notebook to perform statistical analysis and generate the plots based on the benchmark_data.csv and simulation_results.csv files.
- **Command:**

```
jupyter lab
```

- **Action**: Your web browser will open the JupyterLab interface. In the file browser on the left, double-click 7_final_analysis.ipynb to open it. From the menu at the top, click **Run > Run All Cells**.
- **Expected Outcome:** The notebook will display all data tables and plots, including throughput and latency analyses. The final visualizations (throughput_analysis.png, latency_analysis.png, etc.) will be saved to your project directory.

References

References should be formatted using APA or Harvard style as detailed in NCI Library Referencing Guide available at https://libguides.ncirl.ie/referencing
You can use a reference management system such as Zotero or Mendeley to cite in MS Word.

Ben-Sasson, E., Bentov, I., Horesh, Y. and Riabzev, M. (2018) *Scalable, transparent, and post-quantum secure computational integrity*. IACR Cryptology ePrint Archive. Available at: https://eprint.iacr.org/2018/046.pdf.

Ben-Sasson, E., Chiesa, A., Garman, C., Green, M., Miers, I., Tromer, E. and Virza, M. (2014) 'Zerocash: Decentralized Anonymous Payments from Bitcoin'. In: *IEEE Symposium on Security and Privacy*. IEEE. Available at: https://ieeexplore.ieee.org/document/6956581.

Bünz, B., Korf, S., Gabizon, A., Boneh, D. and Drake, J. (2020) 'Zether: Towards Privacy in a Smart Contract World'. In: *Financial Cryptography and Data Security*. Springer. Available at: https://link.springer.com/chapter/10.1007/978-3-030-51280-4_23.

Dwork, C. and Naor, M. (1993) 'Pricing via processing or combatting junk mail'. In: Stinson, D.R. (ed.) *Advances in Cryptology – CRYPTO' 92*. Lecture Notes in Computer Science, vol 740. Berlin: Springer, pp. 10-18. Available at: https://link.springer.com/chapter/10.1007/3-540-48071-4_10.

Goldwasser, S., Micali, S. and Rackoff, C. (1985) 'The knowledge complexity of interactive proof systems'. *SIAM Journal on Computing*, 18(1), pp.186–208. Available at: https://people.csail.mit.edu/silvio/Selected%20Scientific%20Papers/Proof%20Systems/The_Knowledge_Complexity_Of_Interactive_Proof_Systems.pdf.

Houben, R. and Snyers, A. (2018) *Cryptocurrencies and blockchain: Legal context and implications for financial crime, money laundering and tax evasion*. European Parliament. Available at: https://www.europarl.europa.eu/cmsdata/150761/TAX3%20Study%20on%20cryptocurrencies%20and%20blockchain.pdf.

Miers, I., Garman, C., Green, M. and Rubin, A.D. (2013) 'Zerocoin: Anonymous distributed e-cash from Bitcoin'. In: *IEEE Symposium on Security and Privacy*. IEEE. pp. 397-411. Available at: https://www.ieee-security.org/TC/SP2013/papers/4977a397.pdf.

Narayanan, A., Bonneau, J., Felten, E., Miller, A. and Goldfeder, S. (2016) *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*. Princeton, NJ: Princeton University Press. Available at: https://press.princeton.edu/books/hardcover/9780691171692/bitcoin-and-cryptocurrency-technologies .

Ogungbemi, O.S. (2025) 'The Role of Zero-Knowledge Proofs in Blockchain-Based Property Transactions to Ensure Data Privacy and Compliance with UK Regulations'. *Journal of Engineering Research and Reports*, 27(3), pp. 414–435. Available at: https://journaljerr.com/index.php/JERR/article/view/1443.