National College of Ireland

Configuration Manual

MSc Research Project

Msc Cloud Computing

Ikenna Ughanze

Student ID: 23384069

School of Computing

National College of Ireland

Supervisor: Punit Gupta

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Ikenna Ughanze……
Student ID:	……23384069……
Programme:	Msc Cloud computing …… **Year:** ……2024/2025……
Module:	Research Project ……
Lecturer:	Punit Gupta ……
Submission Due Date:	13-09-2025 ……
Project Title:	Privacy-Preserving Public Auditing of Outsourced Cloud Data with Zero Knowledge Proofs ……
Word Count:	167 …… **Page Count: 3**……

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:	Ikenna Ughanze ……
Date:	10-08-2025 ……

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:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Ikenna Ughanze

Student ID: 23384069

1 Introduction

A cloud-based zero-knowledge data integrity audit system that uses ZK-STARK proofs to verify data hasn't been tampered with, without revealing sensitive information.

This system enables organizations to audit data outsourced to the cloud with mathematical certainty while preserving complete privacy. It uses:

- Zero-Knowledge STARK Proofs for privacy-preserving verification
- Cryptographically Secure Random Sampling for 95% confidence corruption detection
- Cost-Effective Auditing by sampling only a fraction of total data

2 Prerequisites

You will need this programming languages and packages installed on your local system

- Python 3.8+ with pip
- Node.js 16+ with npm
- Rust 1.70+ with nightly toolchain

To install rust on Linux/Mac or windows

Linux/macOS:

```bash
curl --proto '=https' --tlsv1.2 -sSf https://sh.rustup.rs | sh
```

Code: “curl --proto '=https' --tlsv1.2 -sSf https://sh.rustup.rs | sh”

Windows:

1. Go to https://rustup.rs/
2. Download and run `rustup-init.exe`
3. Follow the prompts

After installation:

1. **Restart your terminal** or run:

```bash
source ~/.cargo/env
```

2. **Verify it worked:**

```bash
rustc --version
cargo --version
```

3 Steps to run the Audit application on your local system

Step 1: Clone and Setup (clone the github repo)

Code: Git clone https://github.com/Ughanze23/thesis-project.git

Step 2: Install Dependencies

Python Backend:

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

Frontend:

```
cd frontend
npm install
cd ..
```

Rust Verification System:

```
cd verification-rs
cargo build --release
cd ..
```

Note: The requirement.txt file is found in the root of the directory.

Step 3: run the application

1. **Start the Backend Server:**

```
python3 fastapi-server.py
```

The API will be available at `http://localhost:8000`

2. **Start the Frontend (in a new terminal):**

```
cd frontend
npm start
```

The web interface will open at `http://localhost:3000`

3. **Use the Application:**
    - Upload CSV files through the web interface
    - Configure audit parameters (confidence level, corruption rate)
    - Monitor real-time audit progress
    - View verification results

This includes:

- Sample dataset generation
- Data ingestion and block creation
- Random block selection with 95% confidence
- Zero-knowledge STARK proof generation/verification
- Tampering detection testing
- Performance analysis

- Upload the sample-financial data found in the directory to test the audit framework.

References

https://github.com/facebook/winterfell