National College of Ireland

Configuration Manual

MSc Research Project
MSc Cybersecurity

Venkata Nikhil Tata
Student ID: x23263245

School of Computing
National College of Ireland

Supervisor: Khadija Hafeez

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Venkata Nikhil Tata
Student ID:	x23263245
Programme:	MSc Cybersecurity **Year:** 2025
Module:	MSc Practicum/Internship part 2
Lecturer:	Khadija Hafeez
Submission Due Date:	29-01-2025
Project Title:	Enhancing Peer-to-Peer Security with a Two-Stage Blockchain Model: Mitigating Sybil and 51% Attacks
Word Count:	802 **Page Count:** 04

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: Venkata Nikhil Tata

Date: 29-01-2025

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

Venkata Nikhil Tata
Student ID: x23263245

1 Introduction

Purpose of the Manual:
Step-by-step instructions will be provided as part of this configuration manual for setting up and configuring the environment, which is essential to implement the Two-Stage Blockchain Model in NS-3 simulator. The main aim of this setup is to simulate a secure peer-to-peer network and to assess the proposed mechanisms that are meant to mitigate Sybil and 51% attacks.

2 System Requirements

Hardware Requirements

- **Minimum CPU:** 2 cores
- **Minimum RAM:** 4 GB
- **Disk Space:** 15 GB of free space for software installation and data storage.

Software Requirements:

- **Operating System:** Ubuntu 20.04 or later
- **NS-3:** Stable Version
- **SQLite:** Storing node information and transaction data
- **Python3:** For scripting and automation
- **OpenSSL:** For encryption and public key infrastructure (PKI)
- **Development Tools**:
    - **GCC/G++ Compiler** (version 9.0 or later)
    - **CMake** (for NS-3 configuration)

3 Setting Up NS-3

Installing NS-3:

- As a first stage update the systems package index:
    - sudo apt update
- Install the required packages or dependencies (*ns-3 manual: 32.2 NetAnim*, no date)
    - sudo apt install g++ cmake sqlite3 libsqlite3-dev build-essential python3 python3-pip
- Download the NS-3 repository (nsnam, no date)
    - Download the stable version from nsnam.org/releases/ns-3.34/
    - Example: wget https://www.nsnam.org/releases/ns-allinone-3.34.tar.bz2
- Extract the downloaded file
    - tar xjf ns-allinone-3.34.tar.bz2
    - cd ns-allinone-3.34/ns-3.34/
- Build NS-3:

    - ./build.py --enable-examples –enable-tests
    - ./waf configure
    - ./waf build
- Execute the below command in the ns-3.34 folder to check if the installation was successful or not (Agrawal, 2023)
    - ./waf –run hello-simulator

4 Project Setup

- Add the project code files
    - Copy the project code files to the scratch/ directory of NS-3:
        - cp two_stage_p2p_model.cc /home/user/ns-allinone-3.34/ns-3.34/scratch/
- Update Database Configuration
    - Make sure SQLite3 is properly configured (*SQLite Download Page*, no date)
        - Verify the initialization logic in the code:
            - sqlite3 *db;
            - sqlite3_open("user_nodes.db", &db);
    - Also ensure that the database directory is in the project directory
- Configure & Build the project
    - ./waf configure
    - ./waf build
- Run the Simulation
    - ./waf –run scratch/two_stage_p2p_model.cc

5 Simulation Parameters

- **Node Configuration**
    - **Number of nodes:** Create number of nodes for a user in the simulation with the userId.
    - **Node Communication:** Point-to-point links with defined bandwidth and delay.
- **Simulation Time**
    - If needed adjust the simulation time from the two_stage_p2p_model.cc file
        - Simulator::Stop(Seconds(20.0));

6 Visualization

- After executing the code file successfully, a .xml file will be created in the ns-3.34 directory.
- To visualize this file run NetAnim
    - ./netanim
- Open the XML file which is generated as part of the simulation.

- Usually, it is in the scratch directory.

7 Troubleshooting

- Common issues that I have encountered as part of building this project.
  - **Socket binding errors:** Ensure correct port and interface settings
  - **Database Connection Errors:** Verify if the SQLite3 installation and database file paths.
- If any changes are made in the project, it is always recommended to reconfigure and rebuild
  - ./waf clean
  - ./waf configure
  - ./waf build
- For debugging use NS-3 logs for detailed insights
  - NS_LOG_INFO("Message for debugging")

8 High-Level Code Flow for this project

1. **Initialization phase:**
   a. Database Initialization: In this project I have used SQLite to store the user information like userId, public key of the user, count of the nodes, transactions and merkle roots.
   InitializeDatabase();
   b. Generating Key pair: Using ECDSA a key pairs are generated for the users.
   GenerateKeyPair("UserId");
2. **User and node management**
   a. Users are authenticated by their UserId. If there's no user in the database a new user will be created with the userId and key pair for that userId.
   b. Also user is allowed to create only 4 nodes. If he exceeds creating 4 nodes the simulation will not allow him to create a new node.
   IncrementNodeCount(userId);
   c. After creating nodes for user, connections will be created to all the nodes in the network.
   CreateConnections(userId);
3. **Network Setup**
   a. Installation of Network stack: Internet stack will be installed to each node after creation for the communication and unique IP addresses will be assigned to each node.
   InternetStackHelper stack; stack.Install(newNode);
4. **Transaction Management**
   a. Transactions are scheduled dynamically between the users. Both inter and intra transactions are scheduled.
   b. Each transaction includes:
      i. **Signing** using ECDSA.
      ii. **Encryption** using AES-256.
      iii. **Storage** in the database.
      ScheduleTransactionEvents();

c. Broadcast Transactions: Now the transaction will be broadcasted to the other nodes.
   BroadcastTransaction(serializedTransaction);
d. Merkle root generation: After transactions are stored in the database, a Merkle root is calculated to maintain the integrity.
   GenerateMerkleRoot();
   ValidateMerkleRoot();

5. **Network monitoring and Logging**
   a. Enabled packet logging sent, received and dropped.
      EnablePacketLogging(entry.second.Get(i));
   b. Flow Monitoring: Using flow monitor captured the metrics such as Throughput, Latency and Packet loss.
      EnableFlowMonitoring();
      AnalyzeFlowMetrics();
6. **Simulation Execution**
   a. The simulation will be run, and the transactions will be validated at the end.
      Simulator::Run();
      VerifyStoredTransactions();
7. **Cleanup**
   a. At the end the database will be closed, and simulation will be destroyed.
      CloseDatabase();
      Simulator::Destroy();

References

Agrawal, A.K. (2023) 'Installing ns-3.34 in Ubuntu 18.04.6 LTS version', *Medium*, 7 August. Available at: https://adnitr.medium.com/installing-ns-3-34-in-ubuntu-22-04-02-lts-version-2bb43c48683b (Accessed: 3 December 2024).

ns-3 manual: 32.2 NetAnim (no date). Available at: https://www.nsnam.org/docs/release/3.7/manual/manual_113.html (Accessed: 3 December 2024).

nsnam (no date) *Documentation*, *ns-3*. Available at: https://www.nsnam.org/documentation/ (Accessed: 3 December 2024).

SQLite Download Page (no date). Available at: https://www.sqlite.org/download.html (Accessed: 3 December 2024).