INFSCI 2750: Cloud Computing

Mini project 3

**Group17**

*Abhigyan Kishor (abk171)*

*Quinn K Wolter (QuinnKWolter)*

*Saleh Almutairi (saa451)*

April 23, 2023

**Objective:**

In this project, we have implemented the ODB described in the following figure.

A diagram of a diagram of a blockchain

Description automatically generated

# Part 1:

# You will need to prepare your own key value data set and submit all your implementations in Python.

# For the purpose of this project, we have prepared a dataset consisting of the following five key-value pairs:

# k1: v1

# k2: v2

# k3: v3

# k4: v4

# k5: v5

# These key-value pairs are used to build a Merkle Hash Tree, which is essential for the verification process.

# Part 2:

# You will need to show the following two verification results:

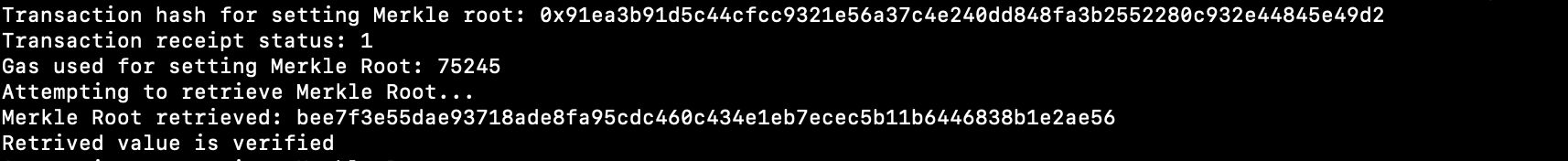
# We have modified the blockchain.py file while we were getting “invalid opcode” error before you made announcement on canvas. The detail explanation of our modifications and how we tackle this error is given at the end.

**No Attack:**

In this scenario, do not run your MC and show your verification result, and the system's integrity verification process is demonstrated under normal operating conditions.

**Verification Process:**

1. The Query Client requests a value for a specific key.
2. The Service Provider returns the value and the corresponding Merkle proof.
3. The Query Client verifies the Merkle proof against the Merkle root stored on the Ethereum blockchain.



**Figure 1:** Result of ’No Attack’ scenario

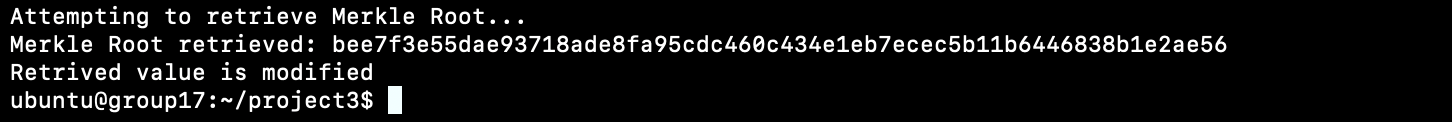
**Attack has happened:**

In this scenario, you will need to run your MC first, followed by performing queries as well as

verification, and showing your verification result.

**Steps:**

1. The Malicious Client modifies the value of a specific key in the database.
2. The Query Client requests the altered key's value and receives the tampered data along with a Merkle proof.
3. The Query Client attempts to verify the received data using the Merkle root on the blockchain.



**Figure 2:** Result of ’Attack has happened’ scenario

# Part 3:

# You will need to submit your table stored in your Cassandra by showing a screen- shot

**A screenshot of a computer

Description automatically generated**

**Figure 3:** Table stored in Our Cassandra