Mini Project 3 Report
INFSCI 2750: Cloud Computing

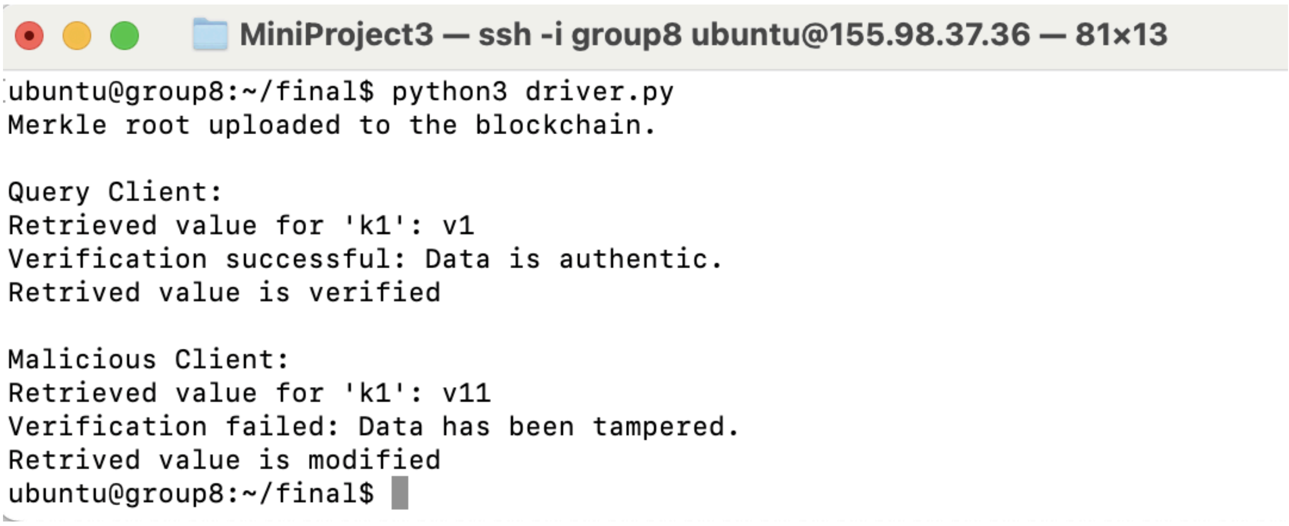
Group 8: Aishwarya Bhargava**,** Anusha Shiva Kumar**,** Harshitha Batta

Overview:

In this project, a proof-of-concept implementation of a Blockchain-assisted Verifiable Cassandra has been implemented, consisting of the following key classes:

- **Data Owner (DO):** DO takes charge of preparing a set of key-value data, and building a local MHT over such data.
- **Database Service Provider (SP):** SP serves as a server program running a Cassandra database. After getting the data from DO, SP will interact with Cassandra to store such data in terms of a table.
- **Ethereum Blockchain:** You will not need to implement anything regarding this part. We will provide relevant codes.
- **Query Client (C):** C will be able to issue query requests to SP. In your implementation, you will need to support key-value queries only. Also, C will be able to verify the resultant queries on his/her side by adopting MHT.
- **Malicious Client (MC):** MC will serve as an adversary to tamper with some data stored in the Cassandra running in SP

The below screenshot shows the "No Attack" and "Attack" Scenarios:

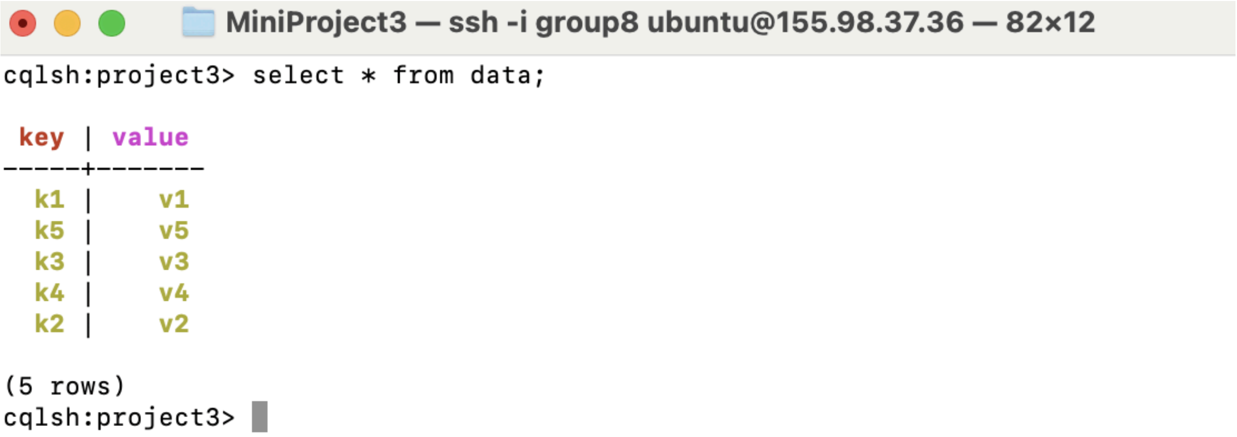


```
ubuntu@group8:~/final$ python3 driver.py
Merkle root uploaded to the blockchain.

Query Client:
Retrieved value for 'k1': v1
Verification successful: Data is authentic.
Retrived value is verified

Malicious Client:
Retrieved value for 'k1': v11
Verification failed: Data has been tampered.
Retrived value is modified
ubuntu@group8:~/final$
```

The below screenshot shows the data that had been added to the database before it was tampered with:

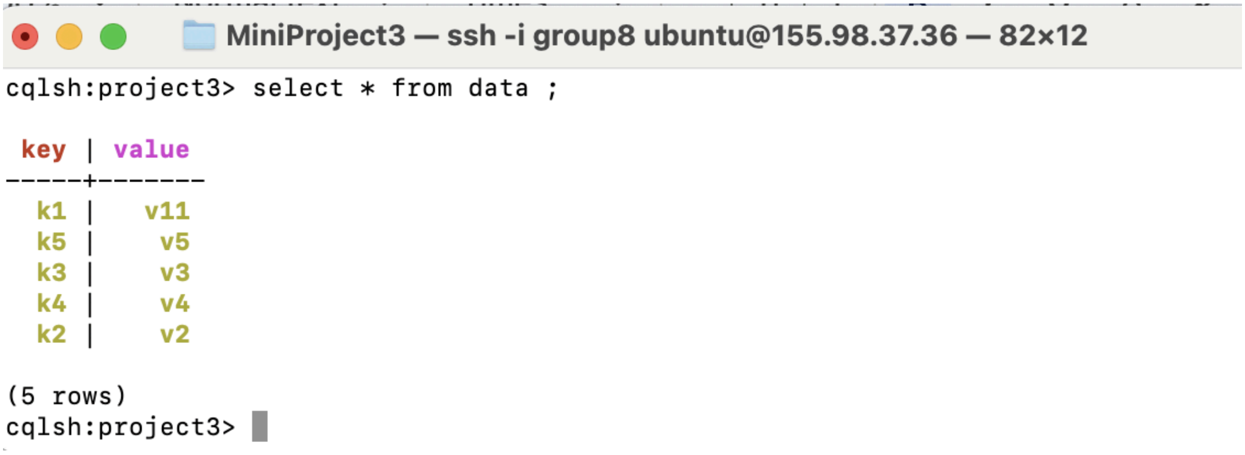


```
cqlsh:project3> select * from data;

 key | value
-----+-------
  k1 |    v1
  k5 |    v5
  k3 |    v3
  k4 |    v4
  k2 |    v2

(5 rows)
cqlsh:project3>
```

The below screenshot shows how the data has been tampered with in the database:

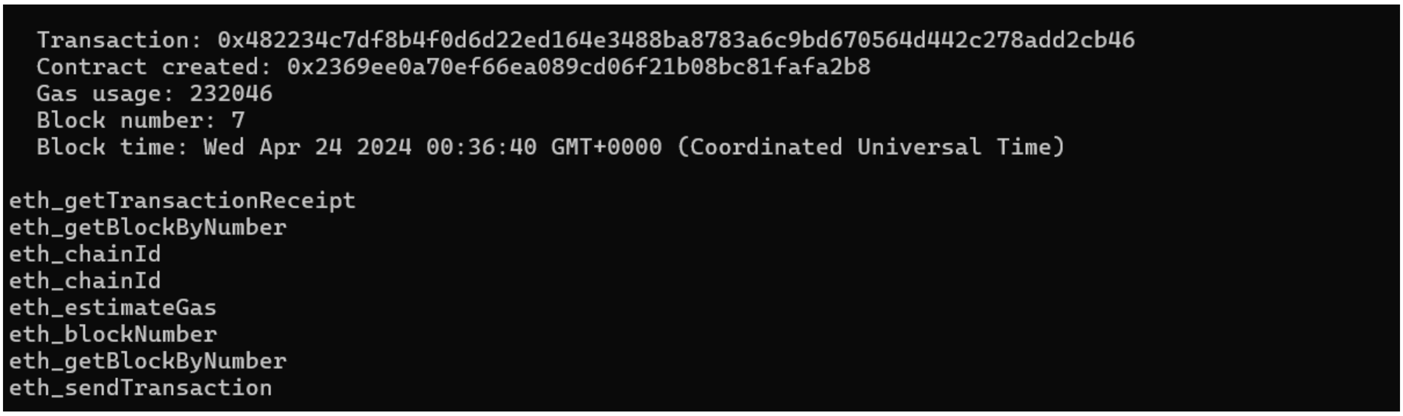


```
cqlsh:project3> select * from data ;

 key | value
-----+-------
  k1 |   v11
  k5 |    v5
  k3 |    v3
  k4 |    v4
  k2 |    v2

(5 rows)
cqlsh:project3>
```

Blockchain Transaction:



```
  Transaction: 0x482234c7df8b4f0d6d22ed164e3488ba8783a6c9bd670564d442c278add2cb46
  Contract created: 0x2369ee0a70ef66ea089cd06f21b08bc81fafa2b8
  Gas usage: 232046
  Block number: 7
  Block time: Wed Apr 24 2024 00:36:40 GMT+0000 (Coordinated Universal Time)

eth_getTransactionReceipt
eth_getBlockByNumber
eth_chainId
eth_chainId
eth_estimateGas
eth_blockNumber
eth_getBlockByNumber
eth_sendTransaction
```

REFERENCES:

[1] Ethereum white paper. https://ethereum.org/en/whitepaper/, [Online].
[2] Merkle tree. https://en.wikipedia.org/wiki/Merkle_tree, [Online].
[3] Andreas M Antonopoulos and Gavin Wood. Mastering Ethereum: building smart contracts and dapps. O’Reilly Media, 2018.
[4] Ralph C Merkle. A digital signature based on a conventional encryption function. In Advances in Cryptology—CRYPTO’87: Proceedings 7, pages 369–378. Springer, 1988.