Blockchain-Assisted Verifiable Cassandra

Introduction:

Outsourced Database Model (ODB) is a paradigm in cloud computing where data owners (DOs) delegate their data to a database service provider (SP) for efficient cloud services. However, the untrusted nature of SPs poses risks to data integrity. To address this, we propose a solution using Merkle Trees (MHT) and the Ethereum blockchain to authenticate queried data from SP. This report presents a proof-of-concept implementation of this solution.

Implementation:

- *Data Owner (DO):* Prepares key-value data set and builds a local MHT over the data.
- *Database Service Provider (SP):* Runs a Cassandra database and stores data received from DO in a table.
- *Ethereum Blockchain:* Stores the Merkle root of the MHT for data integrity verification.
- *Query Client (C):* Issues key-value queries to SP and verifies query results using MHT.

Verification Results:

- *No Attack Scenario:* Verification result without any malicious modifications, confirming data integrity.
- *Attack Scenario:* Verification after running a Malicious Client (MC) that tampers with data. Shows detection of attack through invalidation of modified data set.

Set-01:

```
ubuntu@group4:~/project3_copy$ python3 driver.py
Merkle tree built.
Data inserted to server.
Merkle root uploaded to blockchain: 84a085ed5184f652659a258b74457c8c26c27bb0a3a5c85c79a7dc57400c889a
Merkle root from blockchain: 84a085ed5184f652659a258b74457c8c26c27bb0a3a5c85c79a7dc57400c889a
Verification result: verified
Retrieved value is verified
Data for key k1 modified to v11.
Verification result: not verified
Retrieved value is modified
ubuntu@group4:~/project3_copy$
```

Set-02:

```
ubuntu@group4:~/project3_copy$ python3 driver.py
Merkle tree built.
Data inserted to server.
Merkle root uploaded to blockchain: 179eeadc0f029de9bb9387504cc0b76083e280b95e71dc3a0d0d1aa3473e2b57
Merkle root from blockchain: 179eeadc0f029de9bb9387504cc0b76083e280b95e71dc3a0d0d1aa3473e2b57
Verification result: verified
Retrieved value is verified
Data for key k1 modified to v20.
Verification result: not verified
Retrieved value is modified
ubuntu@group4:~/project3_copy$
```

In the above output screenshots, initially the data is confirmed as untampered ("verified") after the Merkle tree is constructed and the root is uploaded to the blockchain. However, a subsequent modification to the data ("Data for key k1 modified to v20/v11") results in a failed verification ("not verified"), indicating that the system has detected an unauthorized change, suggesting an 'attack' on the data integrity. This output demonstrates the blockchain's mechanism for ensuring data consistency and detecting alterations.

Cassandra Table:

A screenshot of the table stored in Cassandra, containing the key-value data set.

Set-01

Set-02:

Conclusion:

The proof-of-concept implementation demonstrates the use of MHT and the Ethereum blockchain for query authentication in an outsourced database model. The solution provides a tamper-resistant context for data integrity verification, enhancing security in cloud computing environments.