Blockchain Assisted Verifiable Cassandra -SOLC

Approach:

A diagram of a diagram

Description automatically generated

Store data in Cassandra DB.

Generate a Merkle tree and store its root in the blockchain.

Use a simulated malicious user to edit the data in Cassandra DB.

Generate a new Merkle tree and obtain the new Merkle tree’s root.

Compare the new root with the root stored in the blockchain to validate the attack.

Main Components:

Python scripts for interacting with Cassandra and performing blockchain operations.

Solidity smart contract for storing and retrieving the Merkle root hash on the blockchain.

Functions for building Merkle trees, querying data from Cassandra, and validating Merkle proofs.

Key Functions:

build\_merkle\_tree(data): Builds a Merkle tree over the provided data.

store\_data\_in\_cassandra(data): Stores key-value pairs in a Cassandra database.

query\_value\_by\_key(key): Queries the value associated with a given key from the Cassandra table.

get\_merkle\_tree\_from\_cassandra(key\_index): Retrieves data from the Cassandra table and constructs a Merkle tree over its values.

validate\_merkle\_proof(value, merkle\_proof, merkle\_root): Validates a value using its Merkle proof and Merkle root hash.