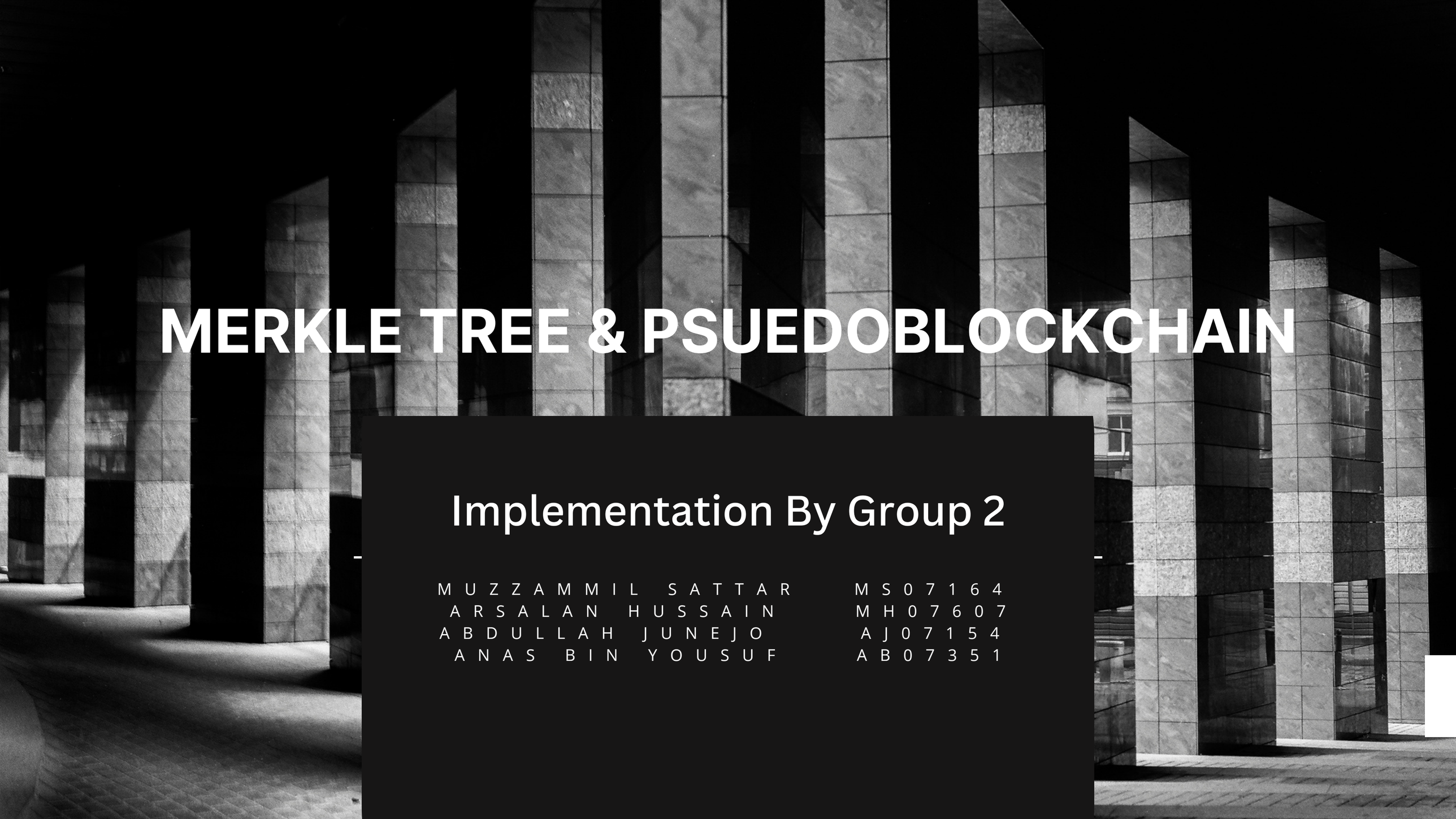


MERKLE TREE & PSUEDOBLOCKCHAIN

Implementation By Group 2

MUZZAMMIL SATTAR	MS07164
ARSALAN HUSSAIN	MH07607
ABDULLAH JUNEJO	AJ07154
ANAS BIN YOUSUF	AB07351

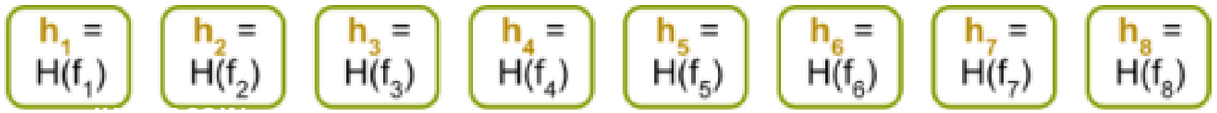


Figure 1: The Zeroth Layer

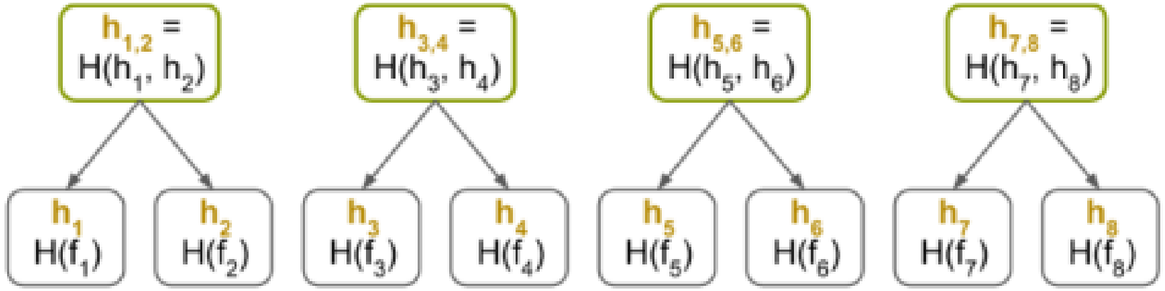


Figure 2: The First Layer

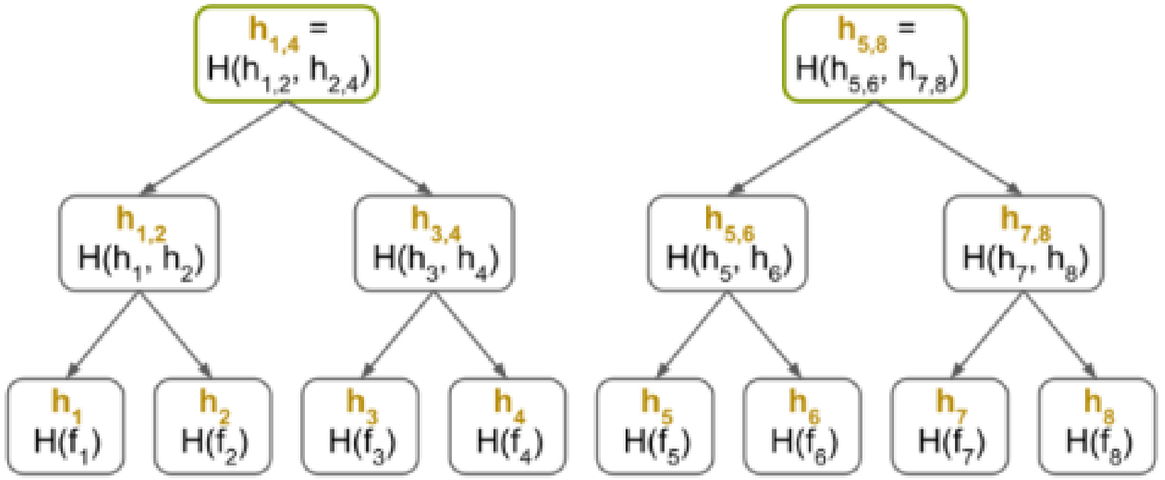


Figure 3: The Second Layer

INTRODUCTION

Merkle Tree is an encryption data structure that uses binary tree of hash values. Hashing starts from leaves of the Trees and this continues recursively to produce parents and ultimately the root hash/node.

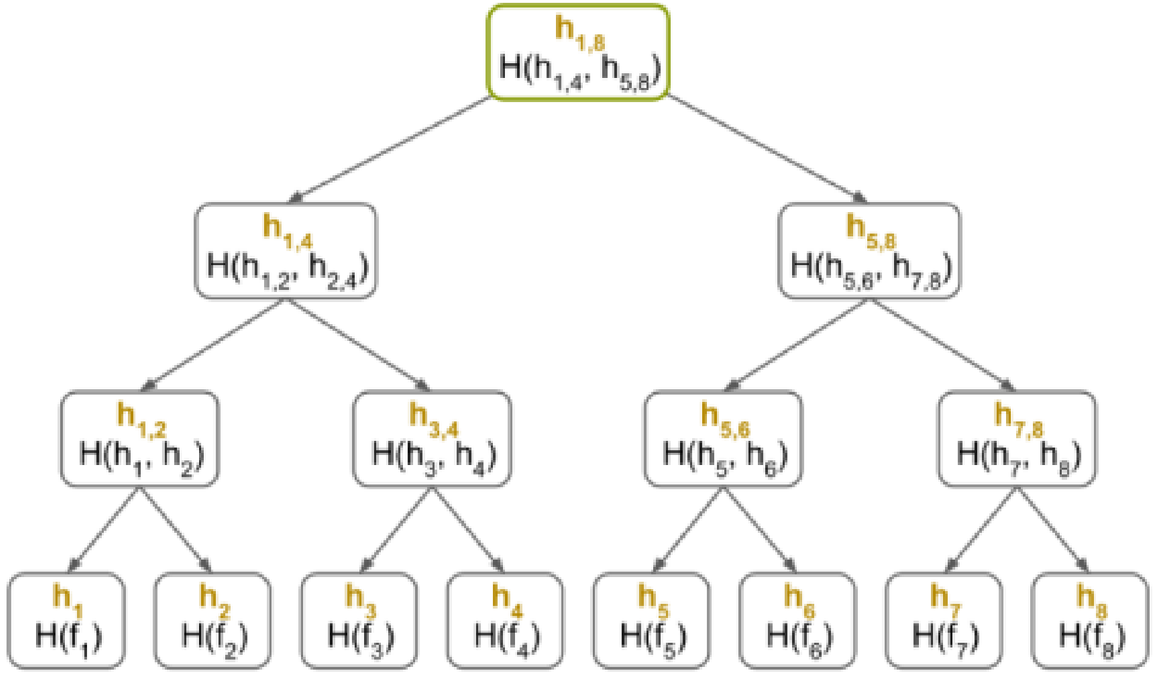
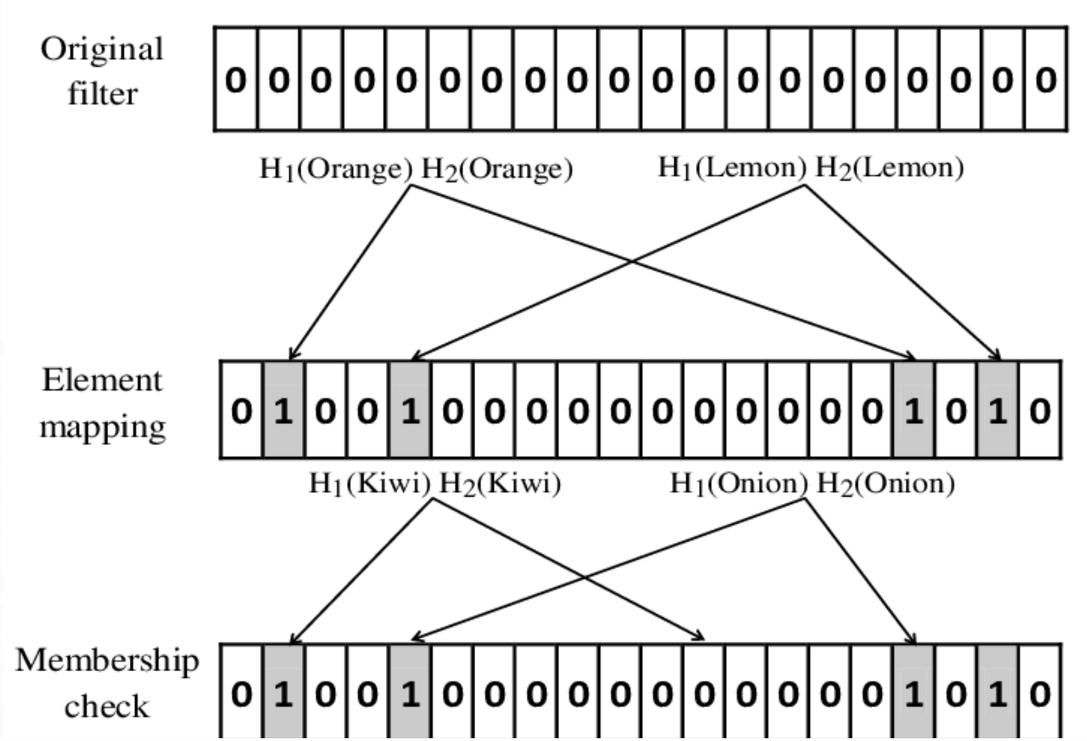


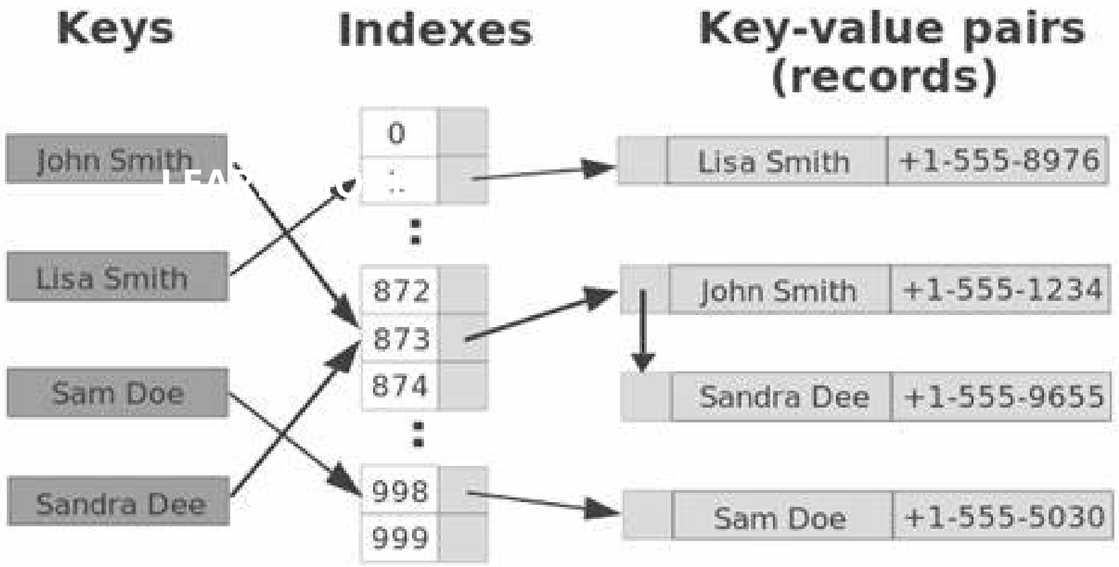
Figure 4: The Final Layer and Root Hash

ALTERNATIVES FOR MERKLE TREE

- **Bloom filter**
  O(K)

- **Hash List**
  O(N)





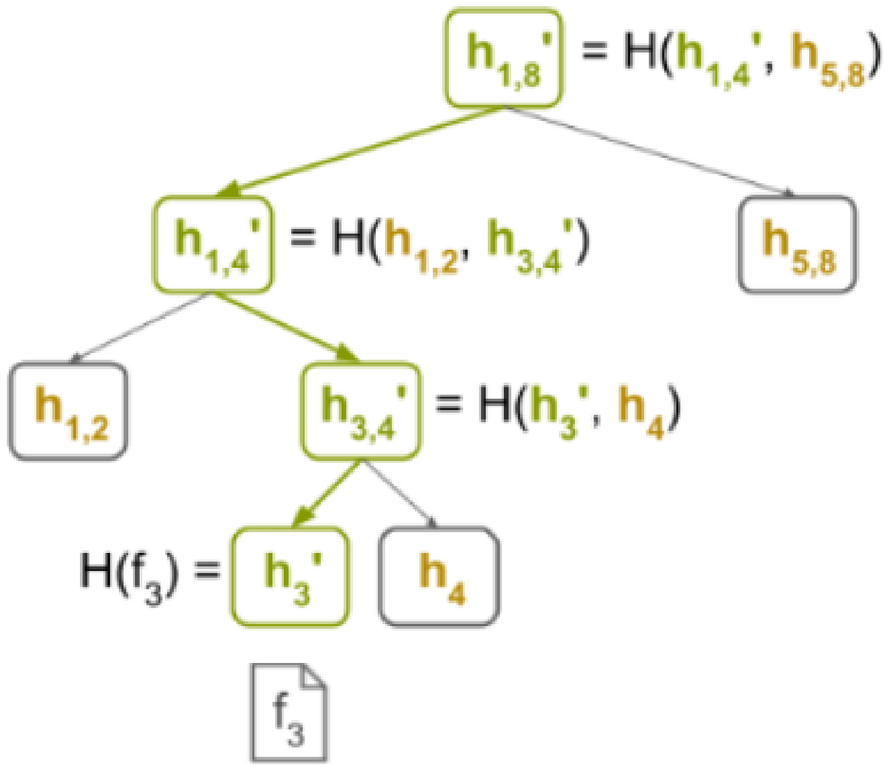


Figure 5: Merkle Proof

FURTHER THEORY

FUNCTIONALITY/MOTIVATION/STRENGTHS

The Merkle Tree has two key functions:

- Encryption: This is where the Merkle Tree (or Block) is created using the decided data/documents and collision-resistant hash function.
- Verification: This is where we use Merkle Proofs to efficiently check if a given data/document belongs to the Tree and also to verify (through hashing) if the given data/document has been altered or corrupted.
- O(logn) time complexity for add, verify_inclusion.
- **MERKLE PROOF**

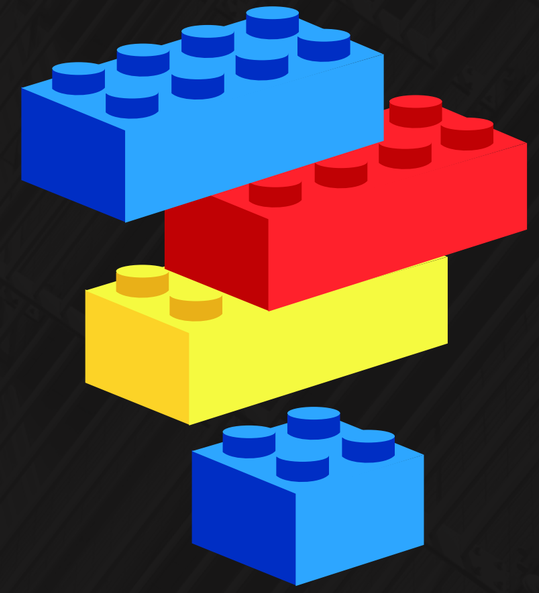
  Merkle Trees are efficient at verifying if a document has been changed as an altered document would produce a hash that is different from the root hash. If the document hasn't been changed then the same root hash would be produced. This is the bases of a Merkle Proof. Furthermore, only a set of hashes would be needed to prove a given leaf's membership in the tree as illustrated

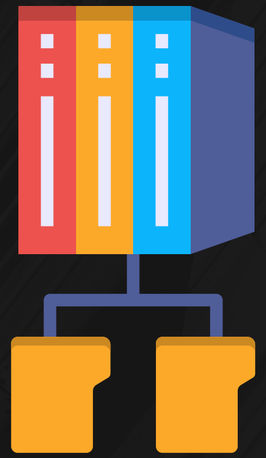
WEAKNESS

- Large Space Complexity O(n) for creation

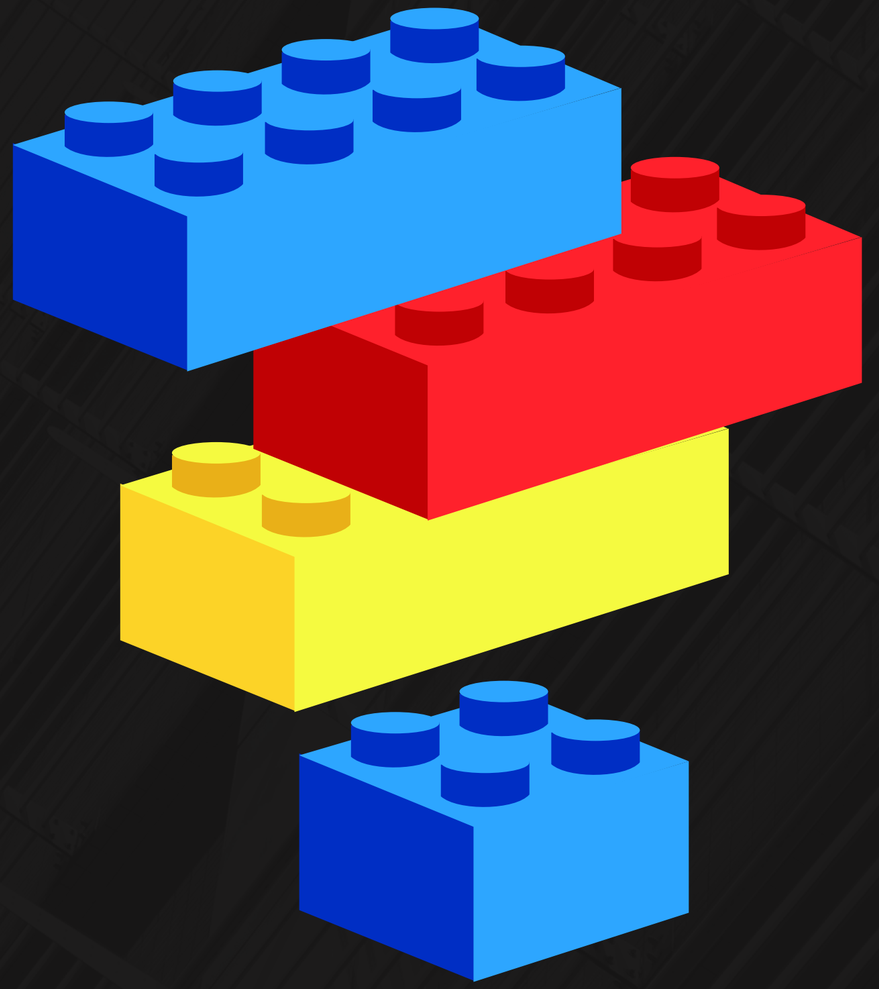
PSUEDOBLOCKCHAIN?

Our Implementation is very far from the actual block chain. It does not rely on many principles such as proof of work, and there is no Auto Correction.
Our Implementation is simply Linked List of Blocks. Each Block Class Object has Merkle Trees Constructed from Leaf Nodes which are documents in our case!



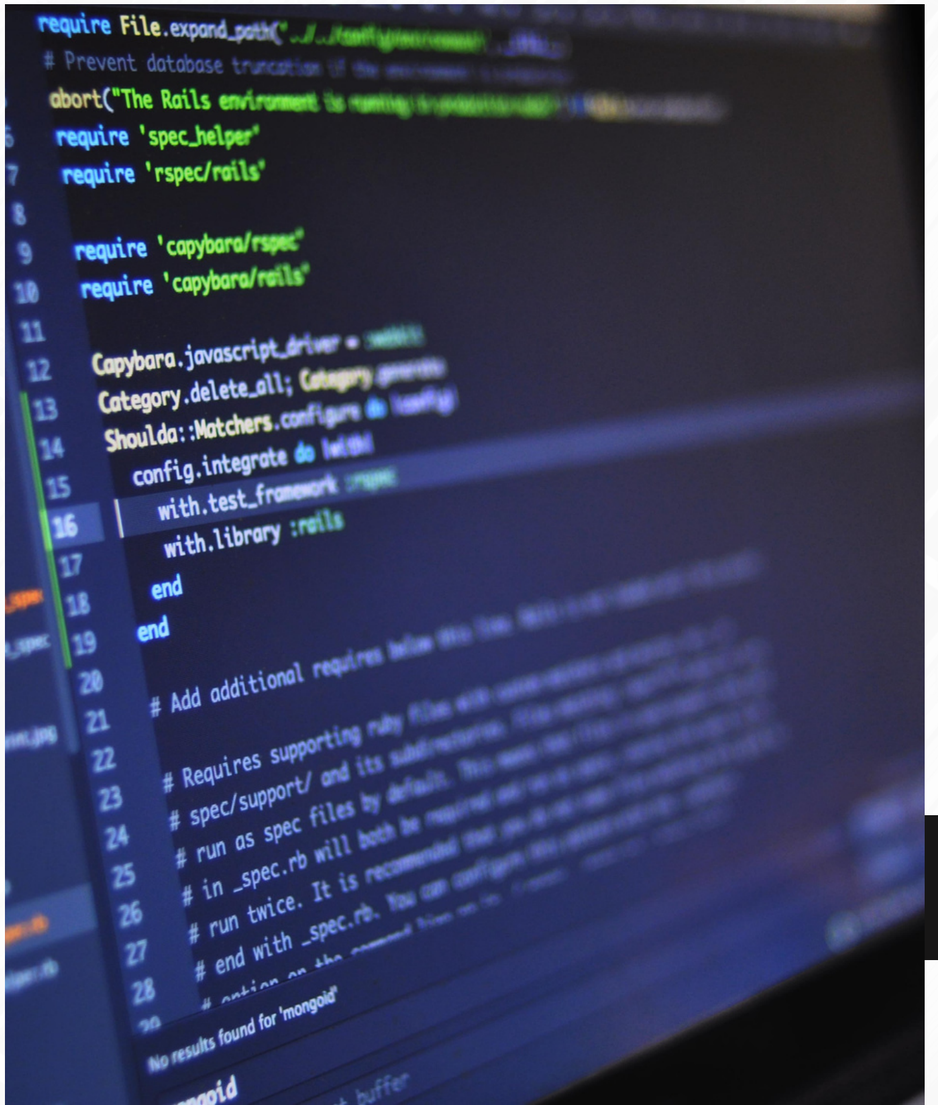


APPLICATION DEMO



REFERENCES/DATASETS

1. https://decentralizedthoughts.github.io/2020-12-22-what-is-a-merkle-tree/
2. https://www.kaggle.com/datasets/arshi2nlp/neobank-transactions
3. https://www.kaggle.com/datasets/demodatauk/full-banking-transaction-log-sample
4. https://www.kaggle.com/datasets/demodatauk/full-banking-transaction-log-sample
5. https://www.kaggle.com/datasets/apoorvwatsky/bank-transaction-data



THANK YOU

Any Questions?