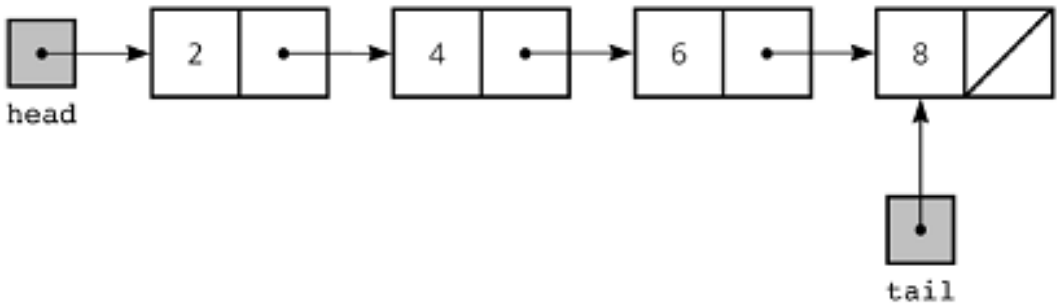

Welcome

More On Blockchain Cryptocurrency

Theory

What is linked list?

- A linked list is a linear data structure where each element is a separate object. Linked list elements are not stored at contiguous location; the elements are linked using pointers.
- Each node of a list is made up of two items - the data and a reference to the next node. The last node has a reference to null. The entry point into a linked list is called the head of the list. It should be noted that head is not a separate node, but the reference to the first node. If the list is empty then the head is a null reference.



Linked List Representation

Link list Example [C and Python]

```c
// Linked list example in C/C++
// A linked list node
struct ListNode
{
    int val;
    struct ListNode *next;
};
```

```python
# Linked list example in Python
# Node class
class Node:
    # Function to initialize the node object
    def __init__(self, v):
        self.val = v   # Assign value
        self.next = None   # Initialize next as null

# Linked List class
class ListNode:
    # Function to initialize the Linked List
    def __init__(self):
        self.head = None
```

Blockchain vs Linked List: Are They Both Same?

How Is Blockchain Similar to a Linked List?

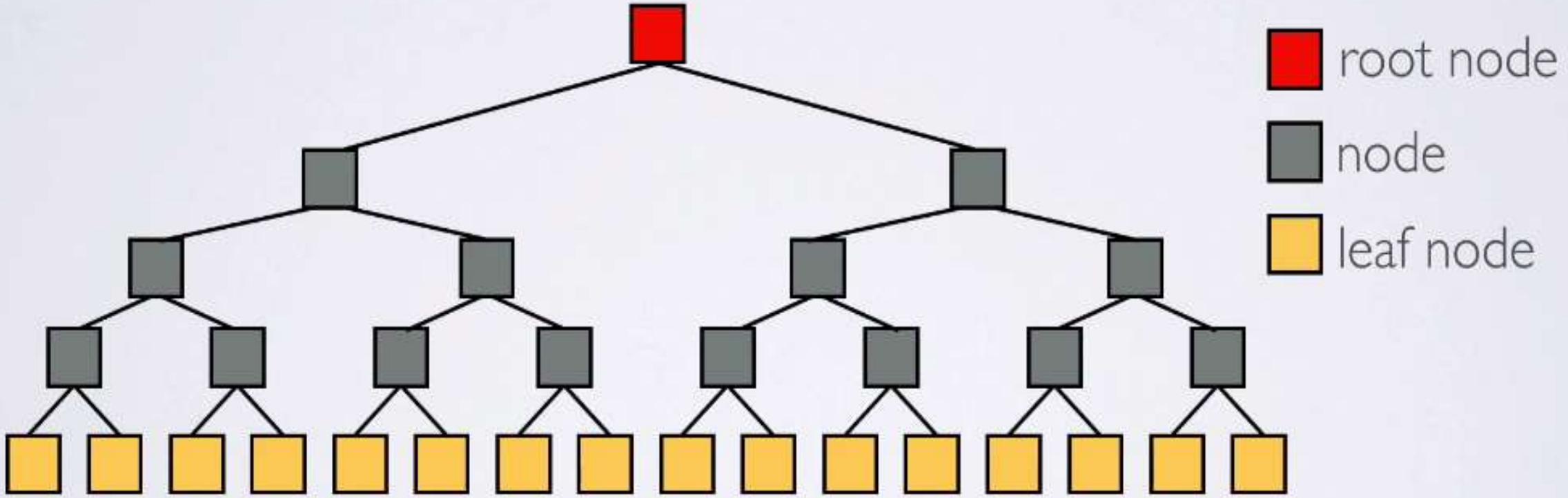
- You must be already getting the idea of how these two have noticeable similarity. Blockchain data structure can easily be said to be a linked list. While the linked list has the pointer function, the blockchain has the hash function. Each block in the blockchain has a unique hash number as well as the hash number of the previous block or the parent block.
- Both of blockchain and linked list adopt the same technology. They both have a genesis block which doesn’t have a previous hash number.

Differences: Blockchain and Linked list?

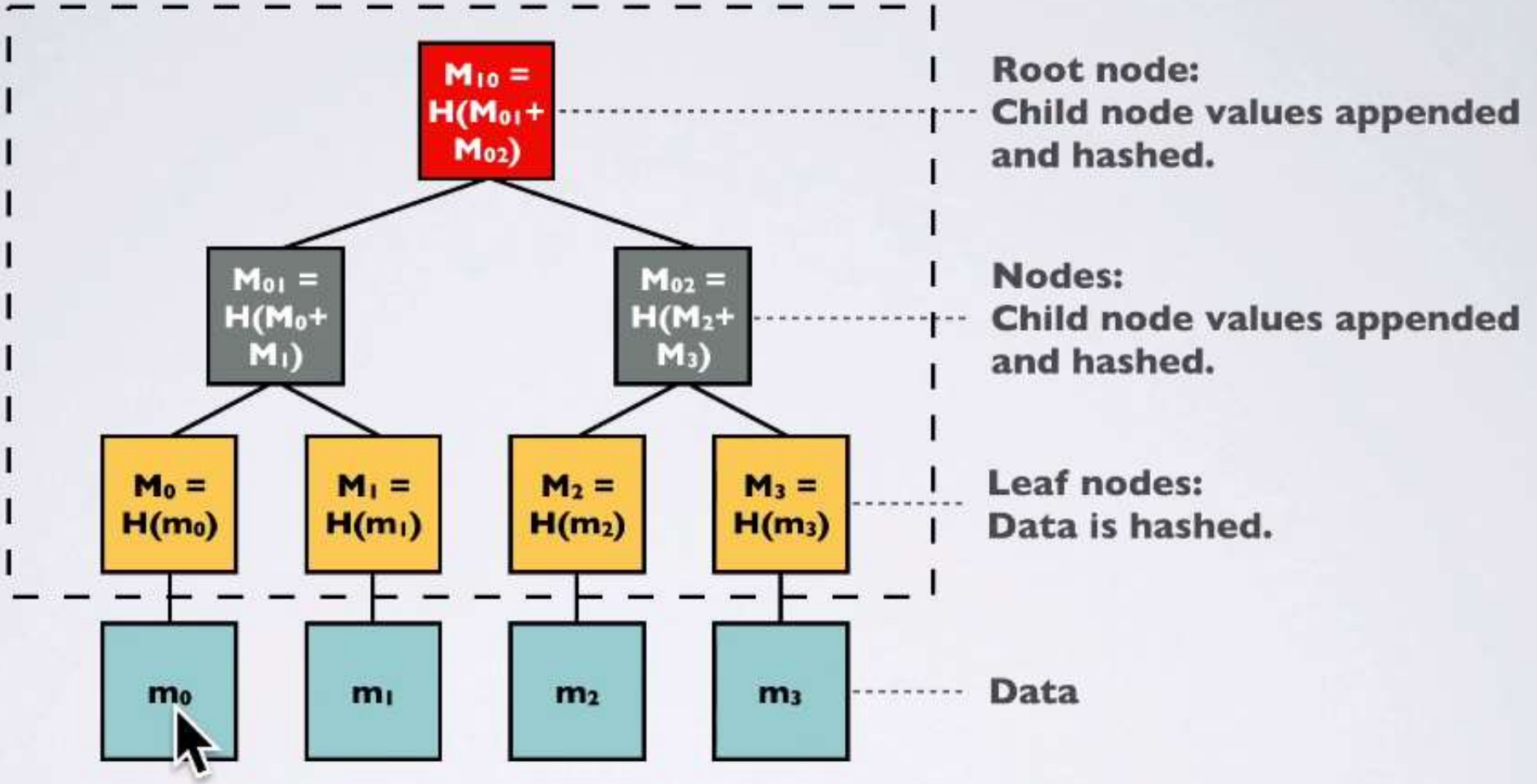
- The first and foremost clash of blockchain vs linked list is that a blockchain has a hash function to identify the ancestor of a particular block. Whereas a linked list does the same function using a pointer function.
- Again, a blockchain is way more complicated in terms of structure. A linked list is a linear way of arranging and storing data. Blockchains, for example, the blockchains of Bitcoin or Ethereum have Merkle trees to store transactions and all the data related to the transactions. Moreover, these Merkle trees (or blocks) have a link to their parent hash with the unique hash number.
- Moreover, a blockchain has some unique features. For instance, a blockchain is decentralized, distributed, an autonomous digital ledger that can have numerous applications in our practical life. Data manipulation and tampering are nearly impossible as the system of the blockchain will allow it in sense. While on the other hand, a linked list is a simple way of structuring data.

Merkle Tree?

- A hash tree or Merkle tree is a tree structure in which each leaf node is a hash of a block of data and each non-leaf node is a hash of its children. This results in a single hash called the Merkle root. If every node has two children, the tree is called a binary hash tree.

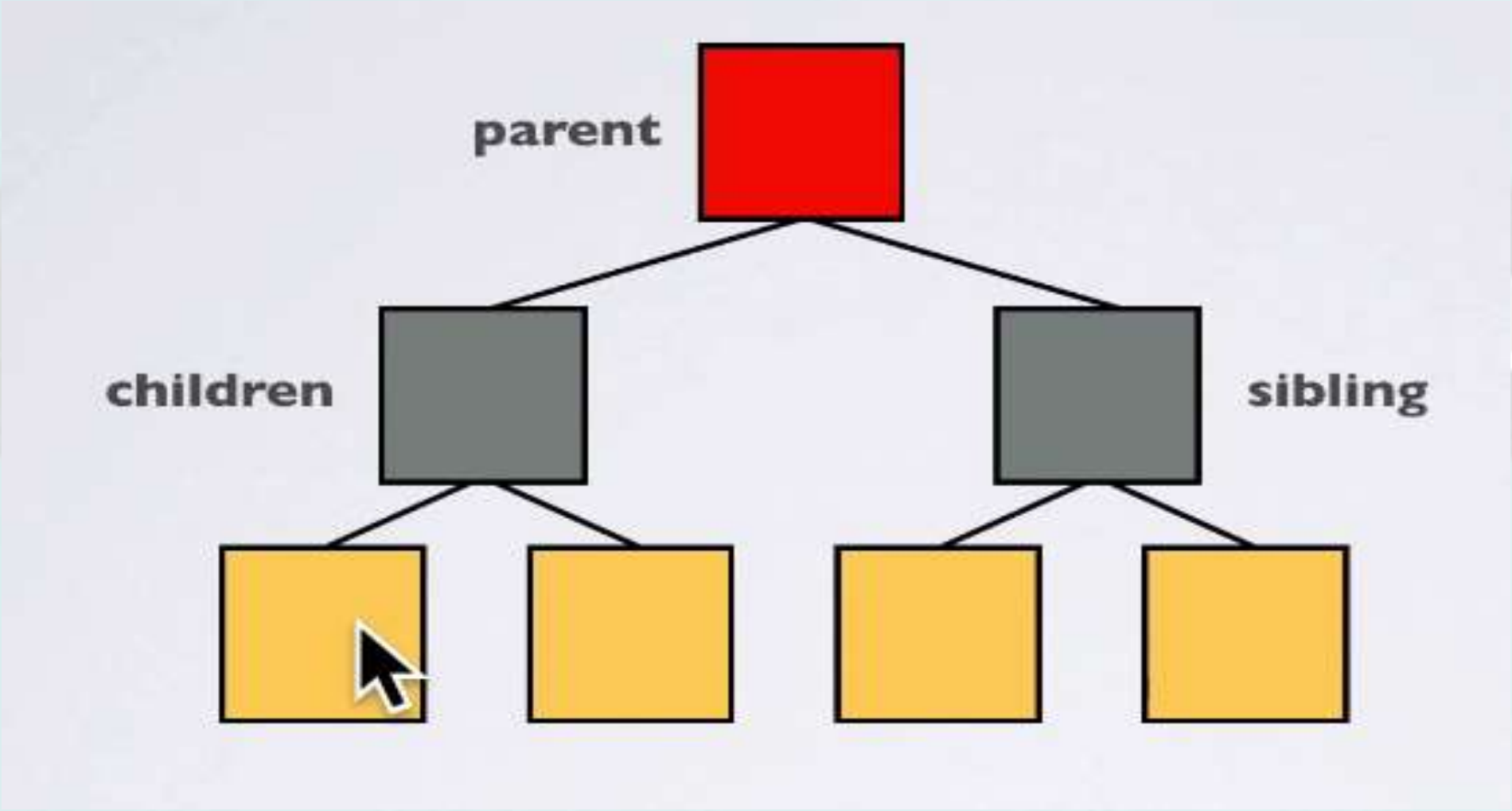


MERKLE TREE



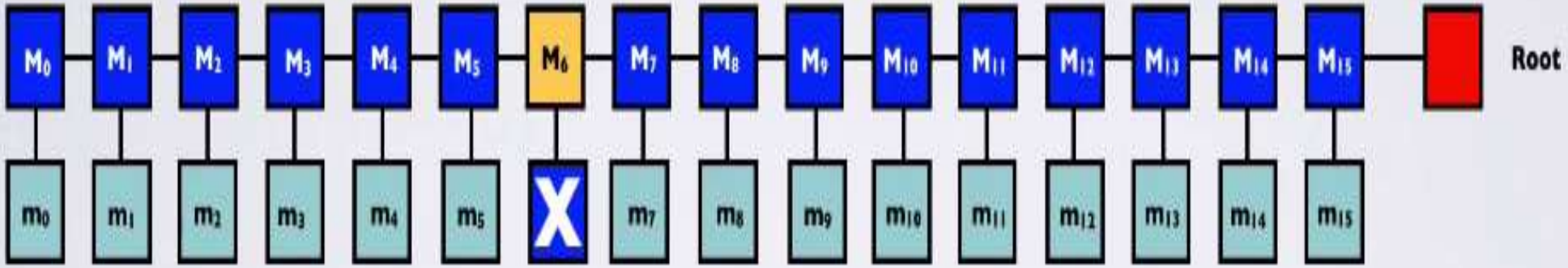
The data (m) itself is not considered part of the Merkle tree but the HASHED data (M) is part of the Merkle tree.

Structure of Merkle Tree

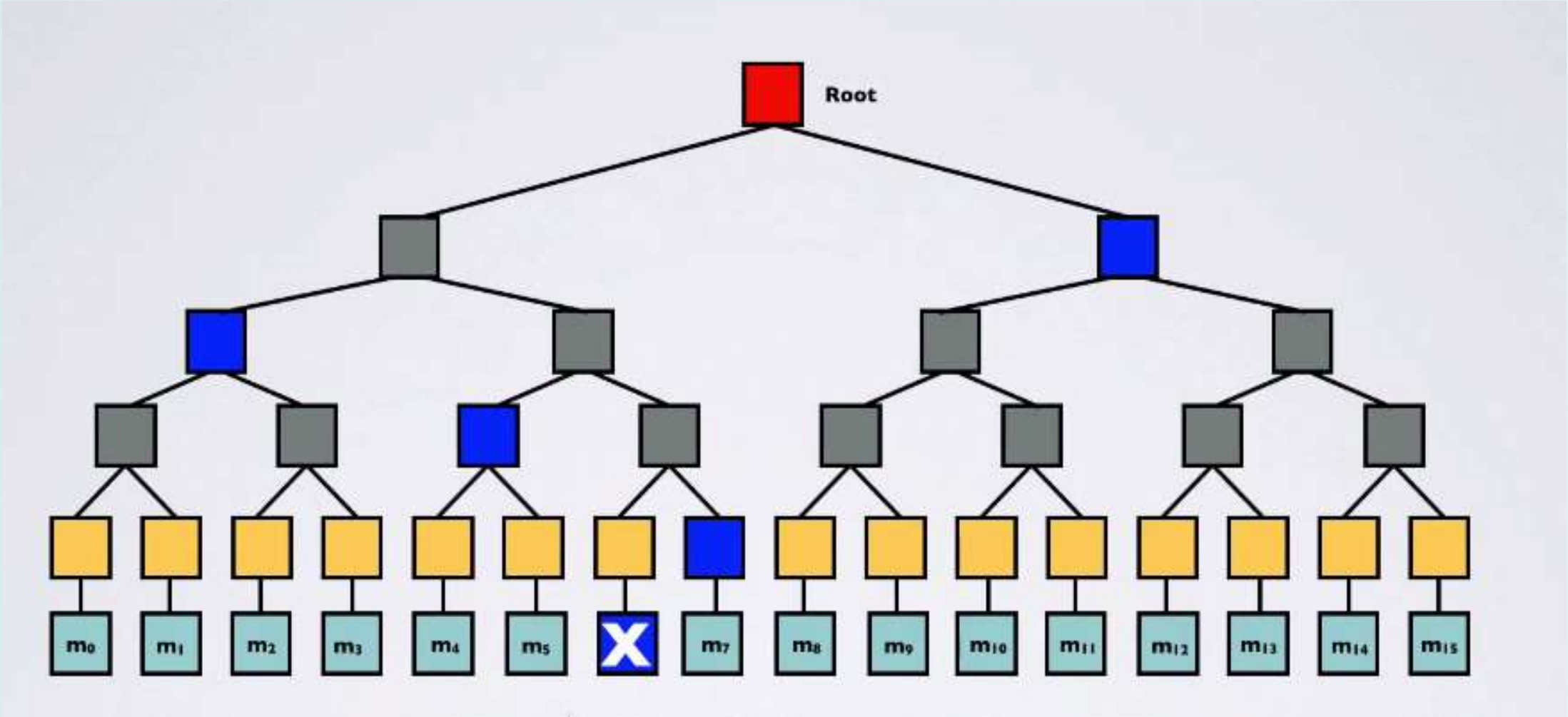


Why Merkle Tree?

Why not hash all messages, append the hashed messages and then hash it all to get one root hash value.

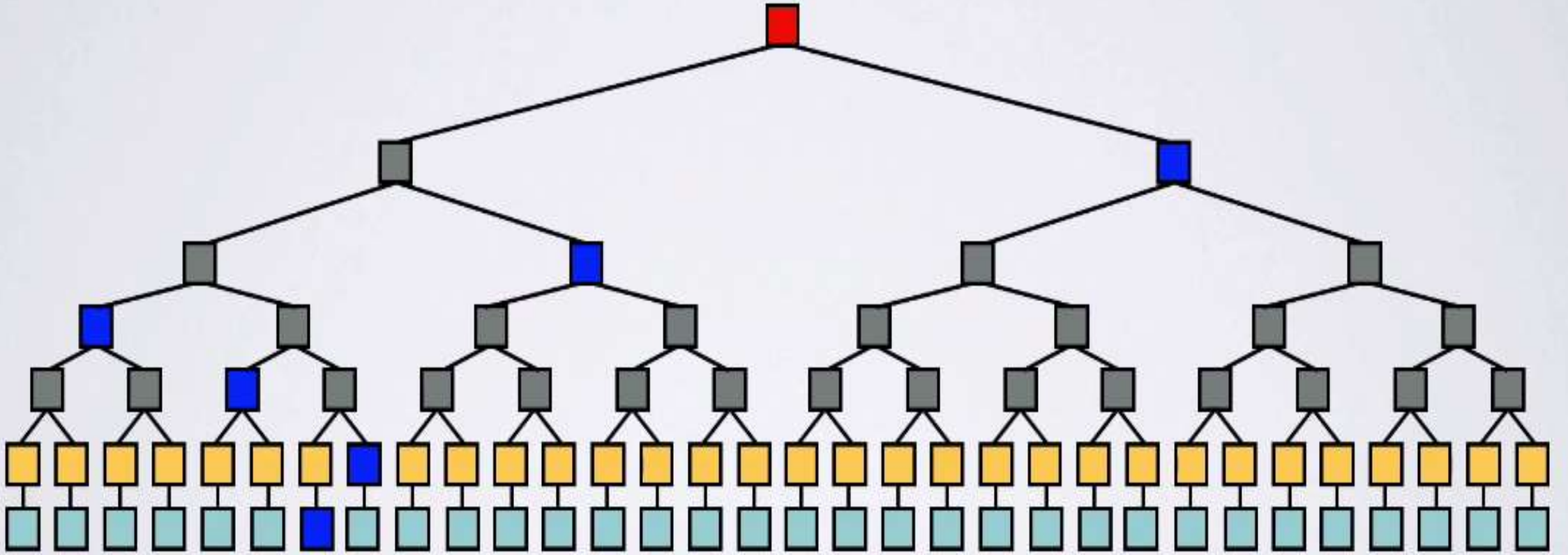


In case of Merkle tree



More Over With large of data

- If a Merkle tree has more leaves less hashed values are needed, in comparison to the number of leaves, to validate if a message is not tampered with.



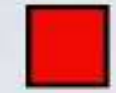
:Properties of Binary Merkle tree:

- A perfect Merkle binary tree has the following properties:
  - The number of leaves is always 2^n (n=0, 1, 2, 3,..).
  - Each node has 0 or 2 children.
  - All leaves are on the same level.

- In a perfect binary tree the following formulas can be applied:

Total number of leaves = L = (N + 1) / 2
Total number of nodes = N = 2L - 1
Total number of levels = LV = $\log_2$(L) + 1
LV = (ln(L) / ln(2)) + 1

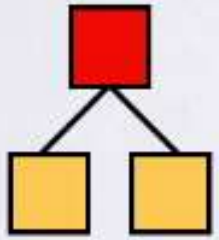
Example of Perfect Binary Tree



Level 1

Number of leaves = L = 1
Number of nodes = N = 1
Number of levels = LV = 1

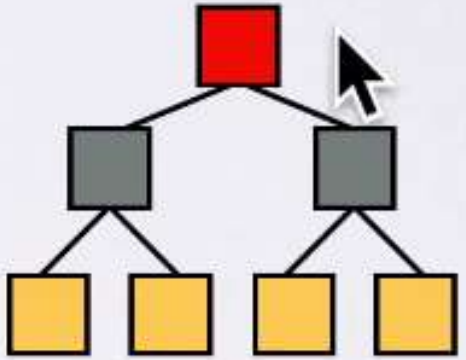
This Merkle tree has only one leaf.
This leaf is also the root.



Level 1

Level 2

Number of leaves = L = 2
Number of nodes = N = 3
Number of levels = LV = 2



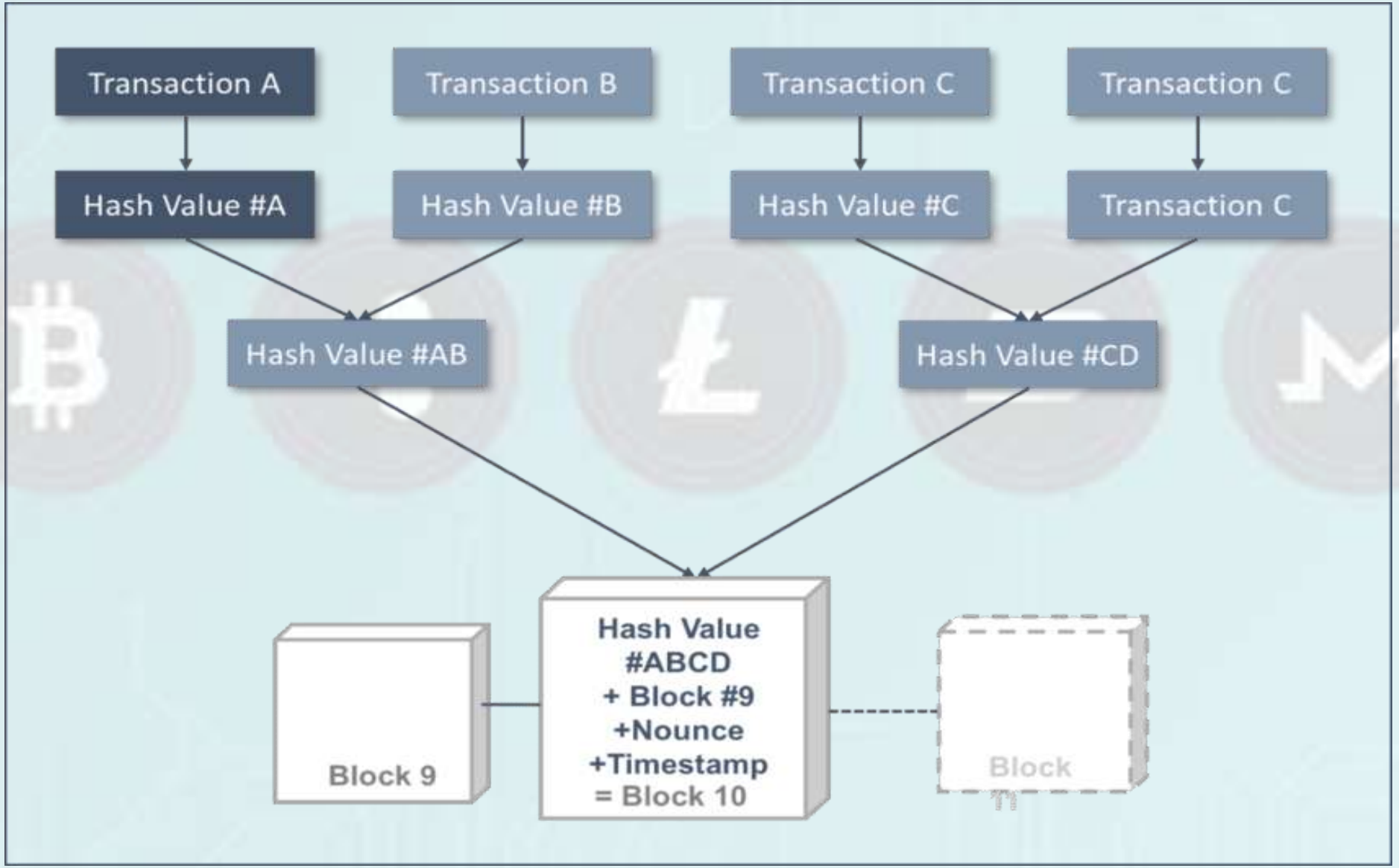
Level 1

Level 2

Level 3

Number of leaves = L = 4
Number of nodes = N = 7
Number of levels = LV = 3

Merkle Trees in Bitcoin



Why Merkle Tree in Cryptocurrency?

Merkle trees are a fundamental part of blockchain technology. A merkle tree is a structure that allows for efficient and secure verification of content in a large body of data. This structure helps verify the consistency and content of the data. Merkle trees are used by both Bitcoin and Ethereum.


Thank you