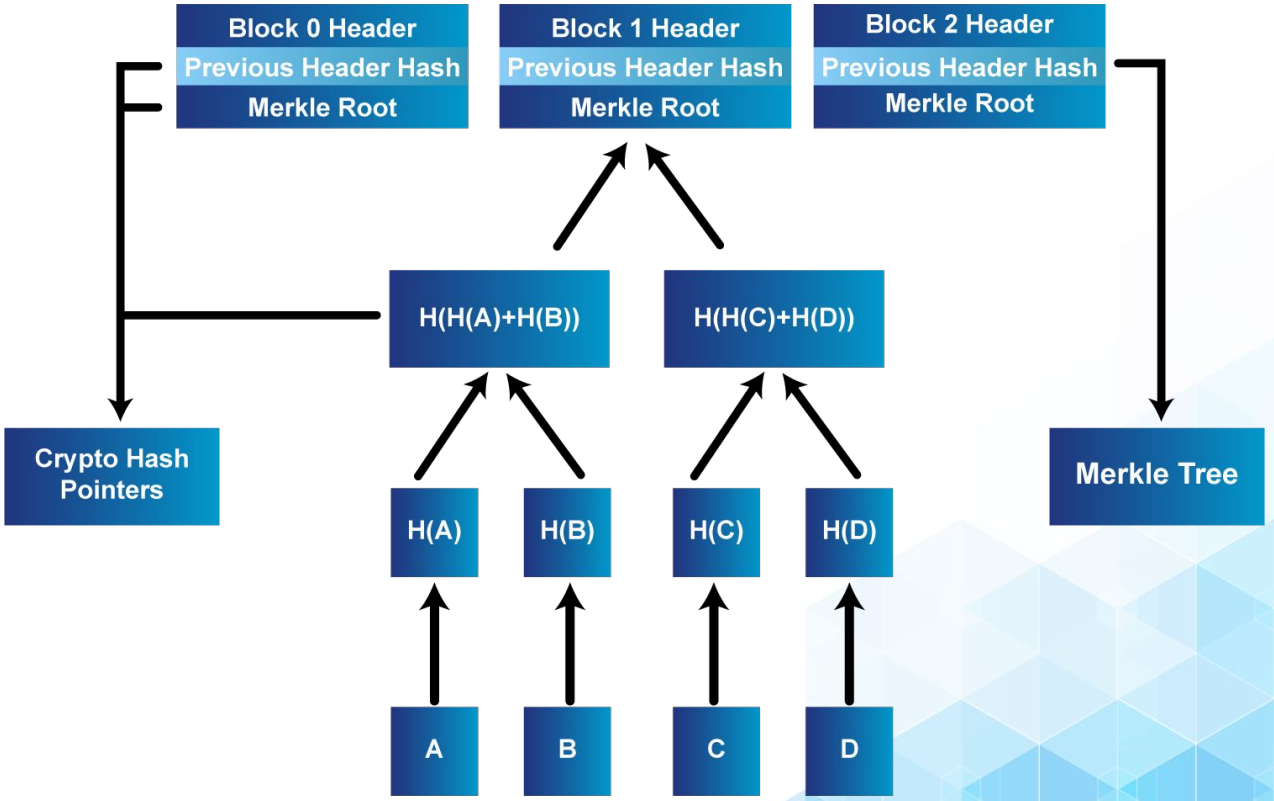
Merkle Tree and Hashing

Merkle Tree

- A Merkle tree is a data structure that uses hashing to store and verify the data.
- In the Merkle tree, each root node contains the hash value of a data block, and non-leaf nodes contain the hash of the child nodes.
- Merkle trees are used for effective data validation in distributed systems.
- In blockchain, each leaf node is the hash of a transaction in the block, and each non-leaf node is a hash of its children.
- Merkle trees are secure because of hashing. If any data is changed, the merkle tree structure becomes incorrect.
- Merkle trees are small in size, making it much easier to maintain and verify the data in the tree.

Merkle Tree



Why is Merkle Tree vital in Blockchain?

- Merkle trees are used at the base layer of the blockchains to store and verify transactions data.
- Merkle trees provide data security and integrity by using hash functions.
- In the case of blockchains, without Merkle trees, the blockchain will become very hard to manage and use.
- The node would need to compare each entry line by line.
- Any discrepancy between the ledgers compromises the security of the network.
- Every verification request would require large packets of information to be sent over the network.
- A lot of processing power is consumed to compare the ledgers to ensure that there have been no changes.

Hashing

- Hashing is the process of having an input item of any length converted into an output item of a fixed length.
- Transactions of different lengths are run through a given hashing algorithm, and all give an output of a fixed length, called a hash.
- The hash size will depend on the hash function used, but the output using a particular hashing algorithm will be of a specific size.
- Cryptographic hash functions are one of the most important techniques in the field of cryptography and are used to accomplish many safety goals such as authentication, digital signatures, generation of pseudo numbers, digital steganography, digital time-stamping, etc.
- Bitcoin uses a cryptographic hashing algorithm- **Secure Hashing Algorithm 256**, often known as **SHA-256**.

Blockchain Council

THANK YOU!

Any Questions?

Visit

community.blockchain-council.org

Mail Us

hello@blockchain-council.org