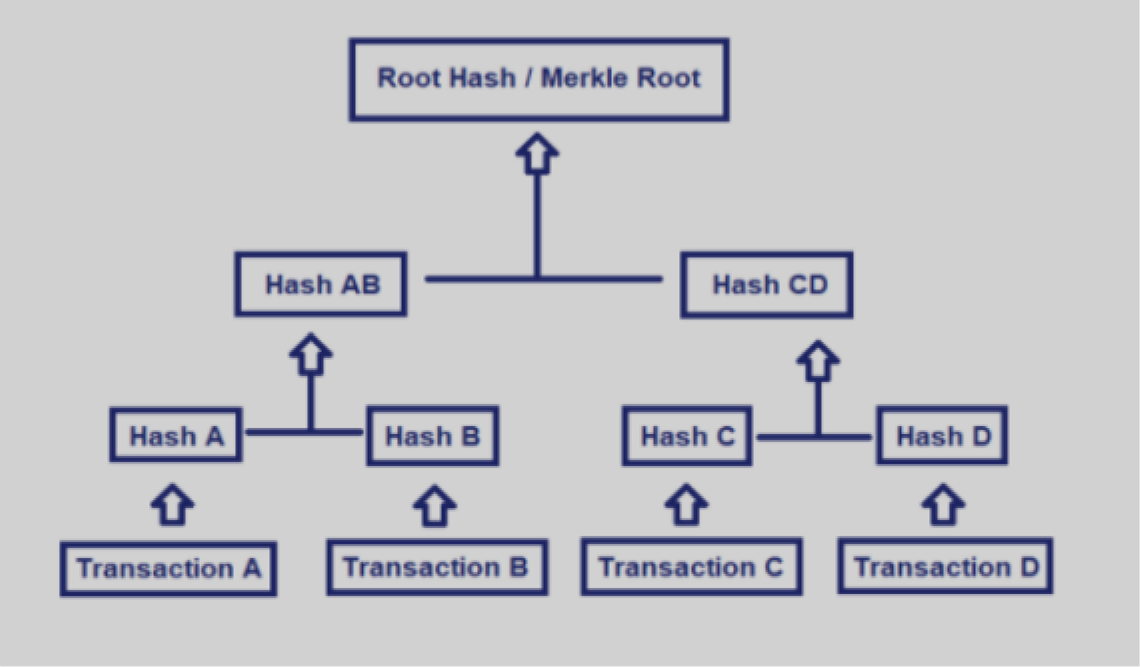
4 marks Explanation:

Q: In Blockchain terms, to prove “Transaction B” and Transaction D are part of the following merkle tree. What all hashes are required to be sent by the prover to the verifier in addition to transaction B and Transaction D itself. Refer below Diagram.



Expert Answer

This solution was written by a subject matter expert. It's designed to help students like you learn core concepts.

Step-by-step

1st step
All steps
Answer only

Step 1/3

Detailed explanation of how the prover can prove that "Transaction B" and "Transaction D" are part of the given Merkle tree and what additional hashes are required to be sent to the verifier IS -

A Merkle tree is a binary tree in which every leaf node represents a data block and every non-leaf node represents the hash of its two child nodes. In the given Merkle tree, each leaf node represents a transaction and the hash of each non-leaf node is the hash of the concatenation of its two child nodes.

To prove that "Transaction B" and "Transaction D" are part of the Merkle tree, the prover needs to provide the following information to the verifier:

- Hash of "Transaction B": The prover needs to send the hash of "Transaction B" to the verifier. This hash can be calculated by applying a cryptographic hash function (e.g., SHA-256) to the value of "Transaction B".

- Hash of the sibling of "Transaction D" in the same level of the tree: The prover needs to send the hash of the sibling of "Transaction D" in the same level of the Merkle tree as "Transaction D". In this case, the sibling of "Transaction D" is "Hash H" (as shown in the image). Therefore, the prover needs to send the value of "Hash H" to the verifier.

EXPLANATION

- Hash of the parent of the hash from step 2 and the hash of the sibling from step 1: The prover needs to send the hash of the parent of the hash from step 2 and the hash of the sibling from step 1. In this case, the parent of "Hash H" and "Hash of Transaction B" is "Hash E" (as shown in the image). Therefore, the prover needs to send the value of "Hash E" to the verifier.

Step 2/3

The verifier can then perform the following steps to verify the inclusion of "Transaction B" and "Transaction D" in the Merkle tree:

1. Hash the value of "Transaction B": The verifier applies the same cryptographic hash function to the value of "Transaction B" as the prover did to obtain the hash of "Transaction B".
2. Hash the value of the sibling of "Transaction D" (i.e., "Hash H"): The verifier applies the same cryptographic hash function to the value of "Hash H" as the prover did to obtain the hash of the sibling of "Transaction D".
3. Hash the values from step 1 and step 2 together: The verifier concatenates the two hashes obtained in steps 1 and 2 and applies the same cryptographic hash function to the concatenation to obtain a new hash.

Step 3/3

4. Compare the resulting hash with the value of the parent hash (i.e., "Hash E"): The verifier compares the resulting hash from step 3 with the value of the parent hash (i.e.,

"Hash E"). If the two values match, then "Transaction B" and "Transaction D" are part of the Merkle tree.

Final answer

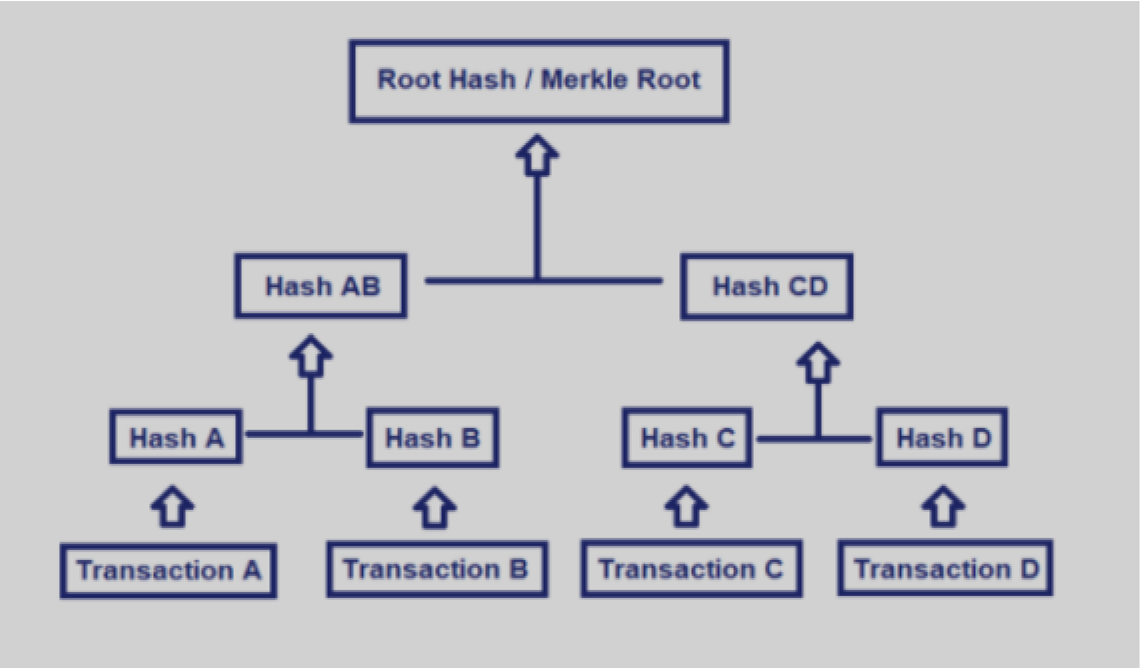
By providing the additional hashes required for verification, the prover can prove the inclusion of "Transaction B" and "Transaction D" in the Merkle tree without revealing any other information about the other transactions in the tree

---

2nd approach:

Question

(0)



Q: In Blockchain terms, to prove “Transaction B” and Transaction D are part of the following merkle tree. What all hashes are required to be sent by the prover to the verifier in addition to transaction B and Transaction D itself. Refer below Diagram.

Expert Answer

Step-by-step

1st step

All steps

Answer only

Step 1/2

Suppose a verifier wants to verify whether transaction E is part of merkel tree or not. For this, the prover sends the hash values Hash F, Hash GH, Hash ABCD, and merkle root to the verifier. First, the verifier finds the hash of transaction E, i.e., hash E using the hash function.

Step 2/2

Then hash E, along with the values hash F, Hash GH, and Hash ABCD, are recomputed to generate the Merkle root/root hash. If the computed Merkle root and original Merkle root match, then we can say the hash E is a genuine leaf of the Merkle tree, and transaction E is part of the tree. If the obtained Merkle root and original Merkle root do not match, then it can be confirmed that transaction E has been tampered with.

Final answer

in given ques, You need to provide the merkle root/root hash, hash A and hash C to the verifier. From transaction B and transaction C, the verifier will generate hash B and hash D, respectively. It will then calculate the hash ABCD, which will be compared with the mekle root. If they match, the verification is successful; otherwise, it is unsuccessful.