The Bitcoin Cryptocurrency: Part 2

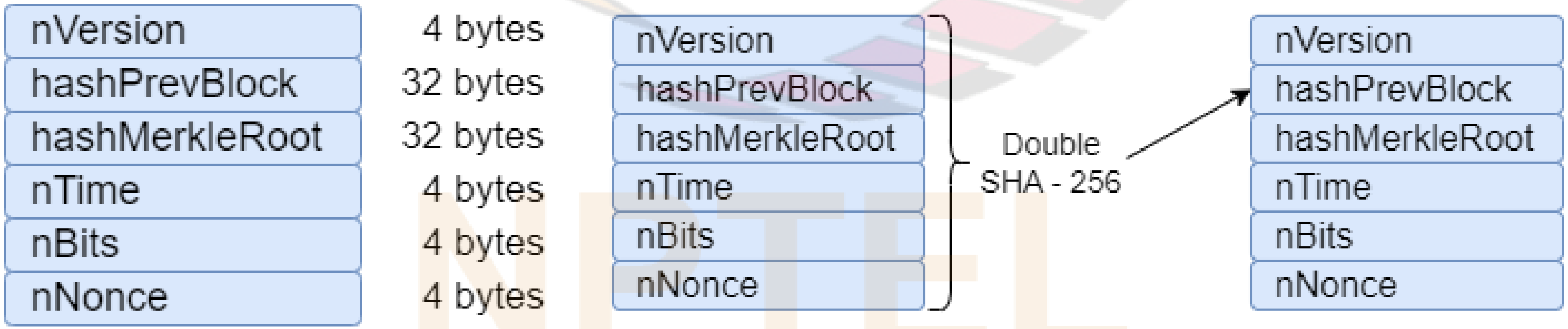
Gaurav S. Kasbekar
Dept. of Electrical Engineering
IIT Bombay

References

- Saravanan Vijayakumaran, "*An Introduction to Bitcoin*", Lecture notes, IIT Bombay, Oct. 4, 2017. Available at: https://www.ee.iitb.ac.in/~sarva/bitcoin.html
- A. Narayanan, J. Bonneau, E. Felten, A. Miller, S. Goldfeder, "*Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*", Princeton University Press, 2016

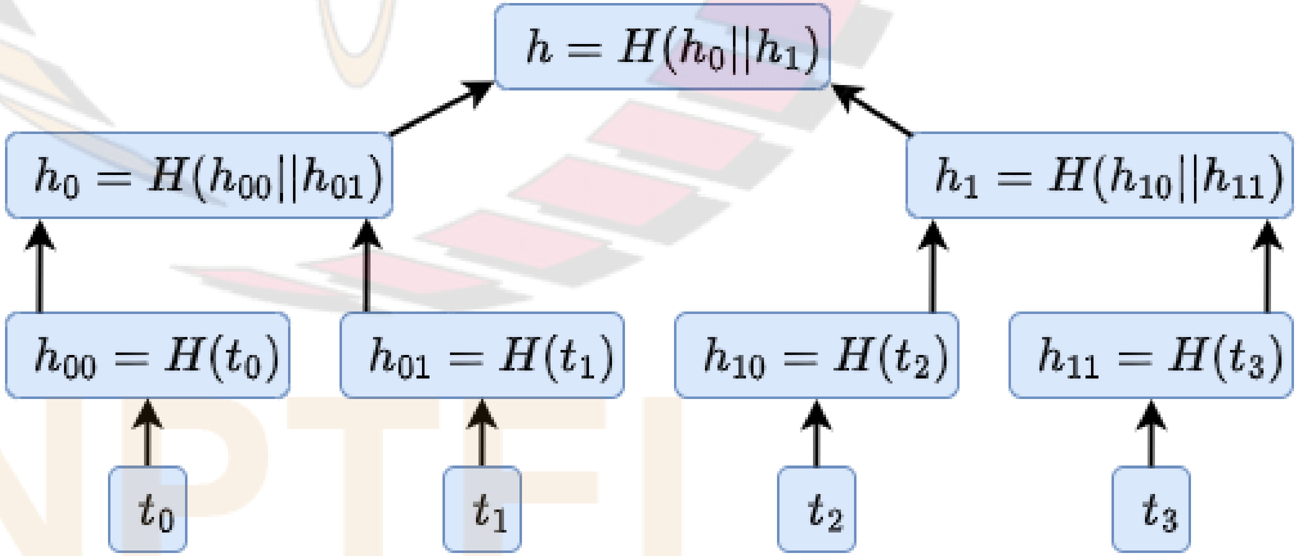
Block Header

- Each block in the blockchain begins with a 80-byte block header (see fig.)
- The prefix “n” of the 4-byte fields indicates that they are integer variables
- The prefix “hash” of the 32-byte fields indicates that they store hash function outputs
- “nVersion” field specifies version of block
  - as Bitcoin system evolved, changes were proposed to fix bugs or add new features
- “hashPrevBlock” field in block header contains double SHA-256 hash of the block header of the previous block in the blockchain
- Since the genesis block has no previous block, its “hashPrevBlock” field was set to all zeros



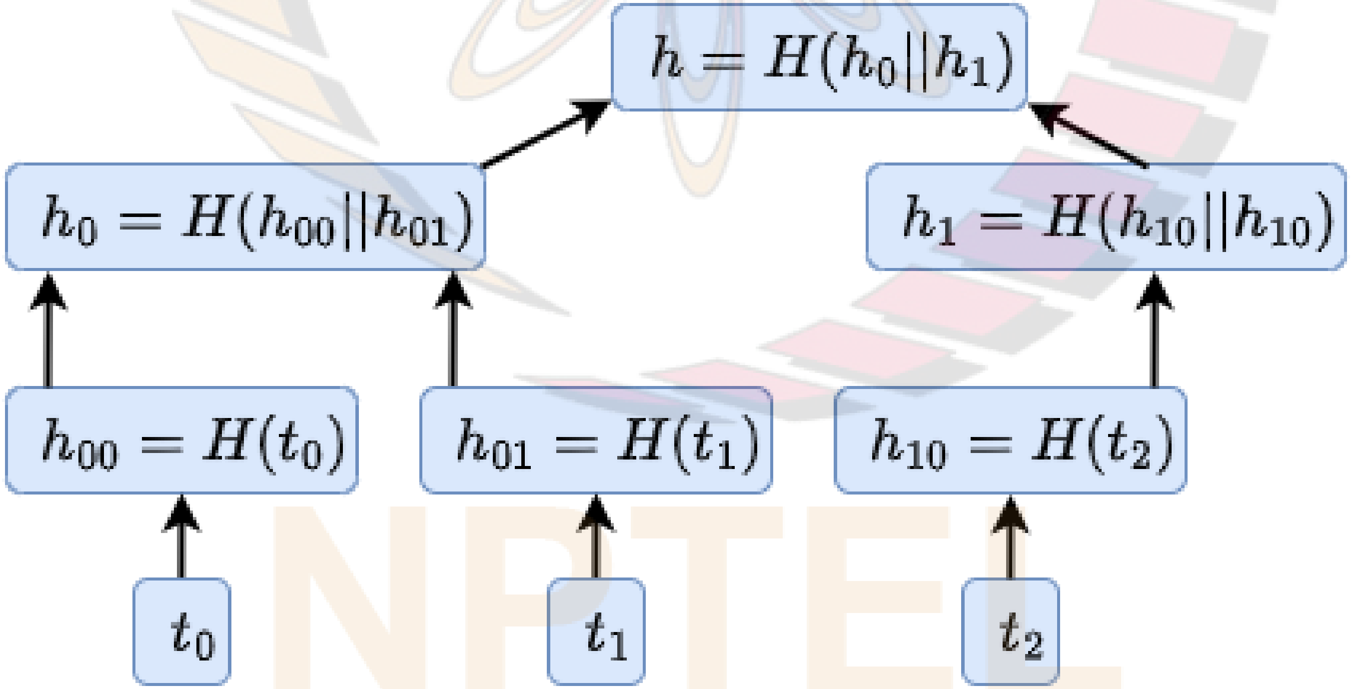
hashMerkleRoot field

- hashMerkleRoot field stores the root hash of a *Merkle tree* formed using the transactions in the block
- Transactions are arranged as a list and double SHA-256 hash of each is computed
- Using these hashes as leaves, a binary tree is created where each node is associated with a double SHA-256 hash of the concatenation of its child hashes
- In fig., the hash value h is the root hash of Merkle tree



hashMerkleRoot field (contd.)

- When the number of transactions is not a power of two, some nodes in Merkle tree have only one child
- In this case the hash of the single child is concatenated with itself and then hashed to derive the hash value associated with parent
- Fig. shows case where there are three transactions



hashMerkleRoot field (contd.)

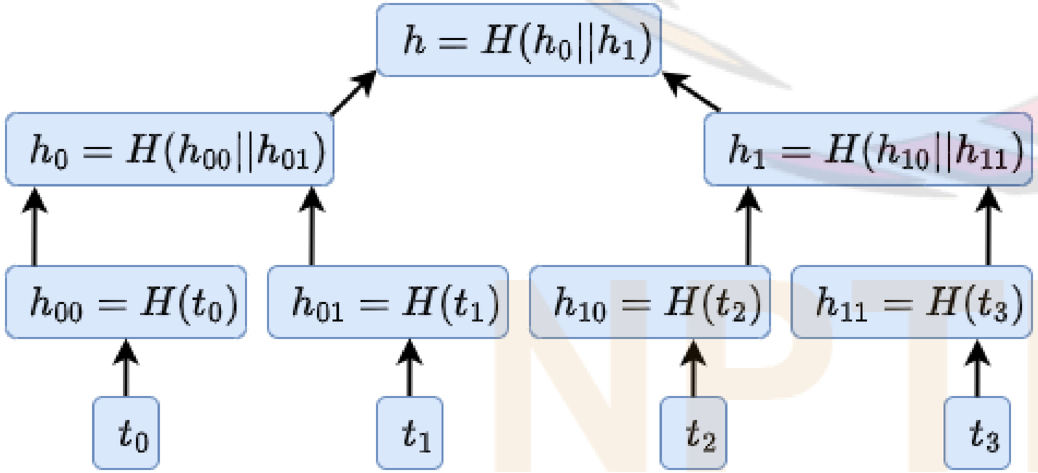
- hashMerkleRoot field is used for *tamper resistance*
- Any change to a transaction in a block will result in a change in the hashMerkleRoot field
- A change in hashMerkleRoot field in turn results in change in double SHA-256 hash of block header
- Recall that the “hashPrevBlock” field of a block header contains double SHA-256 hash of block header of previous block
- Hence, *the “hashPrevBlock” field of a block depends on all the transactions in all the previous blocks all the way up to the genesis block*
- We will later see that this property is used for guaranteeing tamper resistance of the transaction data

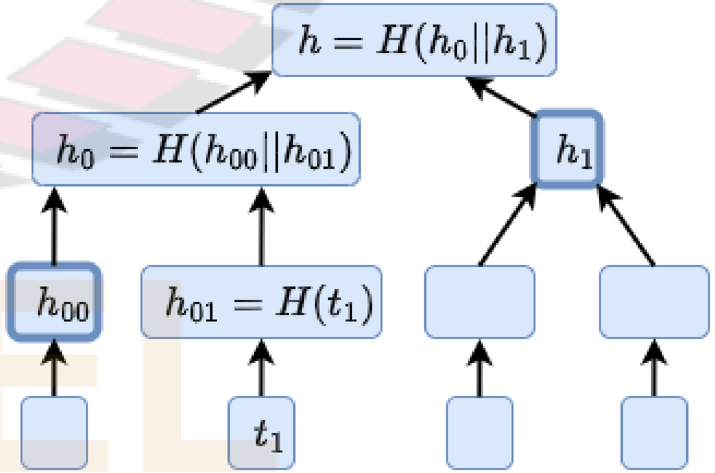
hashMerkleRoot field (contd.)

- Why do we use a Merkle tree of the transactions?
- Alternative (simpler) field that could be included in block header in place of hashMerkleRoot field:
  - ❑ we could have hashed the concatenation of all the transactions in a block and put the resulting hash value in the block header
  - ❑ i.e., if the transactions in a block are $t_0, t_1, \dots, t_{n-1}$, then we could have included the hash value $H(t_0||t_1|| \dots ||t_{n-1})$ in the block header
  - ❑ this would also have guaranteed tamper resistance of the transaction data
- Reason for using Merkle root:
  - ❑ *enables efficient membership proofs of transactions within a block*
  - ❑ there are nodes in the Bitcoin network called simple payment verification (SPV) nodes which store only the block headers and not the whole blocks like full nodes
  - ❑ suppose a SPV node, Bob, needs to verify whether a transaction, say t_1, involving Bob belongs to a block or not (e.g., the transaction may say that Alice transferred x bitcoins to Bob); Bob has the header of the block, but not the full block
  - ❑ Bob contacts a full node; the full node must *efficiently* (i.e., by sending only a small amount of data to Bob) convince Bob that the transaction t_1 indeed belongs to the block
  - ❑ the full node sends a *Merkle branch* to Bob to prove the existence of the transaction t_1 in the block

Example

- Suppose we want to prove that the transaction t_1 was involved in the calculation of the root hash h in fig. on left
- We only need to provide the Merkle branch consisting of the hashes:
  - ❑ h_{00} and h_1
- The hashes h_{01}, h_0 and h can be calculated from t_1 and the Merkle branch
- If the root hash h appears in the hashMerkleRoot field of a block header, then we can be certain that this block contains the transaction t_1





hashMerkleRoot field (contd.)

- In general, the Merkle branch required to prove the existence of a transaction in a block containing n transactions has size:
  - $O(\log_2 n)$
- In contrast, if $H(t_0||t_1|| \ldots ||t_{n-1})$ were included in block header instead of the hashMerkleRoot field, then what data would be required to prove the existence of a transaction in the block?
  - all the transactions $t_0, \ldots, t_{n-1}$ would have to be specified
- Hence required size would be:
  - $O(n)$