Blockchain and Cryptocurrency

By: Syeda Tayyaba Bukhari

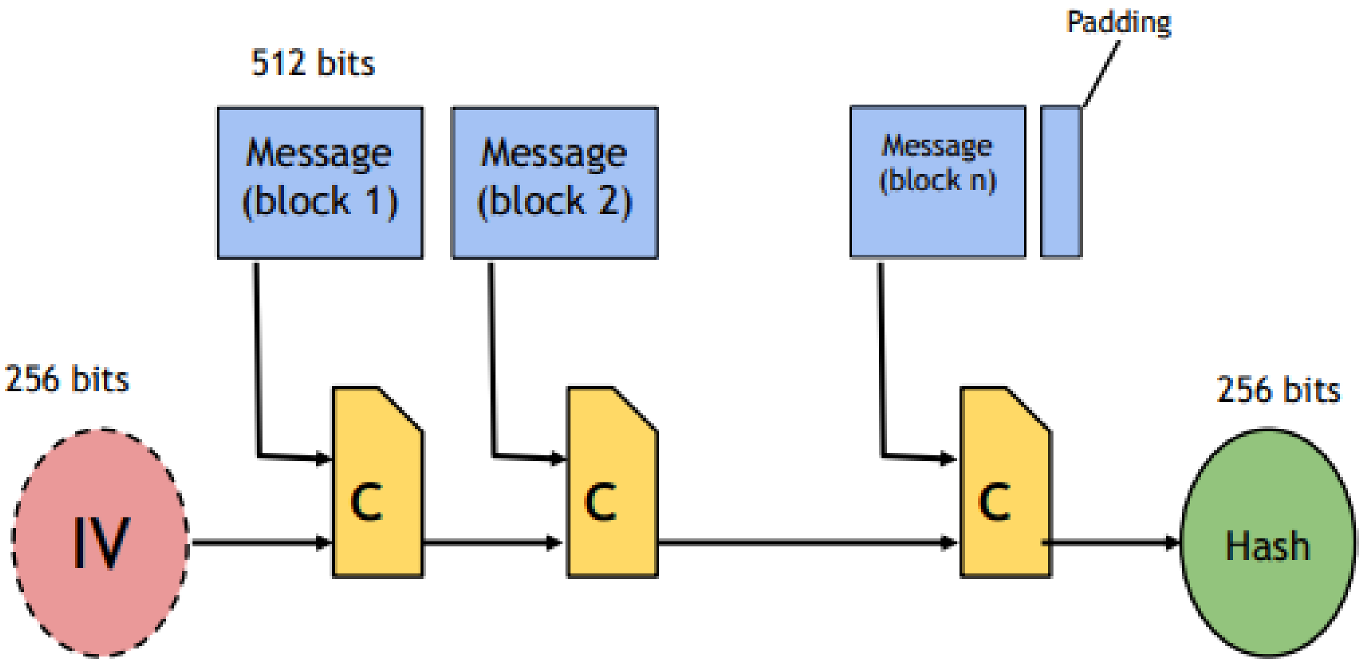
Byzantine Fault Tolerance

- If our system has 1/3 or less traitors, then our system will be still in safe state
- Tolerance factor is 33% (traitors)

Merkle-Damgard transform

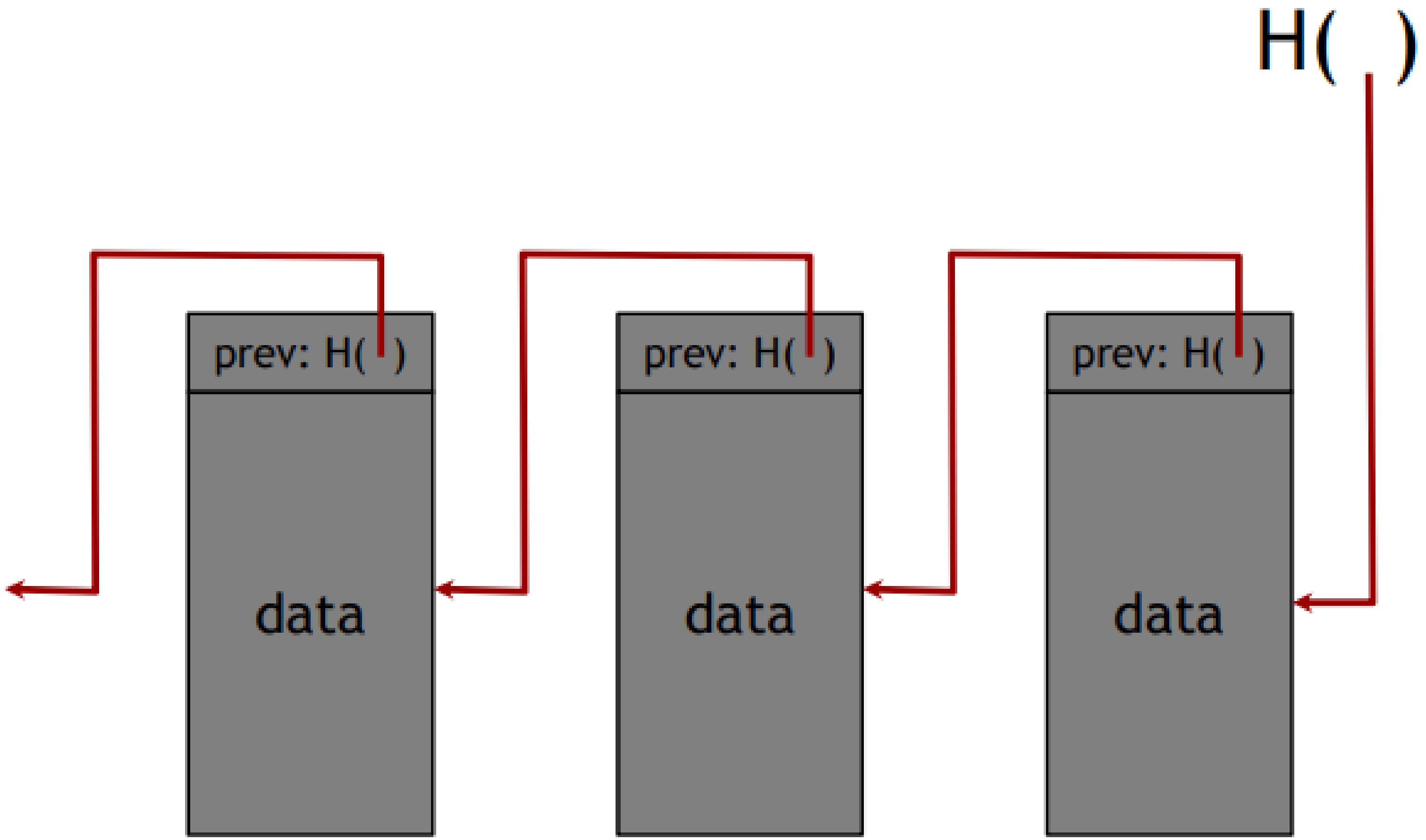
- We require that our hash functions work on inputs of arbitrary length
- as long as we can build a hash function that works on fixed-length inputs, there’s a generic method to convert it into a hash function that works on arbitrary-length inputs.

SHA-256 - Merkle-Damgard transform



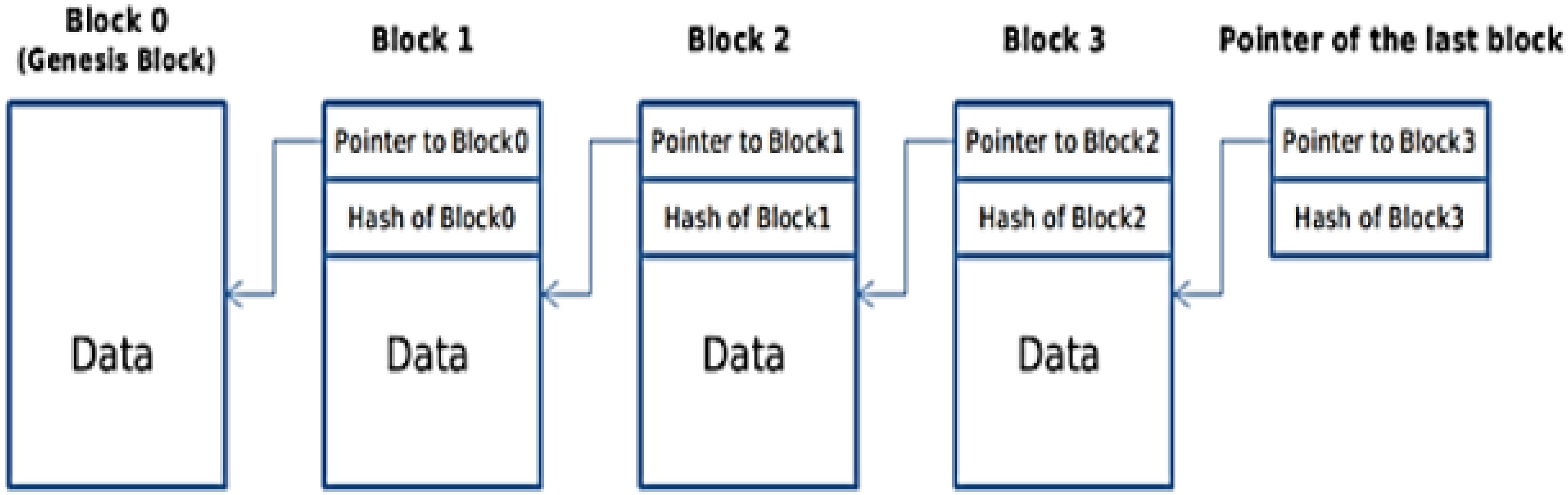
Theorem: If c is collision-free, then SHA-256 is collision-free.

linked list with hash pointers = “block chain”



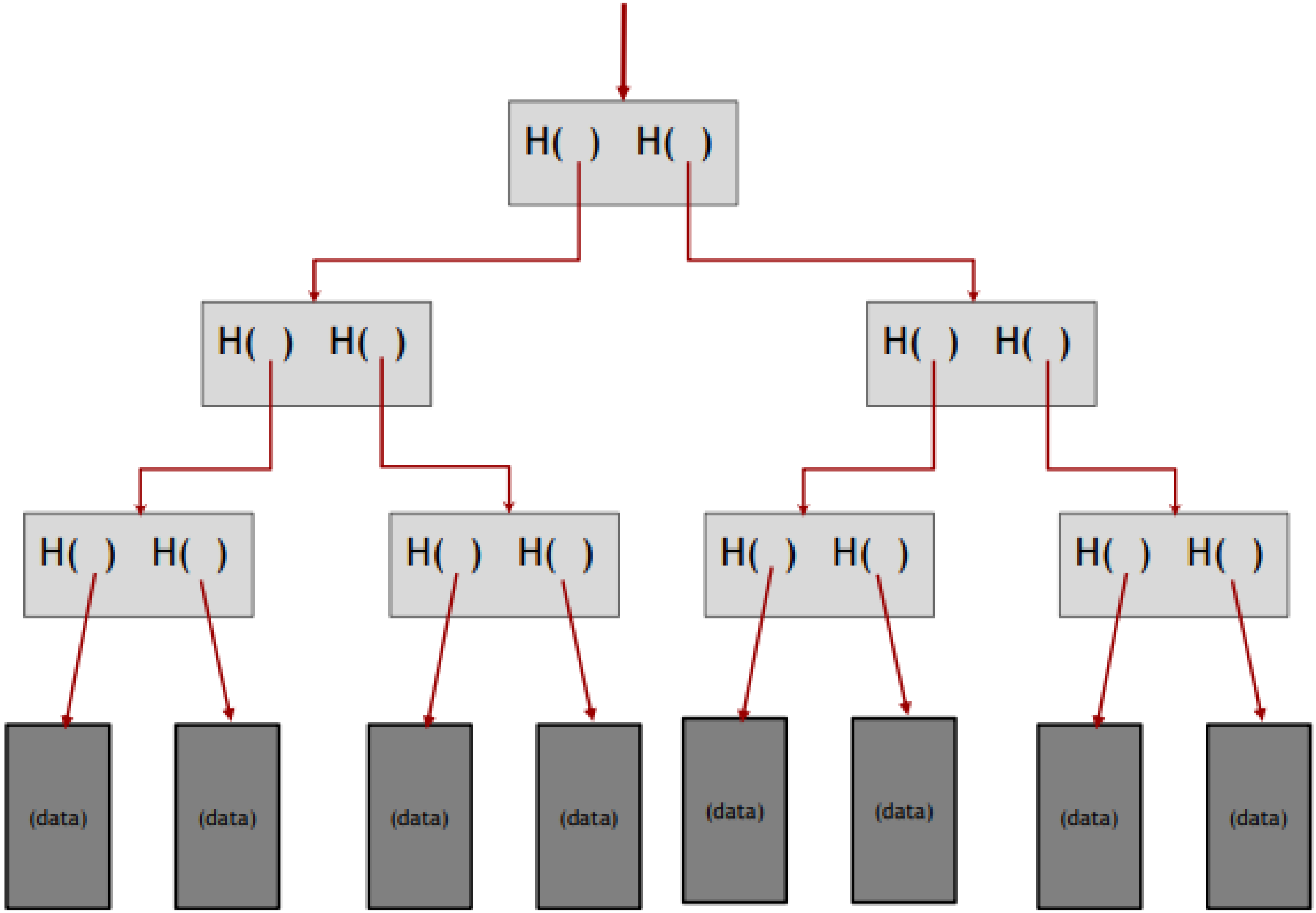
use case: tamper-evident log

linked list with hash pointers = “block chain”

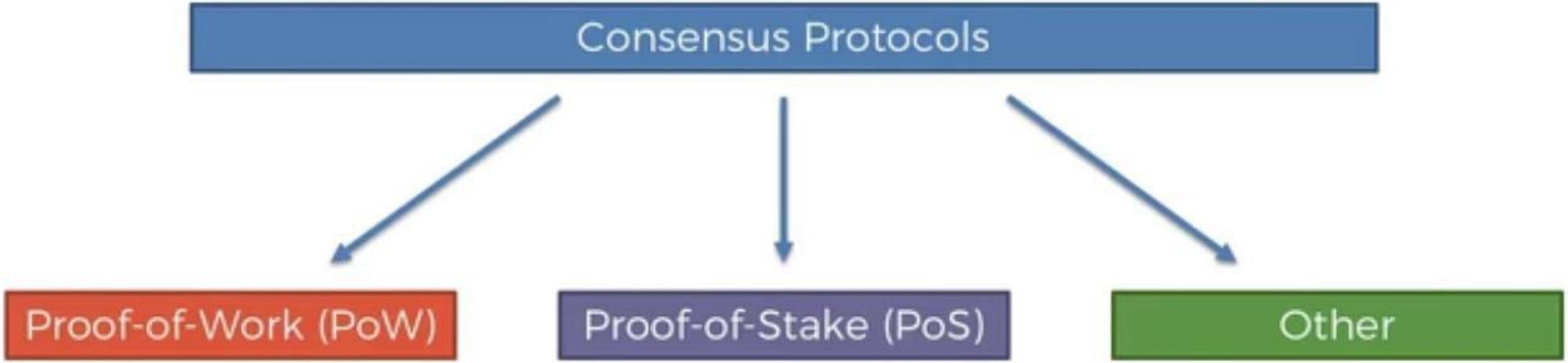


each block not only tells us where the value of the previous block was, but it also contains a digest of that value that allows us to verify that the value hasn’t changed.

binary tree with hash pointers = “Merkle tree”



Consensus Protocol



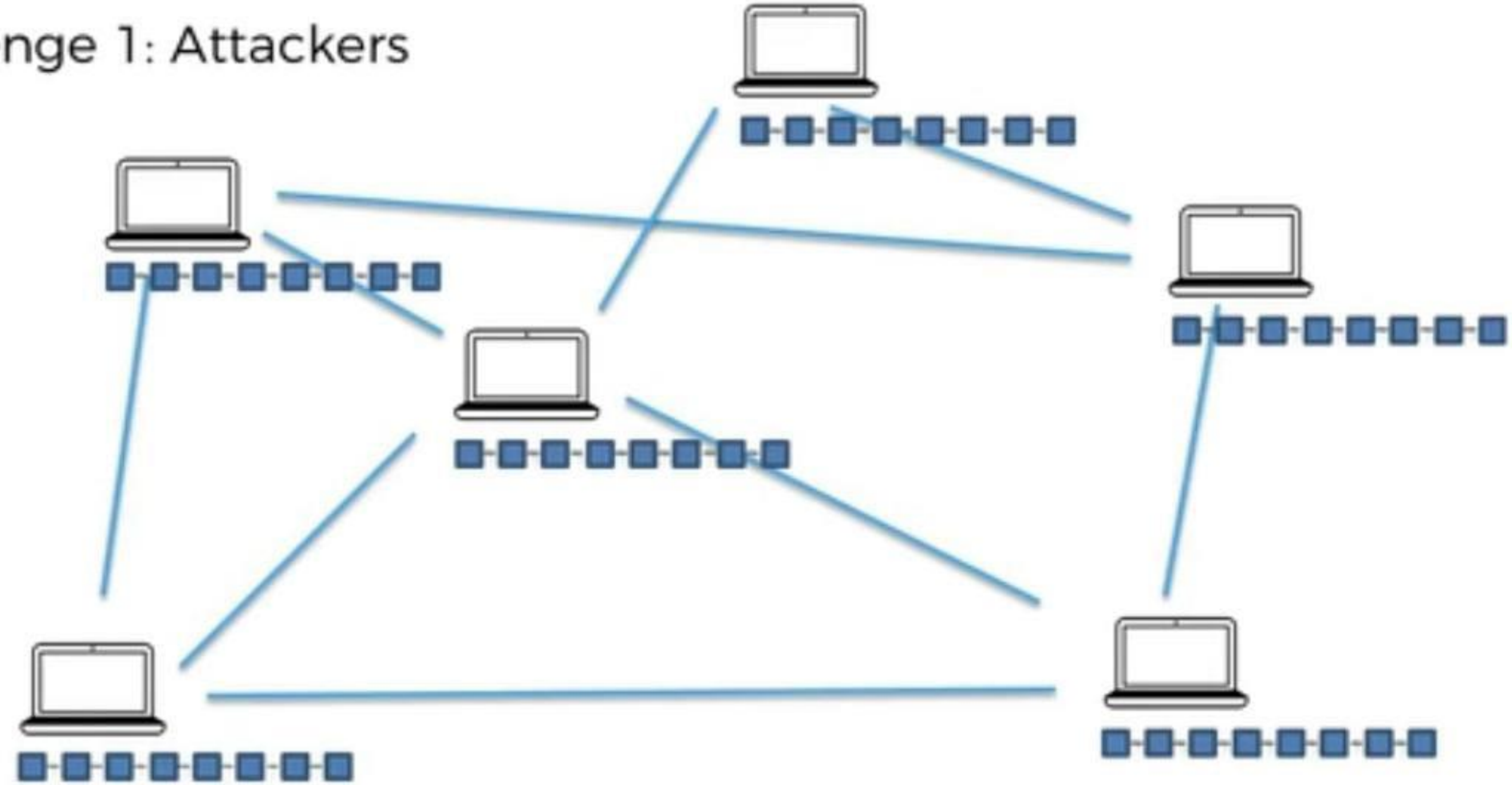
Consensus Protocols

Proof-of-Work (PoW)

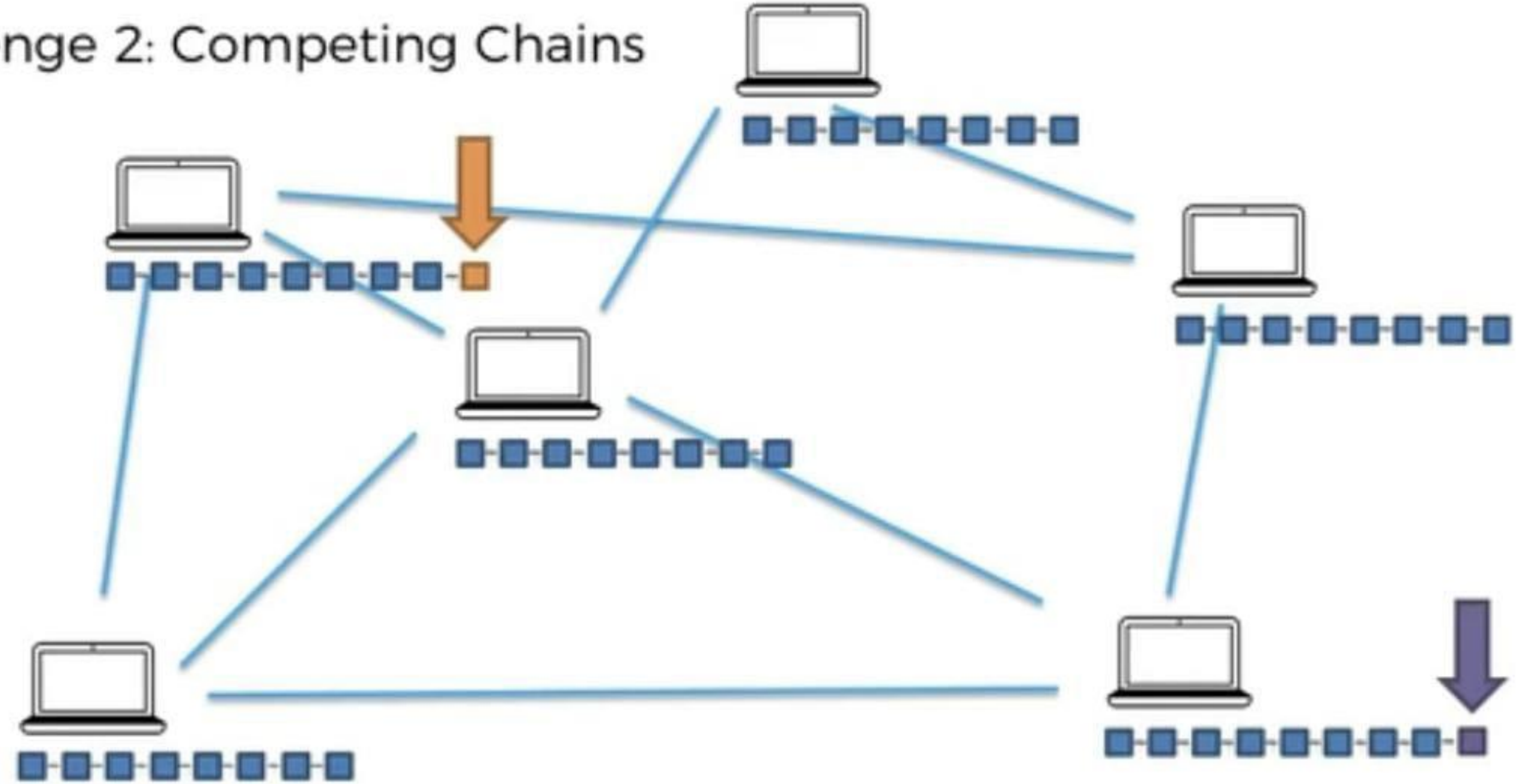
Proof-of-Stake (PoS)

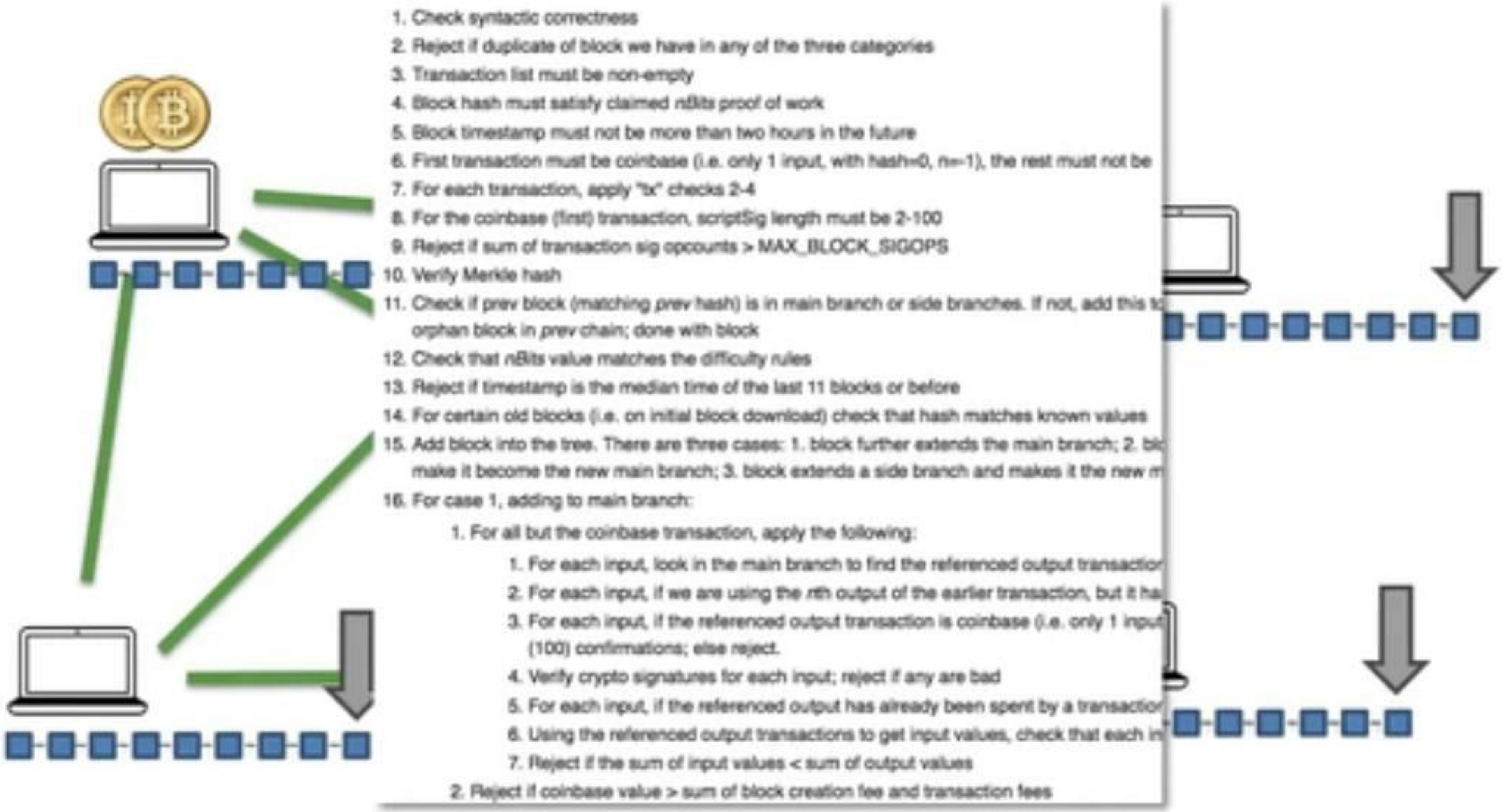
Other

Challenge 1: Attackers

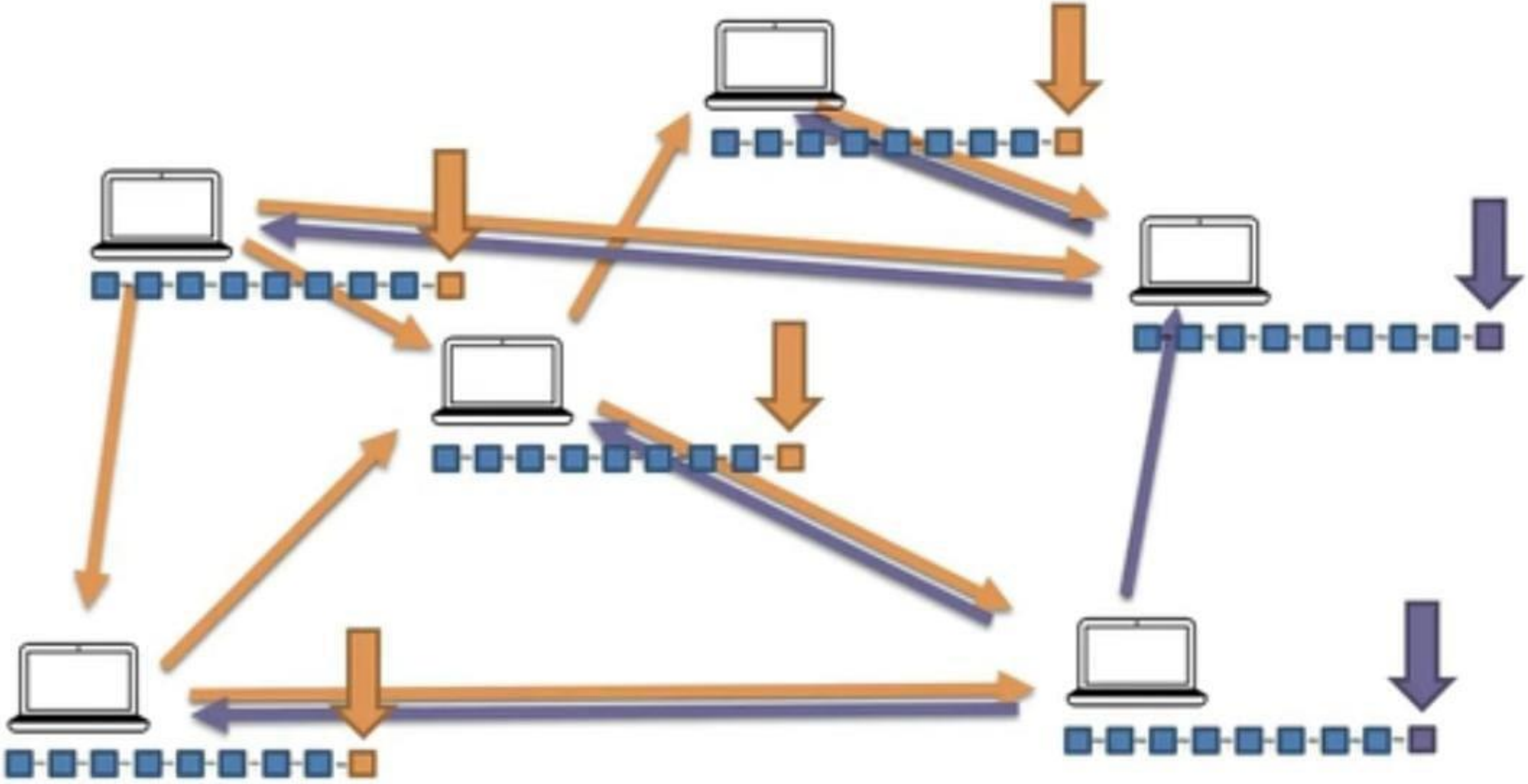


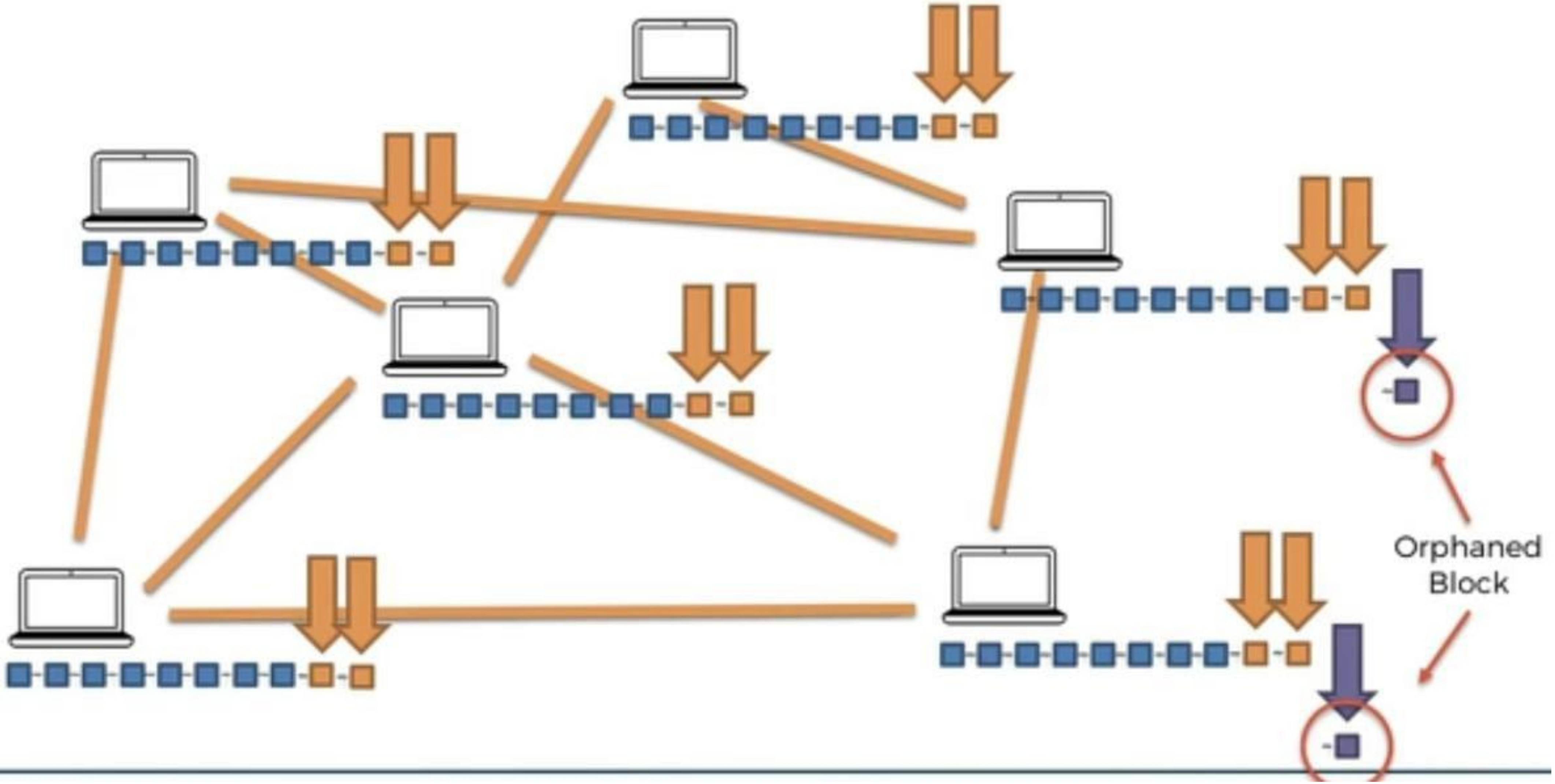
Challenge 2: Competing Chains

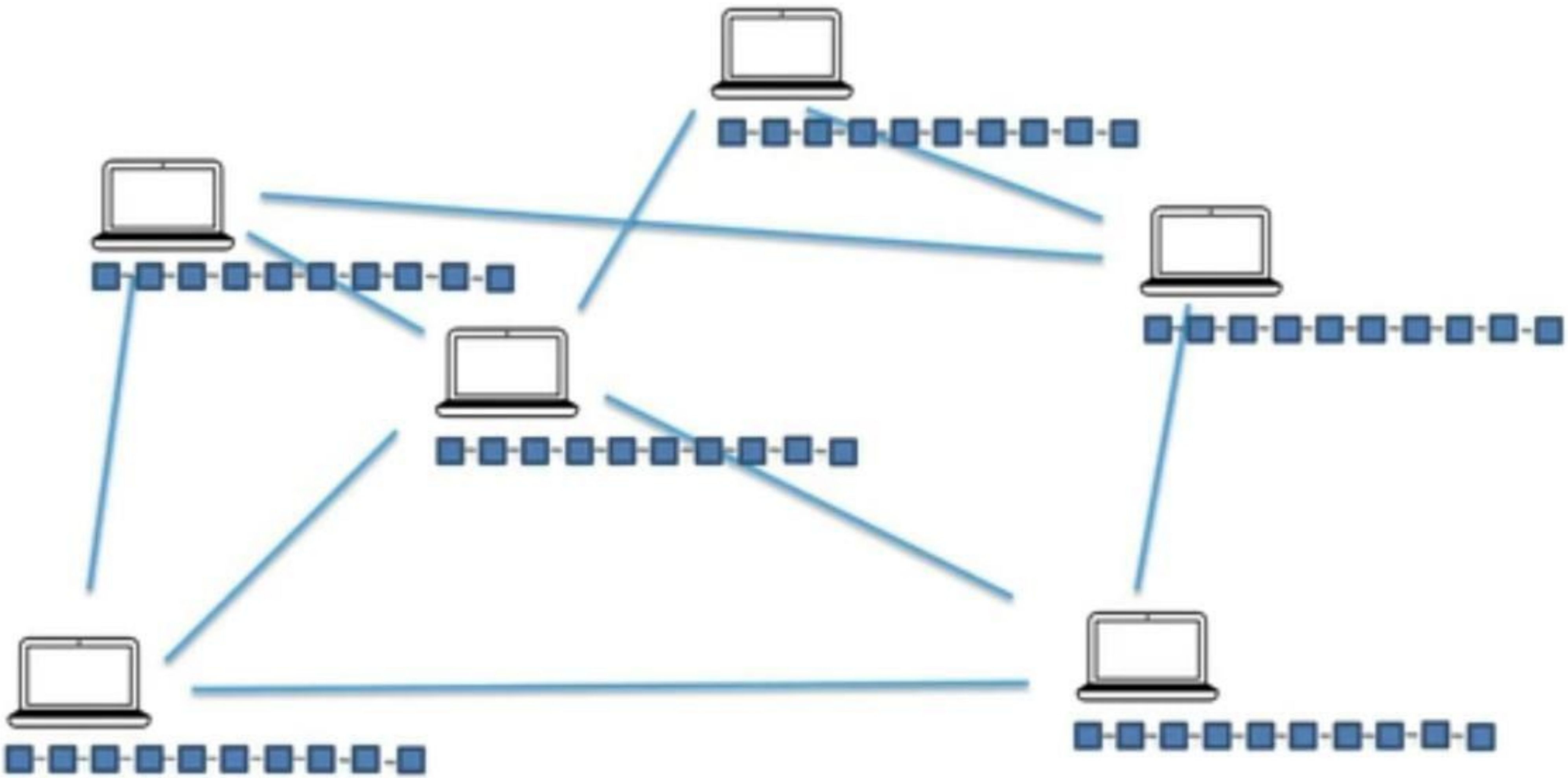




1. Check syntactic correctness
2. Reject if duplicate of block we have in any of the three categories
3. Transaction list must be non-empty
4. Block hash must satisfy claimed *nBits* proof of work
5. Block timestamp must not be more than two hours in the future
6. First transaction must be coinbase (i.e. only 1 input, with hash=0, n=-1), the rest must not be
7. For each transaction, apply "tx" checks 2-4
8. For the coinbase (first) transaction, scriptSig length must be 2-100
9. Reject if sum of transaction sig opcounts > MAX_BLOCK_SIGOPS
10. Verify Merkle hash
11. Check if prev block (matching *prev* hash) is in main branch or side branches. If not, add this t
    orphan block in *prev* chain; done with block
12. Check that *nBits* value matches the difficulty rules
13. Reject if timestamp is the median time of the last 11 blocks or before
14. For certain old blocks (i.e. on initial block download) check that hash matches known values
15. Add block into the tree. There are three cases: 1. block further extends the main branch; 2. bl
    make it become the new main branch; 3. block extends a side branch and makes it the new m
16. For case 1, adding to main branch:
    1. For all but the coinbase transaction, apply the following:
        1. For each input, look in the main branch to find the referenced output transactio
        2. For each input, if we are using the *n*th output of the earlier transaction, but it ha
        3. For each input, if the referenced output transaction is coinbase (i.e. only 1 input
           (100) confirmations; else reject.
        4. Verify crypto signatures for each input; reject if any are bad
        5. For each input, if the referenced output has already been spent by a transactio
        6. Using the referenced output transactions to get input values, check that each in
        7. Reject if the sum of input values < sum of output values
    2. Reject if coinbase value > sum of block creation fee and transaction fees




Orphaned
Block



Next Discussion Module

- What is Bitcoin?
- Bitcoin’s Monetary Policy
- Understanding Mining Difficulty
- Virtual tour of a Bitcoin Mine
- Mining Pools
- Nonce Range
- How Miners Pick Transactions (Part 1)
- How Miners Pick Transactions (Part 2)
- CPUs vs GPUs vs ASICs
- How do Mempools work?
- Orphaned blocks
- The 51% Attack

Acknowledgement and Source:

- https://www.udemy.com/course/build-your-blockchain-az/