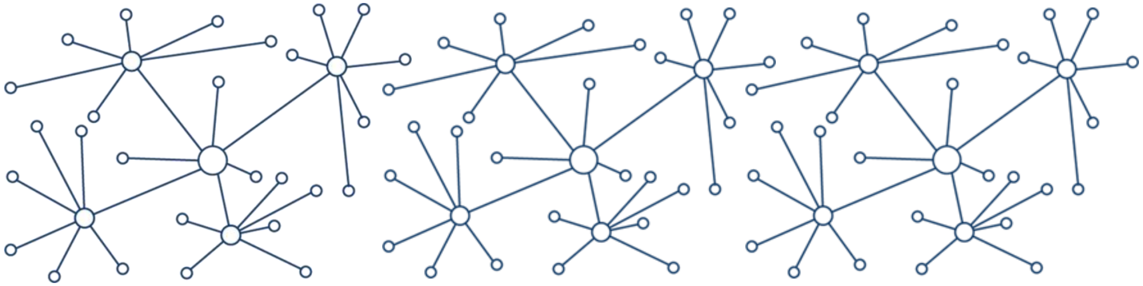


Blockchain Technology

CS423

B. Tech. IV CSE 7th Sem

Lecture#9 and 10 (6 Sept 2022)

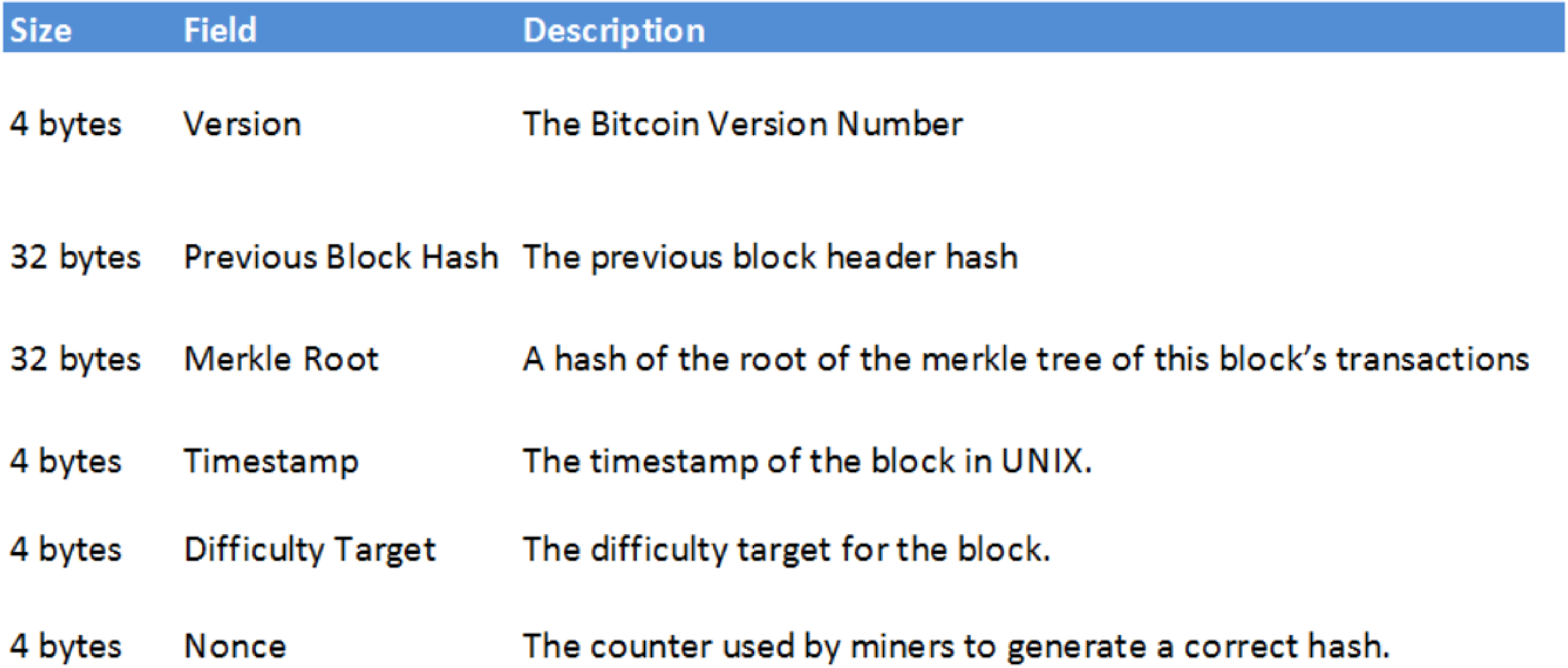
Background – Mining and Trust

Dr. Dhiren Patel

Bitcoin Blockchain

- Bitcoin components (max. supply 21 M)
- Hash function SHA256
- Puzzle to solve (making x leading bits of block hash to 0)
- Difficulty adjustment (auto – approx. every 2 weeks (time it took to find the last 2,016 blocks) to keep av. time between blocks to 10 min)
- Elliptic curve crypto - Secp256k1 is the name of the elliptic curve used by Bitcoin to implement its public key cryptography (wallets)

Blockheader in Bitcoin Block



Size	Field	Description
4 bytes	Version	The Bitcoin Version Number
32 bytes	Previous Block Hash	The previous block header hash
32 bytes	Merkle Root	A hash of the root of the merkle tree of this block's transactions
4 bytes	Timestamp	The timestamp of the block in UNIX.
4 bytes	Difficulty Target	The difficulty target for the block.
4 bytes	Nonce	The counter used by miners to generate a correct hash.

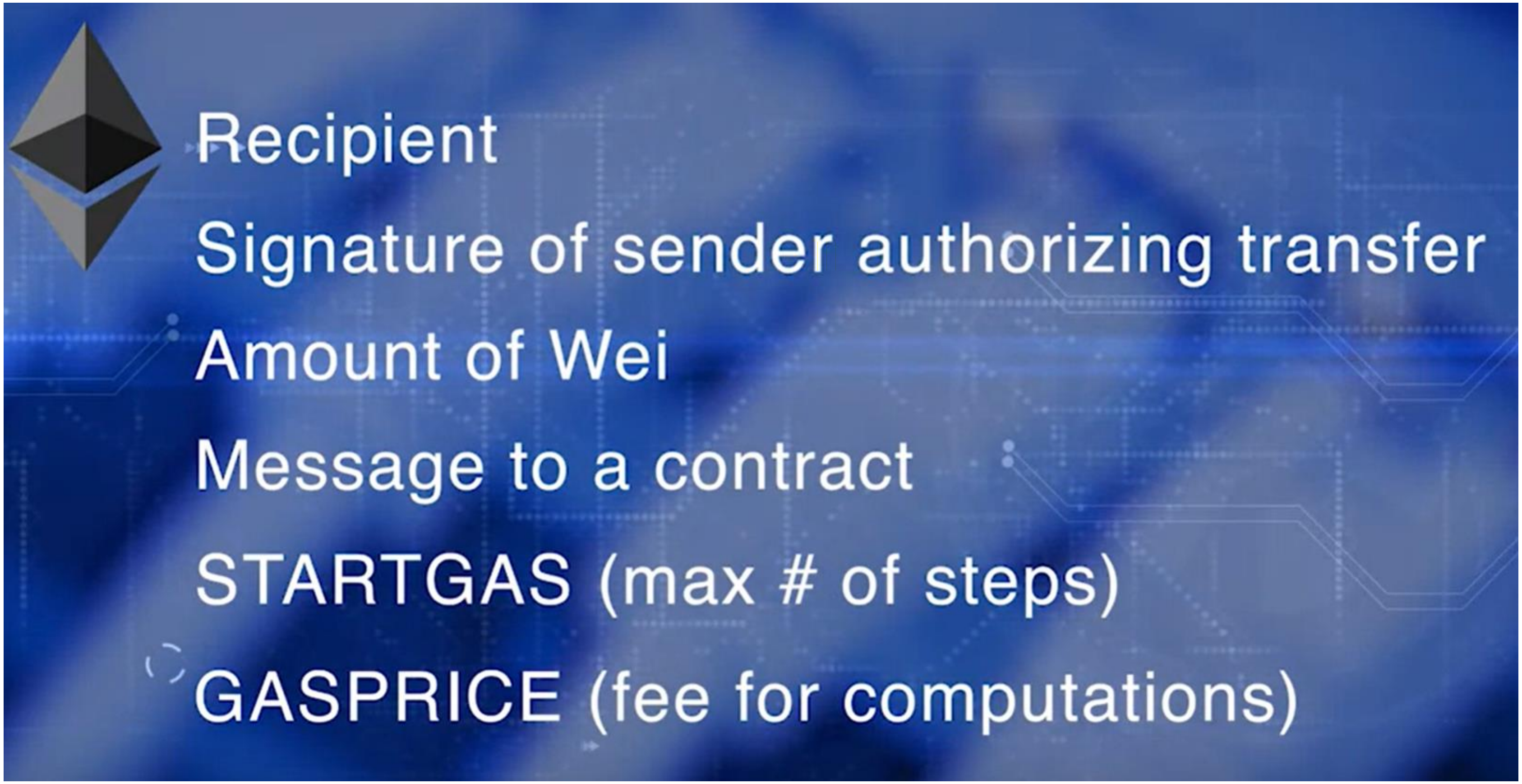
Encryption in Bitcoin Blockchain

- Two techniques are predominantly used for securing the chain and for efficient validation and verification.
- Hashing and asymmetric key encryption (PKC)
- Public-key cryptography, secure hashing, transaction integrity, and block integrity

Ethereum Structure

- Bitcoin blocking state was defined in terms of unspent transaction outputs UTXOs and a reference implementation of the Wallet application
- Ethereum formally introduce the concept of an account as a part of the protocol.
- The account is the originator and the target of a transaction. A transaction directly updates the account balances as opposed to maintaining the state such as in the bitcoin UTXOs.
- It allows for transmit of value and messages and data between the accounts that may result in the state transitions.
- These transfers are implemented using transactions.

Eth transaction



Recipient

Signature of sender authorizing transfer

Amount of Wei

Message to a contract

STARTGAS (max # of steps)

GASPRICE (fee for computations)

Observe (Ethereum Explorer)

- transaction hash, height of the chain,
- timestamp, from and to accounts,
- value transport, gas limit,
- gas used, transaction receipt,
- success in this case and nonce.

Externally Owned Accounts and Contract Accounts

- There are two types of accounts,
- Externally Owned Accounts and Contract Accounts.
- Externally Owned Accounts or EOA are controlled by private keys.
- Contract Accounts or CA are controlled by the code and can be activated only by an EOA.
- An externally owned account is needed to participate in the Ethereum network.
- It interacts with the blockchain using transactions.
- A Contract Account represents a smart contract.
- Every account has a coin balance.

Ether and Wei

- Both types of transaction require fees.
- An account must have sufficient balance to meet the fees needed for the transactions activated.
- Fees are paid in Wei. Wei is a lower denomination of Ether.
- One Ether 10 to the power of 18 Weis.
- A transaction in Ethereum includes the recipient of the message, digital signature of the sender authorizing the transfer, amount of Wei to transfer, an optional data field or payload that contains a message to a contract,
- STARTGAS which is a value representing the maximum number of computational steps the transaction is allowed.
- Gas price a value representing the fee sender is willing to pay for the computations.

Ethereum Node

- an Ethereum node is a computational system representing a business entity or an individual participant.
- An Ethereum full node hosts the software needed for transaction initiation, validation, mining, block creation, smart contract execution and the Ethereum Virtual Machine (EVM).

Smart Contract Execution

- When the target address in a transaction is a smart contract, the execution code corresponding to the smart contract is activated and executed on the EVM.
- The input needed for this execution is extracted from the payload field of the transaction.
- Current state of the smart contract is the values of the variables defined in it.
- The state of the smart contract may be updated by this execution.
- Results of this execution is told in the receipts.

Benefits of Smart Contract

- Trust
- Your documents are encrypted on a shared ledger. There’s no way that someone can say they lost it.
- Backup
- Imagine if your bank lost your savings account. On the blockchain, each and every one of your friends have your back. Your documents are duplicated many times over.

Smart Contracts are not Perfect

- What if bugs get in the code? Or how should governments regulate such contracts? Or, how would governments tax these smart contract transactions?
- Smart contracts are not reversible, meaning that if there is a problem with the contract, it can be difficult or impossible to fix.

Validation

- Transaction validation involves checking the time-stamp and the nonce combination to be valid and the availability of sufficient fees for execution.
- Miner nodes in the network receive, verify, gather and execute transactions.
- The in-work smart contract code are executed by all miners.
- Validated transactions are broadcast and gathered for block creation.

Mining

- mining is the process used to secure the network by validating the computations, collecting them to form a block, verifying them, and broadcasting it
- The proof of work puzzle winner, miner that creates a new block, is incentivized with the base fees of three Ethers, and the transaction fees in Ethereum blockchain. (Eth2 – base fees is removed/reduced).

Mining

- A trustless and distributed consensus system means that if you want to send and/or receive money from someone you don't need to trust in third-party services.
- Mining serves as two purposes:
- To verify the legitimacy of a transaction by avoiding the so-called double-spending;
- To create new digital currencies by rewarding miners for performing the previous task.

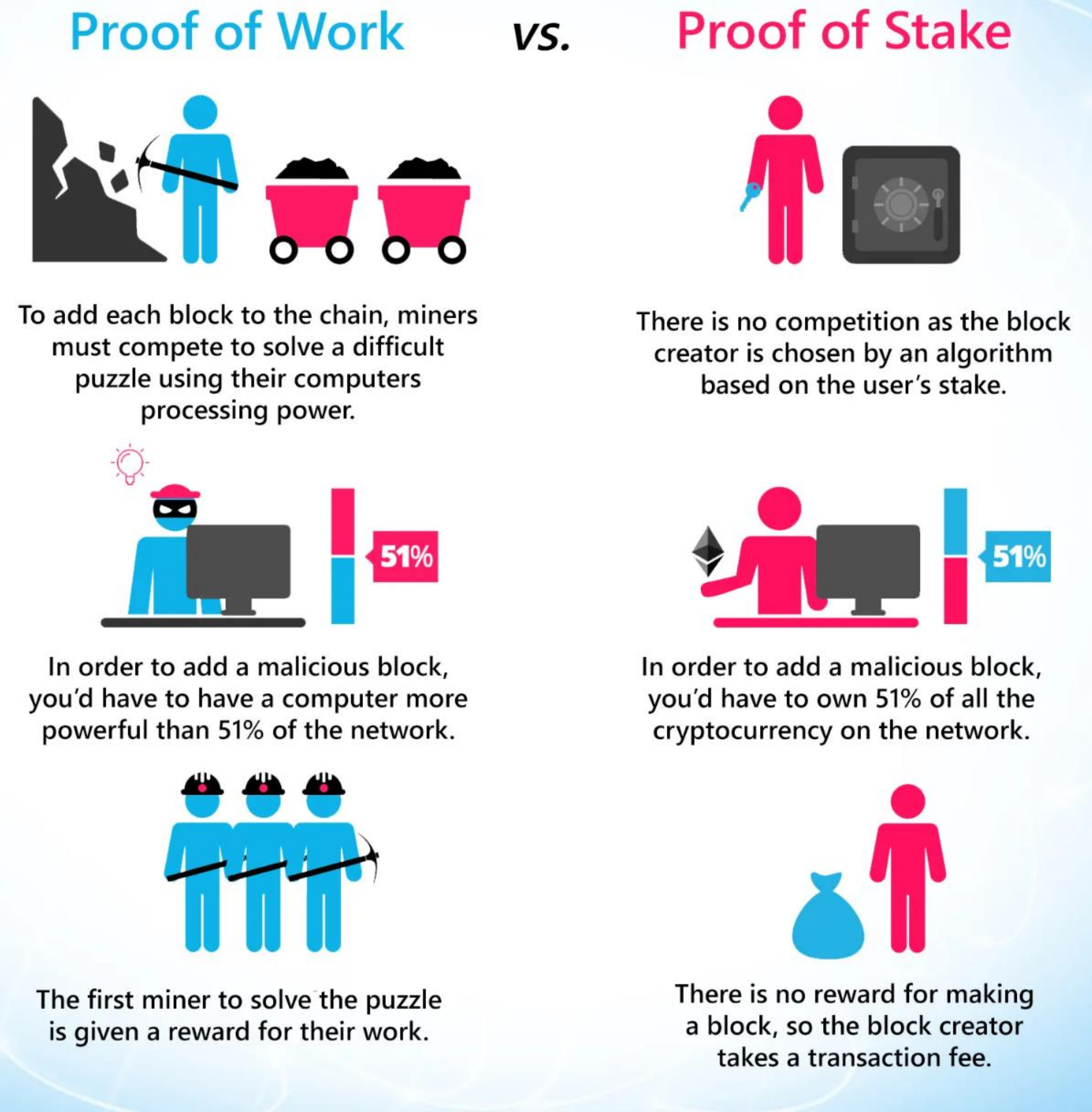
Mining

- From a technical point of view, the mining process is an operation of inverse hashing: it determines a number (nonce), so the cryptographic hash algorithm of block data results in less than a given threshold.
- This threshold, called difficulty, is what determines the competitive nature of mining

PoW v/s PoS

- In POW, the miners solve cryptographically hard puzzles by using their computational resources.
- In POS, instead of miners, there are validators. The validators lock up some of their Ether as a stake in the ecosystem. Following that, the validators bet on the blocks that they feel will be added next to the chain. When the block gets added, the validators get a block reward in proportion to their stake.
- ethereum community wants to exploit the proof of stake method for a more greener and cheaper distributed form of consensus.

I.M.P



Proof of Work vs. Proof of Stake

To add each block to the chain, miners must compete to solve a difficult puzzle using their computers processing power.

There is no competition as the block creator is chosen by an algorithm based on the user's stake.

In order to add a malicious block, you'd have to have a computer more powerful than 51% of the network.

In order to add a malicious block, you'd have to own 51% of all the cryptocurrency on the network.

The first miner to solve the puzzle is given a reward for their work.

There is no reward for making a block, so the block creator takes a transaction fee.

PoW v/s PoS

- Using a Proof-of-Work system, bad actors are cut out thanks to technological and economic disincentives.
- programming an attack to a PoW network is very expensive, and you would need more money than you can be able to steal
- the Casper protocol, a bad validator might lose their deposit. (use the set some circumstances)

What? Why?

- blockchains decentralized network participants, are not necessarily known to each other.
- Credentials cannot be checked by the conventional means such as verifying who you are with your driver's license.
- Participants can join and leave the chain as they wish.
- They operate beyond the boundaries of trust.

What? Why?

- Given this context: how do you identify the peer participants?
- How do you authorize and authenticate the transactions?
- How do you detect forged or faulty transactions?
- Private public key pair and hashing are important foundational concepts in decentralized networks that operate beyond trust boundaries.

Hashing

- What is hashing? A hash function or hashing transforms and maps an arbitrary length of input data value to a unique fixed length value.
- Input data can be a document, tree data, or a block data.
- Even a slight difference in the input data would produce a totally different hash output value.

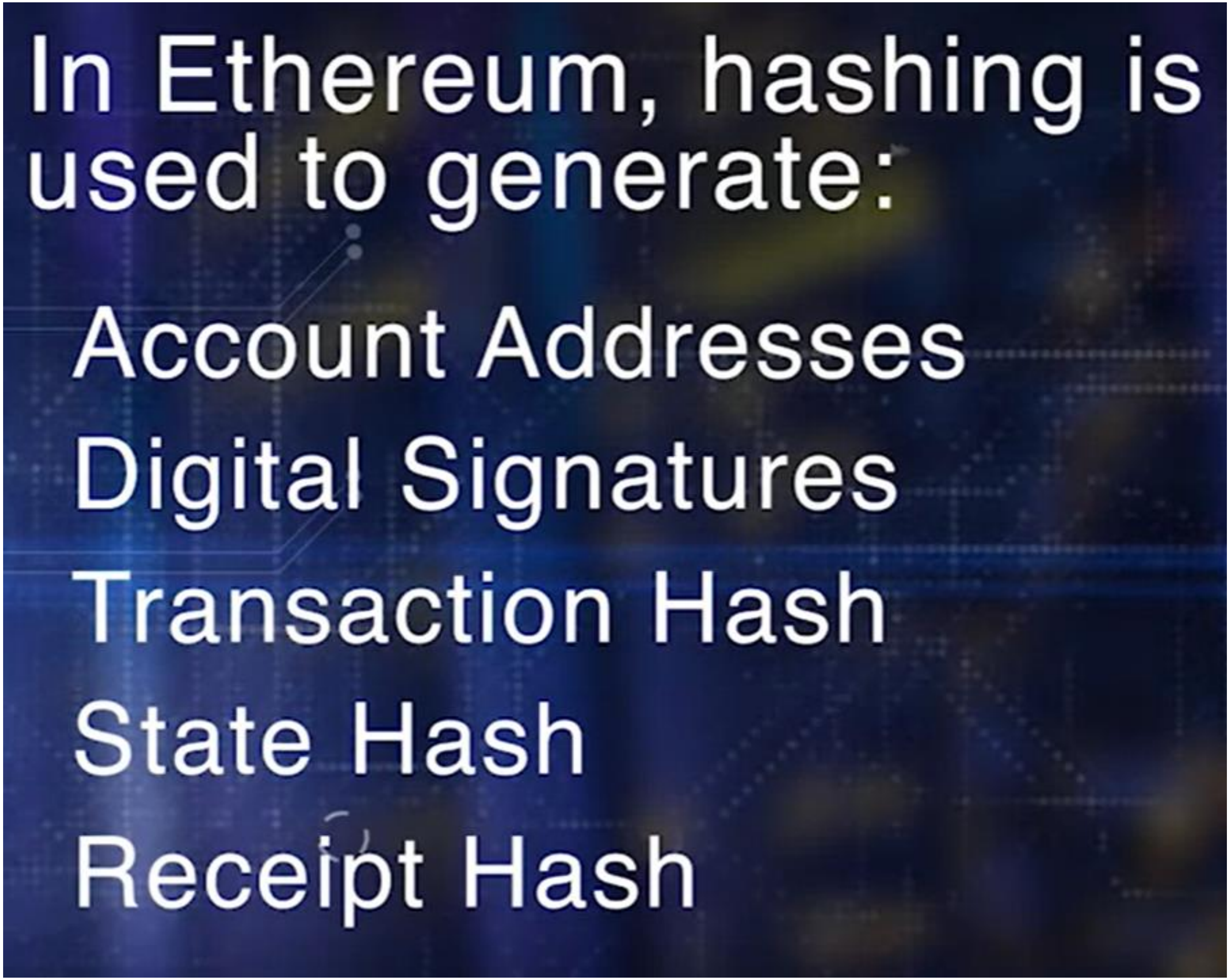
Hashing

- The algorithm chosen for the hash function should be a one-way function and it should be collision free, or exhibit extremely low probability of collision.
- The first requirement is to make certain that no one can derive the original items hashed from the hash value.
- Can you make potatoes out of mashed potatoes?
- The second requirement is to make sure that the hash value uniquely represents the original items hashed.

Hashing

- Odds of a meteor hitting your house is higher than generating two of the same hash values of 256 bits when applying this algorithm!!
- Tree structure helps the efficiency of repeated operations, such as transaction modification and the state changes from one block to the next.
- Log N versus N.

S. D.A.R.T (State Hash, Digital Signature, Account Address, Receipt Hash, Transaction Hash)



In Ethereum, hashing is used to generate:

Account Addresses

Digital Signatures

Transaction Hash

State Hash

Receipt Hash

Transaction Integrity

- To manage the integrity of a transaction we need number one, secure a unique account address.
- We need a standard approach to uniquely identify the participants in the decentralized network.
- Number two, authorization of the transaction by the sender through digital signing.
- And number three, verification that the content of that transaction is not modified. We use a combination of hashing and public key cryptography.

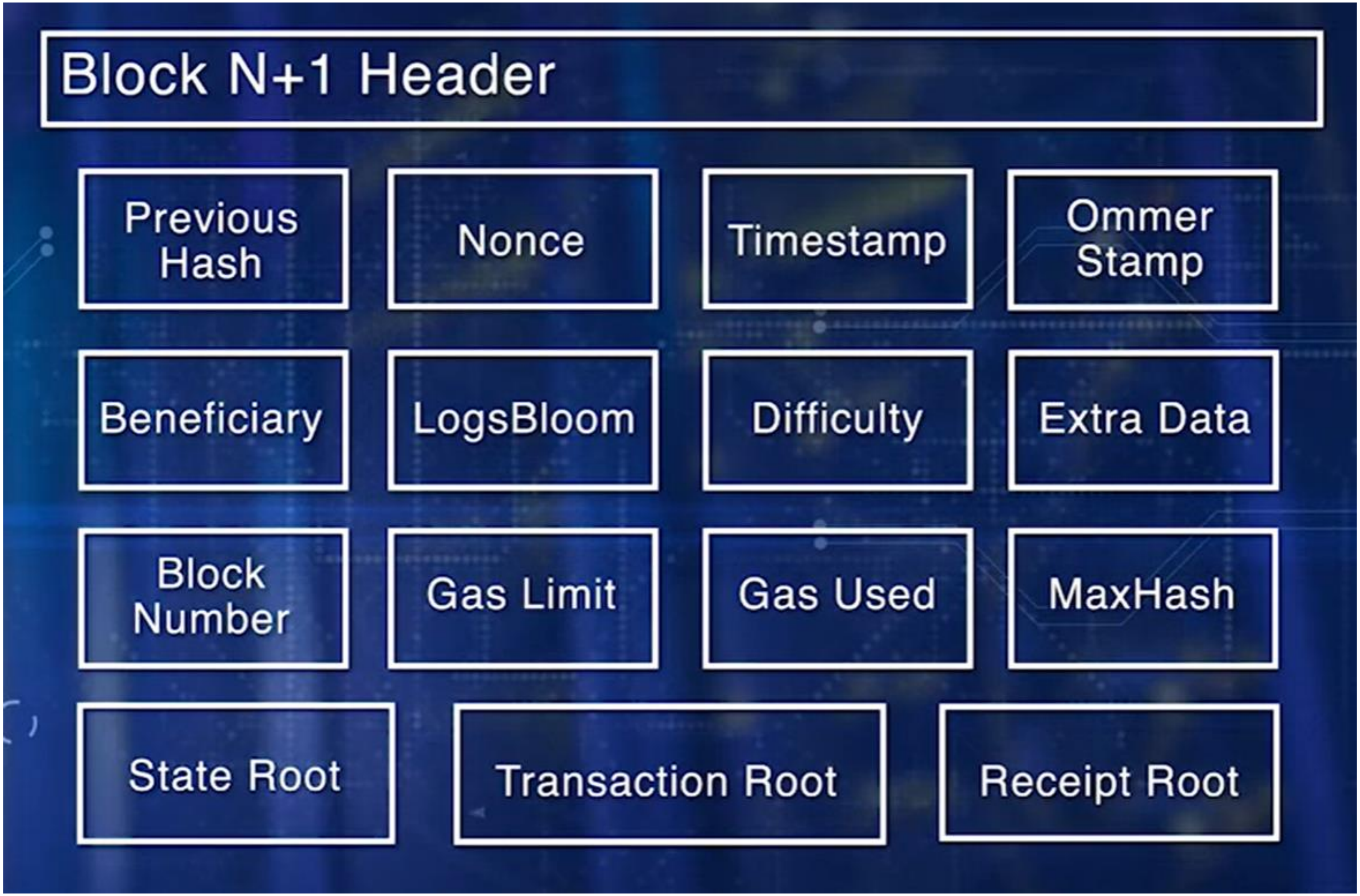
Address of Accounts

- Addresses of accounts are generated using public key, private key pair.
- Step 1, at 256-bit random number is generated, and designated as the private key.
- Kept secure and locked using a passphrase.
- Step 2, an ECC algorithm is applied to the private key, to get a unique public key.
- This is the private public key pair.
- Step 3. Then a hashing function is applied to the public key to obtain account address.
- The address is shorter in size, only 20 bytes or 160 bits.

Non-repudiable

- A transaction for transferring assets will have to be authorized, it has to be non-repudiable, and unmodifiable.
- Step number 1, find the hash of the data fields of the transaction.
- Step number 2, encrypt that hash using the private key of the participant originating the transaction.
- Thus, digitally signing the transaction to authorize and making the transaction non-repudiable.
- Step number 3, this hash just added to the transaction.
- It can be verified by others by decryiptng it using the public key of the sender of the transaction, and recomputing the hash of the transaction.
- Then, compare the computed hash, and the hash received at the digital signature.
- If that is a match, accept the transaction. Otherwise, reject it.

Ethereum Block



Block N+1 Header

Previous Hash

Nonce

Timestamp

Ommer Stamp

Beneficiary

LogsBloom

Difficulty

Extra Data

Block Number

Gas Limit

Gas Used

MaxHash

State Root

Transaction Root

Receipt Root

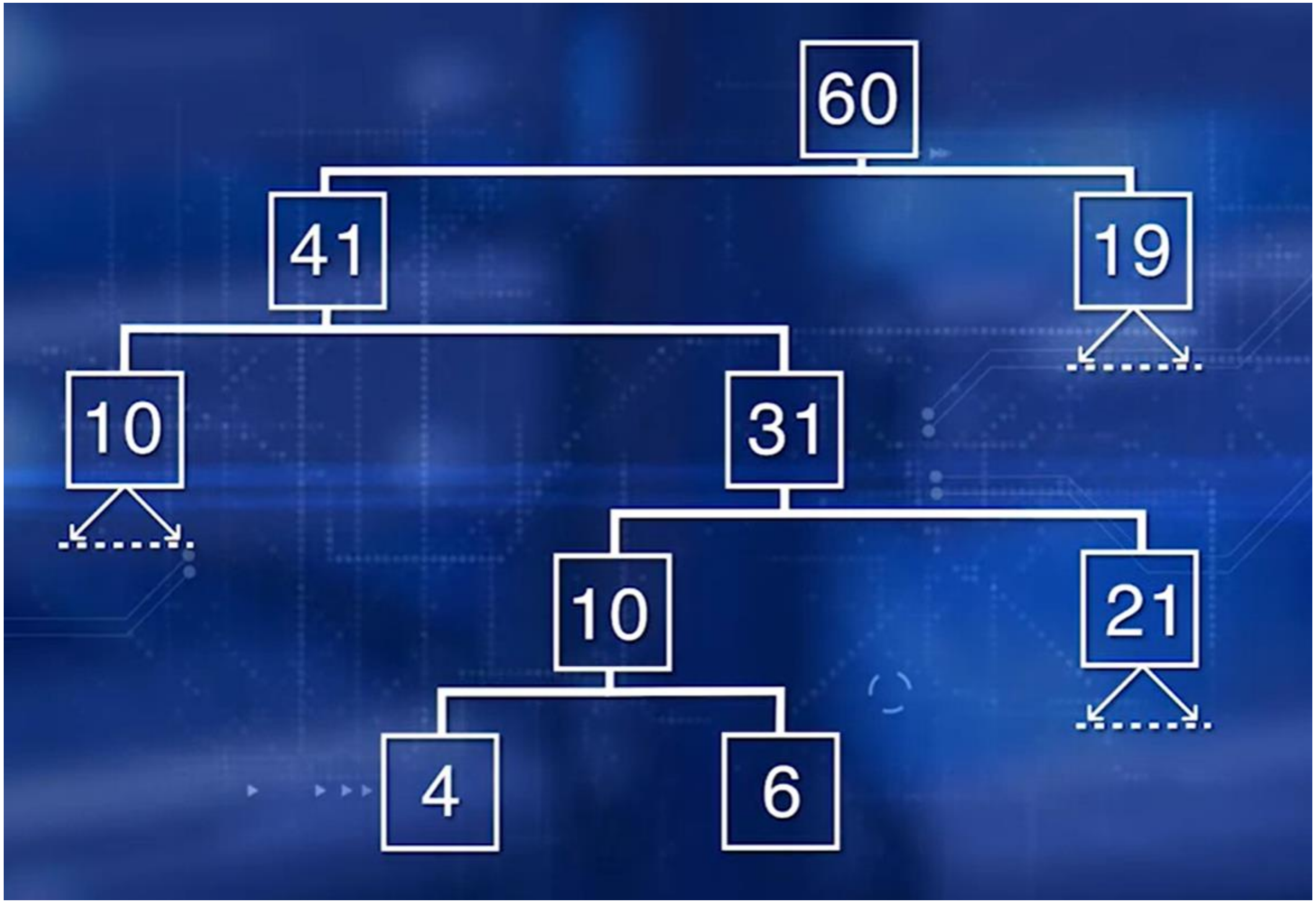
Securing Blockchain

- Integrity of the block is managed by assuring that the block header contents are not tampered with, the transactions are not tempered with, state transitions are efficiently computed, hashed, and verified.
- In Ethereum, the block hash is (the block) of all the elements in the block header, including the transaction root and state root hashes.
- It is computed by applying a variant of SHA-3 algorithm called Keccak and all the items of the block header.

Merkle Tree

- A typical block has about 2,000 transactions in bitcoin and about 100 transaction Ethereum
- Hashes of transaction in a block are processed in a tree structure called Merkle tree hash.
- Merkle tree hash is also used for computing the state root hash, since only the hash of the chained states from block to block have to be re-computed.
- It is also used for receipt hash root.
- The advantage over flat versus tree representation.
- If any transaction is to be verified, only one path to the tree has to be checked. You don't have to go through the entire set of transactions.

Merkle Tree

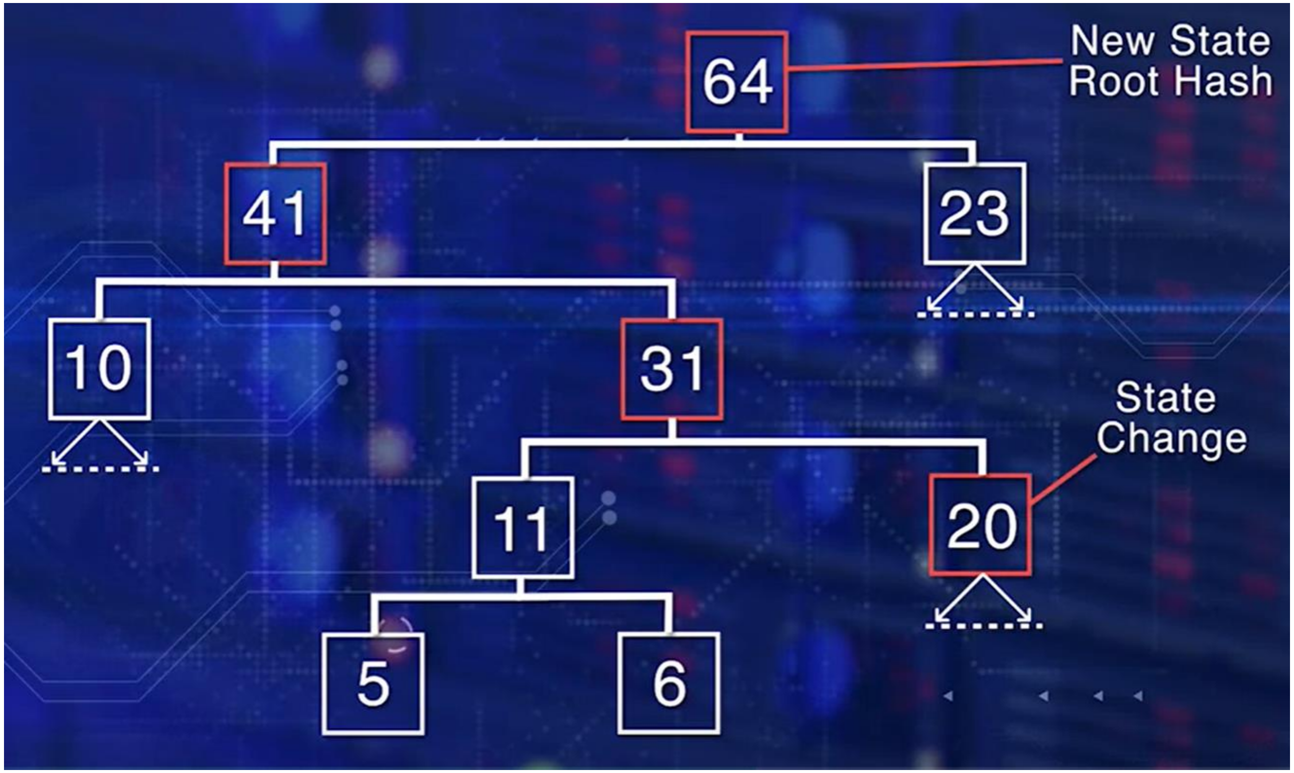


Immutability of Eth Chain

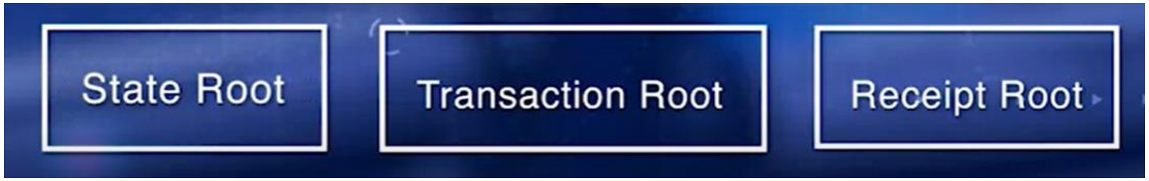
- Block hash serves two important purposes; verification of the integrity of the block and the transactions, formation of the chain link by embedding the previous block hash in the current block header.
- If any participant node tampers with the block, it's hash value changes resulting in the mismatch of the hash values and rendering the local chain of the node in an invalid state.
- Any future blocks initiated by the node would be rejected by other miners due to hash mismatch.
- This enforces the immutability of the chain.

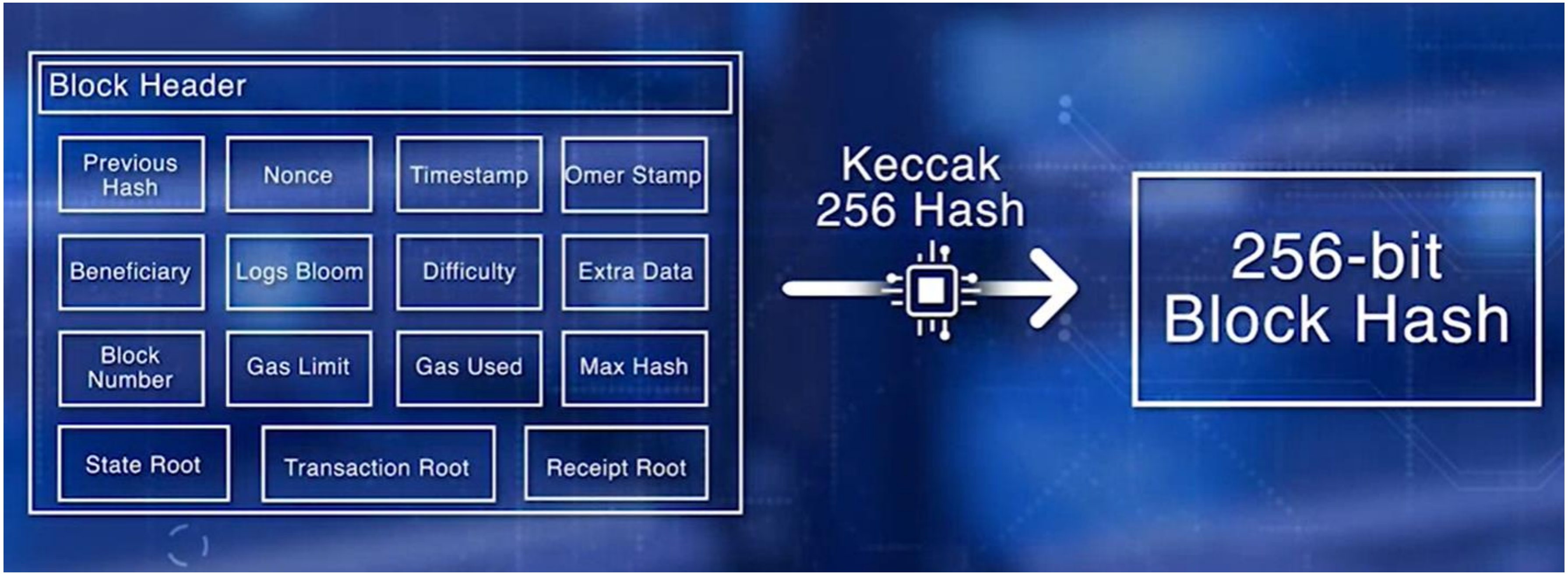
Re-computation

- Every state change requires state root (hash) re-computation



Re-computation, and update





Decentralized Autonomous Organization (DAO)

- The DAO is a leaderless, virtual organization built within a smart contract on the Ethereum blockchain.
- This smart contract sets rules that provide the ability for participants to vote on which ventures would be funded using the Ether (a crypto currency similar to Bitcoin) that each participant contributes to during the creation of the DAO.
- The larger the contribution, the larger the number of votes each participant has.

Blockchain breaches

- blockchain is inherently secure because its principles are founded on cryptography and immutability (i.e., information can be permanently stored on a public ledger without being tampered with).
- But despite its strengths and promise, blockchain is not inherently secure, and even a small oversight can have a significant impact

Trust (Centralized)

- Say, you want to fly out of the Mumbai airport.
- Entry check (only passengers (flyers) with valid tkt and travel document (passport) can enter)
- Check-in Baggage screening (of valid tkt holders)
- Airline counter check-in
- Document verification (of passengers – passport, visa) and boarding pass issue
- Immigration, Border control (passport stamping, visa, support – destination requirements)
- Security check-in (frisking and hand bag screening)
- Boarding Gate security and Aircraft Entry security
- Passenger list to destination (background check)

Trust (decentralized)???

- The airport authority would have pre-established a secure environment for people to arrive and depart.
- This establishes the base trust.
- Then there is additional trust once you - enter and your passport and travel documents are verified, validated, and your baggage is screened.
- Even more trust in you is established when the airline staff checks your boarding pass at the gate and you enter the aircraft to fly.
- < There is nobody checking your credentials and certifying that you are trustworthy. Then, how do you do it?>

Decentralized Trust

- Similar to our airport scenario, trust in a decentralized blockchain is also about securing, validating, verifying, and making sure resources needed for transaction execution are available.
- This is accomplished by securing the chain using specific protocols, validating the transaction and blocks for tamper proofing, verifying the availability of resources for transactions, and executing and confirming the transactions.

Trust

- The Trust Trail is defined by these operations: validate transaction, verify gas and resources, gather transactions, execute transaction to get a new state, form the block, work towards consensus, finalize the block by the bidder, and everyone add the block to their chain and confirm the transactions.

Forks

- debate around whether the network should permit the ability to rewrite history through a “hard fork”
- In that case - the rules of the network would have been bent for a particular scenario and would have set a dangerous precedent for the future
- blockchain is there to stay and hence its adoption will increase, secure implementation is the key!!

Security and Risk management

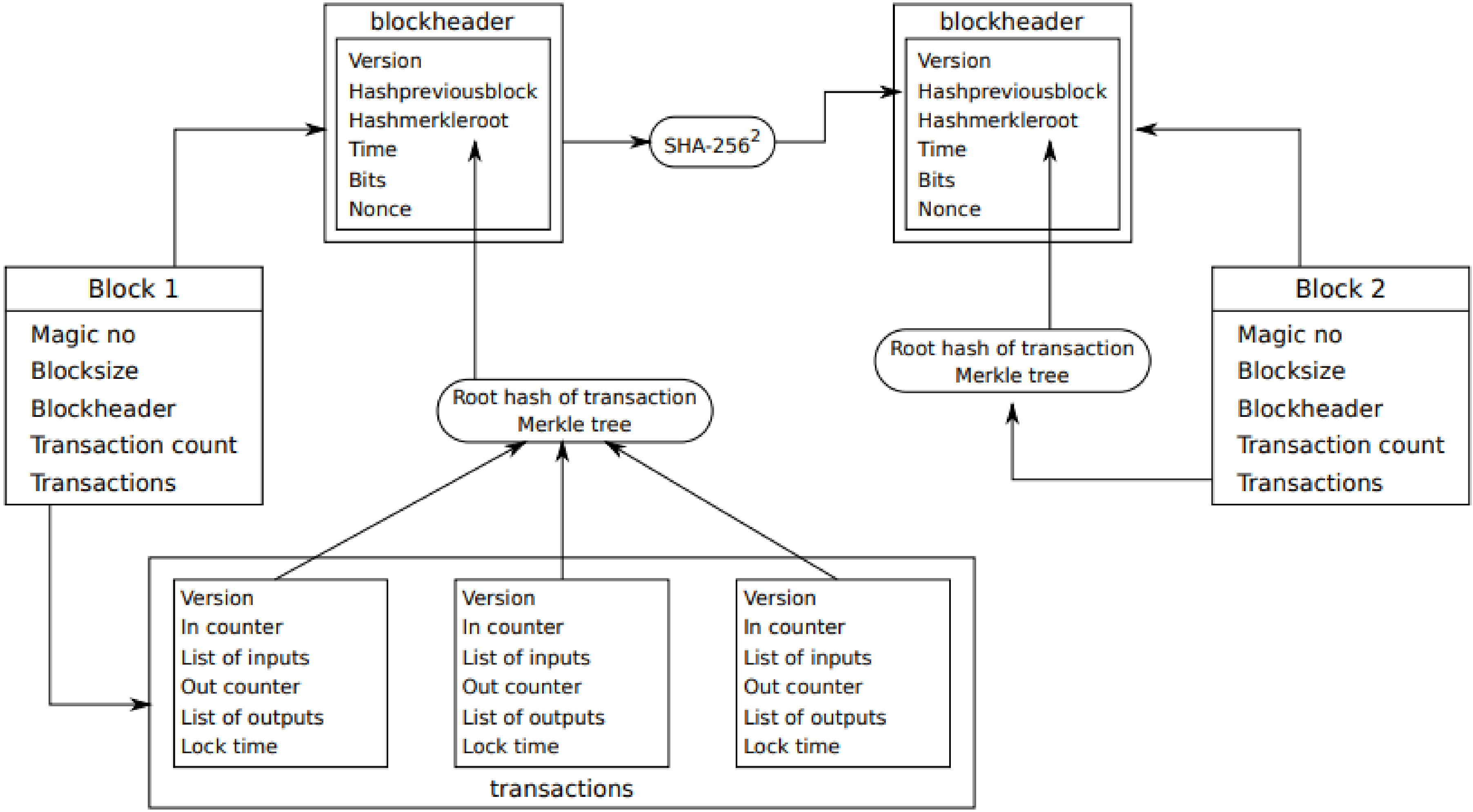
- Poor implementation — inadequate testing creates vulnerabilities in the software code.
- Unauthorized access — inappropriate access to private keys or blockchain related software could be used to steal funds or information.
- Identity management — personally identifiable information may be stolen or a node impersonated to obtain access to a blockchain.

Ethereum check

- The syntax, the transaction signature, time stamp, nonce, gas limit, and sender account balance are validated before execution.
- The fuel, or gas points, and other resources available for smart contract execution, are also validated.
- Transaction signatures and hash are also verified.
- <check - execute transactions>
- Merkle tree hash of the validated transactions is computed.
- This is in Ethereum. This is the transaction root of the block header.
- All miners execute the transaction for either transfer, as well as for execution of smart contracts.
- The state resulting from transaction execution are used in computing the Merkle tree hash of the states, the state root of the block header.
- The receipt root of the block header is also computed.

Blockchain (definition)

- Nakamoto (2008) describes the blockchain as a database modeled by a linear sequence of blocks, each one containing cryptographic hashes corresponding to the previous and current block to ensure continuity and immutability



blockheader

Version
Hashpreviousblock
Hashmerkleroot
Time
Bits
Nonce

SHA-256^2

blockheader

Version
Hashpreviousblock
Hashmerkleroot
Time
Bits
Nonce

Block 1

Magic no
Blocksize
Blockheader
Transaction count
Transactions

Root hash of transaction
Merkle tree

Root hash of transaction
Merkle tree

Block 2

Magic no
Blocksize
Blockheader
Transaction count
Transactions

Version
In counter
List of inputs
Out counter
List of outputs
Lock time

Version
In counter
List of inputs
Out counter
List of outputs
Lock time

Version
In counter
List of inputs
Out counter
List of outputs
Lock time

transactions

Why Blockchain and DLT?

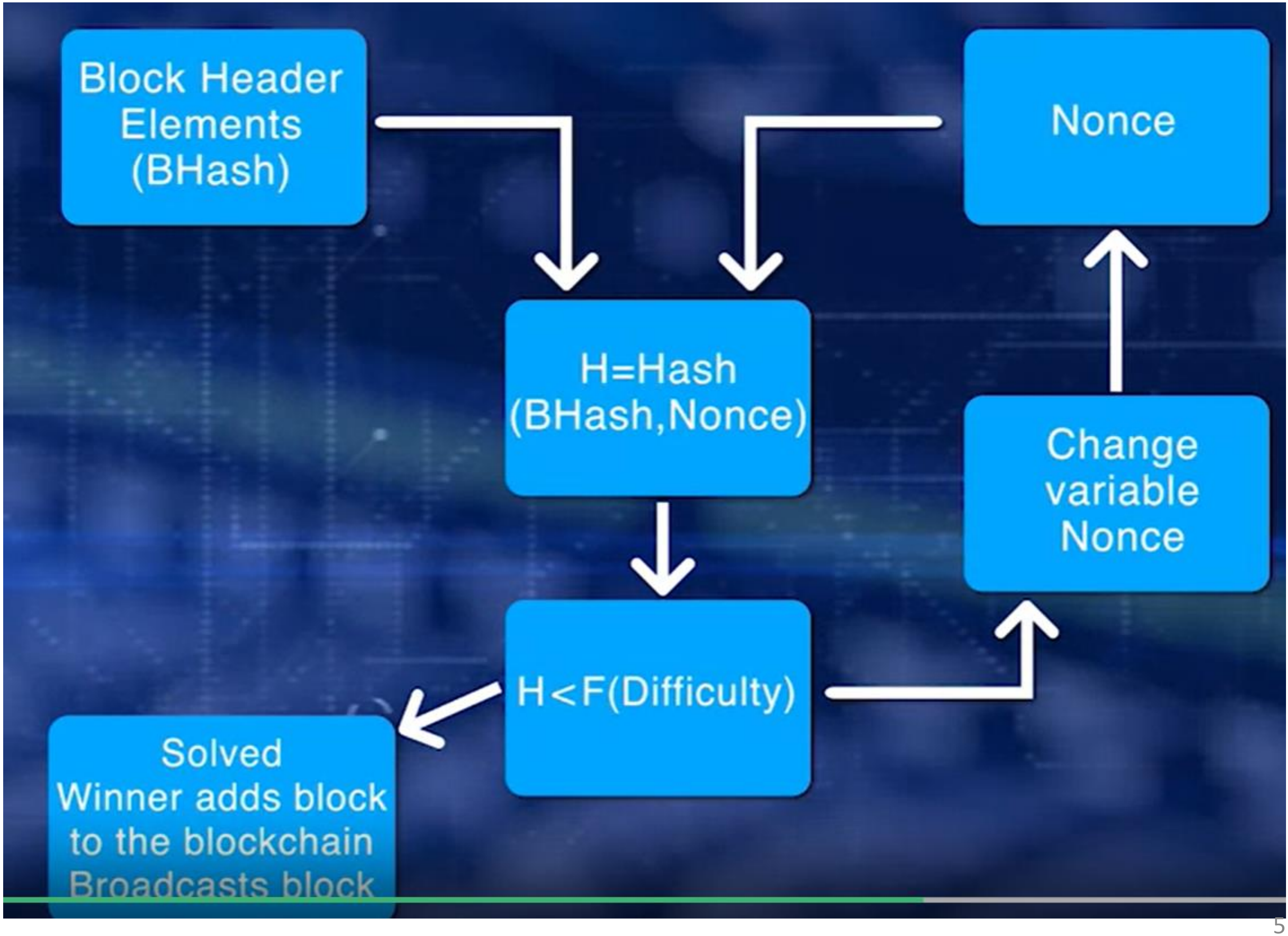
- ledger technologies could save banks $15–20 billion a year by reducing regulatory, settlement and cross-border costs (2017-2022)
- Speed and efficiency are not the only qualities that make distributed ledgers attractive to banks. ‘Regulators will like that blockchain-based transactions can achieve greater transparency and traceability– an “immutable audit trail”.

Consensus Protocol

- A secure chain is a single main chain with a consistent state.
- Every valid block added to this chain, adds to the trust level of the chain.
- What if everyone wants to add their candidate block to the chain?
- Each of the candidate blocks is by a competing miner.
- Which is the next block to be added to the chain?
- Can they agree on the next block?
- Is there a method or a protocol to choose the next block?

Proof of Work (PoW) – BTC and ETH

- Proof of Work uses hashing
- First, compute the hash of the block header elements that is a fixed value, and a nonce that is a variable.
- If hash value is less than 2^128 for bitcoin, and less than function of difficulty for ethereum, the puzzle has been solved.
- If it has not been solved, repeat the process after changing the nonce value.
- If the puzzle has been solved, broadcast the winning block that will be verified by other miners.
- Non-winning miner nodes add the new block to the local copy of the chain, and move on to working on the next block.
- The winner gets an incentive for creating the block.



Block Header Elements (BHash)

Nonce

H=Hash (BHash,Nonce)

Change variable Nonce

H<F(Difficulty)

Solved
Winner adds block to the blockchain
Broadcasts block

Trust and Robustness

- Trust is about executing regular operations correctly and managing exception satisfactory.
- Robustness is the ability to satisfactorily manage exceptional situations.

Double spending

- what if more than one miner solves the consensus puzzle where it close in time to each other?
- What if more than one transaction references as input the same digital asset?
- There's a possibility that digital currency and other consumables are single used digital assets, can be intentionally or inadvertently reused in transactions.
- This situation is called double spending.

Handling exceptions

- In a decentralized network, like a blockchain, there is no intermediary.
- We need a policy and an automatic deterministic way to handle this situation.
- A policy for handling transaction and double spending in Bitcoin is to allow the first transaction that reference the digital asset and reject the rest of the transaction that reference the same digital asset.
- There should be a well-defined processes for handling exception improve trust in the blockchain

Fork

- Background - a minor perturbation in the chain - is handled as a naturally expected occurrence within the block chain.
- On the other hand, occasionally, a minor process adjustment has to be carried out typically by bootstrapping a new software to the already running processes.
- This is soft fork. (sort of - the release of software patches)

Hard fork

- Hard fork implies a major change in the protocol.
- (sort of – a new version of operating system)
- Forks are mechanisms that add to the robustness of the blockchain framework.
- Well-managed forks help build credibility in the blockchain by providing approaches to manage unexpected faults and planned improvements.

Forks

- A **Soft Fork** is a fork where updated versions of the protocol are backwards compatible with previous versions.
- A **Hard Fork** is a change of the protocol that is not backwards compatible with older versions of the client. Participants would absolutely need to upgrade their software in order to recognize new blocks.