What Is a Blockchain?

A blockchain is a distributed database or ledger that is shared among the nodes of a computer network. As a database, a blockchain stores information electronically in digital format. Blockchains are best known for their crucial role in cryptocurrency systems, such as Bitcoin, for maintaining a secure and decentralized record of transactions. The innovation with a blockchain is that it guarantees the fidelity and security of a record of data and generates trust without the need for a trusted third party.

One key difference between a typical database and a blockchain is how the data is structured. A blockchain collects information together in groups, known as blocks, that hold sets of information. Blocks have certain storage capacities and, when filled, are closed and linked to the previously filled block, forming a chain of data known as the blockchain. All new information that follows that freshly added block is compiled into a newly formed block that will then also be added to the chain once filled.

A database usually structures its data into tables, whereas a blockchain, as its name implies, structures its data into chunks (blocks) that are strung together. This data structure inherently makes an irreversible timeline of data when implemented in a decentralized nature. When a block is filled, it is set in stone and becomes a part of this timeline. Each block in the chain is given an exact timestamp when it is added to the chain.

What Are Distributed Ledgers(खातेवही)?

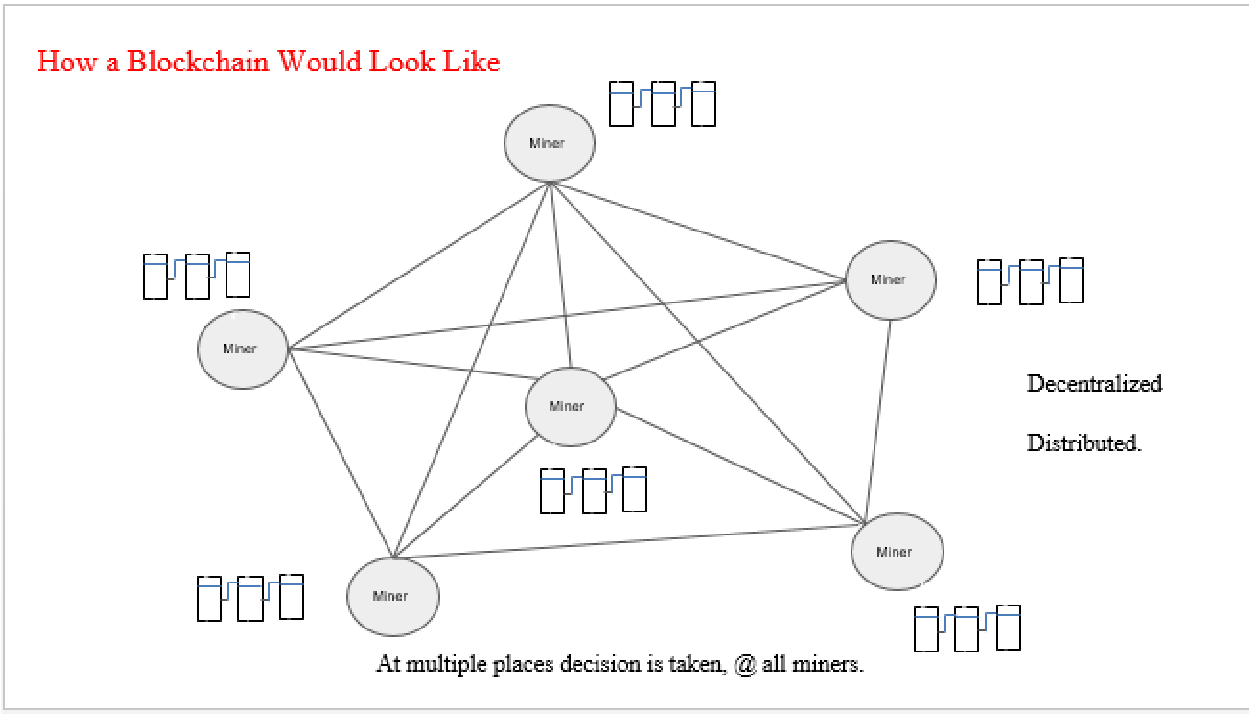
A distributed ledger is a database that is consensually(सहमतीने) shared and synchronized across multiple sites, institutions, or geographies, accessible by multiple people. It allows transactions to have public "witnesses." The participant at each node of the network can access the recordings shared across that network and can own an identical copy of it. Any changes or additions made to the ledger are reflected and copied to all participants in a matter of seconds or minutes.

All the information on the ledger is securely and accurately stored using cryptography and can be accessed using keys and cryptographic signatures. Once the information is stored, it becomes an immutable database, which the rules of the network govern.

Advantages of Distributed Ledgers

While centralized ledgers are prone to cyber attacks, distributed ledgers are inherently harder to attack because all of the distributed copies need to be attacked simultaneously for an attack to be successful. Furthermore, these records are resistant to malicious changes by a single party. By being difficult to manipulate and attack, distributed ledgers allow for extensive transparency.

Distributed ledgers also provide for an easy flow of information, which makes an audit trail easy to follow for accountants when they conduct reviews of financial statements. This helps remove the possibility of fraud occurring on the financial books of a company. The reduction in the use of paper is also a benefit to the environment


How a Blockchain Would Look Like
Miner
Miner
Miner
Miner
Miner
Miner
Decentralized
Distributed.
At multiple places decision is taken, @ all miners.

Why Bitcoin Needs Miners ?

Blockchain "mining" is a metaphor for the computational work that nodes in the network undertake in hopes of earning new tokens. In reality, miners are essentially getting paid for their work as auditors. They are doing the work of verifying the legitimacy of Bitcoin transactions. This convention is meant to keep Bitcoin users honest and was conceived by Bitcoin's founder, Satoshi Nakamoto.1 By verifying transactions, miners are helping to prevent the "double-spending problem."

- What Is Double-Spending?

Jar same currency 2 times use keli tyala double spending problem mhantat. Ha problem physical currency madhe yet nhi pn digital currency like bitcoin madhe yeu shakto.

For overcome this problem: Jar same bitcoin ni 2 transaction request keli tya case madhe Khalil pramane block chain handle karto.

Note: all transaction store in unconfirmed transaction pool and transaction picked from that pool only

1. Case1: Jya miner ne pahile transaction ghet to accept honr and dusar reject honr.
2. Case2: At a time donhi miner ne transaction ghetle tr jo pahile transaction la verify karel to confirm honr and dusara reject honr.
3. Case2: Jar donhi pn sobt verify zale mg tya transaction madhe race condition honr until ek transaction kami miners ne verify kart nhi tovr.

Double-spending is the risk that a cryptocurrency can be used twice or more. Transaction information within a blockchain can be altered if specific conditions are met. The conditions allow modified blocks to enter the blockchain; if this happens, the person that initiated the alteration can reclaim spent coins.

Note: Only 1 megabyte of transaction data can fit into a single bitcoin block. The 1MB limit was set by Satoshi Nakamoto, and this has become a matter of controversy because some miners believe the block size should increase to accommodate more data, which would effectively mean that the Bitcoin network could process and verify transactions more quickly

- Why Mine bitcoin?

  In addition to lining the pockets of miners and supporting the Bitcoin ecosystem, mining serves another vital purpose: It is the only way to release new cryptocurrency into circulation. In other words, miners are basically "minting" currency. For example, as of March 2022, there were just under 19 million bitcoins in circulation, out of a total of 21 million.

  Aside from the coins minted via the genesis block (the very first block, which founder Satoshi Nakamoto created), every single one of those bitcoins came into being because of miners. In the absence of miners, Bitcoin as a network would still exist and be usable, but there would never be any additional bitcoin. However, because the rate of bitcoin "mined" is reduced over time, the final bitcoin won't be circulated until around the year 2140. This does not mean that transactions will cease to be verified. Miners will continue to verify transactions and will be paid fees for doing so in order to keep the integrity of Bitcoin's network.

  To earn new bitcoins, you need to be the first miner to arrive at the right answer, or closest answer, to a numeric problem. This process is also known as proof of work (PoW). To begin mining is to start engaging in this proof-of-work activity to find the answer to the puzzle.

No advanced math or computation is really involved. You may have heard that miners are solving difficult mathematical problems—that's true but not because the math itself is hard. What they're actually doing is trying to be the first miner to come up with a 64-digit hexadecimal number (a "hash") that is less than or equal to the target hash. It's basically guesswork

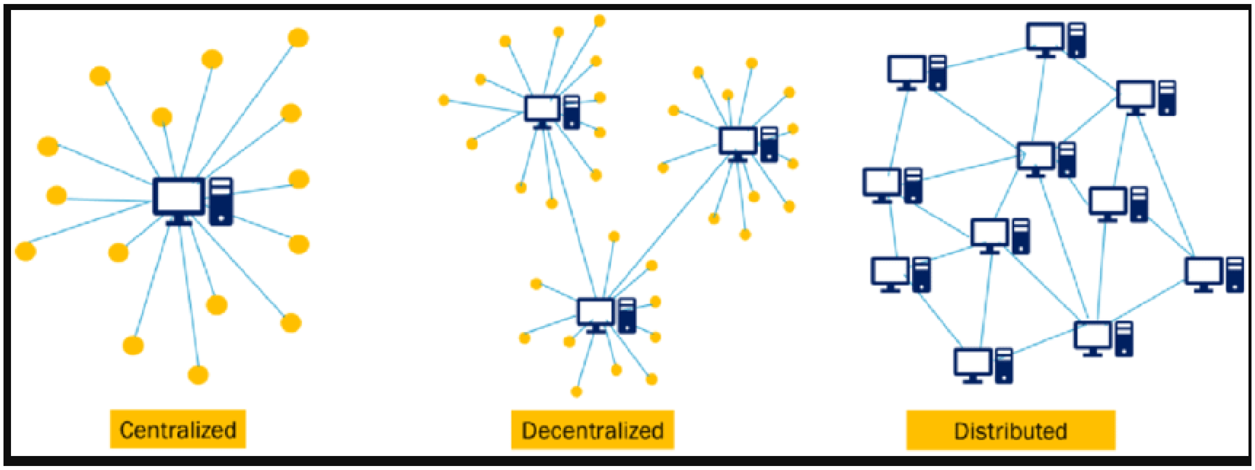
So it is a matter of randomness, but with the total number of possible guesses for each of these problems numbering in the trillions, it's incredibly arduous work. And the number of possible solutions (referred to as the level of mining difficulty) only increases with each miner that joins the mining network. In order to solve a problem first, miners need a lot of computing power. To mine successfully, you need to have a high "hash rate," which is measured in terms gigahashes per second (GH/s) and terahashes per second (TH/s)

Aside from the short-term payoff of newly minted bitcoins, being a coin miner can also give you "voting" power when changes are proposed in the Bitcoin network protocol. This is known as a Bitcoin Improvement Protocol (BIP). In other words, miners have some degree of influence on the decision-making process for matters such as forking. The more hash power you possess, the more votes you have to cast for such initiatives.

- How Much miner earn?

The rewards for Bitcoin mining are reduced by half roughly every four years. When bitcoin was first mined in 2009, mining one block would earn you 50 BTC. In 2012, this was halved to 25 BTC. By 2016, this was halved again to 12.5 BTC. On May 11, 2020, the reward halved again to 6.25 BTC.

Centralized vs Decentralized vs Distributed Netwrok



Discussions about blockchain technology frequently refer to networks as "decentralized," "distributed," and "centralized." But what is a decentralized network? Is it different from a distributed network? And what advantages do these network designs have over centralized networks?

Blockchain is decentralized, distributed technology that collects a digital record of any event and store it in a distributed database that is shared among all the users connected with it.

What Is a Centralized Network?

- centralized networks are built around a single, centralized server/master node, which handles all major data processing and stores data and user information that other users can access. From there, client nodes can be connected to the main server and submit data requests instead of performing them directly. The majority of web services — including YouTube, a mobile app store, or your online banking account — are coordinated by a centralized network owner, meaning that all data transactions within these networks require verification via a third-party authority.
- Centralized networks are currently the most widely used type of network on the web. These networks are dependent on a central network owner to connect all the other satellite users and devices — which means there is a single point of failure that can be deliberately exploited by malicious actors.

What Is a Decentralized Network?

- By contrast, a decentralized network distributes information-processing workloads across multiple devices instead of relying on a single central server. Each of these separate devices serves as a mini central unit that interacts independently with other nodes. As a result, even if one of the master nodes crashes or is compromised, the other servers can continue providing data access to users, and the overall network will continue to operate with limited or zero disruption.
- Decentralized networks are made possible by recent technological advancements that have equipped computers and other devices with a significant amount of processing power and can be synced up and leveraged for distributed processing. However, while decentralized networks are substantially different from centralized networks, it’s important to note that decentralized networks do not distribute data storage and processing evenly across the entire network and still rely on main servers, albeit more than one per network.
- Advantages
    - Increased flexibility/scalability: Since decentralized networks do not have a single point of failure, they can continue to operate even if a master node is compromised or shut down. Furthermore, decentralized networks are easy to scale since you can simply add more devices to the network in order to increase its computing power, and network maintenance typically does not necessitate a full network shutdown.
    - Faster performance: User requests are often completed faster when using a decentralized network because network administrators can create master nodes in regions where user activity is high, as opposed to routing connections over vast expanses to a single centralized server.
    - Enhanced privacy: Decentralized networks enable a greater degree of user privacy, since information saved on the network is disseminated across multiple

    points instead of passing through a single point. This makes data flows more difficult to track across a network, and eliminates the risks of having a single target malicious actors can go after.

- Disadvantages
    - High maintenance costs: Decentralized networks are more fault-tolerant than centralized networks. This makes maintaining these networks typically more costly and labor-intensive. Since a decentralized network relies on multiple devices to underpin the system, this places a commensurate burden on an organization's IT resources. As a result, decentralized systems are often not suitable for organizations that only require a small system, since the cost/benefit ratio isn't favorable under these circumstances.
    - Coordination issues: Since master nodes within a decentralized network act independently and may not communicate with one another, larger organizations may run into coordination issues and have a difficult time directing and achieving collective tasks. While this is a deliberate feature of decentralized networks, it means that not all business models and organizational structures will necessarily benefit from using a decentralized network.

What Is a Distributed Network?

- A distributed network is similar to a decentralized network in the sense that it forgoes a single centralized master server in favor of multiple network owners. However, distributed networks are composed of equal, interconnected nodes, meaning that data ownership and computational resources are shared evenly across the entire network. The term "distributed network" is sometimes used to describe a network that is simply geographically distributed but may follow a top-down node hierarchy model. In most instances, though, the term refers to a network where node locations and computational resources are evenly distributed.
- Because distributed networks do not have a central server or a separate set of master nodes, the burden of data processing is crowdsourced across the network, with all users granted equal access to data. The decision-making process on a distributed network therefore typically involves individual nodes voting to change to a new state, and the final behavior of the system changes in accordance with the aggregate results of the decisions each individual node votes on. The specific processes by which a distributed network votes and makes decisions is contingent on the network's consensus mechanism. All forms of distributed decision-making involve the network's individual components interacting with one another in order to achieve a common goal.

- Due to their geographically scattered nature, distributed networks are consequently extremely fault-tolerant and secure. Their advantages and disadvantages closely mirror those of decentralized networks, but at a higher magnitude.
- Advantages
  - Extreme fault tolerance: With distributed networks, a node can fail independently without affecting the rest of the system, since the computational workload will simply be rebalanced among the remaining nodes. As a result, distributed data systems are significantly more robust than other network architectures that rely on some form of top-down node hierarchy.
  - Speed and scalability: Distributed networks are more scalable than both centralized and decentralized networks. They generally exhibit lower latency as well due to the even distribution of network processing power and data.
  - Enhanced transparency: Since data within a distributed network is shared evenly across the entire network, it is significantly harder to successfully modify, censor, or destroy information on the network. As a result, distributed networks are intrinsically more transparent than other systems, particularly given the fact that they often utilize cryptography to secure their data
- Disadvantages
  - High maintenance costs: As is the case with decentralized networks, distributed networks require more resources to maintain or reconfigure, since any meaningful change requires updating every individual node. And, since the distributed nodes have different latencies and do not follow a common clock, network administrators cannot temporally order commands or logs. As a result, it can be difficult to design and debug algorithms for a distributed network.
  - Coordination Issues: In the absence of a node hierarchy, there are no superior nodes overseeing the behavior of subordinate nodes, and consequently there is no way to regulate individual nodes on the system. It can therefore be difficult to make timely decisions or achieve large-scale tasks. This decentralized chain of command can be an insurmountable issue for certain businesses and organizations. Furthermore, since it is difficult for any individual node to gain a global view of the entire network, it is therefore harder for individual nodes to make well-informed decisions based on the state of other nodes in the system.

CAP Theorem

- In theoretical computer science, the CAP theorem states that it is impossible for a distributed data store (such as a blockchain network) to simultaneously provide more than two out of the three guarantees: Consistency, Availability & Partition tolerance.
- CAP
  - C: Consistency — At any given time, all nodes in the network have exactly the same (most recent) value.
  - A: Availability — Every request to the network receives a response, though without any guarantee that returned data is the most recent.
  - P: Partition tolerance — The network continues to operate, even if an arbitrary number of nodes are failing.
- Due to the nature of distributed data stores (such as blockchain), Partition tolerance is a given fact; there will always be failing/unreachable nodes in the network (not least because of the unstable nature of the internet). CAP Theorem states that one has to choose between C (Consistency) or A (Availability) when in the presence of P (Partition):
- Distributed System can only guarantees 2 out of 3 properties.
- CAP Theorem is violated in Blockchain. In Blockchain, consistency is sacrificed in favor of availability and partition tolerance.Consistency (C) on the Blockchain is not achieved simultaneously with Partition Tolerance (P) and Availability (A), but it is achieved over time. This is called eventual consistency.

Generic Elements of the Blockchain

Addresses: Addresses are unique identifiers that are used in a transaction on the blockchain to denote senders and recipients. An address is usually a public key or derived from a public key. While addresses can be reused by the same user, addresses themselves are unique. In practice, however, a single user may not use the same address again and generate a new one for each transaction. This newly generated address will be unique. Bitcoin is in fact a pseudonymous system. End users are usually not directly identifiable but some research in de-anonymizing bitcoin users have shown that users can be identified successfully. As a good practice it is suggested that users generate a new address for each transaction in order to avoid linking transactions to the common owner, thus avoiding identification.

Transaction: *A transaction is the fundamental unit of a blockchain. A transaction represents a transfer of value from one address to another.*

Block: *A block is composed of multiple transactions and some other elements such as the previous block hash (hash pointer), timestamp, and nonce.*

Peer-to-peer network: *As the name implies, this is a network topology whereby all peers can communicate with each other and send and receive messages.*

Scripting or programming language: *This element performs various operations on a transaction. Transaction scripts are predefined sets of commands for nodes to transfer tokens from one address to another and perform various other functions. Turing complete programming language is a desirable feature of blockchains; however, the security of such languages is a key question and an area of important and ongoing research.*

Virtual machine: *This is an extension of a transaction script. A virtual machine allows Turing complete code to be run on a blockchain (as smart contracts) whereas a transaction script can be limited in its operation. Virtual machines are not available on all blockchains; however, various blockchains use virtual machines to run programs, for example* ***Ethereum Virtual Machine*** *(**EVM**) and* ***Chain Virtual Machine*** *(**CVM**).*

Nodes: *A node in a blockchain network performs various functions depending on the role it takes. A node can propose and validate transactions and perform mining to facilitate consensus and secure the blockchain. This is done by following a consensus protocol. (Most commonly this is PoW.) Nodes can also perform other functions such as simple payment verification (lightweight nodes), validators, and many others functions depending on the type of the blockchain used and the role assigned to the node.*

Smart contracts: *These programs run on top of the blockchain and encapsulate the business logic to be executed when certain conditions are met. The smart contract feature is not available in all blockchains but is now becoming a very desirable feature due to the flexibility and power it provides to the blockchain applications.*

Nodes

- Nodes act as communication endpoints, allowing users to interact with and within the network.
- The Purpose of Blockchain Nodes
    - Maintaining the Blockchain:
        - Since blockchains are decentralized ledgers, there is no one centralized server where their database can be stored. But every blockchain network continuously grows as more blocks keep getting added to the chain.
        - Not only does a blockchain need a decentralized storage space to keep all the data, but it should also be scalable. This requirement is fulfilled by using blockchain nodes. All decentralized networks, such as Ethereum and Bitcoin, store their blockchain data on several thousand nodes across the globe.

        - Each individual node maintains its own copy of the blockchain and keeps syncing it as new blocks are minted
    - Validating a Transaction
        - Being decentralized also means that there is no centralized authority to check and approve transaction orders. This job is also done by the nodes on the blockchain.
        - Some nodes participate in the consensus algorithm to verify transactions' validity, while others are responsible for just storing transaction records.
        - The entire procedure includes nodes receiving a transaction order, checking its authenticity, approving or rejecting it, and then recording it on the ledger.
        - Many blockchain networks also involve the validator nodes sharing the proposed decision regarding the transactions with other nodes to maintain the consensus.
    - Accessing Information
        - All decentralized blockchains are transparent, meaning users can access all information without barriers. To access any information on the blockchain, they have to interact with nodes.
        - When you use a third-party blockchain explorer, such as Etherscan to explore information on transactions on the Ethereum blockchain, you are communicating with the blockchain nodes to access the data.
- How do Blockchain nodes work?
    - As mentioned before, being decentralized, blockchains need a mechanism to approve transactions and make decisions regarding the network. This is achieved by all nodes working together simultaneously and communicating with each other.
    - Each blockchain has its own set of consensus rules. In some blockchains, the approval of one validator node is enough to process the transaction. But some require the approval of the entire committee of nodes to reach a decision.
    - Similarly, when there is a need to change the working mechanism of a network, all the nodes connected to the blockchain cast their votes.
    - Approval of the majority of nodes is needed to carry out the change/upgrade. For example, the recent Ethereum London Hard Fork upgrade was done after the decision received over 85% votes.
- Types of Blockchain Nodes
    - Light Nodes/ SPV
        - Lightweight nodes or "light nodes" do not hold full copies of the blockchain. Light nodes only download blockheaders, saving users significant download time and storage space. Nodes of this nature depend on full nodes to function and are used for simplified payment verification (SPV).
    - Archival/Full Nodes
        - Most often, when someone uses the term "full node," they are referring to an archival full node. This is the primary node type that forms the backbone of a blockchain network. Archival full nodes are servers that host the entire blockchain, with every single transaction recorded in their databases. The main task of these nodes is to validate blocks and maintain consensus.

        - Archival nodes can be broken down further into two subcategories: nodes that can add blocks to the chain and those that cannot.
    - Pruned Full Nodes
        - A pruned full node is one that saves hard disk space for its users by "pruning" older blocks in the blockchain. This type of node will first have to download the entire blockchain from the beginning. After that, it will begin deleting blocks beginning with the oldest and continue until the node only holds the most recent transactions up to a set size limit. If a node operator were to set the size limit to 250 MB, then a pruned full node would hold the most recent 250 MB worth of transactions.
    - Mining Nodes
        - In crypto mining, miners are either full or light nodes that try to prove they've completed the work required to create a new block. This is where the term "proof-of-work" comes from. To accomplish this task, miners have to either be an archival full node themselves, or get data from other nodes to learn the current status of the blockchain and how to work on finding the next block. (Those who seek to run mining nodes might want to take into account crypto mining electricity costs.)
    - Users - Who takes service from the Blockchain using DApps

Consensus Algorithms in Blockchain

- A consensus algorithm is a procedure through which all the peers of the Blockchain network reach a common agreement about the present state of the distributed ledger. In this way, consensus algorithms achieve reliability in the Blockchain network and establish trust between unknown peers in a distributed computing environment. Essentially, the consensus protocol makes sure that every new block that is added to the Blockchain is the one and only version of the truth that is agreed upon by all the nodes in the Blockchain. The Blockchain consensus protocol consists of some specific objectives such as coming to an agreement, collaboration, co-operation, equal rights to every node, and mandatory participation of each node in the consensus process. Thus, a consensus algorithm aims at finding a common agreement that is a win for the entire network. Now, we will discuss various consensus algorithms and how they work.
- Types
    1. Proof of Work (PoW):
        - This consensus algorithm is used to select a miner for the next block generation. Bitcoin uses this PoW consensus algorithm. The central idea behind this algorithm is to solve a complex mathematical puzzle and easily give out a solution. This mathematical puzzle requires a lot of computational power and thus, the node who solves the puzzle as soon as possible gets to mine the next block. For more details on PoW, please read Proof of Work (PoW) Consensus
        - Proof of work (PoW) is a decentralized consensus mechanism that requires members of a network to expend effort solving an arbitrary mathematical puzzle to prevent anybody from gaming the system.

    - Proof of work is used widely in cryptocurrency mining, for validating transactions and mining new tokens.
    - Due to proof of work, Bitcoin and other cryptocurrency transactions can be processed peer-to-peer in a secure manner without the need for a trusted third party.
    - Proof of work at scale requires huge amounts of energy, which only increases as more miners join the network.
    - Proof of Stake (POS) was one of several novel consensus mechanisms created as an alternative to proof of work.
    - How Does Proof of Work Validate a Crypto Transaction?

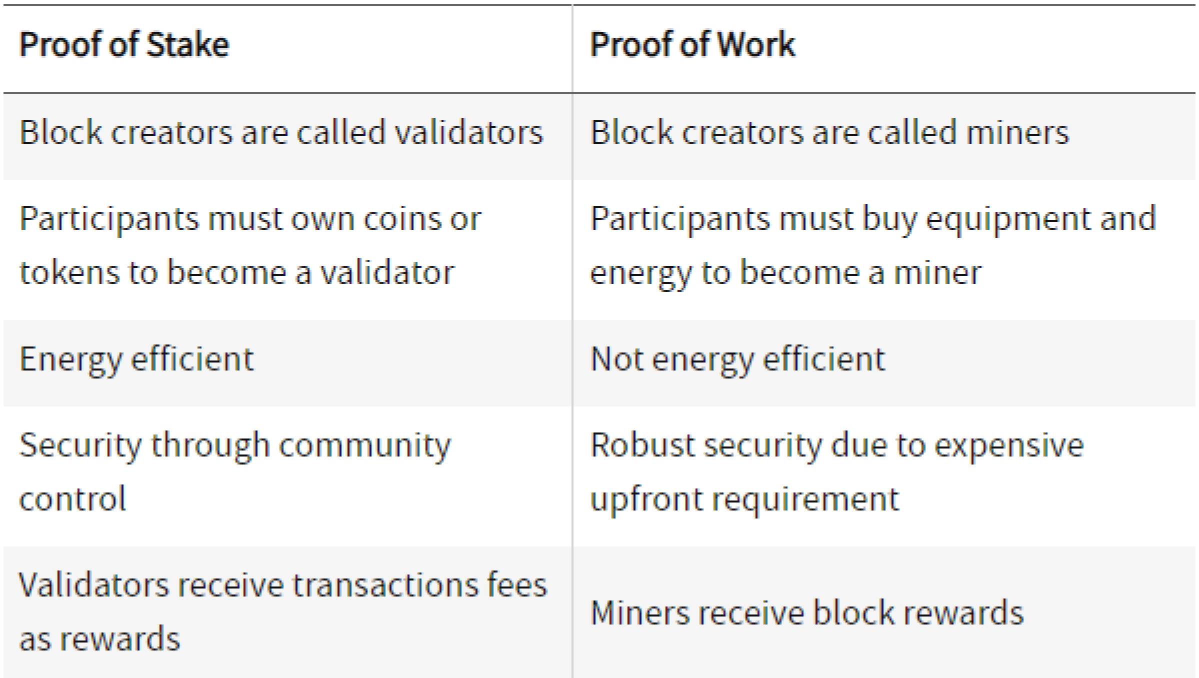
      The work itself is arbitrary. For Bitcoin, it involves iterations of SHA-256 hashing algorithms. The "winner" of a round of hashing, however, aggregates and records transactions from the mempool into the next block. Because the "winner" is randomly-chosen proportional to the work done, it incentivizes everybody on the network to act honestly and record only true transactions.

2. Proof of Stake (PoS): This is the most common alternative to PoW. Ethereum has shifted from PoW to PoS consensus. In this type of consensus algorithm, instead of investing in expensive hardware to solve a complex puzzle, validators invest in the coins of the system by locking up some of their coins as stake. After that, all the validators will start validating the blocks. Validators will validate blocks by placing a bet on it if they discover a block which they think can be added to the chain. Based on the actual blocks added in the Blockchain, all the validators get a reward proportionate to their bets and their stake increase accordingly. In the end, a validator is chosen to generate a new block based on their economic stake in the network. Thus, PoS encourages validators through an incentive mechanism to reach to an agreement.
    - In Proof stake miner should assign some stake for validation. If he mined unauthorized block then there stake or eth will gone.

As a security point of view if Long touted as a threat for cryptocurrency fans, the 51% attack is a concern when PoS is used, but there is doubt it will occur. Under PoW, a 51% attack is when an entity controls more than 50% of the miners in a network and uses that majority to alter the blockchain. In PoS, a group or individual would have to own 51% of the staked cryptocurrency.

It's very expensive to control 51% of staked cryptocurrency. Under Ethereum's PoS, if a 51% attack occurred, the honest validators in the network could vote to disregard the altered blockchain and burn the offender(s) staked ETH. This incentivizes validators to act in good faith to benefit the cryptocurrency and the network.

.



Proof of Stake	Proof of Work
Block creators are called validators	Block creators are called miners
Participants must own coins or tokens to become a validator	Participants must buy equipment and energy to become a miner
Energy efficient	Not energy efficient
Security through community control	Robust security due to expensive upfront requirement
Validators receive transactions fees as rewards	Miners receive block rewards

For avoid many attack PoW check the computation power of each miner that required to mine a perticular block. If power comsumption is low then there is posibility that that mined block is compromised block.

Types of Blockchains

1. Public Blockchain
   a. A public blockchain is a non-restrictive, permission-less distributed ledger system. Anyone who has access to the internet can sign in on a blockchain platform to become an authorized node and be a part of the blockchain network. A node or user which is a part of the public blockchain is authorized to access current and past records, verify transactions or do proof-of-work for an incoming block, and do mining. The most basic use of public blockchains is for mining and exchanging cryptocurrencies. Thus, the most common public blockchains are Bitcoin and Litecoin blockchains. Public blockchains are mostly secure if the users strictly follow security rules and methods. However, it is only risky when the participants don’t follow the security protocols sincerely.
   b. Example: Bitcoin, Ethereum, Litecoin
   c. Advantages: Trustable ,Secure,Open and Transparent
   d. Disadvantages: Lower TPS(transactions per second), Scalability Issues , High Energy Consumption

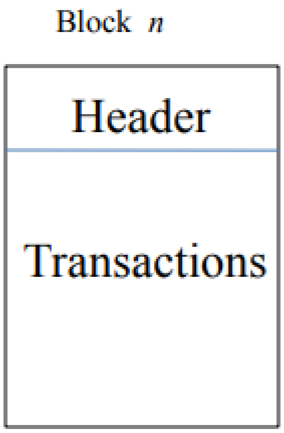
2. Private Blockchain

   a. A private blockchain is a restrictive or permission blockchain operative only in a closed network. Private blockchains are usually used within an organization or enterprises where only selected members are participants of a blockchain network. The level of security, authorizations, permissions, accessibility is in the hands of the controlling organization. Thus, private blockchains are similar in use as a public blockchain but have a small and restrictive network. Private blockchain networks are deployed for voting, supply chain management, digital identity, asset ownership, etc.
   b. Examples of private blockchains are; Multichain and Hyperledger projects (Fabric, Sawtooth), Corda, etc.
   c. Advantages: Speed ,Scalability
   d. Disadvantages: Needs Trust-building,Lower Security , Centralization

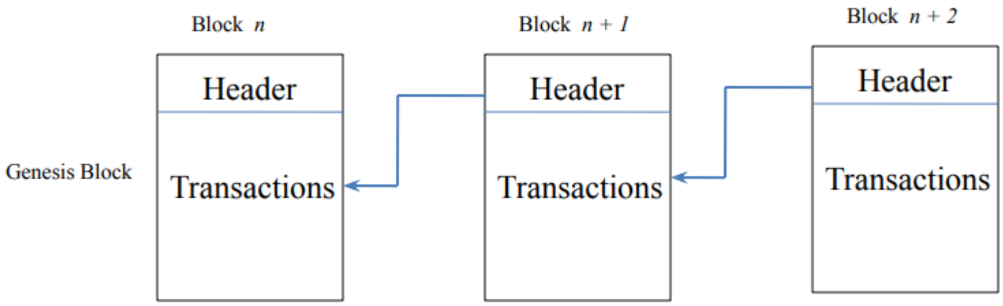
3. Consortium Blockchain
   a. A consortium blockchain is a semi-decentralized type where more than one organization manages a blockchain network. This is contrary to what we saw in a private blockchain, which is managed by only a single organization. More than one organization can act as a node in this type of blockchain and exchange information or do mining. Consortium blockchains are typically used by banks, government organizations, etc.
   b. Examples of consortium blockchain are; Energy Web Foundation, R3, etc.
4. Hybrid Blockchain
   a. A hybrid blockchain is a combination of the private and public blockchain. It uses the features of both types of blockchains that is one can have a private permission-based system as well as a public permission-less system. With such a hybrid network, users can control who gets access to which data stored in the blockchain. Only a selected section of data or records from the blockchain can be allowed to go public keeping the rest as confidential in the private network. The hybrid system of blockchain is flexible so that users can easily join a private blockchain with multiple public blockchains. A transaction in a private network of a hybrid blockchain is usually verified within that network. But users can also release it in the public blockchain to get verified. The public blockchains increase the hashing and involve more nodes for verification. This enhances the security and transparency of the blockchain network.
   b. Example of a hybrid blockchain is Dragonchain.

BlockChain Structure

- Block chain contains blocks of chain
- Each block contain mainly two info
  - Header
  - Transaction

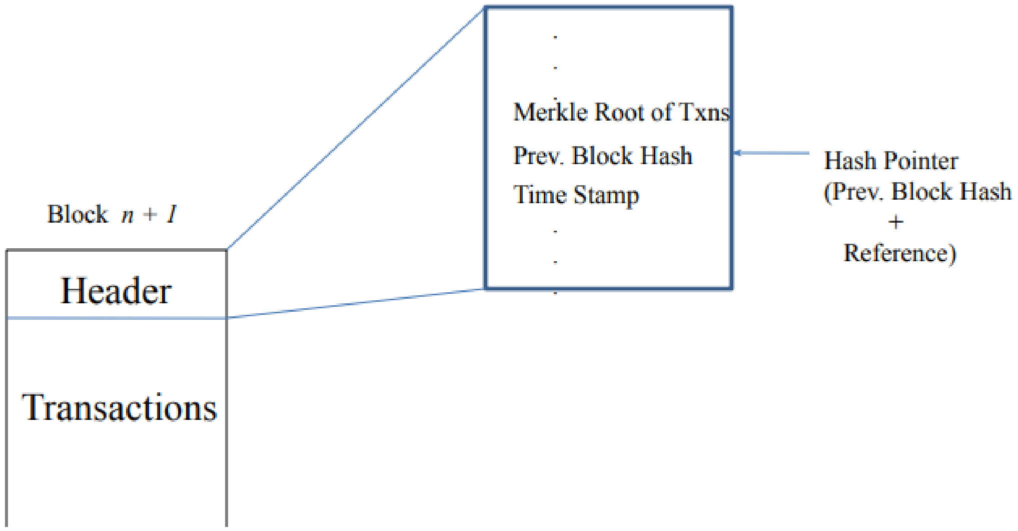


- Block chain is look like a linked list .
- All block connected to each other in reverser manner means “n+1” connect directly to “n” block not n+2 block.

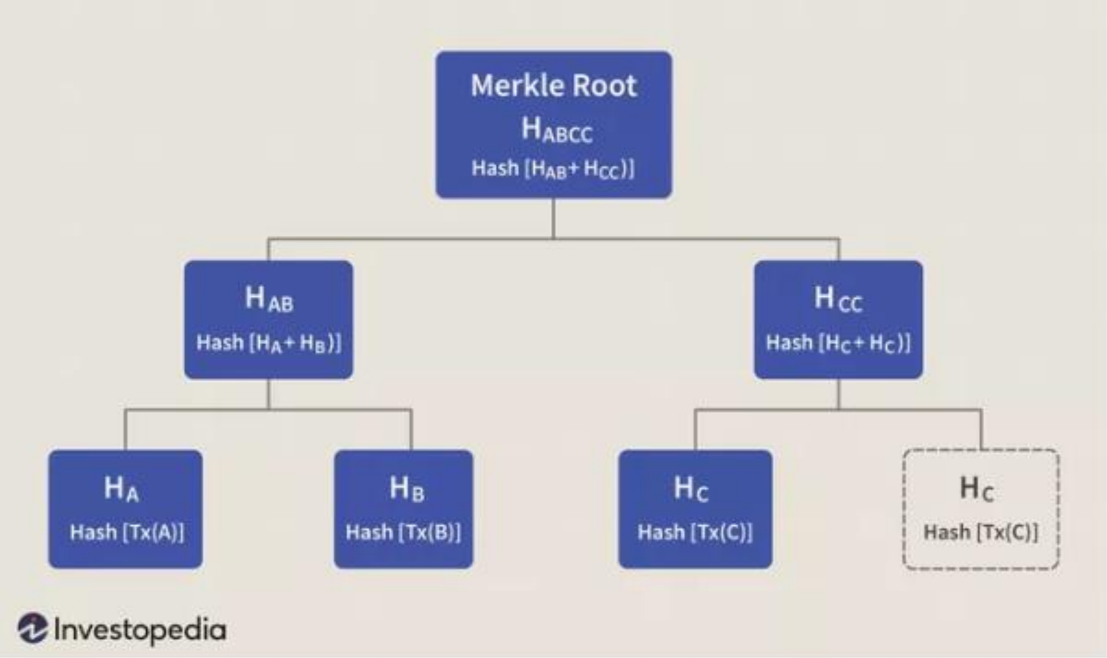


Chain of Blocks – Blockchain

- Access of the block is always possible backward direction. E.g. if you create a new block i.e. “n+2” then this block can access only block “n,n+1”. Even there is “n+3” block is available.
- Inside the blockchain header
  - When discussing the structure of the blockchain, it is often described as a series of blocks that are linked together in a way that protects them against modification. However, it is only the headers of the blocks that are actually linked together in this way.

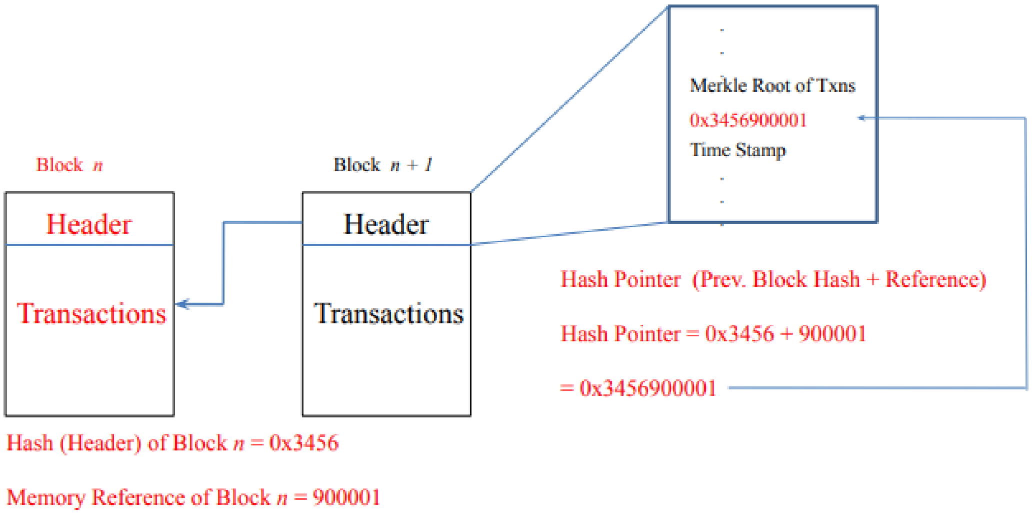


- Header part contain following infor
  - Merkle root of transaction
  - Previous Block hash
  - Time stamp
- Markle root
  - A Merkle root is the hash of all the hashes of all the transactions that are part of a block in a blockchain network.
  - A blockchain is comprised of various blocks that are linked with one another (hence the name blockchain). A hash tree, or the Merkle tree, encodes the blockchain data in an efficient and secure manner. It enables the quick verification of blockchain data, as well as quick movement of large amounts of data from one computer node to the other on the peer-to-peer blockchain network.
  - Every transaction occurring on the blockchain network has a hash associated with it. However, these hashes are not stored in a sequential order on the block, rather in the form of a tree-like structure such that each hash is linked to its parent following a parent-child tree-like relation.
  - Since there are numerous transactions stored on a particular block, all the transaction hashes in the block are also hashed, which results in a Merkle root.
  - For example, consider a seven-transaction block. At the lowest level (called the leaf-level), there will be four transaction hashes. At the level one above the leaf-level, there will be two transaction hashes, each of which will connect to two hashes that are below them at the leaf level. At the top (level two), there will be the last transaction hash called the root, and it will connect to the two hashes below it (at level one).
  - Effectively, you get an upside-down binary tree, with each node of the tree connecting to only two nodes below it (hence the name "binary tree"). It has one root hash at the top, which connects to two hashes at level one, each of which again connects to the two hashes at level three (leaf-level), and the structure continues depending upon the number of transaction hashes.



- The hashing starts at the lowest level (leaf-level) nodes, and all four hashes are included in the hash of nodes that are linked to it at level one. Similarly, hashing continues at level one, which leads to hashes of hashes reaching to higher levels, until it reaches the single top root hash.
- This root hash is called the Merkle root, and due to the tree-like linkage of hashes, it contains all the information about every single transaction hash that exists on the block. It offers a single-point hash value that enables validating everything present on that block.

- Previous Block Hash
  - We compute hash of each block and store in next block header as a previous hash to make connection between them
- Timestamp
  - One of the main uses of timestamp is to establish the parameters of the process of mining is.. This is because these timestamps allow nodes to correctly adjust the mining difficulty to be used for each block generation period. Timestamps help the network determine how long it takes to extract blocks for a certain period, and from there adjust the mining difficulty parameter.
  - Timestamp in the blockchain is used as proof that the particular block is used at what instance of a time, also this timestamp is used as a parameter to verify the authenticity of any block.
- What is Hash Pointer?
  - A regular pointer stores the memory address of data. With this pointer, the data can be accessed easily. On the other hand, a hash pointer is a pointer to where data is stored and with the pointer, the cryptographic hash of the data is also stored. So a hash pointer points to the data and also allows us to verify the data. A hash pointer can be used to build all kinds of data structures such as blockchain and Merkle tree.

    - 
Block n
Block n + 1
Header
Transactions
Header
Transactions
Merkle Root of Txns
0x3456900001
Time Stamp
Hash Pointer (Prev. Block Hash + Reference)
Hash Pointer = 0x3456 + 900001
= 0x3456900001
Hash (Header) of Block n = 0x3456
Memory Reference of Block n = 900001

- Why Hashing ?
    - A hash is a mathematical function that converts an input of arbitrary length into an encrypted output of a fixed length. Thus regardless of the original amount of data or file size involved, its unique hash will always be the same size. Moreover, hashes cannot be used to "reverse-engineer" the input from the hashed output, since hash functions are "one-way" (like a meat grinder; you can't put the ground beef back into a steak). Still, if you use such a function on the same data, its hash will be identical, so you can validate that the data is the same (i.e., unaltered) if you already know its hash.
    - A hash is a function that meets the encrypted demands needed to solve for a blockchain computation.
    - Hashes are of a fixed length since it makes it nearly impossible to guess the length of the hash if someone was trying to crack the blockchain.
    - The same data will always produce the same hashed value.
    - A hash, like a nonce or a solution, is the backbone of the blockchain network.
    - A hash is developed based on the information present in the block header.
    - Hashes are used in several parts of a blockchain system. First, each block contains the hash of the block header of the previous block, ensuring that nothing has been tampered with as new blocks are added
    - It usage SHA256, SHA512
        - If we write the same text again in a data section, it will always give the same output. It is because you are creating a message digest of that one's specific amount of data.
    - Once we did hashed then hash output act as a signature, if content is changed then signature is not match.
- How many transactions we can include in a block ?
    - It all depends on the block size.
- What could be the block size ?

    - It depends on the application
- Bitcoin
    - Bitcoin - 1MB
    - Bitcoin XT (ExTended) - 1 MB - 8 MB
    - Bitcoin Classic - 2 MB
    - Bitcoin Unlimited
    - Bitcoin Cash & Bitcoin Gold - 2MB
- Ethereum block size
    - A final important note is that blocks themselves are bounded in size. Each block has a target size of 15 million gas but the size of blocks will increase or decrease in accordance with network demands, up until the block limit of 30 million gas (2x target block size). The total amount of gas expended by all transactions in the block must be less than the block gas limit. This is important because it ensures that blocks can't be arbitrarily large. If blocks could be arbitrarily large, then less performant full nodes would gradually stop being able to keep up with the network due to space and speed requirements. The larger the block, the greater the computing power required to process them in time for the next slot. This is a centralizing force, which is resisted by capping block sizes.
- WHAT IS GAS?
    - Gas refers to the unit that measures the amount of computational effort required to execute specific operations on the Ethereum network.
    - Since each Ethereum transaction requires computational resources to execute, each transaction requires a fee. Gas refers to the fee required to conduct a transaction on Ethereum successfully.
    - Gas fees are paid in Ethereum's native currency, ether (ETH). Gas prices are denoted in gwei, which itself is a denomination of ETH - each gwei is equal to 0.000000001 ETH (10-9 ETH).

Revolution vs Evolution?

- Blind Signature - Contents are Blinded and Signed
    - Others can only verify the signature and not the content.
- E-Cash
    - eCash was a digital-based system that facilitated the transfer of funds anonymously. A pioneer in cryptocurrency, its goal was to secure the privacy of individuals that use the Internet for micropayments
    - eCash was an electronic platform created to transfer funds anonymously. It was a pioneer in cryptocurrency.
    - eCash was created by Dr. David Chaum and implemented via his company, DigiCash, in 1990.
    - eCash worked on the premise of blind signatures where message content is invisible before signing, resulting in no user being able to link withdrawal and spend transactions.
    - Despite the initial interest and bringing large banks on board to use the system

    - The core concept behind eCash was blind signatures. A blind signature is a type of digital signature in which the message's content is invisible prior to signing. In this manner, no user is able to create a link between withdrawal and spend transactions. The money used in the system was called "CyberBucks."
- Hash-Cash
    - development the proof of work system.
    - This as a countermeasure against spam in emails and also in blogs
    - this technology based on a job called; "Assessment through processing or combating spam"
    - The objective of HashCash is to require computer work for it to be verified. Once said work is verified, the user is allowed to use the resource. Use in email is based on adding an encrypted header to the email. This header has the information generated by the user using the HashCash system. This is a kind of seal that ensures that the mail has passed the proof of work. This seal is an identifier that shows that the sender has used the processor for a small amount of time. Well, it is the only way to generate a genuine stamp for each email you want to send.
    - It is based on the idea that if a certain user has used their processor to generate this stamp, it is unlikely that they are a spammer. Receivers with a very low almost negligible computational cost can verify this. In this way we can guarantee that it is not spam.

Flooding Transactions in Blockchain

- Whether Blockchain is a complete graph ?
    - YES, Nearly Complete Graph
- How Transactions Reaches all the Nodes ?
    - Floods the incoming message to all other outgoing edge/interfaces.
- Each node/Miner will receive the new transactions.
- Achieves C of CAP Theorem
- Spreads the message much faster.
- Simple - No Complex algorithm
- Guaranteed Delivery
- Flooding required lots of bandwidth resulting whole network end up with transaction. To avoid this we use
    - Limit no. of hopes for transaction
    - Selective flooding- discard duplicate transaction
- One node receive transaction using particular path and that node don’t consider that path for next flooding. Its help to reduce duplicate transaction comes into the network
- What is the difference between Flooding and Broadcasting?
    - Sending a packet to all hosts simultaneously is broadcasting. But flooding does not send packets to all hosts simultaneously. The packets would ultimately reach all nodes in the network due to flooding. Flooding may send the same packet along the same link multiple times, but broadcasting sends a packet along a link at most once. Several

copies of the same packet may reach nodes in flooding, while broadcasting does not cause that problem. Unlike flooding, broadcasting is done by specifying a special broadcast address on packets.

- Which Flooding Algorithm is used in Blockchain ?
    - Gossiping
    - The Gossip protocol is a protocol that allows designing highly efficient, secure and low latency distributed communication systems (P2P). The inspiration for its design has been taken from studies on epidemic expansion and algorithms resulting from it.
    - In blockchain networks, this protocol is used by network nodes to share and disclose information quickly and reliably with each other.
    - How it works?
        - In these protocols, for a node to distribute information, it must only be paired with other nodes randomly. Once this occurs, you should only exchange the information received with said nodes, who in turn will distribute the information with other nodes to which they are also paired. Forming a distribution chain to spread the information throughout the network in a timely and efficient way.

Storage

- Blockchain - Not Suitable for large amount of data because block size it limited
- What kind of data can be stored in the blockchain ?
    - Simple Transactions - Ownership Details , SCM details, Academia Details
- Blockchain in Healthcare ? Blockchain in Surveillance ?
    - MRI's, Operation Videos,
    - CCTv Footages
- Its usage IPFS protocol (Interplanetary File System)
    - PFS (Interplanetary File System) is essentially a file system that allows you to store files and track versions over time, much like Git, keeping track of them on a distributed network, somewhat like BitTorrent.
    - This storage system allows direct interaction through a secure and global P2P network.
    - This system allows users to share files and information without barriers. IPFS works well with large files that can consume or require high bandwidth to upload and / or download over the Internet. The rapid adoption of this distributed file system happened in part because IPFS is designed to operate over different protocols, such as FTP and HTTP. To store data, IPFS uses a DHT or distributed hash table. Once we have a hash, we ask the peer network who has the content located in that hash, and we download the content directly from the node that has the data I want
    - Blockchain is a decentralized data management platform that provides immutability, therefore it is a good choice to support file traceability metadata on a distributed file system like IPFS. We can say that thanks to the great similarity IFPS is the best friend of blockchain.
    - IPFS connects all these different blockchains in a way that's similar to how the web connects all these websites together. The same way that you can drop a link on one

page that links to another page, you can drop a link in ethereum [for example] that links to zcash and IPFS can resolve all of that

Miners

- Back bone of the blockchain technology
- Maintains the chain and the content in the block.
- Join the network and listen for transactions
- Validate transactions and construct a block
- Listen for new blocks – validate and re-broadcast a new block when it is proposed
- Store and broadcast the blockchain to the peers

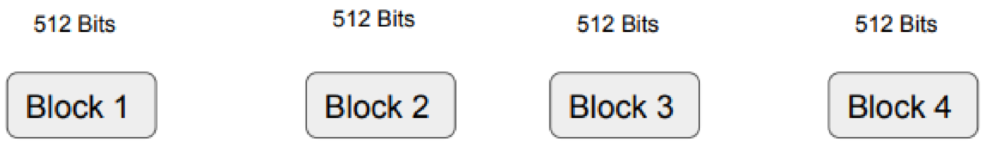
Hashing

- Hash(Input String) = Hash Value
- Hash Value - Shorter and Fixed Length Output
- Provides integrity to the input string
    - Avalanche Effect - If a single bit is changed
- Most extensively used for cryptographic function
- Blockchain Uses Hash Algorithm to protect the integrity of the transactions stored
- Hashing Vs Encryption
    - Hashing is One Way
    - Whilst Encryption is Two Way
- Blockchain Uses Encryption, No
- General Properties
    - Input can be any string of any size
    - Uniqueness - Always produces the same hash value for the same input.
        - $H(a) = b$
        - $H(b) \neq b$
    - Should Produce Fixed Size Output
    - It should be efficiently Computable
- Cryptographic Properties - Additional Properties
    - Collision Resistance
        - Collison can happen but probability is too less.
        - Because SHA-256
        - SHA-256 generate Hexadecimal values as a output because it is very difficult to represent in binary 2^256 bit.
    - Hiding
        - For output value of “Y” there is not any feasible way to get input “X”
        - Minimum Entropy: It is measure of how predictable an outcome is.
        - A hash Value should have a very high min-entropy value.
    - Puzzle Friendliness
        - A Hash Function Produces n bit output for x input

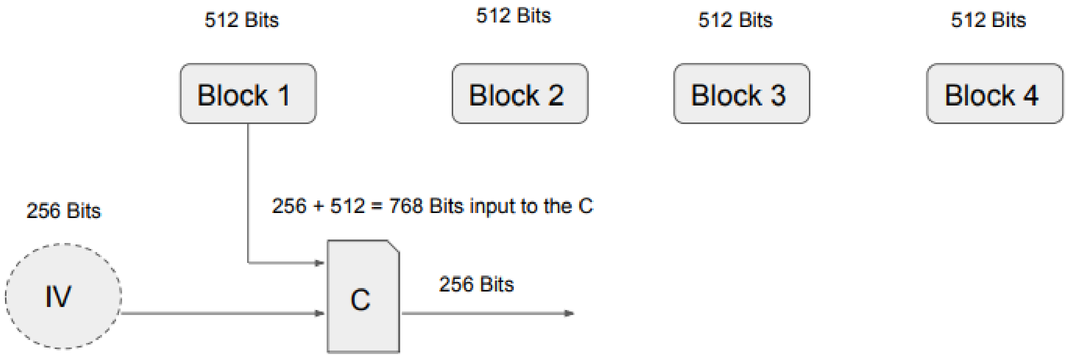
        - Then, it is infeasible to find x in time significantly less than 2^n
        - For SHA 256 bits, one should try 2^256 time to find x
        - If a computer calculates 10K hashes per second, then it would take 1027 (one Octillion) to calculate 2^128 hashes.
        - Solving strategy is difficult
        - The size of Hash value determines how difficult the puzzle is

SHA-256

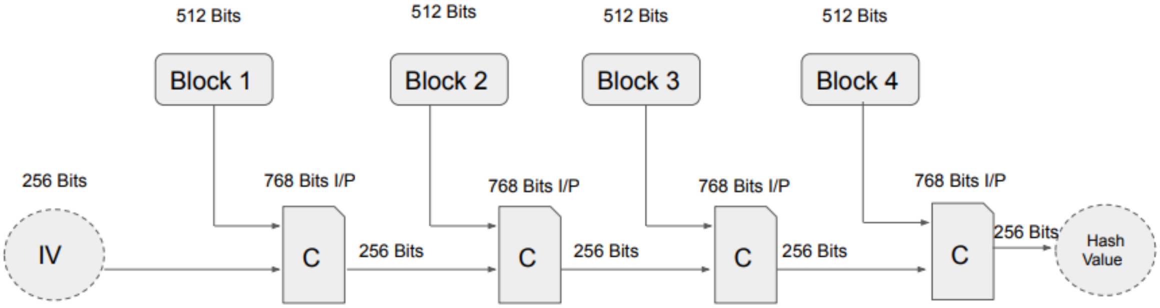
- SHA-256 uses Compression Function
    - Which compresses the given variable length input to fixed length output.
- Compression Function should be of collision resistance
- SHA-256 can take either fixed length or variable length inputs
- For Variable Length Inputs
    - Compression function uses Merkle-Damgard transform
    - Because of M-D Transform, SHA-256 works on variable length inputs
- Any fixed length input hash function can be converted to variable length input hash function using M-D Transform
- SHA-256 - Most Commonly Used Hash Function
- Real World Application - Bitcoin uses SHA-256
- M-D Transform
    - Takes input of length m and produces output of a smaller length n
    - The input m is divided into blocks of length m-n
    - The output of previous block will act as input to the compression function.
    - What about the very first block ?
        - There is no previous block to the first block.
        - Hence, an Initialization Vector (IV) is used
        - IV - A random text , adds randomness to the hash value produced.



Divide the input into equal sized blocks



- Compression Function



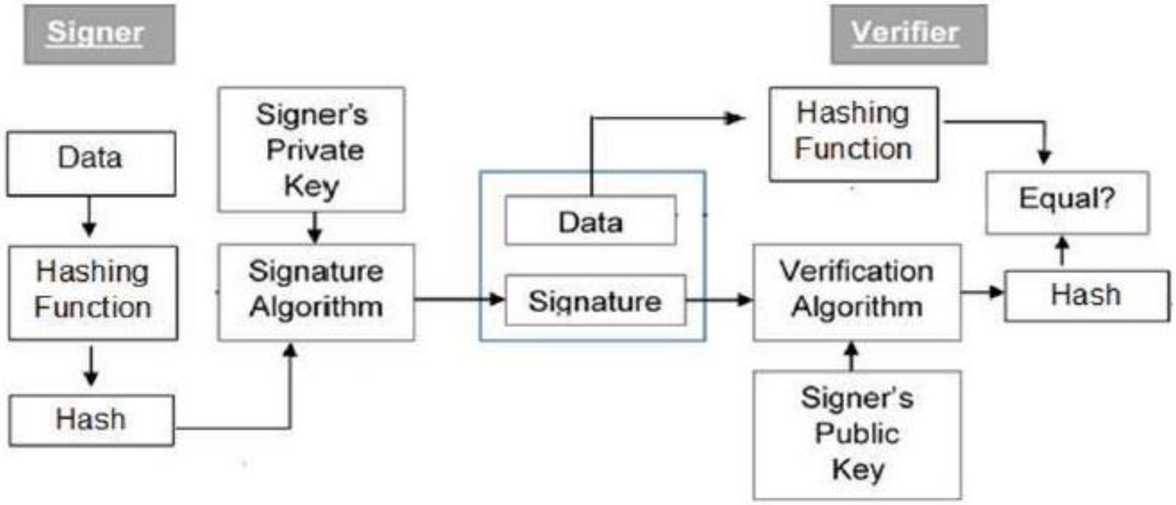
- Hash Function in Blockchain
  - Distributed environment - data in multiple places
  - Changing data and the corresponding hash in one place is not enough
  - You need to change in all repositories - impossible.
  - So it prevent from tampering data.

Digital Signature

- Key Generation Algorithms: Digital signature is electronic signatures, which assure that the message was sent by a particular sender. While performing digital transactions authenticity and integrity should be assured, otherwise, the data can be altered or someone can also act as if he was the sender and expect a reply.
- Signing Algorithms: To create a digital signature, signing algorithms like email programs create a one-way hash of the electronic data which is to be signed. The signing algorithm then encrypts the hash value using the private key (signature key). This encrypted hash along with other information like the hashing algorithm is the digital signature. This digital signature is appended with the data and sent to the verifier. The reason for encrypting the hash instead of the entire message or document is that a hash function converts any arbitrary input into a much shorter

fixed-length value. This saves time as now instead of signing a long message a shorter hash value has to be signed and moreover hashing is much faster than signing.

- Signature Verification Algorithms : Verifier receives Digital Signature along with the data. It then uses Verification algorithm to process on the digital signature and the public key (verification key) and generates some value. It also applies the same hash function on the received data and generates a hash value. Then the hash value and the output of the verification algorithm are compared. If they both are equal, then the digital signature is valid else it is invalid.
- Digital Signature on block chain
    - Every Transaction in blockchain should be signed.
    - Miners will check for the signature in the transaction for the authenticity.
    - No one except you knows your signature - Reproducing is difficult
        - But, verification of the message can be done based on the signature.
    - Signature should be corresponding to the particular document.
    - Digital Signature – Ex
- Digital Signature Scheme
    - Generate Keys(keysize) : sk, pk
        - Generates key pairs sk and pk
        - sk - Secret Key, pk - Public Key
        - sk is kept private while, pk is made public
        - Anyone with pk can verify the sk
    - Digital Signature: sign(sk, Message)
    - Verify Signature : Verify(pk, Message)
- Digital Signatures using Keys
    - Symmetric Key
        - Single Key
        - Same key is used for encryption and decryption
        - shared key, the key should be shared between the communicating players.
    - Asymmetric Key
        - Two Keys
        - One for the encryption while another one for the decryption
        - One will be private and the other will be public



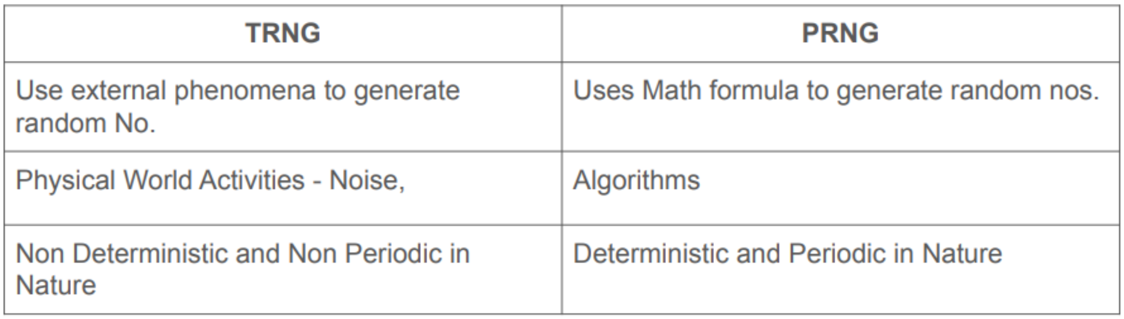
Model of Digital Signature

Keys and Wallets

- You should create identity for the users
  - Address - Gives Identity
  - Keys - Gives integrity to the transactions.
- How to maintain it ?
  - Wallets.
- Wallet stores all the keys and the address generated
- Wallet - a Secured Independent Entity
- Who will create wallet?
  - For creating wallet we never relay on 3rd party, because we compromise security
  - User can also create Wallet but it is not provide guarantee wallet follow entropy or randomness
- So, Block chain provide wallet, its act as Key Distribution Center (KDC)
  - A Fully Trusted Environment
  - Keys are generated Once the authentication of Player is done.
  - miners should create the keys.

PKI-Using Blockchain

- Public-Key Infrastructure (PKI) is the cornerstone technology that facilitates secure information exchange over the Internet
- Use in block chain
    - Address for users
    - Public and Private key pairs for the maintaining the integrity.
    - Steps
        - Step 1: Generate Private Key
        - Step 2: From Private Key Generate Public Key
        - Step 3: From Public Key Generate Address
- First Step : Private Key (k)
    - Private Key: a 256 bits Random Number between 1 to 2^256
        - Assuming that you need 256 bit address.
    - For generating random number
        - Don't use Operating System Random function
        - Don't write code yourself for random number generation
        - It can not guarantee the randomness (min entropy)
        - Use a cryptographically-secure pseudo-random number generator (CSPRNG)
            - Provides sufficient entropy
    - Classical Ways to Generate Random No
        - True Random Number Generators(TRNGs)
        - Pseudo-Random No. Generator (PRNGs)

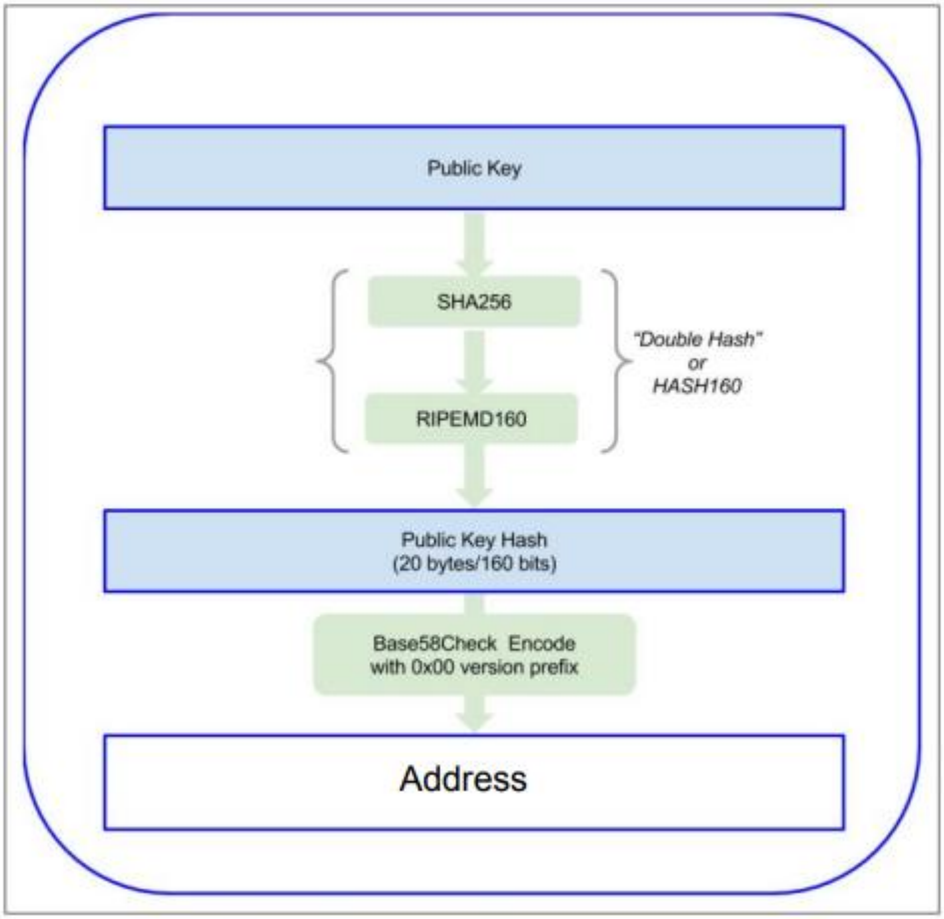


TRNG	PRNG
Use external phenomena to generate random No.	Uses Math formula to generate random nos.
Physical World Activities - Noise,	Algorithms
Non Deterministic and Non Periodic in Nature	Deterministic and Periodic in Nature

- Quantum Random Number Generator(QRNG)
    - Traditional random number generators like PRNG and TRNG use predictable inputs which are deterministic. These inputs have higher probability of repeating which creates predictability. Hence, making the entire system vulnerable.
    - quantum random number generator (QRNG) uses the principles of quantum mechanics to generate truly random numbers. By fact, quantum physics is fundamentally random in nature and is confirmed by theory and experimental research.
    - Quantum Random Number Generator is a highly-sophisticated engineering innovation which involves the power of complex deep-tech technologies (such as semiconductors, optoelectronics, high precision electronics and quantum physics) working together to create the highest level of randomness possible.

- Second Step : Public Key (K)
    - For creating public key using private key we preferred Elliptic Curve Cryptography (ECC).
    - RSA (Rivest, Shamir, Adleman) we can also use but it need more computation power.
    - Elliptic Curve Multiplication - Public Key Generation Method
        - K = k * G
        - Where K - Public Key, k - Private Key and G is a generator point in the elliptic curve
    - Bitcoin as well as ethereum uses well known secp256k1 standard elliptic curve.
    - secp256r1 curve alternative

- Third Step : Address (A)
    - Create the address for the user using SHA algorithm
    - SHA256(K)
        - Where K-Public key
        - SHA gives a 256 bit value.
    - But, remembering a 256 bit value is difficult.
        - E99423A4ED27608A15A2616A2B0E9E52CED330AC530EDCC32C8FFC6A526AEDD
    - For using shorter version other then 256bit use
        - RACE Integrity Primitives Evaluation Message Digest (RIPEMD)
        - Produces 160 bit data - 20 byte
    - A = RIPEMD160(SHA256(K))
    - Now A is shorter But, it is not human readable.
        - It is in an hexa decimal form
            - E99423A4ED27608A15A2616A2B0E9E52CE
    - Make it human readable.
        - Use Base58Check
        - Its generate readable message
    - After Base58Check
        - hMirt546nngXqyPEz532S8fLwbozud8
    - there is default convention bitcoin(BTC) and ethereum follow
        - BTC addresses range from 27 to 34 alphanumeric characters, with each one beginning with 1, 3, or bc1. The beginning character in a Bitcoin address depends on the type of address (P2PKH, P2SH, or Bech32). All address formats work with one another. For example, you can send BTC from an address that starts with “1” to an address that starts with “3” or “bc1” or vice versa.
        - Ethereum address start with 0x. An Ethereum address is a 42-character hexadecimal address derived from the last 20 bytes of the public key controlling the account with 0x appended in front. e.g., 0x71C7656EC7ab88b098defB751B7401B5f6d8976F



Transaction Life Cycle

- Everything is transaction in blockchain.
- Transaction
    - Created
    - Propagated
    - Validation
    - Added to the Blockchain.

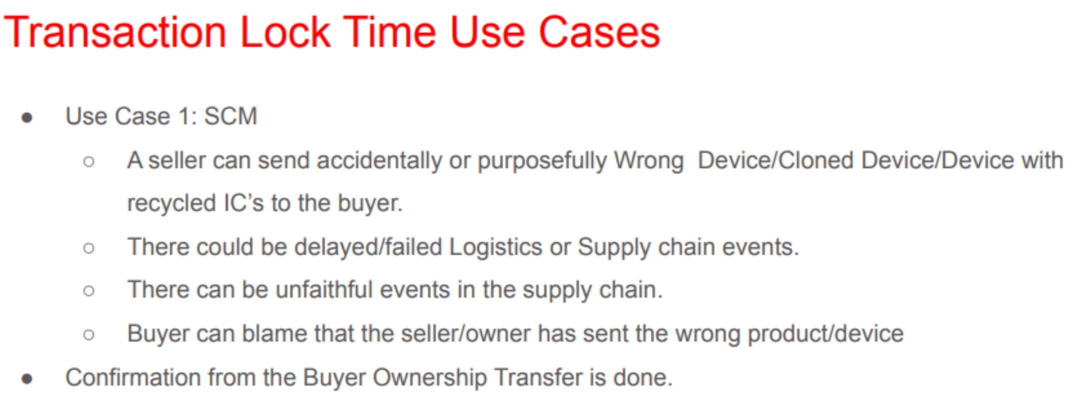
      (Peer-to-peer refers to the direct exchange of some asset, such as a digital currency, between individual parties without the involvement of a central authority)
- In public network we have separate block chain network (P2P). All miner are present in (P2P) private network where as all user/registered user access block chain outside the private network or through public network.
- Transaction Creation
    - Register user request a new transaction via something called a wallet.
- Authorization (Signing the Transaction)
    - Generated transaction sign with digital signature and pass to the network.
    - Digital signature reuired Private/public key authentication.
- Validation and Propagation

    - Every computer in the network checks (validate) the transaction against some validation rules that are set by the creators of the specific blockchain network.
    - The transaction is send (flood) to all participation computers in the specific node of blockchain network.
    - In validation miner can found Transaction may be bad or good. If good then flood to next. If bad then drop from network and notify user about this.
    - There are 3 cases possible in Transaction
        - Good transaction: accepted for further process.
        - Bad Transaction : Reject and notify the user.
        - Bad Transaction and Compromised miner: In this case miner allow bad transaction and forward that transaction to next process. But in next process other Miner validate and confirm this is bad transaction and reject it.
- Transaction include in block
    - Validated transactions are stored into a block and are sealt with a lock (hash).
- Consensus Phase
    - The consensus protocol makes sure that every new block that is added to the Blockchain is the one and only version of the truth that is agreed upon by all the nodes in the Blockchain.
    - The Blockchain consensus protocol consists of some specific objectives such as coming to an agreement, collaboration, co-operation, equal rights to every node, and mandatory participation of each node in the consensus process. Thus, a consensus algorithm aims at finding a common agreement that is a win for the entire network.
- Added to the Blockchain
    - This block becomes part of the blockchain when other computers in the network validate if the lock on the block is correct.
    - Now the transaction is part of the blockchain and can not be altered in any way.

Structure of a Transaction

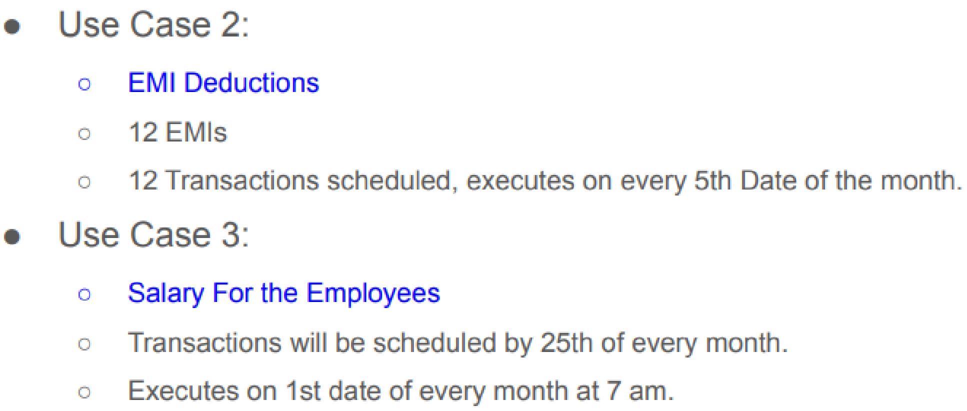
- No specific structure.
- Depends on the application.
- Some of the default fields
    - Committed or not Committed (Failed Transaction)
        - Transaction is authenticated, but do not have enough balance to buy the product
        - Still you have to pay the transaction fee.
    - Transaction Lock time
    - Time Stamp
    - Keys or Signatures
    - Fees
- Transaction Lock Time
    - The time when you want your transaction will added to the block chain.

- Default value of Transaction Lock Time is zero, means immediate add this transaction to the block.
- Value more then zero
    - This mean you want to consider your transaction after some time, not immediately.
    - Like post date cheque.
- For example if a user specified locktime block height @ 664777 and if the current block height is 664700 then the user have to wait until the block 664777 is reached. Only after the block 664777 has been mined miners will attempt to include their transaction in a block. In Bitcoin the block time is 10 minutes so more or less the user have to wait for 770 hours (close to a month) for their transaction to get confirmed.
- Why use this function?
    - First of all most of the Bitcoin transaction does not require this function and they do not use a set locktime. So any transactions that does not use this feature will have a locktime set to 0x00000000.
    - If you do not want your transaction to be locked until specific block height or time you can simply set the lock time to less than the current block height or the UNIX time. This will set the locktime field to 0x00000000. This way your transaction will be made final. So when exactly do we need this function?
    - In simple LockTime function is used to lock a transaction until a specific block height or point of time. Setting locktime means the transaction will be confirmed only when the required time or block height has been met. The transaction won’t be valid until the specified function is met.
    - Nodes validate every transaction and if a transaction contains locktime script then they will simply reject it. The transaction that has locktime specified will only get added to the blockchain after the set time or block height that has been elapsed and not before it.
    - Think of this feature as a post dated cheque written with a future date. The cheque will not be cleared prior to that specified date. Similarly with locktime the coins cannot be spent until a specified time or until certain block height has been past. This will simply lock the UTXO for a predetermined amount of time.
- Exploiting the Transaction Lock Time Variable
    - Block Height
    - Date and Time
    - Only after some event
        - When a event occurs, Miners should check for the pending transactions related to it.
        - May have to change the Transaction Structure as well.
            - o reflect the pending transactions
    - Only on the approval from the other party



Transaction Lock Time Use Cases

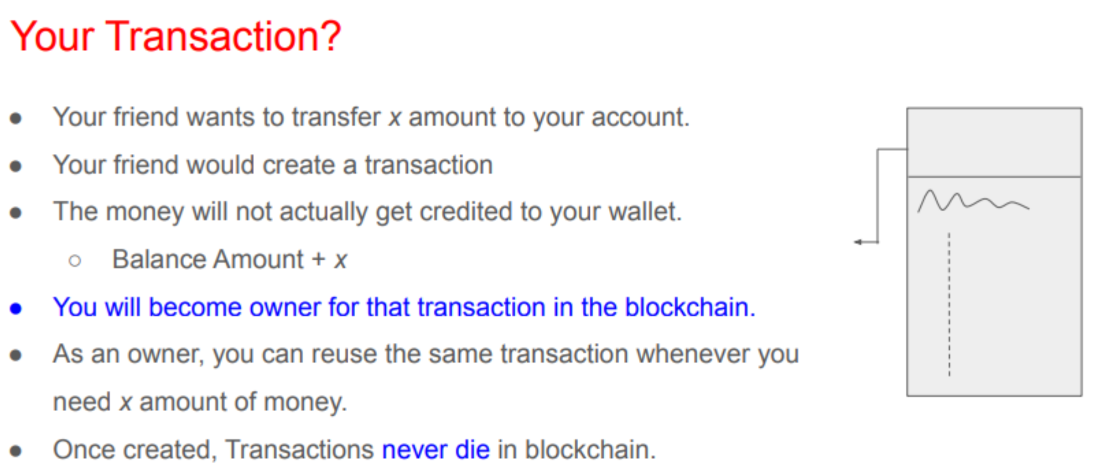
- Use Case 1: SCM
    - A seller can send accidentally or purposefully Wrong Device/Cloned Device/Device with recycled IC's to the buyer.
    - There could be delayed/failed Logistics or Supply chain events.
    - There can be unfaithful events in the supply chain.
    - Buyer can blame that the seller/owner has sent the wrong product/device
- Confirmation from the Buyer Ownership Transfer is done.



- Use Case 2:
    - EMI Deductions
    - 12 EMIs
    - 12 Transactions scheduled, executes on every 5th Date of the month.
- Use Case 3:
    - Salary For the Employees
    - Transactions will be scheduled by 25th of every month.
    - Executes on 1st date of every month at 7 am.

Transaction Pool

- Wallets
    - Wallet does not contains digital currencies.
    - It has access to the Transactions which you own.
        - Ownership matters here.
        - Every transaction in blockchain has an owner
        - Wallet Does not have access to other Transactions
    - Everything in block chain is transaction
    - Transactions which you own are scattered around in the blockchain
    - There is huge number of transaction present in block chain of block, so searching respective transaction is wallet job.
- Example.

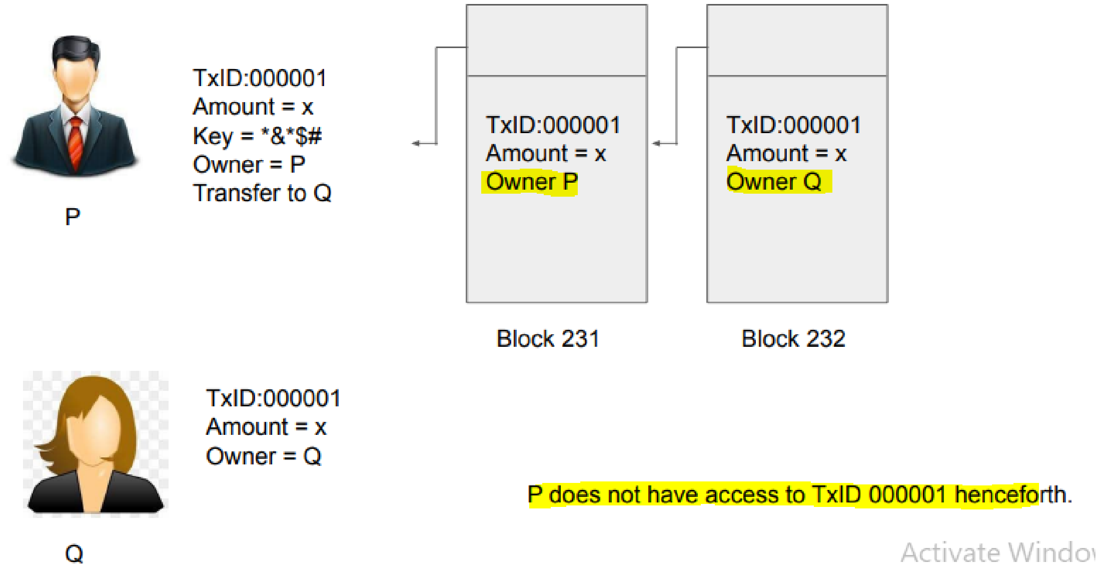


Your Transaction?

- Your friend wants to transfer *x* amount to your account.
- Your friend would create a transaction
- The money will not actually get credited to your wallet.
  - Balance Amount + *x*
- You will become owner for that transaction in the blockchain.
- As an owner, you can reuse the same transaction whenever you need *x* amount of money.
- Once created, Transactions never die in blockchain.

○

○ P Initiates transaction to Q

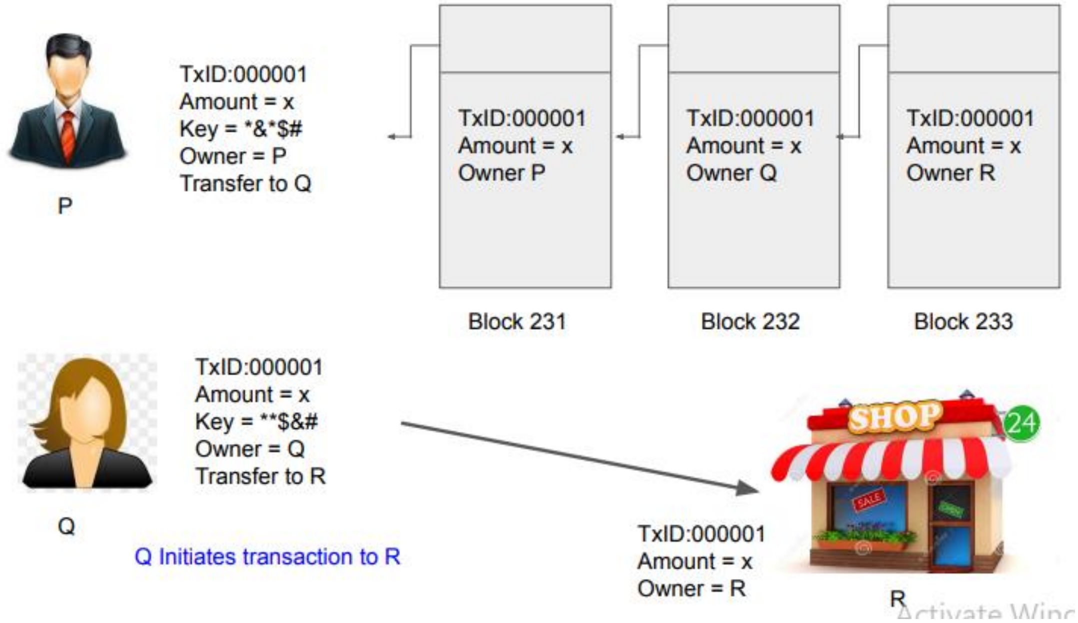


TxID:000001
Amount = x
Key = *&*$#
Owner = P
Transfer to Q

TxID:000001
Amount = x
Owner = Q

P does not have access to TxID 000001 henceforth.

○



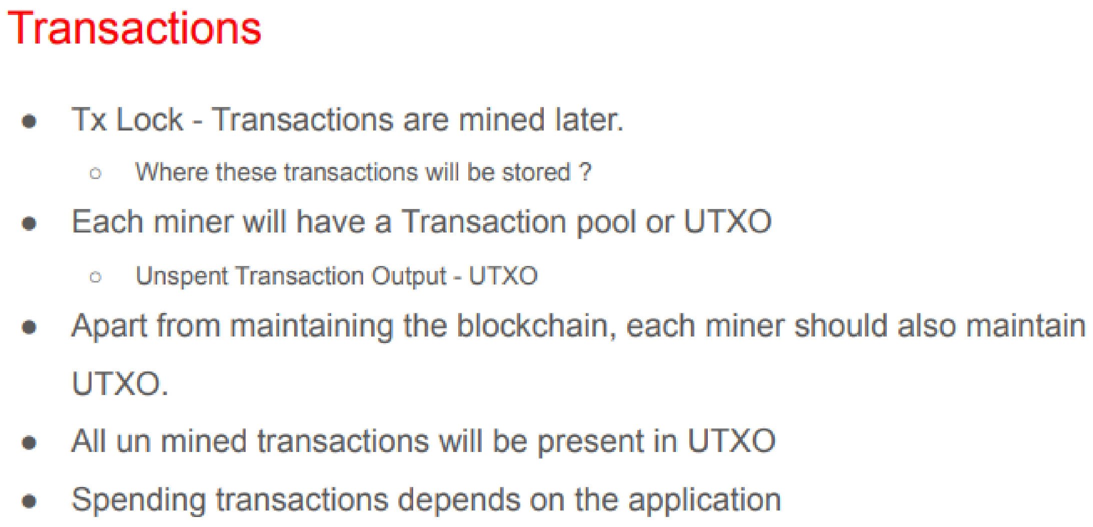
TxID:000001
Amount = x
Key = *&*$#
Owner = P
Transfer to Q

TxID:000001
Amount = x
Key = **$&#
Owner = Q
Transfer to R

Q Initiates transaction to R

○

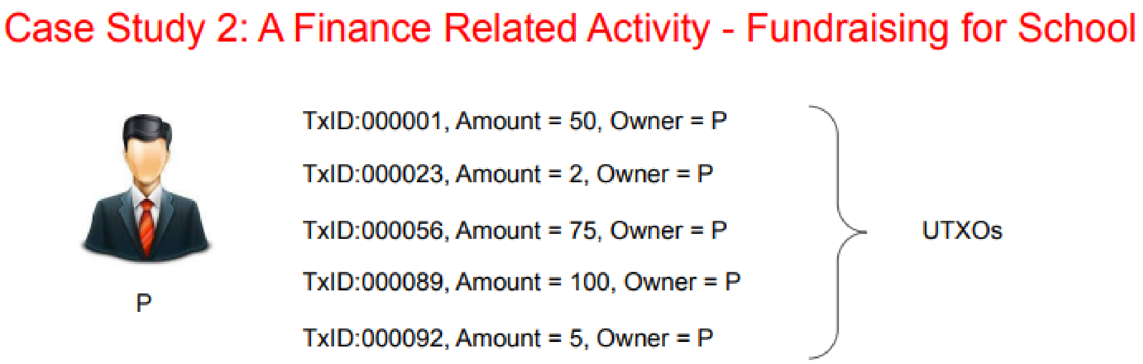
- Advantage
  - Tracking and Tracing
- Finding your transaction
  - Transactions you own are scattered around in the blockchain
  - You to Wallet: How much money I have ?
  - Your Wallet: Checks all the transactions you own in the blockchain
  - Wallet scans and aggregates all the transactions you own



Transactions

- Tx Lock - Transactions are mined later.
  - Where these transactions will be stored ?
- Each miner will have a Transaction pool or UTXO
  - Unspent Transaction Output - UTXO
- Apart from maintaining the blockchain, each miner should also maintain UTXO.
- All un mined transactions will be present in UTXO
- Spending transactions depends on the application

-



Case Study 2: A Finance Related Activity - Fundraising for School

TxID:000001, Amount = 50, Owner = P

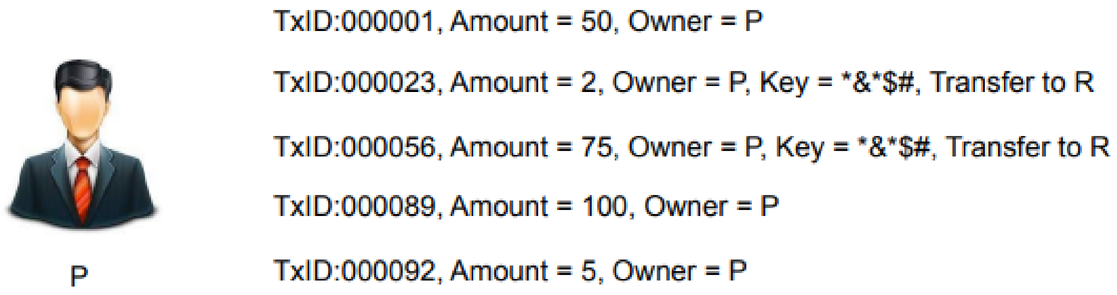
TxID:000023, Amount = 2, Owner = P

TxID:000056, Amount = 75, Owner = P

TxID:000089, Amount = 100, Owner = P

TxID:000092, Amount = 5, Owner = P

-
  - Now, You want to purchase a book for the school which costs Rs. 77
  - You will ask Wallet to search for your transactions which sums up Rs. 77
    - 75 Rs and 2 Rs.



TxID:000001, Amount = 50, Owner = P

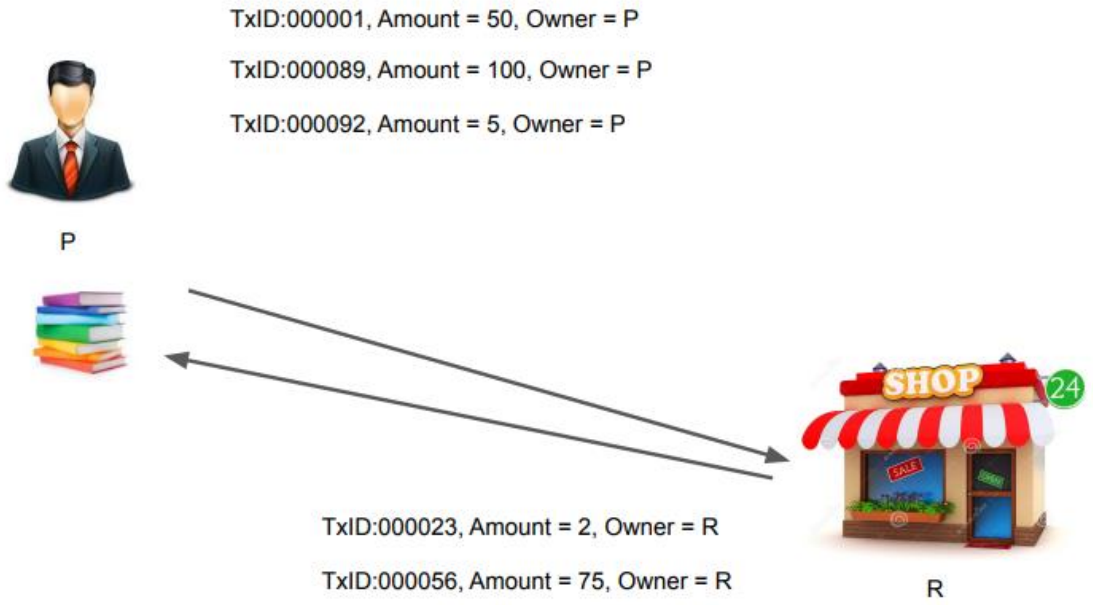
TxID:000023, Amount = 2, Owner = P, Key = *&*$#, Transfer to R

TxID:000056, Amount = 75, Owner = P, Key = *&*$#, Transfer to R

TxID:000089, Amount = 100, Owner = P

TxID:000092, Amount = 5, Owner = P

-

- 

TxID:000001, Amount = 50, Owner = P

TxID:000089, Amount = 100, Owner = P

TxID:000092, Amount = 5, Owner = P

TxID:000023, Amount = 2, Owner = R

TxID:000056, Amount = 75, Owner = R

If you Do Not have Exact Change ?

TxID:000001, Amount = 50, Owner = P

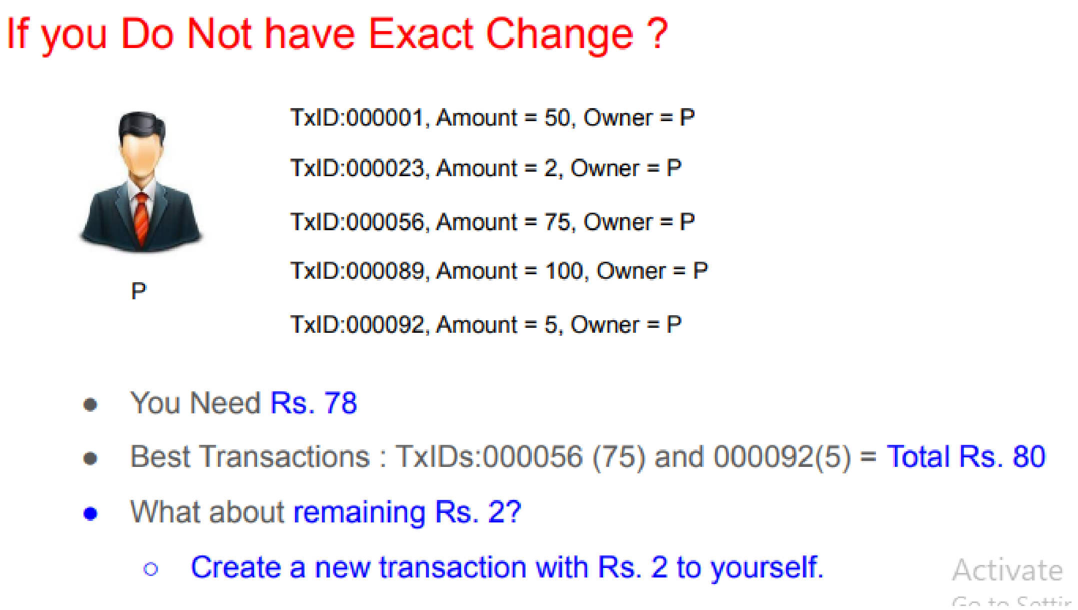
TxID:000023, Amount = 2, Owner = P

TxID:000056, Amount = 75, Owner = P

TxID:000089, Amount = 100, Owner = P

TxID:000092, Amount = 5, Owner = P

- Now, You want to purchase a book for the school which costs Rs. 78
- Your wallet should check all possibilities and should find the best option.
- Best Transactions : TxIDs:000056 (75) and 000092(5)



If you Do Not have Exact Change ?

TxID:000001, Amount = 50, Owner = P

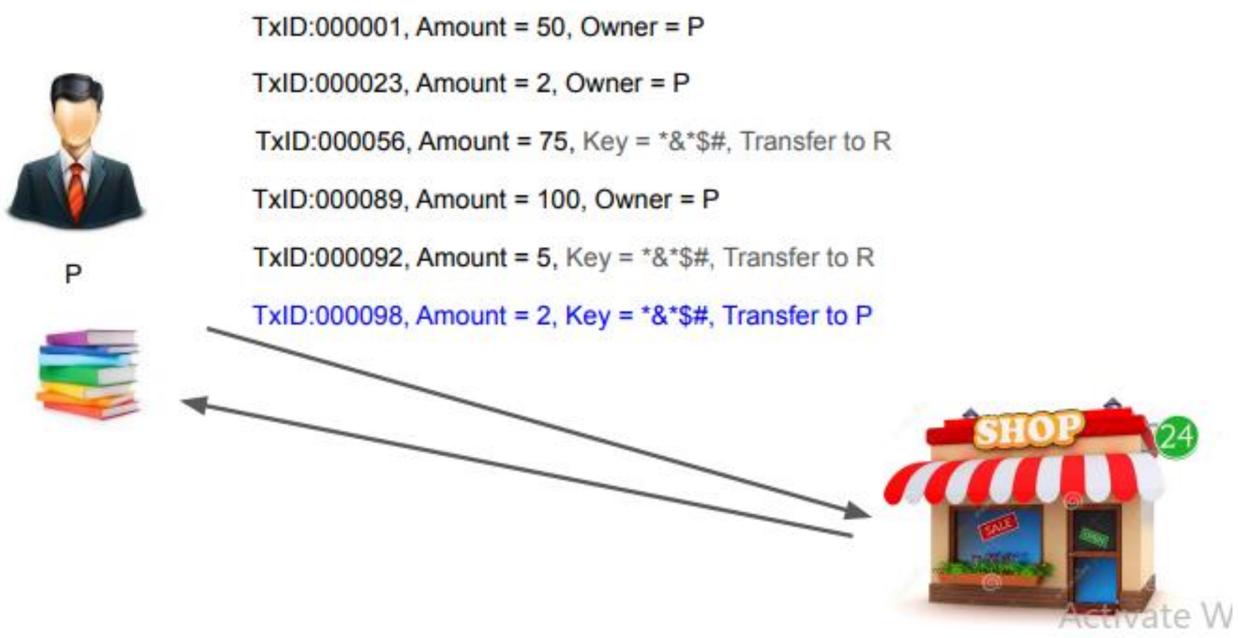
TxID:000023, Amount = 2, Owner = P

TxID:000056, Amount = 75, Owner = P

TxID:000089, Amount = 100, Owner = P

TxID:000092, Amount = 5, Owner = P

- You Need Rs. 78
- Best Transactions : TxIDs:000056 (75) and 000092(5) = Total Rs. 80
- What about remaining Rs. 2?
  - Create a new transaction with Rs. 2 to yourself.



TxID:000001, Amount = 50, Owner = P

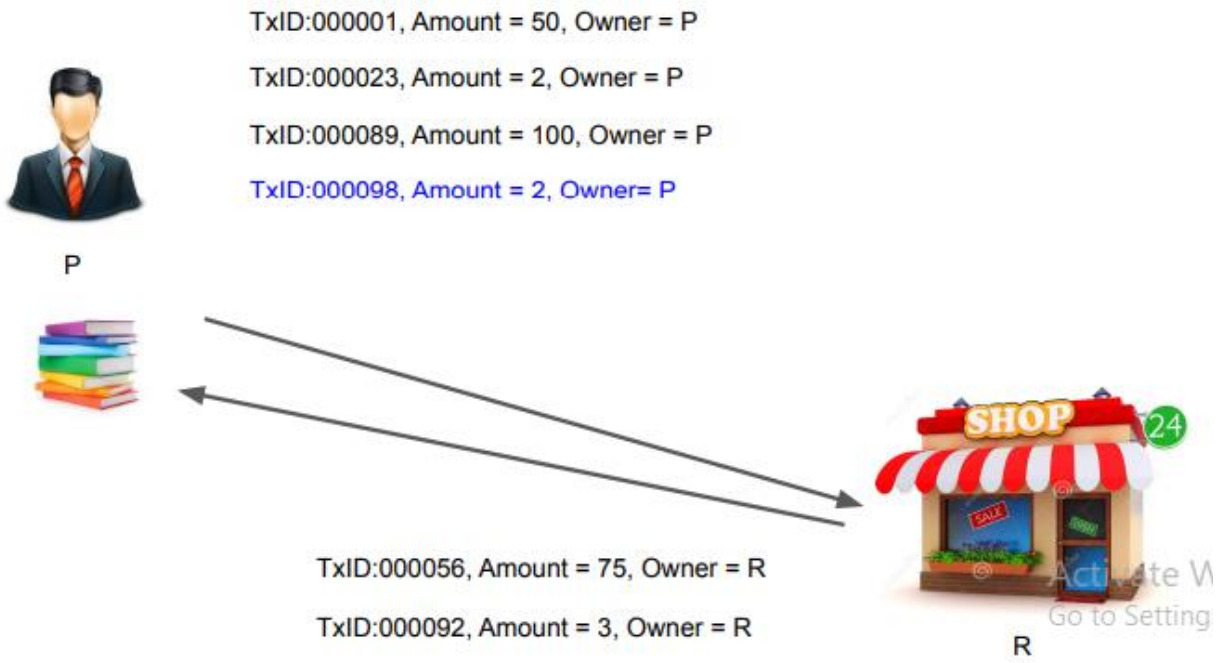
TxID:000023, Amount = 2, Owner = P

TxID:000056, Amount = 75, Key = *&*$#, Transfer to R

TxID:000089, Amount = 100, Owner = P

TxID:000092, Amount = 5, Key = *&*$#, Transfer to R

TxID:000098, Amount = 2, Key = *&*$#, Transfer to P



TxID:000001, Amount = 50, Owner = P

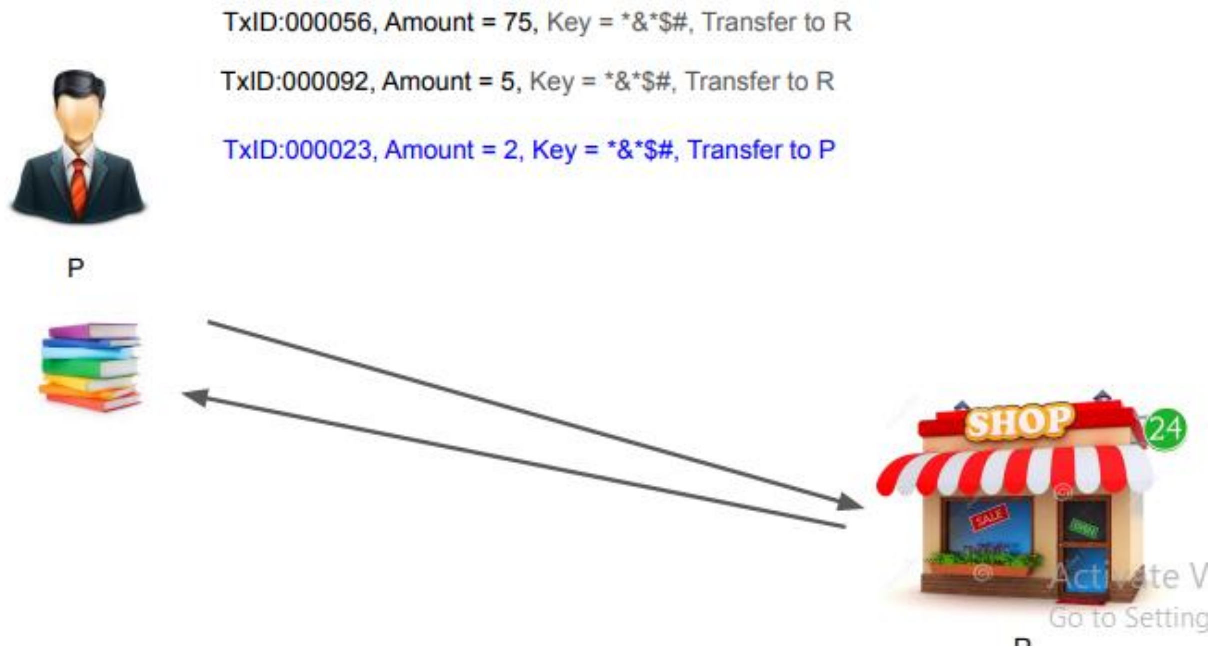
TxID:000023, Amount = 2, Owner = P

TxID:000089, Amount = 100, Owner = P

TxID:000098, Amount = 2, Owner= P

TxID:000056, Amount = 75, Owner = R

TxID:000092, Amount = 3, Owner = R



TxID:000056, Amount = 75, Key = *&*$#, Transfer to R

TxID:000092, Amount = 5, Key = *&*$#, Transfer to R

TxID:000023, Amount = 2, Key = *&*$#, Transfer to P

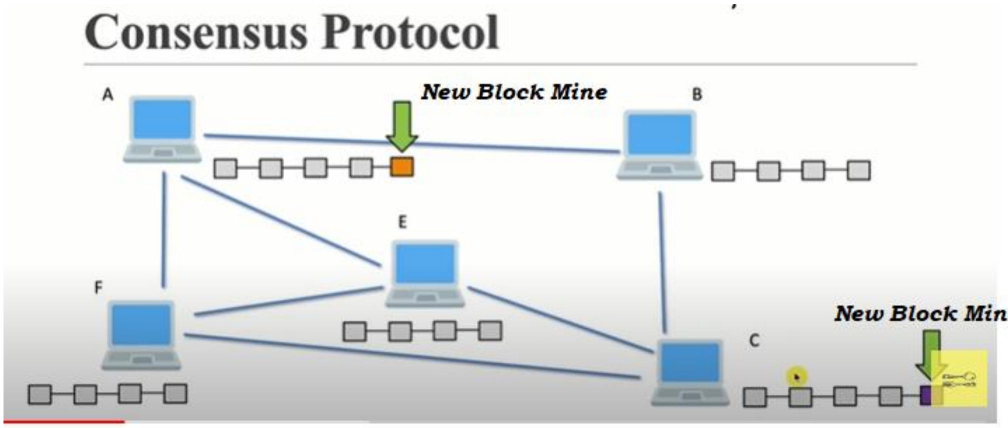
Common attack on Block Chain

- Sybil Attack
    - Sybil Attack is a type of attack seen in peer-to-peer networks in which a node in the network operates multiple identities actively at the same time and undermines the authority/power in reputation systems.
    - The main aim of this attack is to gain the majority of influence in the network to carry out illegal(with respect to rules and laws set in the network) actions in the system
    - A single entity(a computer) has the capability to create and operate multiple identities(user accounts, IP address based accounts). To outside observers, these multiple fake identities appear to be real unique identities.
    - How the Bitcoin network prevents sybil attack ?
        - Bitcoin network uses the Proof of Work(PoW) consensus algorithm to prove the authenticity of any block that is added to the blockchain. A considerable amount of computing power is required to do the work which provides incentive to the miners to do honest work(a bitcoin reward; currently 12.5 bitcoins for every block mined) and no incentive for the faulty work. The transactions are verified by every node and rejected as invalid if faulty transactions are included in the block. A type of sybil attack, called the 51% attack is also practically impossible in the bitcoin network because of so many miners, it is very difficult for a single organization to control 51% of the miners.
- Dusting Attack?
    - A dusting attack refers to a relatively new kind of malicious activity where hackers and scammers try and break the privacy of Bitcoin and cryptocurrency users by sending tiny amounts of coins to their wallets.
    - The transactional activity of these wallets is then tracked down by the attackers, who perform a combined analysis of different addresses to deanonymize the person or company behind each wallet
    - how it happened?
        - Malicious actors realized that cryptocurrency users don't pay much attention to these tiny amounts showing up in their wallet addresses. So they began "dusting" a large number of addresses by sending a few satoshis to them (i.e., a small amount of LTC, BTC or other cryptocurrencies). After dusting different addresses, the next step of a dusting attack involves a combined analysis of those addresses in an attempt to identify which ones belong to the same crypto wallet.
        - The goal is to eventually link the dusted addresses and wallets to their respective companies or individuals. If successful, the attackers may use this knowledge against their targets, either through elaborated phishing attacks or cyber-extortion threats.
    - Solution is: Wallet user don’t confirm such dusty transaction.

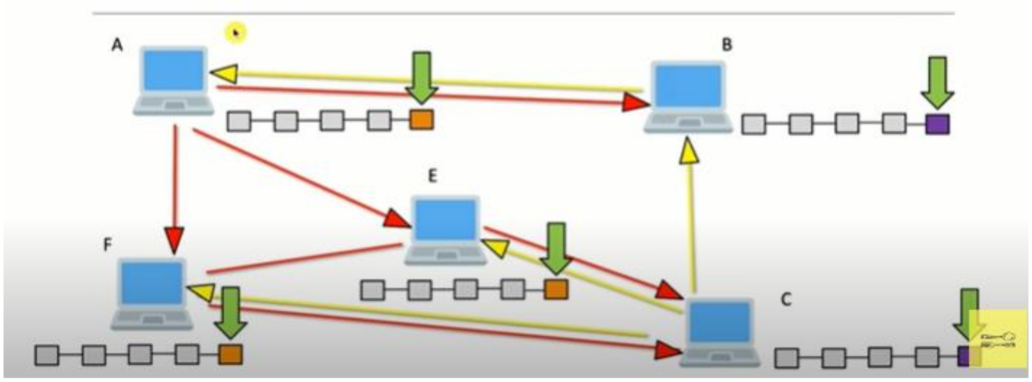
- What Is a 51% Attack?
    - A 51% attack is a potential attack on a blockchain network, where a single entity or organization is able to control the majority of the hash rate, potentially causing network disruption. In such a scenario, the attacker would have enough mining power to intentionally exclude or modify the ordering of transactions. They could also reverse transactions they made while being in control - leading to a double-spending problem.
    - A successful majority attack would also allow the attacker to prevent some or all transactions from being confirmed (transaction denial of service) or to prevent some or all other miners from mining, resulting in what is known as a mining monopoly.
    - Changing the block's reward, creating coins out of thin air, or stealing coins that never belonged to the attacker are also deemed impossible events.
    - How likely is a 51% attack?
        - Since a blockchain is maintained by a distributed network of nodes, all participants cooperate in the process of reaching consensus. This is one of the reasons they tend to be highly secure. The bigger the network, the stronger the protection against attacks and data corruption.

- Byzantine Generals Problem/Byzantine Fault Tolerance
    - What is Byzantine Fault Tolerance?
        - Byzantine Fault Tolerance(BFT) is the feature of a distributed network to reach consensus(agreement on the same value) even when some of the nodes in the network fail to respond or respond with incorrect information. The objective of a BFT mechanism is to safeguard against the system failures by employing collective decision making(both – correct and faulty nodes) which aims to reduce to influence of the faulty nodes
        - means some of faulty node passing miss info within network.
    - Solution
        - Even one or some node are malicious but majority of node are perfect so whole network don't have any effect of that node.
        - Ignoring that node and work smoothly.

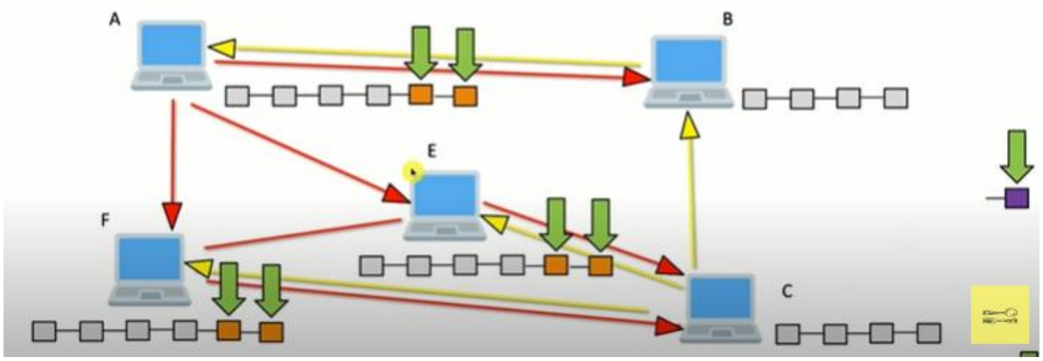
- Competing Chain Problem (longest chain rule)
    - suppose multiple miner will create block simultaneously in the network



    - ***At the same time new block mine by different miner***
    - respective miner will transfer this block to respective close miners.
    - This lead to conflict between miner, some miner have blockchain with A new block and some are with C new block. So what to do now?



    - Consensus protocol handle this problem by considering long chain.
    - Entire network which block chain is longest that will be accepted and other one is rejected.
    - Means both miner wait until any miner will mine next block and add block in there block chain. Here those who create new block that block chain is selectd.



    - B and C discard there block and add new two block in it