blockchained
{'india'}

presents

60 Days of Code with

Polkadot

Instructions:

1. Ensure that the task is completed on the very same day it is assigned. If you face any problem, do share it with us in the group at the same moment, so that the doubts are cleared at the earliest.

2. Every week we will complete one module. Each module is further divided into 5 days. We have kept the weekends free so you can catch up on the work over the weekend and attend the doubts session/workshops which will be organised.

3. If you are not able to complete the task, ensure that they are covered in the very week so that the process is not delayed much.

4. We will have live workshops every week. The schedule of the same will be shared at the start of the week and the same is added to your weekly curriculum. Do ensure you are attending these workshops as they are live sessions by industry experts. It is going to help you only in your journey. If by any reason you are not able to attend, inform the team beforehand.

5. Please submit your assignment as these will be necessary to participate in the hiring track.

6. Please update your daily progress by posting on twitter and sending a message on the group When posting on your twitter accounts please tag us @blockchainedind and @Polkadot, along with the use of following hashtags:
#60DaysofCode #BuildInPublic #Polkadotdevbootcamp

If you have any questions or doubts, please reach out to us on the group or email us at hriday@blocumen.com and manav@blocumen.com.

Content

Module 1: Introduction to blockchain
(June 3 to June 10)

Day 1: Understanding the basics of blockchain technology
Day 2: Different Types of blockchains
Day 3: Bitcoin: the first blockchain-based cryptocurrency
Day 4: Ethereum: A decentralized platform for smart contracts
Day 5: Consensus Algorithms and how they work

Workshop Details
Advanced Topics (optional)

Module 1: Introduction to blockchain

Day 1

Understanding the basics of blockchain technology

1.Blockchain concept and context :

i) How did it all start?

The first blockchain started as a research project. Satoshi Nakamoto wrote the paper Bitcoin: 'A Peer to Peer Electronic Cash System in 2009' in which he described a way of transferring value from one person to another without the need for physical currency or bank. Bitcoin itself is a very simple blockchain. At its core, it's just a list of accounts and the value held within them.

ii) How did it improve?

In 2014 people realized that blockchains could be used to run small programs and a way to transfer money safely.

The Bitcoin protocol isn't particularly well equipped for programs, so in 2015 Vitalik Buterin co-created Ethereum which allowed developers to create *smart contracts* on a blockchain. Ethereum vastly improved the usability of blockchains and attracted the attention of companies like Microsoft, Bank of America, and Google.

iii) So what exactly is blockchain?

Watch this video to understand what exactly is blockchain?
https://youtu.be/G9R2qSWhDsw

Basically a blockchain is a “cryptographically secure transactional singleton machine with shared-state”.
Let's break into three pieces and take cryptocurrency as example :

i)Singleton machine :- Single global truth that everyone believes in.
(Ex: Global Truth- A valid transaction verified by mechanisms accepted by everyone)

ii) “With shared-state” means that the state stored on this machine is shared and open to everyone.
(Ex: A valid transaction obtained from singleton machine is stored in all the members of blockchain)

iii) Cryptographically Secure :

It means that the creation of digital currency is secured by complex mathematical algorithms that are obscenely hard to break.
(Ex: A valid transaction obtained from singleton machine and is stored in all the members of blockchain has a feature of immutability i.e the users who have the data stored with them cannot change them at any cost)

Think of a firewall of sorts. They make it nearly impossible to cheat the system.
(e.g. create fake transactions, erase transactions, etc.)

In a nutshell, A information that is universally accepted/Verified (Singleton Machine) is broadcasted to all the users and every user stores this information with them(Shared state) , and this stored info cannot be manipulated at any cost (Cryptographically Secure)

iv) Blockchain vs Centralized server:

Blockchain networks look like regular networks, with a group of computers all connected. However, they differ in the way that programs and applications run. On a regular network when a user wants to find the results of an application, they send the request to a particular server who runs a program and returns the answer to the user. An issue with this process is that there isn't any way to verify that the response the user got from the server is correct, or that the program the server ran was the program the user requested.

On a blockchain network when a request is sent from the user to the network, all the computers run the program and agree on the answer before sending it back to the user. This way the user can verify that the answer is the correct answer and that the correct program was run in the first place. Calling a program is referred to as a *transaction*. All transactions are saved in a database and shared with all the other computers on the blockchain network. Because of this, users can prove that they performed a certain action.

A layman's introduction to blockchain by Jaspreet Bindra, Advisor at Blockchained India & SVP at Mahindra Group

https://www.youtube.com/watch?v=8fbhI1qVj0c&ab_channel=TEDxTalks

2. How does a distributed ledger work?

https://www.youtube.com/watch?v=SSo_EIwHSd4&ab_channel=SimplyExplained

Definition: A distributed ledger is a database that is consensually shared and synchronized across multiple sites, institutions, or geographies, accessible by multiple people.

Let's split the definition to understand better using a fictional cryptocurrency as

Example:

Ledger - A ledger can be considered as a digital register/Database which can store any content. Assume a new transaction detail "Trump paid 10 paise to Kim jung Un" is stored in the digital database or 'ledger' present in north korea.

Assume countries like India, China, Northkorea, and the United States also have ledgers with them. To achieve 'distributed' status all the above countries copies/ syncs the same transaction - 'Trump paid 10 paise to Kim jung Un' into their ledgers.

Synchronized ledger - Whenever any new transaction is made, that transaction details replicated to all the ledgers of countries India, China, United states and NorthKorea

(Note: new transaction - 'Kim paid 5 paisa to Trump'..etc , replication achieves synchronization with others)

Why Synchronization?

If all the ledgers belonging to countries are synchronized and if someone manipulates the data in their ledger, We can easily find out that black sheep by comparing with other synchronized ledgers.

Consensually shared- A new transaction generated is called valid transaction only if it meets certain rules, and these rules are universally accepted.Every one who is generating the transaction need to follow certain rules like sender must be verified and receiver must be verified (Such that north korea cannot generate whatever it wants and say it as a transaction).

Hence the transaction which abided by the universal rule and is shared/Replicated within all the ledgers constitutes a distributed ledger.

But wait! How can we ensure that the person who is writing the transaction is following the universally acceptable rule?

You might say that as it is universally acceptable , so we can ask other countries whether the particular transaction is valid or not. If a country didn't follow the universal rule the validity of other countries can help reject the transaction.

You must have heard about DDOS attack i.e distributed denial of service , generally this occurs when millions of fake requests generated by malicious bots hit the website and the website loses its control and crashes In this case the genuine requests cannot survive as the fake requests have destroyed the System Now if we don't want that to repeat in our use case , let us punish malicious dude !

In this case each country needs to put some of their valuable men at stake to prove their

Seriously, if they made a faulty transaction , the UN can jail those valuable men they kept at stake !

The below link has awesome explanation of whole blockchain:-

https://hackernoon.com/wtf-is-the-blockchain-1da89ba19348

Day 2

Different Types of Blockchains

There are basically two types of blockchain network related to access control.

1. Public Blockchain
2. Private(and consortium) Blockchain

Reference Video: https://www.youtube.com/watch?v=FcfPU3rYVAk

Public Blockchain

Public Blockchain is accessible to everyone – anyone who wants to read, write can join the blockchain and can perform respective operations. The information once validated in the network cannot be changed and no single entity can have control over the network. Bitcoin is one of the first public blockchain networks to prove the value can be moved anywhere around the world without banks or other third parties.

Take the example of Bitcoins, which are traded on such peer-to-peer networks. This cryptocurrency can be quite practical when the idea is to dissimulate criminal activities, since it's virtual, meaning that it can travel across borders in a seemingly anonymous way. "Seemingly" because we cannot discard what the benefits of a Blockchain, public or not, are: Every single transaction is recorded across all nodes of the network, which provides a complete transaction history accessible by anybody. Now, it's not a routing number and a bank account number that are recorded but rather an (unusable because just made-up) I'D like this one: 1BsghtSLRHtKNngkdBEengR76b53CHAtpyT. The trick is then to convert Bitcoins into fiat currencies since it must be done via a centralized exchange platform like Coinbase/Binance. And transactions from Coinbase to a bank account can be flagged.

Private Blockchain

These blockchains work in the same way as public blockchain but with restricted access. The restriction is applied to the users who are authorized to join the network and operate. They may have one or more entities that control the network.

Private blockchains are amazing for using at a privately-held company or organization that wants to use it for internal use-cases. By doing so, you can use the blockchain effectively and allow only selected participants to access the blockchain network. The organization can also set different parameters to the network, including accessibility, authorization, and so on!

Example: Hyperledger Fabric — hosted under Linux Foundation — is a private, permissioned and open source blockchain solution. Private means that blockchain networks are not publicly accessible and only invited parties can join the network. Permissioned means each party is clearly identified and every transaction is authenticated, authorized, validated and tracked.They are crafted to take advantage of blockchains without sacrificing the authority aspect of a centralized system.

Consortium Blockchain:

A consortium blockchain is a blockchain where the consensus process is controlled by a pre-selected set of nodes; for example, one might imagine a consortium of 15 financial institutions, each of which operates a node and of which 10 must sign every block in order for the block to be valid. The right to read the blockchain may be public, or restricted to the participants, and there are also hybrid routes such as the root hashes of the blocks being public together with an API that allows members of the public to make a limited number of queries and get back cryptographic proofs of some parts of the blockchain state. These blockchains may be considered "partially decentralized"

A consortium blockchain (also known as Federated blockchains) is a creative approach to solving the needs of organizations where there is a need for both public and private blockchain features. In a consortium blockchain, some aspects of the

organizations are made public, while others remain private.

The consensus procedures in a consortium blockchain are controlled by the preset nodes. More so, even though it's not open to mass people, it still holds decentralized nature. How? Well, a consortium blockchain is managed by more than one organization. So, there is no one single force of centralized outcome here.

To ensure proper functionality, the consortium has a validator node that can do two functions, validate transactions, and also initiate or receive transactions. In comparison, the member node can receive or initiate transactions.

In short, it offers all the features of a private blockchain, including transparency, privacy, and efficiency, without one party having a consolidating power.

Example: Let's take an Oil & Gas consortium made of twenty companies for example. Each company operates one node on the network. In a consortium Blockchain, a percentage of the consortium's participants must sign every block in the chain before the block can be validated. So, it could be ten or fifteen companies required to validate a transaction, before the same transaction is validated throughout the network for all participants. A federated Blockchain is less decentralized than a public Blockchain and less centralized than a private Blockchain in regard to write permissions.--Harshil

Public Blockchain	Private Blockchain	Consortium or Federated Blockchain
• Anyone can run BTC/LTC full node	• Anyone can't run a full node	• Selected members of the consortium can run a full node
• Anyone can make transactions	• Anyone can't run a full node	• Selected members of the consortium can make transactions
• Anyone can review/audit the blockchain	• Anyone can't review/audit the blockchain	• Selected members of the consortium can review/audit the blockchain

Day 3

Bitcoin: the First Blockchain-Based Cryptocurrency

Bitcoin is a decentralized digital currency that is exchanged between two parties without involving intermediaries like banks or other financial institutions. As defined in a whitepaper released by the hidden inventor of Bitcoin, Satoshi Nakamoto, Bitcoin is “a purely peer-to-peer version of electronic cash that would allow online payments to be sent directly from one party to another without going through a financial institution”.To understand Bitcoin, one needs to understand the underlying structure, the manner of operation of the Bitcoin ecosystem

How Does Bitcoin Work?

Bitcoin achieves elimination of intermediaries with the help of its underlying technology, blockchain.

Currently if you have to transfer funds to someone, one of the possible ways is by giving cash or alternatively use a trusted intermediary (example, a bank). Both the mechanisms, whether it be physical cash (with the central bank of the country as the guarantor) or electronic transfer, involve an intermediary (in the later case, a bank or another financial institution). When intermediaries are involved, there are transaction costs.

How the blockchain technology helps achieve elimination of intermediaries is by replacing trust that intermediaries bring to the table with cryptographic proof by the use of CPU computing power.

This cryptographic trust is built into Bitcoin through a wallet, a public key and a private key in the program.

Anyone can create a Bitcoin wallet for free by downloading the Bitcoin program. Each wallet contains a public key and a private key.

The public key is like an address or an account number via which any person can receive Bitcoins.

A private key is like a digital signature via which a person can send Bitcoins. The name suggests that private keys should be only held and known by the owner and public keys can be shared with anyone for receiving Bitcoins. That is where you would have heard in the news about Bitcoins being lost either due to a private key not being accessible or stolen by hackers.

Owners of Bitcoin addresses are not explicitly identified, but all transactions on the blockchain are public.

Since the inception of Bitcoin in 2009, each and every transaction that has occurred is stored in a ledger, which is considered immutable, non-tamperable and irreversible.

Bitcoin transactions are verified via telecommunication network nodes through cryptography and are then recorded in a decentralized distributed ledger called blockchain. This is one of the distinguishing aspects of Bitcoin from some other crypto assets, where there is centralized exchange (like the stock exchange) through which all transactions need to be routed or validated.

Mining in Bitcoin

Mining refers to the process of validating and recording transactions on the Bitcoin network. The primary purpose of Bitcoin mining is twofold: validating transactions to prevent fraud and adding new blocks to the blockchain, thereby creating new Bitcoins in a decentralized manner.

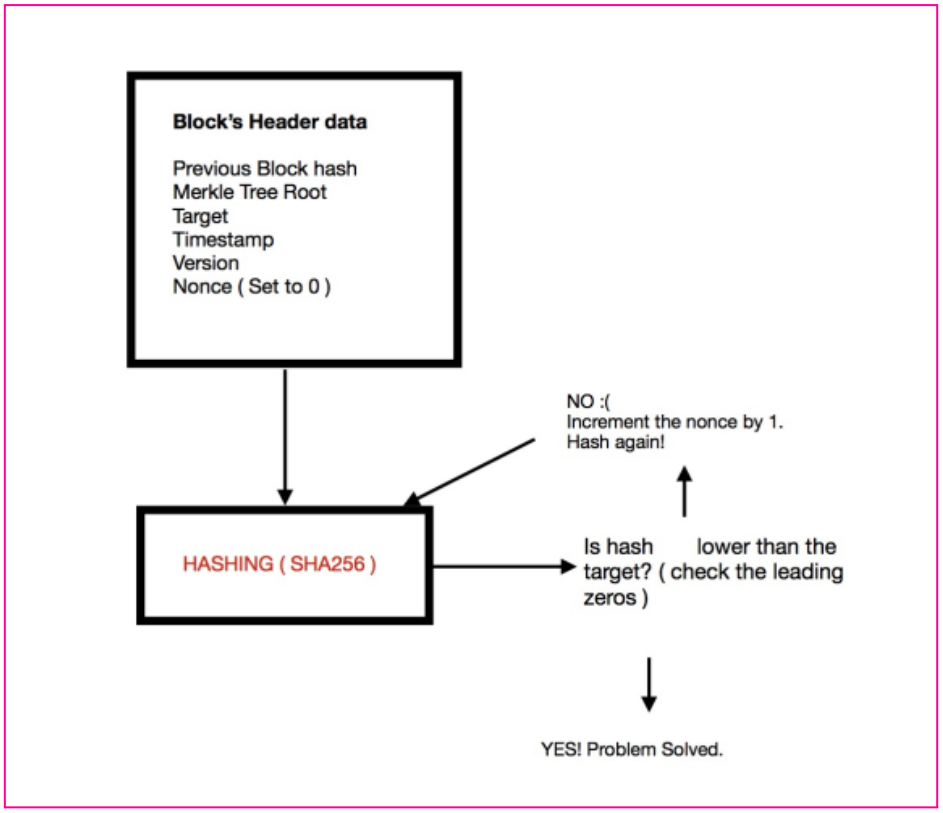
The process of bitcoin mining involves the verification of new transactions against the Bitcoin network, which results in the production of new bitcoins.

Miners validate new transactions and record them on the global ledger (blockchain). On average, a block (the structure containing transactions) is *mined* every 10 minutes. Miners compete to solve a difficult mathematical problem based on a cryptographic hash algorithm. The solution found is called the *Proof-Of-Work*. This proof proves that a miner did spend a lot of time and resources to solve the problem. When a block is 'solved', the transactions contained are considered *confirmed*, and the bitcoin concerned in the transactions can be spent. So, if you receive some bitcoin on your wallet, it will take approximately 10 minutes for your transaction to be confirmed.

Miners receive a reward when they solve the complex mathematical problem. There are two types of rewards: new bitcoins or transaction fees. The amount of bitcoins created decreases every 4 years (every 210,000 blocks to be precise). Today, a newly created block creates 12.5 bitcoins. This number will keep going down until no more bitcoin will be issued. This will happen around 2140, when around 21 millions bitcoins will have been created. After this date, no more bitcoin will be issued.

Miners can also receive rewards in the form of transaction fees. The winning miner can 'keep the change' on the block's transactions. As the amount of bitcoin created with each block diminishes, the transactions fees received by the miner will increase. After 2140, the winning miner will only receive transaction fees as his reward.

In a picture, mining is:



Why do we need mining?

https://youtu.be/4XXIOb6toy8

As you can see, mining is like a lottery. There is no way to predict which nonce will solve the problem.

In the case of Bitcoin, the hash function used is called SHA256. A hash algorithm always produces the same arbitrary length data given the same inputs. It is impossible to compute the same hash with two different inputs (collision). It is also impossible to predict the output of any given data in advance.

SHA256 always produces an output 256 bits long. Mining is finding the *nonce*, the only input that changes every time we run the hash function. It is very easy to prove that the nonce found indeed produces a valid hash. All the information is available, everyone can run the hash function and confirm if the hash is valid or not. Because it

is also impossible to predict what the nonce will be, it also acts as a proof that the miner worked to get a valid hash (Hence => *Proof-of-Work*).

Day 4

Ethereum: A Decentralized Platform for Smart Contracts

Ethereum is considered by many to be the second most popular cryptocurrency, surpassed now only by Bitcoin. According to The Motley Fool, the Enterprise Ethereum Alliance (EEA) has some big-name founding members, including Microsoft, Intel, and JPMorgan Chase.

“The adoption of Ethereum by the corporate world,” says CNBC, “means it could eventually be bigger than its early stage rival.” That means it’s time to get to know the Ethereum platform, including its features and applications, and what makes Ethereum different from Bitcoin.

What is Ethereum?

Ethereum is a blockchain-based computing platform that enables developers to build and deploy decentralized applications—meaning not run by a centralized authority. You can create a decentralized application for which the participants of that particular application are the decision-making authority

Ethereum Features

- Ether: This is Ethereum’s cryptocurrency.
- Smart contracts: Ethereum allows the development and deployment of these types of contracts.

- Ethereum Virtual Machine: Ethereum provides the underlying technology—the architecture and the software—that understands smart contracts and allows you to interact with it.
- Decentralized applications (Dapps): A decentralized application is called a Dapp (also spelled DAPP, App, or DApp) for short. Ethereum allows you to create consolidated applications, called decentralized applications.
- Decentralized autonomous organizations (DAOs): Ethereum allows you to create these for democratic decision-making.

These are Ethereum's essential features. Before going deep into the Ethereum tutorial, let's discuss each of these features in more detail.

1. Ether

Ether (ETH) is Ethereum's cryptocurrency. It is the fuel that runs the network. It is used to pay for the computational resources and the transaction fees for any transaction executed on the Ethereum network. Like Bitcoins, ether is a peer-to-peer currency. Apart from being used to pay for transactions, ether is also used to buy gas, which is used to pay for the computation of any transaction made on the Ethereum network.

Also, if you want to deploy a contract on Ethereum, you will need gas, and you would have to pay for that gas in ether. So gas is the execution fee paid by a user for running a transaction in Ethereum. Ether can be utilized for building decentralized applications, building smart contracts, and making regular peer-to-peer payments.

2. Smart Contracts

Smart contracts are revolutionizing how traditional contracts work, which is why you need to use the tutorial to become more familiar with them. A smart contract is a simple computer program that facilitates the exchange of any asset between two

parties. It could be money, shares, property, or any other digital asset that you want to exchange. Anyone on the Ethereum network can create these contracts. The contract consists primarily of the terms and conditions mutually agreed on between the parties (peers).

The smart contract's primary feature is that once it is executed, it cannot be altered, and any transaction done on top of a smart contract is registered permanently—it is immutable. So even if you modify the smart contract in the future, the transactions correlated with the original contract will not get altered; you cannot edit them.

The verification process for the smart contracts is carried out by anonymous parties in the network without the need for a centralized authority, and that's what makes any smart contract execution on Ethereum a decentralized execution.

The transfer of any asset or currency is done in a transparent and trustworthy manner, and the identities of the two entities are secure on the Ethereum network. Once the transaction is successfully done, the accounts of the sender and receiver are updated accordingly, and in this way, it generates trust between the parties.

Smart Contracts Vs. Traditional Contract Systems

In conventional contract systems, you sign an agreement, then you trust and hire a third party for its execution. The problem is that in this type of process, data tampering is possible. With smart contracts, the agreement is coded in a program.

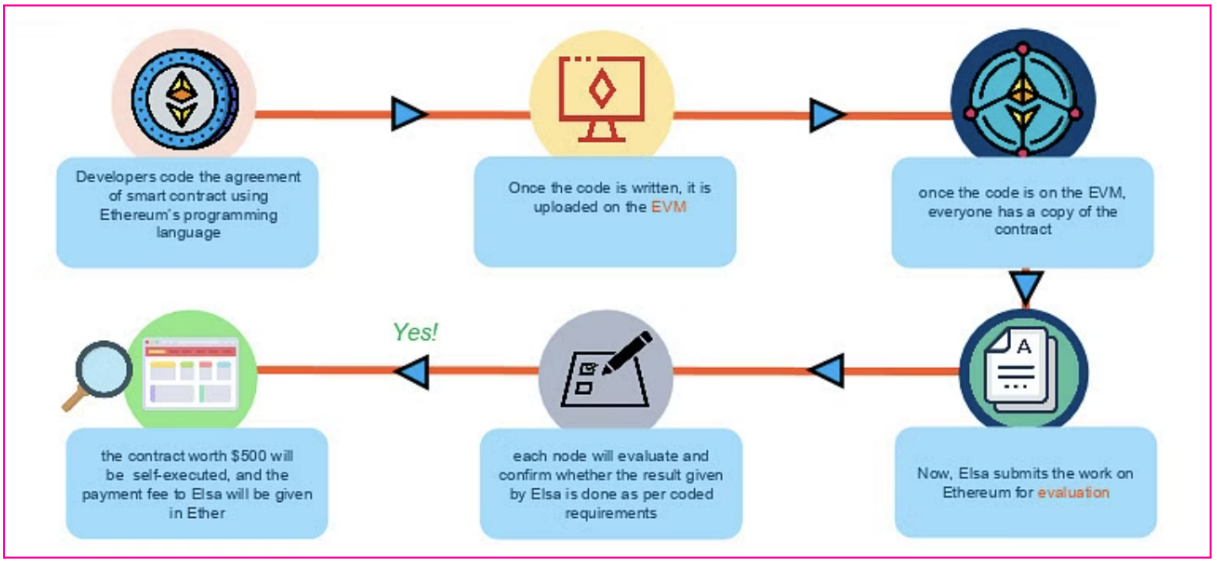
A centralized authority does not verify the result; it is confirmed by the participants on the Ethereum blockchain-based network. Once a contract is executed, the transaction is registered and cannot be altered or tampered, so it removes the risk of any data manipulation or alteration.

Let's take an example in which someone named Zack has given a contract of $500 to someone named Elsa for developing his company's website. The developers code the agreement of the smart contract using Ethereum's programming language.

The smart contract has all the conditions (requirements) for building the website. Once the code is written, it is uploaded and deployed on the Ethereum Virtual Machine (EVM).

EVM is a runtime compiler to execute a smart contract. Once the code is deployed on the EVM, every participant on the network has a copy of the contract. When Elsa submits the work on Ethereum for evaluation, each node on the Ethereum network will evaluate and confirm whether the result given by Elsa has been done as per the coding requirements.

Once the result is approved and verified, the contract worth $500 will be self-executed, and the payment will be paid to Elsa in ether. Zack's account will be automatically debited, and Elsa will be credited with $500 in ether.



3. Ethereum Virtual Machine

EVM, as mentioned above in this Ethereum tutorial, is designed to operate as a runtime environment for compiling and deploying Ethereum-based smart contracts. EVM is the engine that understands the language of smart contracts, which are written in the Solidity language for Ethereum. EVM is operated in a sandbox environment—basically, you can deploy your stand-alone environment, which can act as a testing and development environment. You can then test your smart contract (use it) “n” number of times, verify it, and once you are satisfied with the performance and the functionality of the smart contract, you can deploy it on the Ethereum main network.

Any programming language in the smart contract is compiled into the bytecode, which the EVM understands. This bytecode can be read and executed using the EVM. Solidity is one of the most popular languages for writing a smart contract. Once you write your smart contract in Solidity, that contract gets converted into the bytecode and gets deployed on the EVM, thereby guaranteeing security from cyberattacks.

a) How Does EVM Work?

Suppose person A wants to pay person B 10 ethers. The transaction will be sent to the EVM using a smart contract for a fund transfer from A to B. To validate the transaction; the Ethereum network will perform the proof-of-work consensus algorithm.

The miner nodes on Ethereum will validate this transaction—whether the identity of A exists or not, and if A has the requested amount to transfer. Once the transaction is confirmed, the ether will be debited from A’s wallet and will be credited to B’s wallet, and during this process, the miners will charge a fee to validate this transaction and will earn a reward.

All the nodes on the Ethereum network execute smart contracts using their respective EVMs.

b) Proof of Work

Every node in the Ethereum network has:

- The entire history of all the transactions—the entire chain
- The history of the smart contract, which is the address at which the smart contract is deployed, along with the transactions associated with the smart contract
- The handle to the current state of the smart contract

The goal of the miners on the Ethereum network is to validate the blocks. For each block of a transaction, miners use their computational power and resources to get the appropriate hash value by varying the nonce. The miners will vary the nonce and pass it through a hashing algorithm—in Ethereum, it is the Ethash algorithm.

This produces a hash value that should be less than the predefined target as per the proof-of-work consensus. If the hash value generated is less than the target value, then the block is considered to be verified, and the miner gets rewarded.

When the proof of work is solved, the result is broadcast and shared with all the other nodes to update their ledger. If other nodes accept the hashed block as valid, then the block gets added to the Ethereum main blockchain, and as a result, the miner receives a reward, which as of today stands at three ethers. Plus, the miner gets the transaction fees that have been generated for verifying the block. All the transactions that are aggregated in the block—the cumulative transaction fees associated with all the transactions are also rewarded to the miner.

c) Proof of Stake

In Ethereum, a process called proof of stake is also under development. It is an alternative to proof of work and is meant to be a solution to minimize the use of expensive resources spent on mining using proof of work. In proof of stake, the miner—who is the validator—can validate the transactions based on the number of crypto coins he or she holds before actually starting the mining.

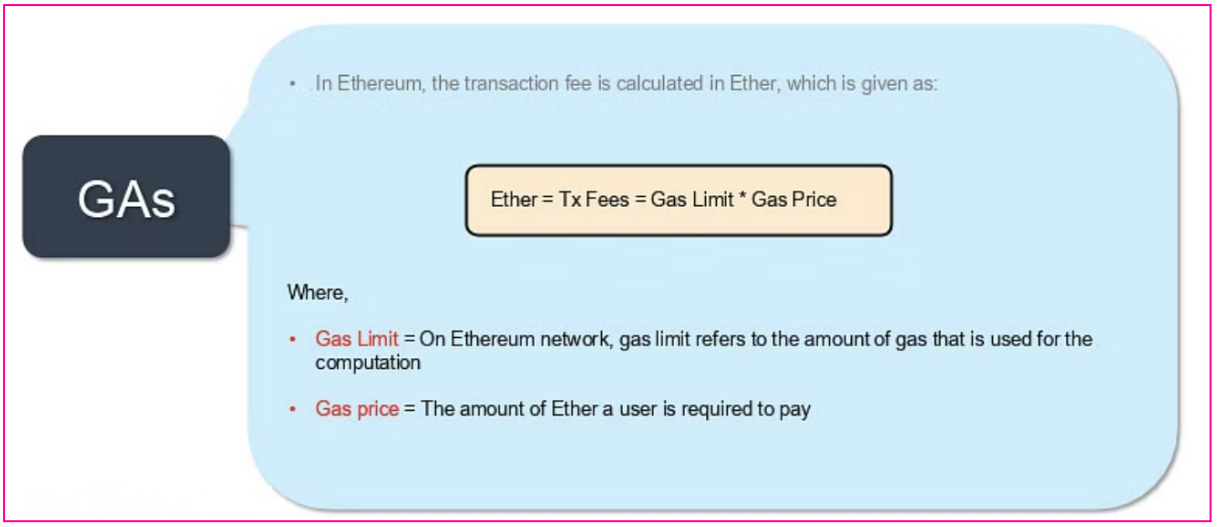
So, based on the accumulation of crypto coins the miner has beforehand, he or she has a higher probability of mining the block. However, proof of stake is not widely used as of now compared to proof of work.

d) Gas

Just like we need fuel to run a car, we need gas to run applications on the Ethereum network. To perform any transaction within the Ethereum network, a user must make a payment, in this case paying out ethers, to get a transaction done, and the intermediary monetary value is called gas.

On the Ethereum network, gas is a unit that measures the computational power required to run a smart contract or a transaction. So, if you must do a transaction that updates the blockchain, you would have to shell out gas, and that gas costs ethers.

In Ethereum, the transaction fees are calculated using a formula (see screenshot below). For every transaction, there is gas and its correlated gas price. The transaction fees equal the amount of gas required to execute a transaction multiplied by the gas price. “Gas limit” refers to the amount of gas used for the computation and the amount of ether a user is required to pay for the gas.



To understand the gas limit and price, let's consider an example using a car. Suppose your vehicle has a mileage of 10 kilometers per liter and petrol costs $1 per liter. Under these parameters, driving a car for 50 kilometers would cost you five liters of petrol, which is worth $5. Similarly, to perform an operation or to run code on Ethereum, you need to obtain a certain amount of gas, like petrol, and the gas has a per-unit price, called gas price.

If the user provides less than the amount of gas to run an operation, then the process will fail, and the user will be given the message "out of gas." And Gwei, as noted above, is the lowest denomination of ether used for measuring a unit of a gas price.

e) How is Ethereum Mining Different from Bitcoin Mining?

Description	Bitcoin	Ethereum
Hashing Algorithm	SHA-256	Ethash
Time is taken to mine a block	An average of 10 minutes	An average of 12-15 seconds
Reward	12.5 BTC	3 ETH
USD - 04/10/2019	1 Bitcoin = 5249.03	1 Ether = 180.89

Transaction Validation and Consensus Mechanisms

In the Blockchain network, a transaction is performed between two accounts and the transaction is flooded across the entire network. This Flooding of transaction in the network is known as Gossip Protocol. All the nodes in the network validate the transaction and add it to their transaction pool in case the transaction is valid

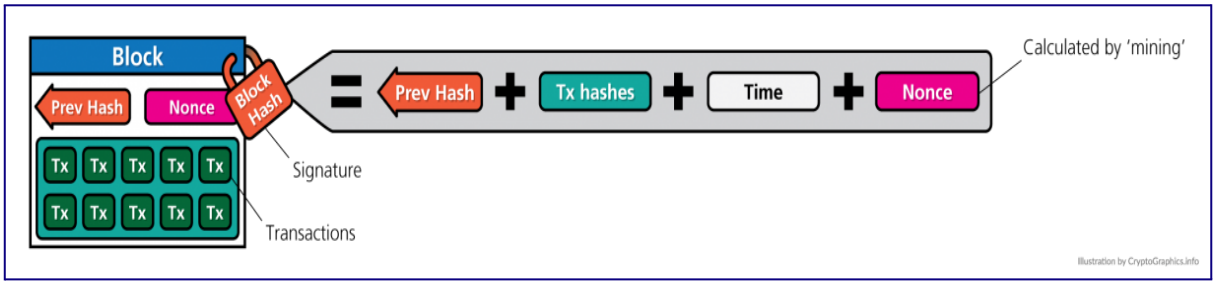
otherwise reject it. Each node has its own transaction pool that it maintains. All the transactions that are valid are placed inside the transaction pool.

In blockchain at a given point of time, it is not necessary for all the nodes to have the same transactions in their transaction pool. There could be various reasons for this to happen such as network latency, broken connection, less computational power etc. Bitcoin blockchain also defines the frequency at which blocks should be created inside the network which is set to 10 minutes in the current Bitcoin system.

The miner node takes the unconfirmed transactions in the mining pool and creates a block. Once a block is created, it gets flooded across the network using the gossip protocol. All the nodes receive the block and first validate the block if the block passes the validation they add it to their respective blockchain otherwise not.

Before going through the validation process we will have a look at the Block Anatomy.

A Block is divided into two – a Header and Body.



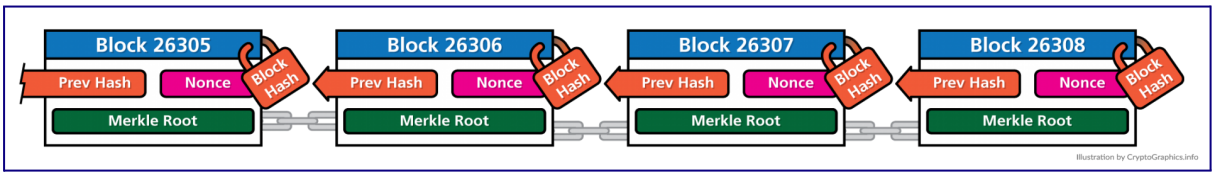
A Blocks Header is Comprised of the Following Components:

i) Merkle root - aggregation of all the hash values of the transactions into a single hash value.

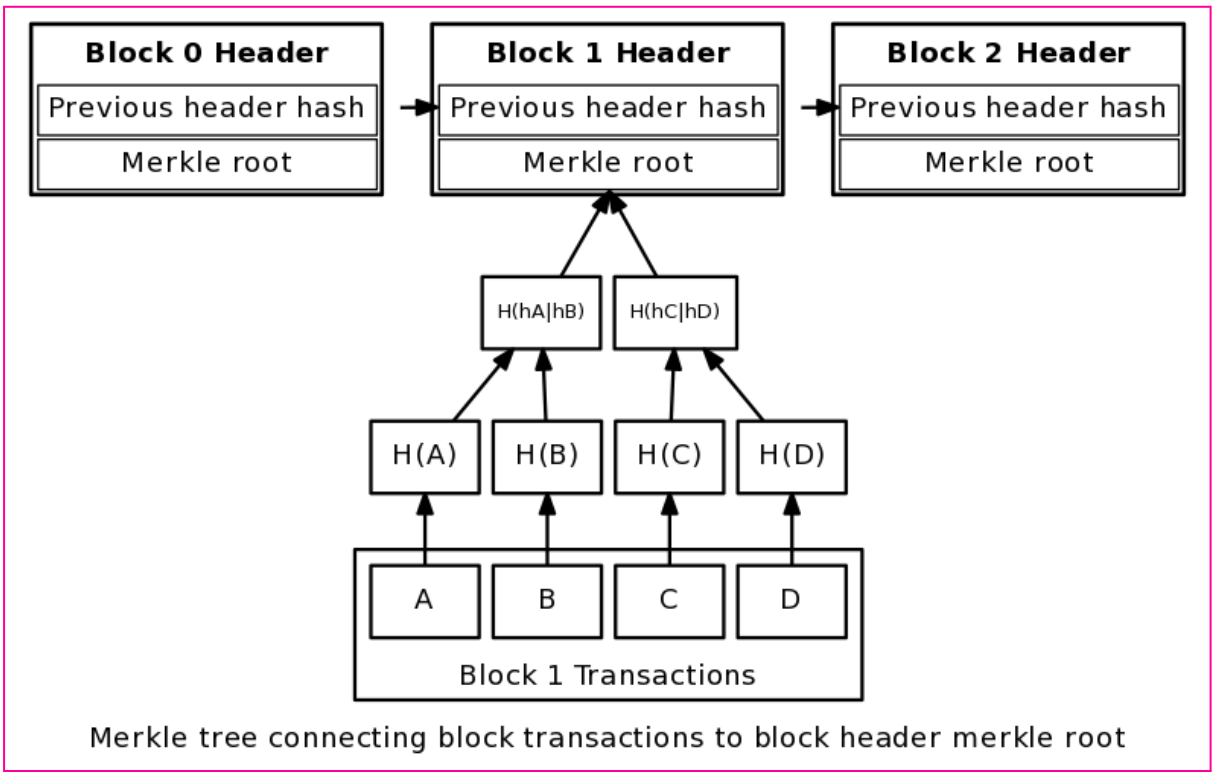
ii) Timestamp - Timestamp of the block creation time.

iii) Nonce - Random Value that is altered/updated to try different permutations to achieve the required difficulty level. You will learn more about this in the upcoming section.

iv) Transaction counter - Count of the number of transactions in a block.



Now you must be thinking what the block body contains, So it contains the **Merkle Tree.**



Merkle tree connecting block transactions to block header merkle root

A Merkle tree stores all the transactions in a block by producing a digital fingerprint of the entire set of transactions. It allows the user to verify whether a transaction can be included in a block or not.

Merkle trees are created by repeatedly calculating hashing pairs of nodes until there is only one hash left. This hash is called the Merkle Root, or the Root Hash. The Merkle Trees are constructed in a bottom-up approach.

Every leaf node is a hash of transactional data, and the non-leaf node is a hash of its previous hashes. Merkle trees are in a binary tree, so it requires an even number of leaf nodes. If there is an odd number of transactions, the last hash will be duplicated once to create an even number of leaf nodes.

The above example is the most common and simple form of a Merkle tree, i.e., Binary Merkle Tree. There are four transactions in a block: TX1, TX2, TX3, and TX4. Here you can see, there is a top hash which is the hash of the entire tree, known as the Root Hash, or the Merkle Root. Each of these is repeatedly hashed, and stored in each leaf node, resulting in Hash 0, 1, 2, and 3. Consecutive pairs of leaf nodes are then summarized in a parent node by hashing Hash0 and Hash1, resulting in Hash01, and separately hashing Hash2 and Hash3, resulting in Hash23. The two hashes (Hash01 and Hash23) are then hashed again to produce the Root Hash or the Merkle Root.

Day 5

Consensus Algorithms and how they work

1. Proof of Stake (PoS):

This is the most common alternative to PoW. Ethereum is shifting 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 increases 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 an agreement.

2. Proof of Burn (PoB):

With PoB, instead of investing into expensive hardware equipment, validators 'burn' coins by sending them to an address from where they are irretrievable. By committing the coins to an unreachable address, validators earn a privilege to mine on the system based on a random selection process. Thus, burning coins here means that validators have a long-term commitment in exchange for their short-term loss.

Depending on how the PoB is implemented, miners may burn the native currency of the Blockchain application or the currency of an alternative chain, such as bitcoin. The more coins they burn, the better are their chances of being selected to mine the next block.
While PoB is an interesting alternative to PoW, the protocol still wastes resources needlessly. And it is also questioned that mining power simply goes to those who are willing to burn more money.

3. Proof of Capacity:

In the Proof of Capacity consensus, validators are supposed to invest their hard drive space instead of investing in expensive hardware or burning coins. The more hard drive space validators have, the better are their chances of getting selected for mining the next block and earning the block reward.

4. Proof of Elapsed Time:

PoET is one of the fairest consensus algorithms which chooses the next miner using fair means only. It is mainly used in a permissioned Blockchain network where permission is required for accessing the network.

In this, every individual on the network is supposed to wait for a random amount of time. The participant who has finished waiting for the given set of time will get a chance to be on the ledger to create a new block.

5. Proof of Activity:

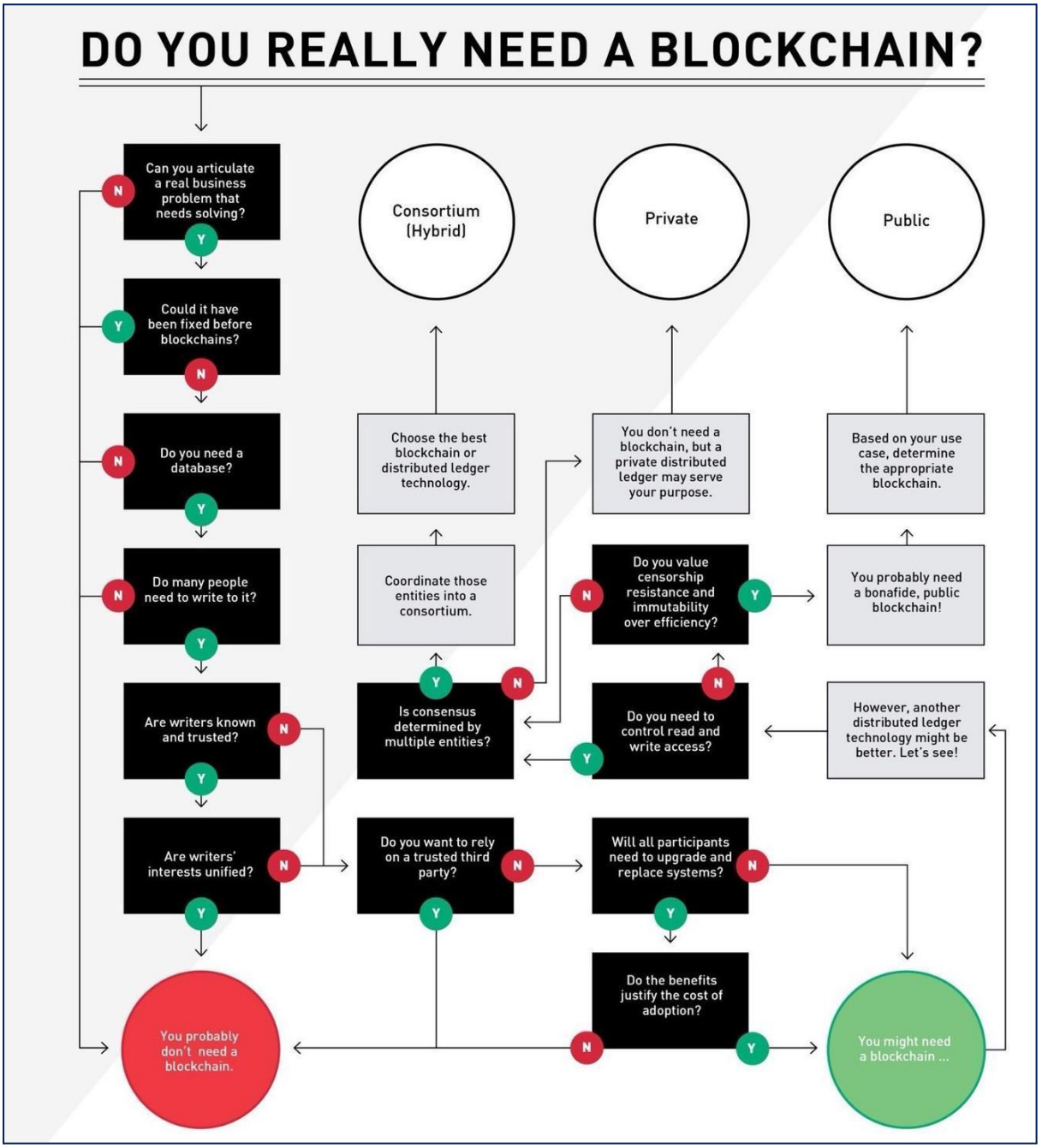
There also exist other consensus algorithms like Proof of Activity, Proof of Weight, Proof of Importance, Leased Proof of Stake, etc. It is therefore important to wisely choose one as per the business network requirement because Blockchain networks cannot function properly without the consensus algorithms to verify each and every transaction that is being committed.

List of consensus algorithms:

https://medium.com/hackernoon/consensuspedia-an-encyclopedia-of-29-consensus-algorithms-e9c4b4b7d08f

Do you actually need a blockchain?

https://medium.com/swlh/hyperledger-chapter-3-when-to-use-the-blockchain-technology-a5c414221bdf



Blockchain Terms: Block, Gas, Gas Price, Merkle Patricia Tree

1. Node - user or computer within the blockchain architecture (each has an independent copy of the whole blockchain ledger)

2. Transaction - smallest building block of a blockchain system (records, information, etc.) that serves as the purpose of blockchain

3. Block - a data structure used for keeping a set of transactions which is distributed to all nodes in the network

4. Chain - a sequence of blocks in a specific order

5. Miners - specific nodes which perform the block verification process before adding anything to the blockchain structure

6. Consensus (consensus protocol) - a set of rules and arrangements to carry out blockchain operations

7. Gas - Gas is the unit of calculation that indicates the fee for a particular action or transaction. For example, if your transaction is to simply add two numbers, that is 3 units of work. If it is multiplication, that would be 5 units of work and so on. This unit of work is called gas.

8. Gas Price - The Gas Limit is the maximum amount of Gas that a user is willing to pay for performing this action or confirming a transaction. For example, you can say "I am willing to pay 3 Gwei per gas" for my transaction. If your transaction takes 100000 gas and you set the gas price to 3 Gwei, you end up paying 300000 Gwei for your transaction.

9. Gas Limit - The price of Gas (Gas Price) is the amount that the user is willing to spend on each unit of Gas. As a developer, you don't want to blindly execute a transaction and realize your transaction took hundreds of dollars worth of Ether. To avoid this situation, you can specify a gas limit which indicates the maximum amount of gas you are willing to buy to execute your transaction.

Don't confuse this with the Block gas limit. Block gas limit is the maximum cap applied to each block in Ethereum. Currently, a block can only include transactions whose total sum of gas is less than 8 million. The reason you have a limit for each block is to prevent someone from just writing an infinite loop in which case the

transaction wouldn't complete it's execution resulting in the block not getting mined.

Gas Explained:

To better understand how gas works in Ethereum, let's use an analogy. Suppose you are going on a road trip. Before you do so you go through these steps:

- You go to the gas station and specify how much gas you want to fill up in your car.
- You get that gas filled up in your car.
- You pay the gas station the amount of money you owe them for the gas.

Now, let's draw parallels with Ethereum.

Driving the car is the operation that you want to execute, like executing a function of a smart contract.

- The gas is well….gas.
- The gas station is your miner.
- The money that you paid them is the miner fees.

All the operations that users want to execute in ethereum must provide gas for the following:

- To cover its data aka intrinsic gas.
- To cover its entire computation.

10. Nonce – Nonce is the central part of this Proof of Work. The Nonce is a random whole number, which is a 32-bit (4 byte) field, which is adjusted by the miners, so that it becomes a valid number to be used for hashing the value of block. Nonce is the number which can be used only once. Once the perfect Nonce is found, it is added to the hashed block. Along with this number, the hash value of that block will get rehashed and create a difficult algorithm.

Workshop Details

10th June: **Live workshop on “How Blockchain Functions" along with the doubt session for Module 1.**

Optional: If you want to go deeper in understanding cryptography you can go through the study material (not important for the bootcamp)

Advance Topics

Cryptographic Primitives: Symmetric Cryptography, Asymmetric Cryptography, Hashing Function

Reference video: https://www.youtube.com/watch?v=AQDCe585Lnc

Cryptography

Asymmetric cryptography, also known as public-key cryptography, is one of the key components of blockchain technology. This form of cryptography allows everyone to verify the integrity of transactions, protect funds from hackers and much more. But how does it work?

To understand asymmetric cryptography it is important to first understand the meaning of cryptography.

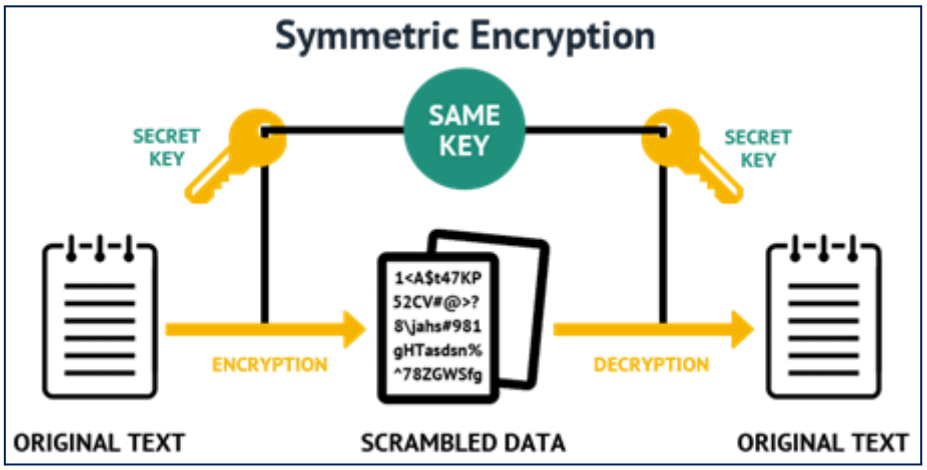
Cryptography is a method of using advanced mathematical principles in storing and transmitting data in a particular form so that only those for whom it is intended can read and process it. Encryption is a key concept in cryptography—It is a process whereby a message is encoded in a format that cannot be read or understood by an eavesdropper. The technique is old and was first used by Caesar to encrypt his messages using Caesar cipher. A plain text from a user can be encrypted to a cipher text, then sent through a communication channel and no eavesdropper can interfere with the plain text. When it reaches the receiver end, the cipher text is decrypted to the original plain text.—SSL2BUY

Asymmetric cryptography is one of these methods and is a more advanced version of symmetric cryptography, which we will explain first.

Symmetric cryptography

Symmetric cryptography is a 'simple' form of cryptography which uses a single key to encrypt and decrypt data. This key can be almost anything, ranging from a number to a word to a random string of characters. This key is then used to encrypt the data after which the data can get sent across a network safely. To decrypt the data the receiver needs the key (the same one that the sender used to encrypt the data).

To get a better idea of how this works, we have created a visualization of the process below. First of all the sender encrypts a message with the shared key, the sender then sends the message without having to worry that anybody without the shared key is able to read the message. The receiver then receives the encrypted message and decrypts it (with the same shared key).



Symmetric cryptography visualized

The main downside of this cryptography method is that the key needs to be shared with everyone who needs to access the data, which can often be quite hard.

Asymmetric Cryptography

Now you are familiar with symmetric cryptography, let's dive into asymmetric cryptography. Asymmetric cryptography is similar to symmetric cryptography, but is a bit more complex and also has a solution to the main downside of symmetric cryptography.

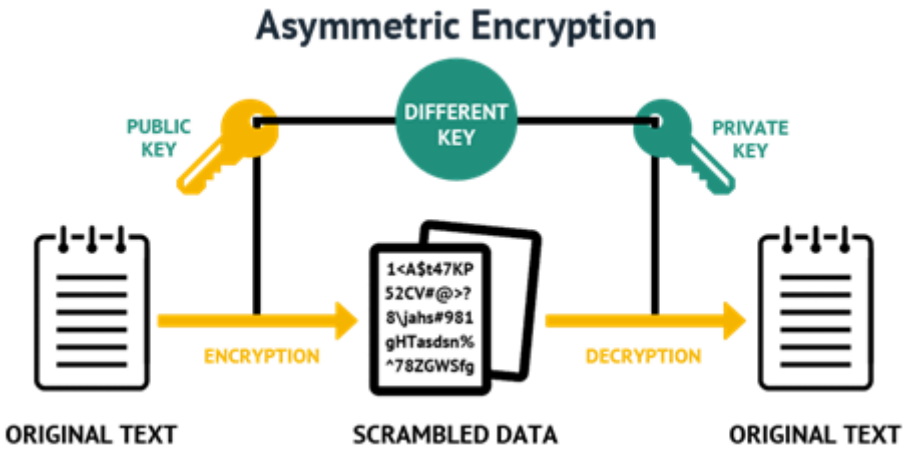
The main distinction from symmetric cryptography is the usage of keypairs. Asymmetric cryptography uses keypairs, instead of a shared key, in order to encrypt and decrypt data.

Keypairs consist of 2 parts, a public key and a private key.

A public key can be seen as a username, it is available to everyone, can be shared with everyone, and everyone can view the history of the account with that username. The username is tied to a password (private key), but there is absolutely no way to derive the password (private key) from a username. It is also not possible to authorize any action on the account with just the username.

A private key can be seen as a password to an account with a certain username. It is not publicly available and should not be shared with anyone. The private key is used to authorize actions on the accounts. Unlike with 'normal accounts', to access the account, or to authorize any action on the account, only the private key is needed.

In the graphic below you can see how these keys work in practice, when sending a message to somebody securely. First of all the sender encrypts the message with the public key of the receiver, the sender can then send the (encrypted) message safely, as the only way to view the message is to decrypt it with the corresponding private key which only the receiver has. The receiver then receives the message and is able to decrypt it using the private key.



Asymmetric cryptography visualized

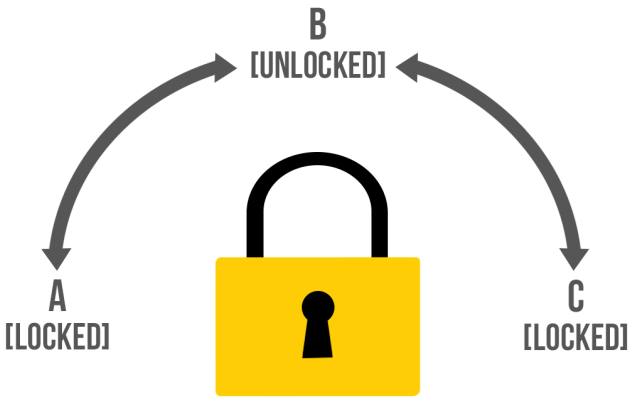
Due to the usage of keypairs asymmetric cryptography is a (much) safer way to encrypt data and make sure only those who are supposed to receive it are able to

receive it. These keypairs also allow themselves to be used for authentication purposes, which we will talk more about below.

Digital Signatures

Digital signatures are essentially signatures that provide integrity using asymmetric cryptography. They are widely used in many protocols for authentication purposes and have already proven to be both very useful and secure.

Digital signatures are incorruptible and easily verifiable thanks to their usage of asymmetric cryptography. Since they use asymmetric cryptography (and a private key is only linked to a single person) digital signatures also have the quality of non-repudiation, meaning they can be as legally binding as a normal signature. But how can you verify the integrity of certain messages, or how could you ensure others are able to verify the integrity of your message?



Public and private key functionality visualized

The answer to these questions is rather simple and can be explained with the image above. The lock in the image above has 2 keys, the public key, which can only turn counterclockwise, and a private key, which can only turn clockwise. We assume the public key of each lock is widely available.

Let's say Alice wants to send an unencrypted message to Bob, but Bob wants to ensure Alice has sent the message. Alice puts the message in a box and then puts the lock on the box. She then locks the lock using her own private key (from B to C). After locking the box she sends it to Bob. Bob then receives the box and can verify the integrity (and sender) of the message using Alice's public key. If Alice's public key

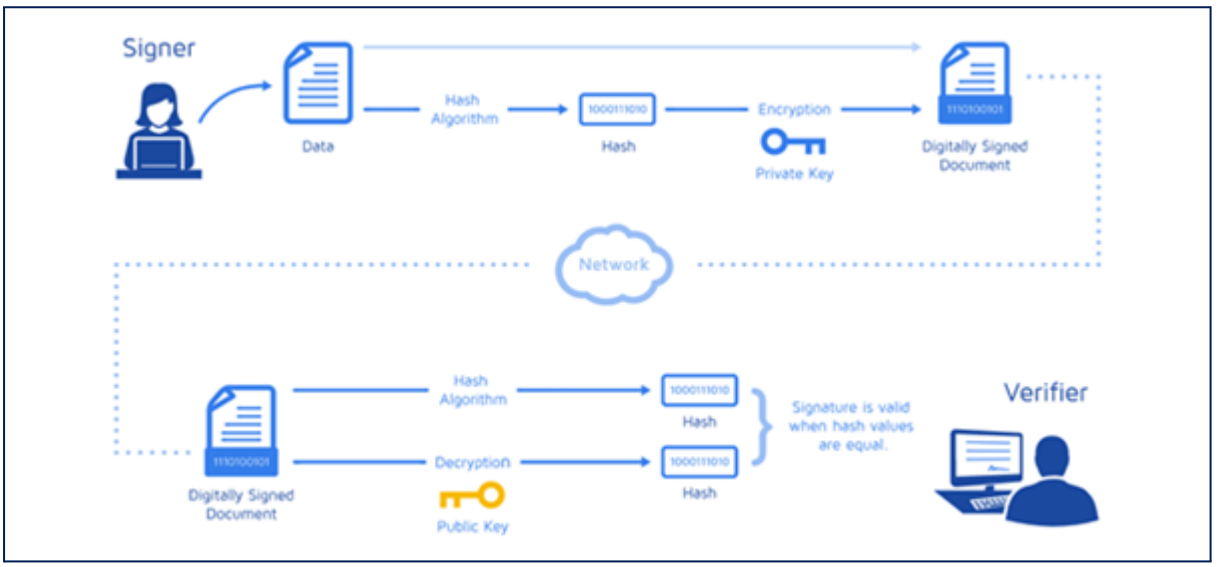
does not open the box the integrity of the message and box should be considered violated.

If you encrypt ("lock") something with your private key, anyone can decrypt it ("unlock"), but this serves as a proof you encrypted it: it's "digitally signed" by you. — Panayotis Vryonis

If Alice wants to send an encrypted message to Bob, and only Bob should be able to both decrypt the message and verify the identity of the sender, she would have to take some extra steps. First of all Alice puts the message in a box and puts a lock on the box, she then locks the box using her own private key (from B to C). She then puts the (locked) box in a new box and puts a new lock on the box. She locks the new lock using Bob's public key (from B to A). After Alice is done she sends the box to Bob. Once Bob receives the box he can open the otterbox with his private key, he can then open the inner box with Alice's public key. If the inner box can not be opened with Alice's public key the the integrity of the message and box should, yet again, be considered violated.

These 'locks' are called digital signatures.

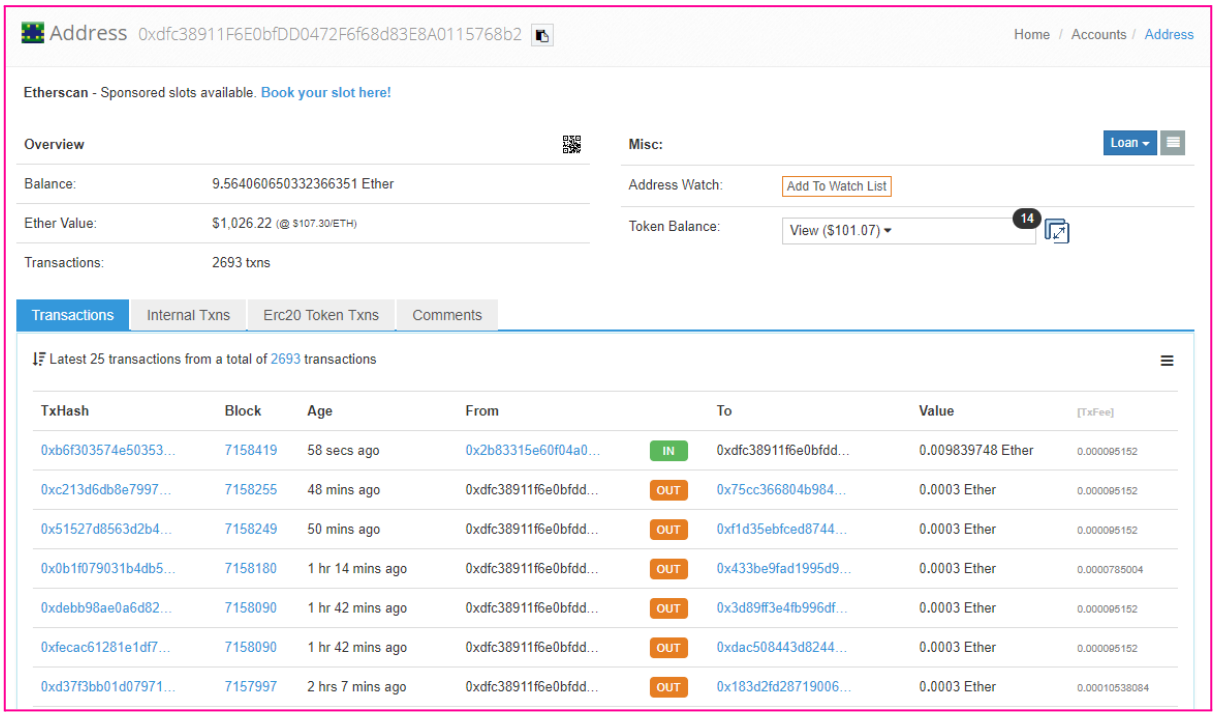
Note that in reality both a signature generation algorithm, which takes a key and data, and a signature verification algorithm, which takes a message and signature, are needed to do the things explained above.



The usage in blockchains

While both asymmetric cryptography and digital signatures have been popular ever since they were invented, they are probably most known for their implementations in blockchains. Since blockchains are essentially widely distributed ledgers, it is of the essence that the cryptography used is both reliable and functional.

Most cryptocurrencies use keypairs (and thus asymmetric cryptography) to manage 'addresses' on the blockchain. The public key is the address, which 'holds' the tokens and it can be viewed by anyone. The private key is used to access the address and authorize actions for the 'address'. Below is an example of an arbitrary address on the Ethereum blockchain.



Arbitrary Ethereum address

Digital signatures are also widely used in cryptocurrencies. They can be used to sign transactions more safely (offline) and are also used in multisignature contracts and wallets. These multisignature contracts and wallets require digital signatures from multiple (different) private keys before any action can be executed.

SHA-256

Presently SHA-256 is the most secure hashing function. This function expresses the possible combinations or values that results from the given input data. SHA stands for Secure Hashing Function, and 256 expresses the numerical quantity of the fixed bit length. This means that the target is correct 256 bit, and as mentioned, Bitcoin uses a 65-hexadecimal hash value.

Using the SHA-256 function makes it (nearly) impossible to duplicate a hash. That's because there are just too many combinations to try and process. Therefore, a significant amount of computational work is required. So much so that Bitcoins are no longer mined with personal computers. Presently, to mine, you need to rely on Application-Specific Integrated Circuits or ASICs. Achieving this target has a probability of 2^256. If you remember your exponents, you will deduce this is an incredibly difficult variable to hit.

Furthermore, using this hash function means that such a hash is intentionally computationally impractical to reverse. The intentional result is that it requires a random or brute-force method to solve for the input.

A Case in Point

Consider the following, if I have 1 six-sided dice, I have a 1 in 6 chance of rolling a 6. However, the more sides my dice has (say 256 sides), the more my chances of rolling a 6 decrease. That's 1 in 256, which is still better than your odds of using brute-force on an extent hash.

A hash rate is then the speed at which hashing operations take place during the mining process. If the hash rate gets too high and miners solve the target too quickly, this increases the potential for a collision. When that happens, the difficulty of the hash needs adjusting accordingly. For example, at present, a Bitcoin is mined/hashed about every 10 minutes.

Collision Resistance

Due to the complexity and sensitivity of SHA-256, reversing the hash sequence in an effort to find the original input data is basically impossible. The difficulty of meeting SHA-256 means that this hash is extremely secure because it is"collision resistant." Collision resistance expresses the likelihood of two different networks solving the same hash at the same chance is minuscule.

Therefore, given the possible permutations of SHA-256, the probability of a collision is negligible. Below is a comparison of two different hash outcomes. The first only uses the single hash function (SHA-1), while the second uses the double hash function (SHA-256). And as you can see, the double hash function produces a much more complicated hash and as a consequence is far more collision resistant.

www.thesslstore.com.cer	(application/x-x509-ca-cert) - 2042 bytes
SHA-1	2377b8642eac9f622b826f7a83500704f3cc3488
SHA-256	0feb43ff09dbea9487421fd3930527828703995e9f1d95d01656e3474cff2493

Here are a few examples of other cryptographic hash functions and when collision resistance broke, and it will become evident why SHA-256 is currently the favored hash:

- **MD 5:** It produces a 128-bit hash.
- Collision resistance was broken after ~2^21 hashes.
- **SHA 1:** Produces a 160-bit hash.
- Collision resistance broke after ~2^61 hashes.
- **SHA 256:** Produces a 256-bit hash.
- This is currently being used by Bitcoin.
- **Keccak-256:** Produces a 256-bit hash.
- Currently used by Ethereum.

If you want to more deep down into how SHA-256 works then you can explore the below link

https://medium.com/bugbountywriteup/breaking-down-sha-256-algorithm-2ce61d86f7a3

ECDSA

Elliptic Curve Digital Signature Algorithm or ECDSA is a cryptographic algorithm used by Bitcoin to ensure that funds can only be spent by their rightful owners.

A few concepts related to ECDSA:

i) Private Key: A secret number, known only to the person that generated it. A private key is essentially a randomly generated number. In Bitcoin, someone with the private key that corresponds to funds on the block chain can spend the funds. In Bitcoin, a private key is a single unsigned 256 bit integer (32 bytes).

ii) Public Key: A number that corresponds to a private key, but does not need to be kept secret. A public key can be calculated from a private key, but not vice versa. A public key can be used to determine if a signature is genuine (in other words, produced with the proper key) without requiring the private key to be divulged. In Bitcoin, public keys are either compressed or uncompressed. Compressed public keys are 33 bytes, consisting of a prefix either 0x02 or 0x03, and a 256-bit integer called *x*. The older uncompressed keys are 65 bytes, consisting of constant prefix (0x04), followed by two 256-bit integers called *x* and *y* (2 * 32 bytes). The prefix of a compressed key allows for the *y* value to be derived from the *x* value.

iii) Signature: A number that proves that a signing operation took place. A signature is mathematically generated from a hash of something to be signed, plus a private key. The signature itself is two numbers known as *r* and *s*. With the public key, a mathematical algorithm can be used on the signature to determine that it was originally produced from the hash and the private key, without needing to know the private key. Resulting signatures are either 73, 72, or 71 bytes long (with approximate probabilities of 25%, 50%, and 25%, respectively--although sizes even smaller than that are possible with exponentially decreasing probability).

What is double-spending?

As the term suggests, double-spending means spending the same money twice. With physical cash such as coins and notes, this simply isn't possible and therefore isn't an issue.

Let's look at this example: You go to Dmart and buy groceries worth Rs.100. You pay in cash and hand over a Rs.100 note. As soon as the cashier puts the cash in the register, you can't re-spend it unless you physically steal it.

Why is double-spending an issue for digital money?

Digital money is different from cash. When you make a transaction with digital cash, you are broadcasting the transaction to all the 'nodes' in the network (nodes are computers that run the software on which the currency is supported). These nodes need to receive and confirm the transaction, which takes time. Hence the problem: what's to stop someone copying a transaction and rebroadcasting it before it's been confirmed on the network?

To prove that no attempts to double-spend have occurred, the blockchain provides a way for all nodes to be aware of every transaction. With bitcoin, all transactions are publicly announced to all nodes. They can then agree on a single history of the order in which they were received. Bitcoin's solution to double-spending is that if the majority of the nodes agree on which transaction was first to be received, later attempts to double-spend are irrelevant.

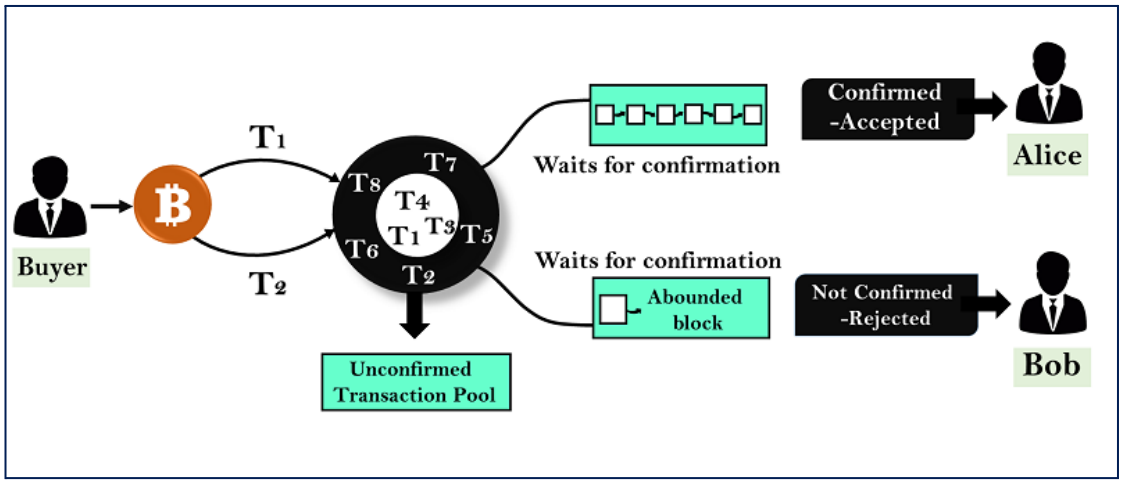
- Double-spending occurs when a blockchain network is disrupted and cryptocurrency is essentially stolen. The thief would send a copy of the currency transaction to make it look legitimate, or might erase the transaction altogether.

- Although it is not common, double-spending does occur. What is much more likely, however, is cryptocurrency being stolen from a wallet that wasn't properly secured.

- The most common method of double-spending is when a blockchain thief will send multiple packets to the network, reversing the transactions so that it looks like they never happened.

Let us suppose you have 1 BTC and try to spend it twice. You made the 1 BTC transaction to Alice. Again, you sign and send the same 1 BTC transaction to Bob. Both transactions go into the pool of unconfirmed transactions where many unconfirmed transactions are stored already. The unconfirmed transactions are

transactions which are not picked by anyone. Now, whichever transaction first got confirmations and was verified by miners, will be valid. Another transaction which could not get enough confirmations will be pulled out from the network. In this example, transaction T1 is valid, and Alice will receive the bitcoin.



.