Unit 1

Blockchain

- Blockchain is defined as a distributed, replicated peer-to-peer network of databases
- It allows multiple non-trusting parties to transact without a trusted intermediary and maintains an ever-growing, append-only, tamper-resistant list of time-sequenced records.
- Blockchain is a type of distributed ledger that sits on the internet for recording transactions and maintaining a permanent and verifiable record-set of Information.

History of Blockchain

- This technology was first published in 2008 by Satoshi Nakamoto in white paper titled “A Peer-to-peer Electronic Cash System”.
- The thought behind the design was to create a decentralized digital currency that is free from government regulation whereby two people can confidently trade directly with one another without the need for mediators or intermediaries.
- In 2009, The concept became a reality when Satoshi Nakamoto implemented the first application of the Blockchain we all know as Bitcoin.

Bitcoin and Blockchain

- Though Bitcoin and Blockchain are often referred to interchangeably, they are not the same.
- Blockchain is the underpinning technology that the Bitcoin was built on.
- Approximately there are around 15,000 mainstream cryptocurrencies in the market.

Evolution of Blockchain

Preblockchain – The Early Years	The 1950s	> First computers developed and adopted
	The 1960s	> 1969: Arpanet, the early Internet on the peer-to-peer network
	The 1970s	> 1973: Public-key cryptography implemented by Clifford Cocks > 1977: RSA, the public-key cryptosystem that is widely used for secure data transmissions, is released. > 1979: Ralph Merkle patents the concept of hash trees now called Merkle tree
	The 1980s	> 1982: IBM Personal Computer launched with DOS operating system
	The 1990s	>1991: Staurt Haber and W Scott Stornetta work on cryptographically secure chain. >1997: Proof-of-work with Hashcash >1996: Nick Szabo introduced bit gold as a mechanism for decentralized digital currency and smart contract. >2000: Stefan Knost introduced a general cryptographic theory for secured chains.

Evolution of Blockchain

Blockchain 1.0:	2008	> Oct 31: Satoshi Nakamoto releases Bitcoin white paper - a concept on the peer to peer payment system. > Bitcoin.org registered in August
	2009	> Jan 03: Bitcoin Genesis blocked mined. > Jan 12: hal Finney receives first Bitcoin transaction, thus launching the first application of a public blockchain > Oct 12: Bitcoin registered open souce code > Oct 31: Bitcoin Market - Bitcoin recognized as a digital currency
	2010	> May 22: First Bitcoin purchase - 10,000 BTC for a $25 pizza > Nov 06: Bitcoin marketplace surpassed $1 million

Evolution of Blockchain

Blockchain 1.0:	2011	> Namecoin, the first Bitcoin fork - Litecoin, releases as an alternative to bitcoin with different mining algorithm and faster transaction speed > Bitcoin reaches parity with the US dollar (1BTC = 1USD)
	2012	> Diaspora, the first decentralized social network > Ripple, a permissioned blockchain, is launched. A payment protocol focusing on integration with banking systems. > The bitcoin foundation launched in September
	2013	> March 28: Bitcoin marketplace surpasses $1 Billion > May 02: First bitcoin ATM unveiled > The University of Nicosia in Cyprus accepts Bitcoin > Mastercoin(the first Altcoin) is one of the earliest. > Vtalik Buterin releases Ethereum white paper.

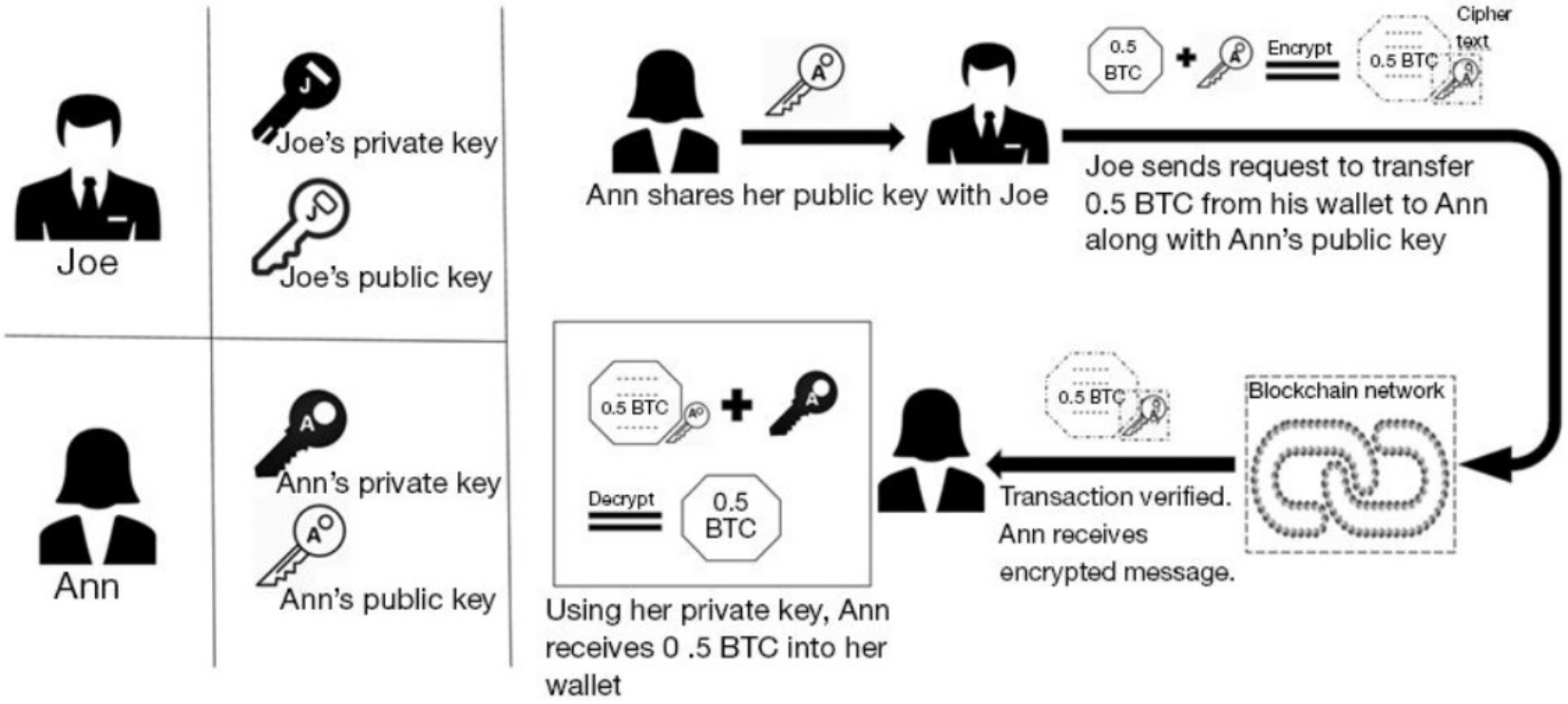
Evolution of Blockchain

Blockchain 2.0: Ethereum and Smart Contract	2014	> Establishment of R3: a consortium of over 40 financial Institutions committed to implementing blockchain technology > Ethereum Blockchain is funded by crowdsale > PayPal announces Bitcoin integration > Microsoft accepts Bitcoin
	2015	> Genesis block in Ethereum created > Linux Foundation unveils Hyperledger to boost blockchain development > Visa, Citi, Nasdaq, Capital One and Fiserv invest $30M in Blochcian startup Chain.com

Evolution of Blockchain

Blockchain 3.0 - Distributed Applications	2016	> Bug in Ethereum DAO code exploited, causing theft of $50M in ether
	2017	> EOS unveiled by Block.one as a new Blochchain protocol for industry-scale decentralized applications.
Blockchain 4.0	2018-future	> Current Bitcoin marketplace is valued above $2T > TRON, a blockchain platform for the entertainment Indusry > Business-oriented hybrid blockchain projects > Integration with IoT, AI and Big Data

Blockchain Architecture



Blockchain Architecture

- Components of Blockchain
  - Node
  - Ledger
  - Wallet
  - Nonce
  - Hash
  - Mining
  - Consensus Protocol

Node

- A Node is an electronic device that is connected to the internet.
- In Blockchain, Any computer or hardware device that is connected to the blockchain network is a node.
- All the nodes in the network have a copy of the blockchain ledger and are interconnected.
- Types of node : A full node, A partial node

Ledger

- A ledger is blockchain technology refers to the database of information that is immutable.
- The traditional database works on the CURD principle. (Create, Read, Update, Delete)
- Ledger works on the principle of Append-Only.

Wallet

- A Wallet in the blockchain world is a digital wallet that allows users to manage cryptocurrency.
- Using a blockchain wallet, One can receive and send cryptocurrency.

Mining

- Mining is the mechanism whereby nodes in the cryptocurrency world validate new transactions and add them to the blockchain ledger.

Consensus Protocol

- Consensus protocols are a set of rules whereby nodes in a network can achieve agreement on the data value or state of the network such that it benefits the network as a whole and does not focus on individual interests.
- Consensus protocols are used in blockchain to ensure that all transactions are validated before being added to the blockchain

Encryption

- Encryption is a part of human history. Since the Roman emperor Gaius Julius Cesar (100–44 BC), many people have tried to create secure encryptions.
- The first attempt to create a cryptography system that is considered safe today was made by Frank Miller (1842–1925), who proposed one-time pads to enhance Vigenere encryption.
- These problems were partially solved by the German Army during World War II with Enigma. Enigma is a machine that encrypts and decrypts a text by identifying the positions of three rotors and a few patches. This allowed the German Army, particularly the U-boats, to have months of valid codes on a few sheets of paper. This obviously was way less secure than the perfect Vigenere cipher; in fact, it was breached during WWII.

Vigenere cipher

- First described by Giovan Battista Bellaso in 1553, the cipher is easy to understand and implement, but it resisted all attempts to break it until 1863, three centuries later.

- Plaintext: ATTACKATDAWN
- Key: LEMONLEMONLE
- Ciphertext: LXFOPVEFRNHR

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Types of encryption

- Symmetric Encryption
- Asymmetric Encryption

Symmetric Encryption

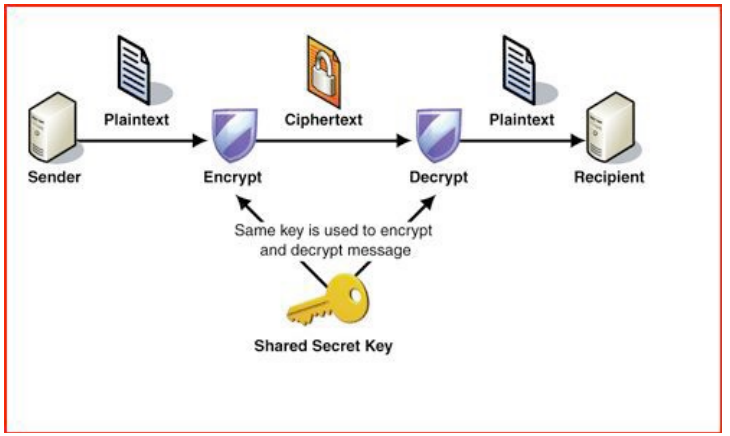
- A systems where by knowing the algorithm and the key, you can both encrypt and decrypt a message. This kind of encryption is known as symmetric encryption.
- The big advantage of this kind of encryption is that it is easy because it does not require complex math and much calculation to execute.
- On the other hand, it makes critical the key exchange moment and key management. In fact, the key has to be exchanged before the transmission can start between the parties, and it has to be done securely.

Symmetric Encryption

- As for the key management problem, since both parties know the same secret (in fact, this kind of cryptography is also called shared secret), if you have multiple people that have to communicate with each other, you will need *n(n-1)/2* keys
- This means that in a group of 20 people, you'll need 190 keys.

Symmetric Encryption

- The following is the schema of a communication using a symmetric encryption.
- The types of symmetric encryption that are used are as follows:
  - Stream cipher (ex, RC4) (WEP, WPA, SSL, TSL)
  - Block cipher (AES, DES, 3DES)

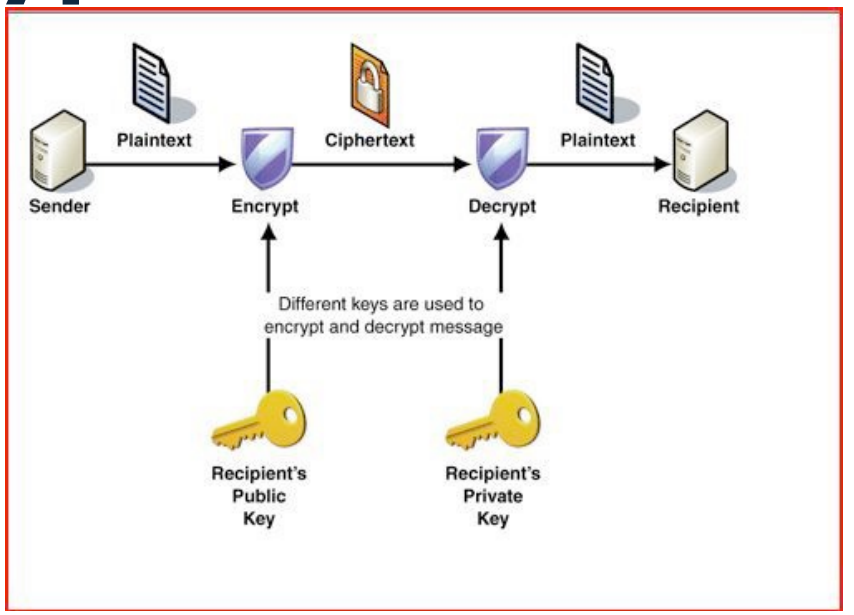


ASymmetric Encryption

- Asymmetric encryption has some core differences from symmetric encryption.
- The first that you can immediately notice is that in asymmetric encryption there are two keys: one public key to encrypt and a private key to decrypt.
- From this concept, one of the names of asymmetric encryption is derived: public key encryption.
- This approach does simplify greatly the key exchange and key management. For the key management, you only need a pair of keys (private/public) for each person. So if you have 20 people that have to communicate between themselves, you'll only need 20 pairs of keys.

ASymmetric Encryption

- Asymmetric encryption also allows you to sign messages, encrypting them with your private key, so that anyone with your private key can decrypt the message, understanding that it is coming from you and granting non-repudiation.
- The process in an exchange of secure data with an asymmetric encryption is as seen here.
- The types of asymmetric encryption that are used are as follows:
  - Diffie-Hellman (IPSec)
  - RSA Algorithm
  - Elliptic Curve Cryptography



comparison

Domain	Symmetric	Asymmetric
Able to grant	Confidential	Confidential, Offering Integrity, Authentication, and Non-repudiation
Needed keys	A single shared key	A public Key and A private Key
Key Exchange	Complex	Simple
Scalability	Not scalable, keys increase exponentially	Scalable
Key Size	small	Big
Implementation Speed	Fast	Slow
Best for	Bulk Data	Small amount of data, key exchange, digital envelopes, digital signatures and digital certificate

hashing

- While encryption is about confidentiality, hashing is about integrity and authentication.
- Hashing algorithms reduce any amount of data to a fixed length value known as the hash value.
- This hash value is a sort of fingerprint of the initial data.
- Due to the algorithms used to create hash values, even small changes in the initial data will create huge changes in the hash value.
- This makes it harder to guess the initial data with a trial-and-error approach.
- Hashing is a one-way algorithm.

hashing

- Since we can have initial data of the desired length, and the output will be of fixed length, there is a possibility that different initial data will have the same hash value. This is called collision.
- A well-designed algorithm should prevent collision; but the only way to create an algorithm that is collision-free would need a hash value longer than the text, making it pointless.

hashing

- It is not possible to extract the original data by knowing the hash value.
- For some hashing algorithms like MD5 that today have a huge number of collisions, you can simply Google a hash value, which will often give you a short string that matches that hash value.
- This does not mean that the original message that created that hash value will be the one you have found on Google.

hashing

- Hashing can be used to check if a file you have received is the same as the one that has been sent.
- It can also be used to check if a password given by the user is the right one to authenticate the user without really knowing the password, since you only stored the algorithm.

MD5

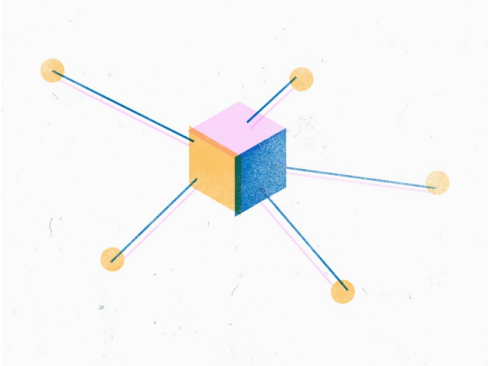
- Message Digest 5 is the most famous MD hashing algorithm and is one of the best-known of all hashing algorithms.
- It produces a 128-bit output(32 characters), processing the initial data in 512-bit blocks.
- Currently, MD5 is deprecated due to its high number of collisions.

SHA

- The Secure Hash Algorithm is another series of algorithms like MD5. The various algorithms that constitute the SHA series are:
  - SHA0 – This is the first version of SHA and was withdrawn shortly after publication since a significant flaw was identified in the hashing algorithm itself.
  - SHA1 – This produces 160-bit output and today it is considered at risk of breaking since it doesn't offer much security.
  - SHA2 – This is set of algorithms that can create outputs of 224-bit, 256-bit, 384 bits and 512 bit. It is considered safe since it can offer fom 112 to 256 bits of security.
  - SHA3 – This set of algorithms can create output like the SHA2 but with entirely different algorithm, which is based on the Keccak algorithm

Centralization vs Decentralization

- Decentralization is an important concept that is not unique to Bitcoin.
- The notion of competing pattern of centralization versus decentralization arises in a variety of different digital technologies
- In order to understand how it plays out in blockchain, it is useful to understand the central conflict between these two.

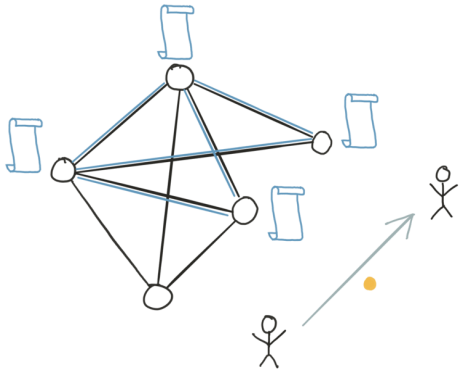


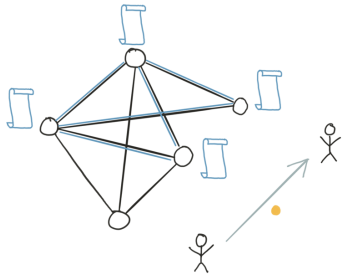
Centralization vs Decentralization

- On the one hand we have the Internet, a famously decentralized system that has historically competed with and prevailed against “walled-garden” alternatives like AOL’s and CompuServe’s information services.
- Then there’s email, which at its core is a decentralized system based on the Simple Mail Transfer Protocol (SMTP), an open standard. While it does have competition from proprietary messaging systems like Facebook or LinkedIn mail, email has managed to remain the default for person-to-person communication online.

Centralization vs Decentralization

- In the case of instant messaging and text messaging, we have a hybrid model that can't be categorically described as centralized or decentralized.
- Finally there's social networking: despite numerous concerted efforts by hobbyists, developers and entrepreneurs to create alternatives to the dominant centralized model, centralized systems like Facebook and LinkedIn still dominate this space.
- In fact, this conflict long predates the digital era and we see a similar struggle between the two models in the history of telephony, radio, television, and film.



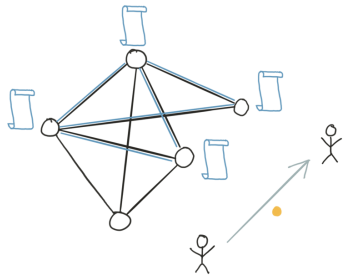


Centralization vs Decentralization

- Decentralization is not all or nothing; almost no system is purely decentralized or purely centralized.
- For example, email is fundamentally a decentralized system based on a standardized protocol, SMTP, and anyone who wishes can operate an email server of their own.
- Yet, what has happened in the market is that a small number of centralized webmail providers have become dominant.
- Similarly, while the Bitcoin protocol is decentralized, services like Bitcoin exchanges, where you can convert Bitcoin into other currencies, and wallet software, or software that allows people to manage their bitcoins may be centralized or decentralized to varying degrees.

Centralization vs Decentralization

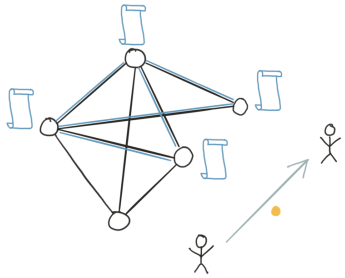
- With this in mind, let's break down the question of how the Bitcoin protocol achieves decentralization into five more specific questions:
- 1. Who maintains the ledger of transactions?
- 2. Who has authority over which transactions are valid?

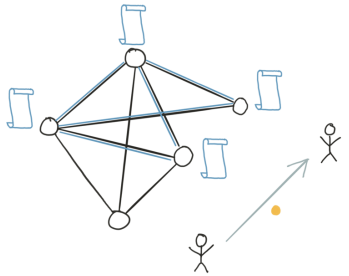


- 3. Who creates new bitcoins?
- 4. Who determines how the rules of the system change?
- 5. How do bitcoins acquire exchange value?

Centralization vs Decentralization

- Different aspects of Bitcoin fall on different points on the centralization/decentralization spectrum.
- The peer-to-peer network is close to purely decentralized since anybody can run a Bitcoin node and There's a fairly low barrier to entry.
- You can go online and easily download a Bitcoin client and run a node on your laptop or your PC. Currently there are several thousand such nodes. Bitcoin mining is technically also open to anyone, but it requires a very high capital cost.





Centralization vs Decentralization

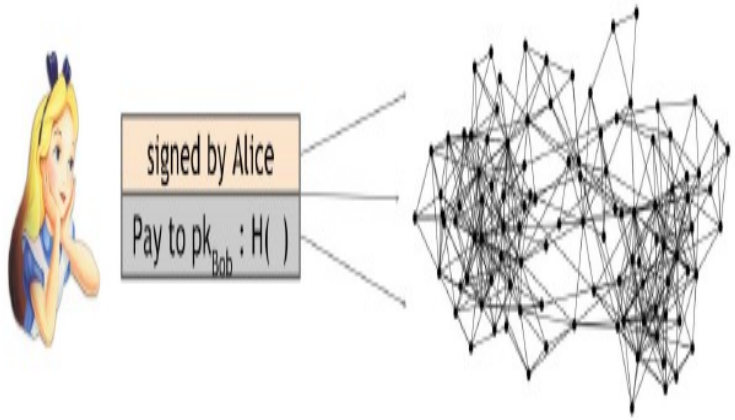
- Because of this there has been a high degree of centralization, or a concentration of power, in the Bitcoin mining ecosystem. Many in the Bitcoin community see this as quite undesirable.
- A third aspect is updates to the software that Bitcoin nodes run, and this has a bearing on how and when the rules of the system change.
- One can imagine that there are numerous interoperable implementations of the protocol, as with email.
- But in practice, most nodes run the reference implementation, and its developers are trusted by the community and have a lot of power

Distributed Consensus

- Distributed consensus has various applications, and it has been studied for decades in computer science.
- The traditional motivating application is reliability in distributed systems
- It can be used to build a massive, distributed key-value store.
- A distributed key-value store in turn would enable many applications.
- For example, we could use it to build a distributive domain name system, which is simply a mapping between human understandable domain names to IP addresses. We could build a public key directory, which is a mapping between email addresses (or some other form of real-world identity) to public keys.

Distributed Consensus

- A distributed consensus protocol has the following two properties:
  - It must terminate with all honest nodes in agreement on the value
  - The value must have been generated by an honest node
- Cryptocurrencies are peer-to-peer system. When Alice wants to pay Bob, what she actually does is broadcast a transaction to all of the bitcoin nodes that comprise the peer-to-peer network.



Distributed Consensus

- Incidentally, you may have noticed that Alice broadcasts the transaction to all the Bitcoin peer-to-peer nodes, but Bob's computer is nowhere in this picture.
- It's of course possible that Bob is running one of the nodes in the peer-to-peer network.
- In fact, if he wants to be notified that this transaction did in fact happen and that he got paid, running a node might be a good idea.
- Nevertheless, there is no requirement that Bob be listening on the network; running a node is not necessary for Bob to receive the funds.
- The bitcoins will be his whether or not he's operating a node on the network.

Distributed Consensus

- the nodes must agree on exactly which transactions were broadcast and the order in which these transactions happened.
- This will result in a single, global ledger for the system.
- So at any given point, all the nodes in the peer-to-peer network have a ledger consisting of a sequence of blocks, each containing a list of transactions, that they've reached consensus on.
- Additionally, each node has a pool of outstanding transactions that it has heard about but have not yet been included on the block chain.
- For these transactions, consensus has not yet happened, and so by definition, each node might have a slightly different version of the outstanding transaction pool

Distributed Consensus

- At regular intervals, say every 10 minutes, every node in the system proposes its own outstanding transaction pool to be the next block.
- Then the nodes execute some consensus protocol, where each node's input is its own proposed block.
- Now, some nodes may be malicious and put invalid transactions into their blocks, but we might assume that other nodes will be honest.
- If the consensus protocol succeeds, a valid block will be selected as the output. Even if the selected block was proposed by only one node, it's a valid output as long as the block is valid.
- Now there may be some valid outstanding transaction that did not get included in the block, but this is not a problem. If some transaction somehow didn't make it into this particular block, it could just wait and get into the next block.

Game Theory

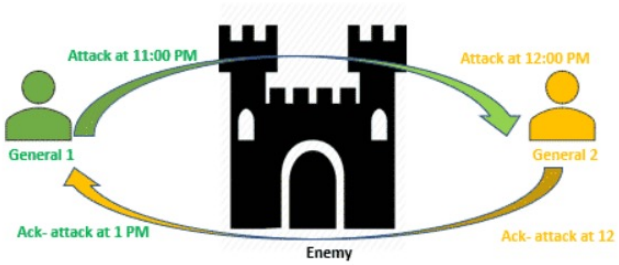
- Game Theory is a study of strategies involved in complex games.
- It is the art of making the best move, or opting for a best strategy in a given situation based on the objective.
- It is the method of modeling real-life situations in the form of a game and analyzing what the best strategy or move of a person or an entity could be in a given situation for a desired outcome.

Examples of Game-Theory

- Nash Equilibrium
- Prisoner’s Dilemma
- Byzantine General’s Problem
- Zero-Sum Games

Two General Problem

- This problem seems to be very simple, but this was unsolved as of today. So, let's understand this in detail.
- This problem states a scenario where two generals are attacking a common enemy, both the generals has its own army and they will be able to defeat the enemy only if they both attack at same time, if any one of them does not attack then they will not be able to win this battle.
- Now the problem here is the communication between two generals, for them to communicate they need to exchange the messages.



Two General Problem

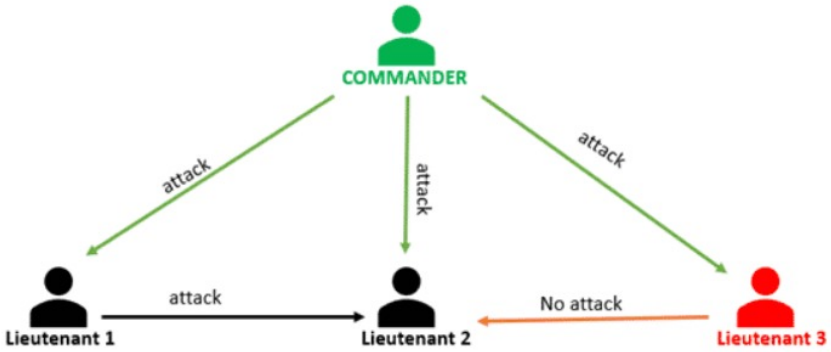
- First general sends a messenger across the enemy camp that need to share the time of the attack to second general, now there may be chance that messenger is captured by enemy army and they distort the message and the correct timing details is not passed to second general as shown in above example.
- Once the information is received by second general then acknowledgement of that need to be send to first general and again that messenger can be captured by army and messenger share some other timing of the attack and this acknowledgement cycle will keep on going. So, this problem seems to be unsolved.

Byzantine Generals Problem

- Byzantine Generals Problem is advance version of “Two general problem” where there can be many generals and they need not to agree only on time of attack but here one or more than one general can be traitor.
- So the question comes how consensus can be reached, answer to that is, consensus is reached when 2/3 of the actors are honest. If the traitors are more than 1/3, consensus is not reached, the armies do not coordinate their attack and the enemy wins.

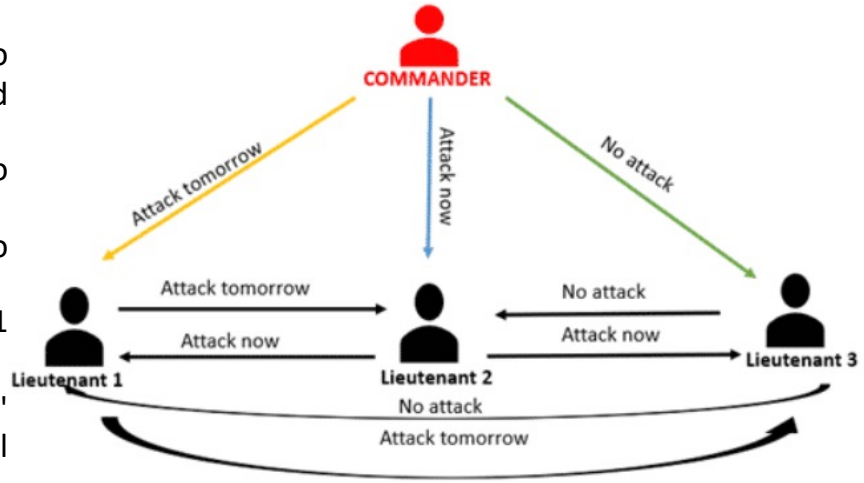
Byzantine Generals Problem

- This should be clear from below diagram where we are seeing Lieutenant 2’s point of view.
- In this scenario, Lieutenant 3 is traitor. As shown in above diagram,
  - Commander sends "attack" command to all Lieutenants
  - Lieutenant-1 sends "attack" command to Lieutenant-2
  - Lieutenant-3 sends "no attack" command to Lieutenant-2
- Now Lieutenant-2 has 2 "attack" command and 1 "no attack" command so majority is "attack" so he will go with "attack" command.



Byzantine Generals Problem

- Let's consider another scenario where commander is traitor.
- In this scenario, commander is traitor. As shown in diagram,
  - Commander sends "attack tomorrow" command to Lieutenant-1, "attack now" command to Lieutenant-2 and "no attack" command to Lieutenant-3
  - Lieutenant-1 sends "attack tomorrow" command to Lieutenant-2 and Lieutenant-3.
  - Lieutenant-2 sends "attack now" command to Lieutenant-1 and Lieutenant-3.
  - Lieutenant-3 sends "no attack" command to Lieutenant-1 and Lieutenant-2.
- Now Lieutenant-1 will have "attack tomorrow", "attack now" and "no attack" command. since there is no majority so he will retreat.
- same apply to other Lieutenants as well.



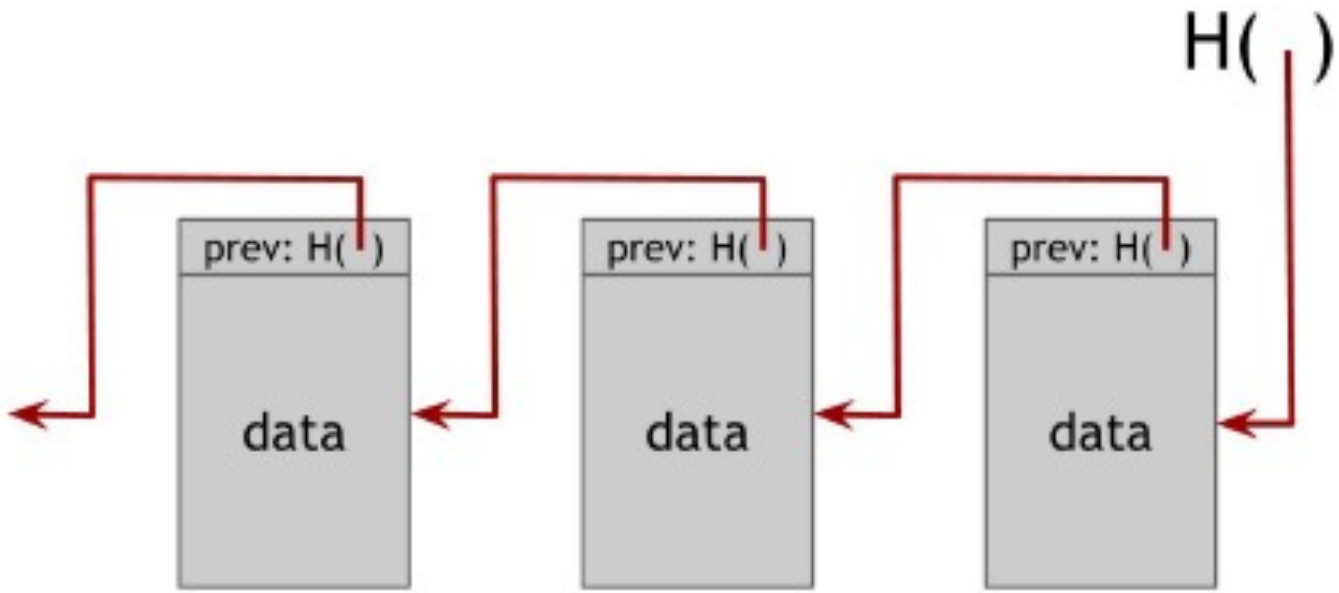
Computer Science Engineering

- It is clever engineering with the concepts from computer science that stitches the components of cryptography, game theory and many others to build a blockchain.
- Main concepts are, Hash Pointers & Merkle Tree.

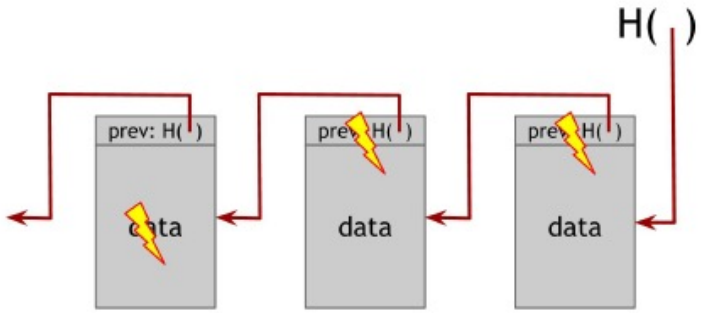
Hash-pointers

- A hash pointer is simply a pointer to where some information is stored along with a cryptographic hash of the information.
- Regular pointer gives a way to retrieve the information, hash pointer also gives you a way to verify that information hasn't changed.
- We can use hash pointers to build all kinds of data structures.
- Intuitively, We can take a familiar data structure that uses pointer such as a binary search tree and implement it with hash pointers.

Blockchain with hash pointers



Blockchain with hash pointers

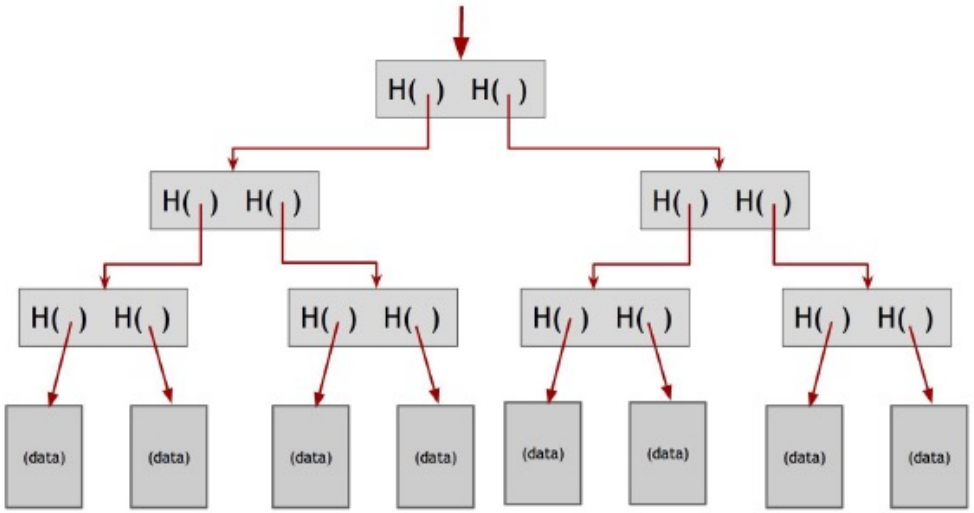


- This provides blockchain a tamper-evident property.
- Let’s discuss what happens if an adversary wants to tamper with data that’s in the middle of the chain.

Merkle Trees

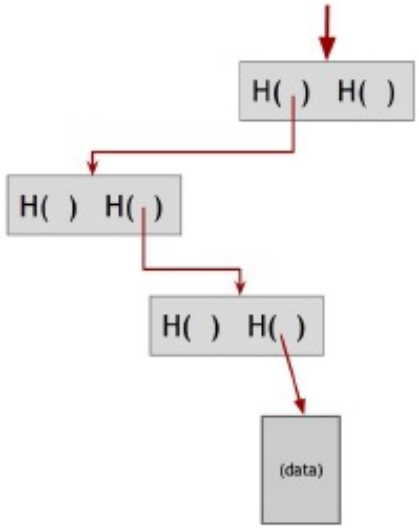
- Another useful data structure that we can build using hash pointer is a binary tree.
- A binary tree with hash pointers is known as Merkle Tree, named after it’s inventor Ralph Merkle.

Merkle Trees



Proof of membership

- Another nice feature of Merkle tree is that, unlike the blockchain that we built before, it allows a concise proof of membership.
- If someone wants to prove that a certain data block is a member of the Merkle Tree, It is completely possible.
- They need to basically show us the data block and the path from that data block to the root. Rest of the tree can be ignored, as the block on this path are enough to allow us to verify the hashes all the way up to the root of tree.



Proof of Non-membership

- The way it is possible to prove membership, similarly Merkle tree also enable us to verify non-membership.
- We can prove that a particular block is not in the merkle tree by showing a path to the item that's just before where the item in question would be and showing the path to the item that is just after where it would be.
- If these two items are consecutive in the tree, then this serves as a proof that the item in question is not included.
- For it to be included, it would need to be between the two items shown, but there is no space between them as they are consecutive.

Blockchain Transaction Process

- Every New Transactions get broadcast to the network so that all the nodes are aware of it at the time it took place.
- Transactions may get validated by the nodes to accept or reject by checking the authenticity.
- The nodes may then group multiple transactions into blocks to share with the other nodes in the network.
- The generation of new block takes place after every node agreeing upon a block, this is called consensus.

Blockchain Transaction Process

- There is no notion of a global time due to network latency, packet drops and geographic locations, but the system still works because the blocks are added one after another in an order.
- All blocks are time stamped in the order they arrive and gets added in the blockchain.
- Once the nodes in the network unanimously accept a block, then that block gets into the blockchain and it includes the hash of the block that was created right before it.

Proof of Work vs Proof of Stake

- These are two widely known and well accepted consensus algorithm available in blockchain.
- Let’s understand them thoroughly.

Proof of Work

Proof of Work

- Originated in 1993 by Cynthia Dwork and Moni Naor
- Discourages DDoS but after 2009, It was adopted for trustless and distributed consensus
- It works in such a way that first miner who solves each block’s problem gets rewarded

How it Works

- Transactions are bundled together into what we call a block
- Miners verify that transactions within each block are legitimate
- To do so, miners should solve a mathematical puzzle
- A reward is given to the first miner who solves each blocks problem
- Verified transactions are stored in the public blockchain

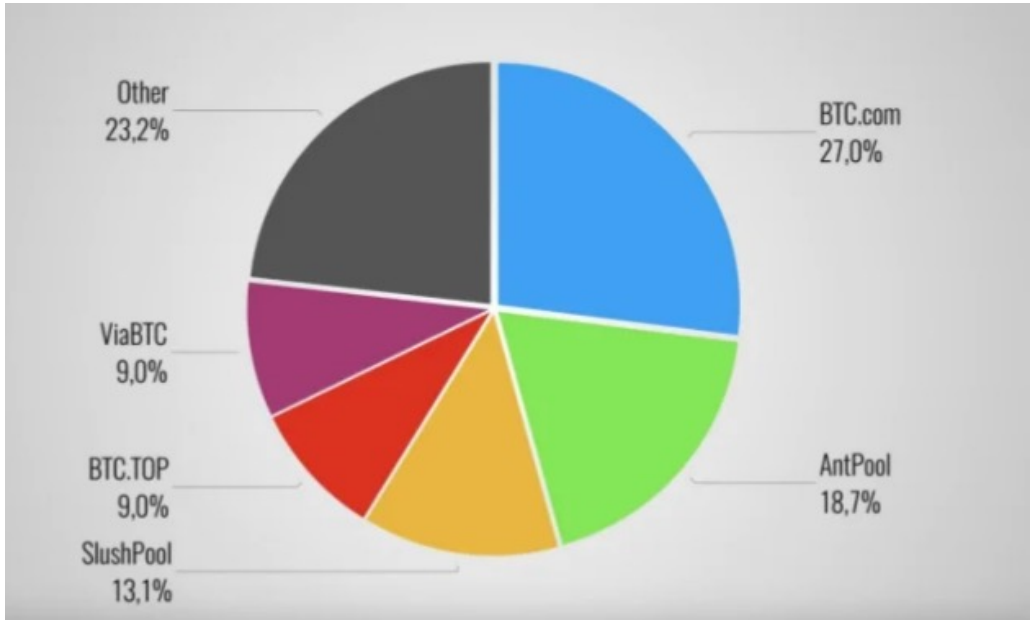
Understanding Mining

- Mining process is an operation of finding the has value
  - Determine a number (nonce)
  - Cryptographic Hash Algorithm of block data results in less than a given threshold

Breaking Point

- Needs computing power i.e. high electricity usage
  - Bitcoin Energy Farms alone consumed 54TWh (~5M US Households or power entire Hungary/Ireland)
- Higher Rewards are given to people with better and more equipment
- Mining Pools make blockchain more centralised than decentralised
  - Miners create a mining pool to combine hashpower and share the profits evenly.

Concern


Other
23,2%
BTC.com
27,0%
ViaBTC
9,0%
BTC.TOP
9,0%
AntPool
18,7%
SlushPool
13,1%

Proof of Stake

Proof of Stake

- Originated in 2011 by Quantum Mechanic (bitcointalk.org)
- It was created with a thought that letting everyone compete with each other for mining is wasteful(PoW)
- Creator of a new block is chosen in a deterministic way, depending on its wealth, also defined as stake.
- There is no particular reward for creation for blocks, rather the validators take the transaction fees.

How it works?

- Transactions are bundled together into what we call a block
- Validators will stake(personal wealth as security deposit) to be randomly selected in a deterministic way
- Validators will mint/forge a new block.
- Validators take the transaction fees inside a block. They lose a part of stake if they verify fraud transactions.
- Verified transactions are stored in the blockchain

Block Selection Variants

- Cryptocurrencies using Proof of Stake often start by selling pre-mined coins or they launch with the Proof of Work algorithm and later switch over to Proof of Stake.
- Where in Proof of Work-based systems more and more cryptocurrency is created as rewards for miners, the Proof-of-Stake system usually uses transaction fees as a reward.

Block Selection Variants

- Users who want to participate in the forging process, are required to lock a certain amount of coins into the network as their stake.
- The size of the stake determines the chances for a node to be selected as the next validator to forge the next block - the bigger the stake, the bigger the chances.
- In order for the process not to favor only the wealthiest nodes in the network, more unique methods are added into the selection process.
- The two most commonly used methods are ‘Randomized Block Selection’ and ‘Coin Age Selection’.

Block Selection Variants

- In the Randomized Block Selection method the validators are selected by looking for nodes with a combination of the lowest hash value and the highest stake and since the size of stakes are public, the next forger can usually be predicted by other nodes.

Block Selection Variants

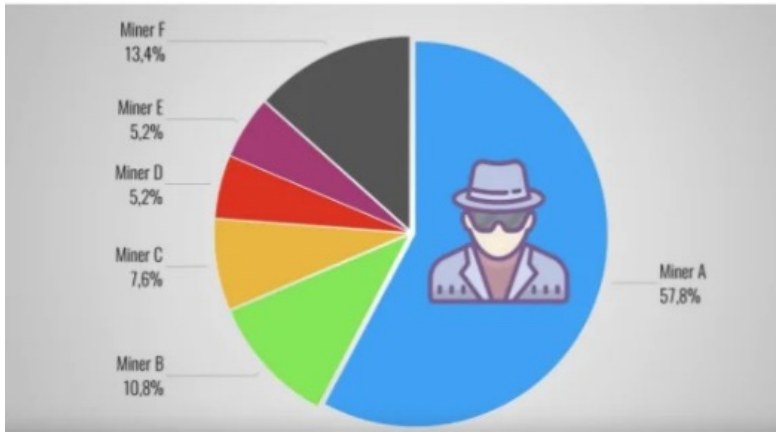
- The Coin Age Selection method chooses nodes based on how long their tokens have been staked for.
- Coin age is calculated by multiplying the number of days the coins have been held as stake by the number of coins that are staked.
- Once a node has forged a block, their coin age is reset to zero and they must wait a certain period of time to be able to forge another block - this prevents large stake nodes from dominating the blockchain.

Additional Points

- Stake vs Chance relation is linear : In PoW, the price of mining equipment reduce with high quantity, where as here it’s always linear. In this way, its more efficient
- Validators lose a portion of stake if they verify fraudulent transaction: As long as the stake is higher from the transaction fees, we can trust that validators to correctly do their job.
- If node stop being a validator: Validator stake plus transaction fees will be released after a certain period of time.

The 51% Attack

Proof of Work (BTC)



Proof of Stake (BTC)

51% × market cap

= $79,826,299,343.76

Who uses it