Introduction to Bitcoin Theory @ 5thwork.com

- Chapter 1: Abstract (Approx 45 mins)
  - 00 Abstract read-through
  - 01 Peer-to-peer cash
  - 02 Digital signatures and trusted third parties
  - 03 Abstract Assessment No.1
  - 04 Peer to Peer network
  - 05 Time Chain and Proof of Work
  - 06 CPU Power
  - 07 Abstract Assessment No.2
  - 08 Cooperation in the network
  - 09 Network structure
  - 10 Messaging between nodes
  - 11 Abstract Video
  - 12 Abstract Assessment No.3

- Chapter 2: Introduction (Approx 45 mins)
  - 00 Introduction read-through
  - 01 Commerce on the internet
  - 02 Non reversible transactions
  - 03 Introduction Assessment No.1
  - 04 Privacy in commerce
  - 05 The paradigm of fraud acceptance
  - 06 What is needed…
  - 07 Introduction Assessment No.2
  - 08 Protecting sellers from fraud
  - 09 Proposed solution
  - 10 Security and honesty
  - 11 Introduction Video
  - 12 Introduction Assessment No.3

- Chapter 3: Transactions (Approx 45 mins)
  - 00 Section read-through
  - 01 Electronic Coins
  - 02 Spending a coin
  - 03 Transactions Assessment No.1
  - 04 Payee verification
  - 05 Existing solutions
  - 06 First Seen Rule
  - 07 Transactions Assessment No.2
  - 08 Broadcasting Transactions
  - 09 Achieving Consensus
  - 10 Proof of acceptance
  - 11 Transactions Video
  - 12 Transactions Assessment No.3

● Chapter 4: Timestamp Server (Approx 15 mins)
○ 00 Section read-through
○ 01 Timestamped Hashes
○ 02 A chain of timestamped hashes
○ 03 Timestamp Server Video
○ 04 Timestamp Server Assessment No.1

● Chapter 5: Proof of Work (Approx 60 mins)
○ 00 Section read-through
○ 01 Hashcash
○ 02 Scanning random space
○ 03 Proof of Work Assessment No.1
○ 04 Nonce
○ 05 Immutable Work
○ 06 Chained effort
○ 07 Proof of Work Assessment No.2
○ 08 One CPU, one vote
○ 09 The majority decision
○ 10 The honest chain
○ 11 Proof of Work Assessment No.3
○ 12 Attacking the longest chain
○ 13 Controlling the block discovery rate
○ 14 Proof of Work Video
○ 15 Proof of Work Assessment No.4

● Chapter 6: Network (Approx 45 mins)
○ 00 Section read-through
○ 01 Running the Network
○ 02 Network Assessment No.1
○ 03 The longest chain
○ 04 Simultaneous blocks
○ 05 Network Assessment No.2○ 06 Breaking the tie
○ 07 Missed messages
○ 08 Network Video
○ 09 Network Assessment No.3

● Chapter 7: Incentive (Approx 45 mins)
○ 00 Section read-through
○ 01 The Coinbase Transaction
○ 02 Coin Distribution
○ 03 Incentive Assessment No.1
○ 04 Mining analogy
○ 05 Transaction fees
○ 06 The end of inflation
○ 07 Incentive Assessment No.2
○ 08 Encouraging honesty
○ 09 The attacker’s dilemma
○ 10 Incentive Video
○ 11 Incentive Assessment No.3

● Chapter 8: Reclaiming Disk Space (Approx 30 mins)
○ 00 Section read-through
○ 01 Spent transactions
○ 02 The Merkle Tree
○ 03 Reclaiming Disk Space Assessment No.1
○ 04 Compacting blocks
○ 05 Block Headers
○ 06 Reclaiming Disk Space Video
○ 07 Reclaiming Disk Space Assessment No.2

● Chapter 9: Simplified Payment Verification (Approx 45 mins)
○ 00 Section read-through
○ 01 Full network nodes○ 02 Merkle Branches
○ 03 Simplified Payment Verification Assessment No.1
○ 04 Transaction acceptance
○ 05 Verification during attack situations
○ 06 Maintaining an attack
○ 07 Simplified Payment Verification Assessment No.2
○ 08 Invalid Block Relay System
○ 09 Businesses running nodes
○ 10 Simplified Payment Verification Video
○ 11 Simplified Payment Verification Assessment No.3

● Chapter 10: Combining and Splitting Value (Approx 30 mins)
○ 00 Section read-through
○ 01 Dynamically sized coins
○ 02 Inputs and Outputs
○ 03 Combining and Splitting Value Assessment No.1
○ 04 A typical example
○ 05 Fan-out
○ 06 Combining and Splitting Value Video
○ 07 Combining and Splitting Value Assessment No.2

● Chapter 11: Privacy (Approx 45 mins)
○ 00 Section read-through
○ 01 Traditional Models
○ 02 Privacy in Bitcoin
○ 03 Privacy Assessment No.1
○ 04 Public records
○ 05 Stock Exchange Comparison
○ 06 Key Re-use
○ 07 Privacy Assessment No.2
○ 08 Linking inputs
○ 09 Linking the owner○ 10 Privacy Video
○ 11 Privacy Assessment No.3

● Chapter 12: Calculations (Approx 60 mins)
○ 00 Section read-through
○ 01 Attacking the chain

○ 02 Things the attacker cannot achieve…
○ 03 The only thing the attacker can achieve…
○ 04 The Binomial Random Walk
○ 05 The Gambler’s Ruin
○ 06 Exponential odds
○ 07 Waiting for confirmation…
○ 08 Attack via proof of work
○ 09 Vanishing probabilities
○ 10 Calculations Video
○ 11 Calculations Assessment No.1

● Chapter 13: Conclusion (Approx 15 mins)
○ 00 Section read-through
○ 01 Conclusion
○ 02 Conclusion Video

Chapter 1: Abstract (Approx 45 mins)

00 Abstract read-through

- A purely peer-to-peer version of electronic cash would allow online payments to be sent directly from one party to another without going through a financial institution.
- Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending. We propose a solution to the double-spending problem using a peer-to-peer network.
- The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work.
- The longest chain not only serves as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power.
- As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they’ll generate the longest chain and outpace attackers.
- The network itself requires minimal structure. Messages are broadcast on a best effort basis, and nodes can leave and rejoin the network at will, accepting the longest proof-of-work chain as proof of what happened while they were gone.

01 Peer-to-peer cash

- Unlike in the legacy banking model, where for two parties to transact, both parties must employ the services of a trusted third party (e.g. a bank), in Bitcoin, money is exchanged peer-to-peer using the Bitcoin protocol.
- Transactions can involve effectively unlimited numbers of peers thanks to the flexibility of the protocol which is limited only by the economics of constructing and verifying each transaction rather than arbitrary parameters.

- Wallets allow users to create transactions that sign ownership records of digital coins and assign them to new owners. The records of these exchanges make up the history of Bitcoin transactions, sometimes referred to as a ledger.

- Transactions are recorded on a public ledger visible to all parties without the need for a financial institution's involvement. This does not mean that banks will not use Bitcoin or be a part of the Bitcoin ecosystem. In fact, banks that use Bitcoin will have the advantages of its low cost transaction verification as a competing advantage in financial markets.

> What is needed is an electronic payment system based on cryptographic proof instead of trust, allowing any two willing parties to transact directly with each other without the need for a trusted third party.

> The solution to the double-spending problem requires using a peer-to-peer distributed timestamp server to generate computational proof of the chronological order of transactions.

> The system is secure as long as honest nodes collectively control more CPU power than any cooperating group of attacker nodes.

02 Digital signatures and trusted third parties

> Digital signatures provide part of the solution, but the main benefits are lost if a trusted third party is still required to prevent double-spending. - Satoshi Nakamoto, Bitcoin Whitepaper

- Digital signatures are a means for the owner of a coin on the ledger to establish their intent to use that coin in a transaction.
- Digital signatures work by generating a unique hash of the message and signing it using the sender's private key.
- The hash generated is unique to the message, and changing any part of the message will completely change the hash thus rendering the digital signature invalid.
- a digital signature is a technique that binds a person to digital data. This can be independently verified by the receiver.
- One of Bitcoin's core innovations is that it allows for digital signatures to be validated without needing a third party who has knowledge of the identity of the transacting parties.
- When the user builds a transaction, a highly flexible scripting language is used to define the conditions under which that coin can be spent. Incorporating signatures into these conditions using Elliptic Curve Digital Signature Algorithm (ECDSA), the signature type used in Bitcoin, provides a means for the user or users to provably show they control a private key. This key is linked to a script based puzzle which the user provides when they receive the coin. To spend it, the user must provide the right script incorporating the necessary proofs to correctly solve the script. It is upon the peers participating in the transaction to determine its content, create the signatures and submit it to the network for validation.
- While the transaction processors who log the transaction have no stake in the value of the Bitcoin being exchanged, they are paid a transaction fee for the service of validating and recording the transaction. Because they are not a party to the transaction, they become a simple third party, to whom no trust needs to be given.
- Transaction processors are not required to record all transactions onto the ledger, but can choose to do so if the users attach a sufficient fee for them to consider it worthwhile and so long as the transaction is actually valid.

03 Abstract Assessment No.1

04 Peer to Peer network

- We propose a solution to the double-spending problem using a peer-to-peer network. - Satoshi Nakamoto, Bitcoin Whitepaper
- Double spending refers to the act of signing a coin to create a transaction and submitting that transaction to the network before using the same coin to create a different transaction which pays to a different recipient, effectively spending the same coin twice.
- This problem is solved in Bitcoin through the creation of a peer-to-peer network of nodes whose role it is to gather, validate and timestamp all of the transactions that take place. It is through this network that the double spending problem is addressed by accepting only the first-seen of such a pair of transactions.
- The Bitcoin network is a global piece of infrastructure that is built by enterprises who compete for the right to extend the ledger by adding new transactions. Each transaction can only be processed once and inputs used in a transaction are consumed. Once a transaction has been submitted to the network, it is broadcast to all nodes within a few seconds making it almost impossible to perform double spends without the assistance of a fraudulent node.

05 Time Chain and Proof of Work

- The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work. - Satoshi Nakamoto, Bitcoin Whitepaper
- The word Time chain can be used to refer to the nature of the Bitcoin block chain as a chain of time stamped events in history. As transactions are received into the network, nodes capture and collate them into logs. These logs, or ‘blocks’, are made up of a timestamp applied to a sequential list of transactions and represent a consensus agreement of the proof of both existence and validity of all the transactions they contain.
- Proof of Work is the term used when explaining the rules that decide who gets to update transactions on the Bitcoin blockchain. Put simply, in order to gain the right to update the next block of transactions, you need to provide proof that you have solved a computational challenge that is hard yet can be easily verified by the network. By doing this you provide proof that you have done the work to solve it.
- Think of this like starting a jigsaw puzzle, it’s hard to solve and you will make many attempts to fit the pieces but once you complete the puzzle it is very easy for it to be validated.
- As new transactions are received, nodes add them into a block template which contains all the transactions they have accepted which have not been put into a valid block, and perform hash based work on a difficulty puzzle that must be solved to form a valid block. The solution represents proof that the node proposing the block has performed the work necessary for that block to be valid.
- Hashing means taking an input of data of any length and transforming it in such a way that it produces a repeatable but essentially random output of a fixed length. In Bitcoin, the transactions are run through a hashing algorithm called SHA-256 which gives an output of a fixed length of 256-bits.
- In this way, anything from a short message to a large file can be hashed and the hash distributed to several parties. At any time, those parties can verify the data block by hashing it and checking that it matches the hash output they received earlier. Only the original data can be used to generate that same hash.
- In Bitcoin, the nodes compete by generating as many hashes as needed to find one with the right properties. In this case a fixed length string of 64 hexadecimal characters less than a particular amount, looking something like this: 000000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f

- Hashes with this many leading zeros are not easy to find and represent an expenditure of energy on CPU power. Nodes compete for the right to create the next timestamp, or block, in the chain which is granted by solving this hash puzzle.
- When a Bitcoin block template is being hashed, the block header contains

1. a time stamp,
2. a reference to the block it builds upon,
3. a hash that represents all of the transactions in the block, a difficulty setting and a field called ‘Nonce’ or ‘number used once’. This Nonce is changed rapidly to generate new messages for the hashes being created during the proof of work process.

- As the network expands, the puzzle’s difficulty is adjusted to keep the average block time as close to 10 minutes as possible. If nodes add their CPU power to the pool performing proof of work, the puzzle becomes increasingly hard to solve. Over time this means that changing blocks which have had several subsequent blocks built on top of becomes almost impossible through the accumulation of proof of work on top of them.

06 CPU Power

- The longest chain not only serves as proof of the sequence of events witnessed, but proof that it came from the largest pool of CPU power. - Satoshi Nakamoto, Bitcoin Whitepaper
- The longest chain of Proof-of-Work is the chain of blocks generated by the largest pool of CPU power.
- At scale, this means far more than just the capacity of nodes to solve the difficult proof of work problem, as a node must be able to keep up with the rest of the network in the tasks of downloading, validating and storing all of the transactions taking place in the world in real time.
- The hashing machinery used to solve proof of work votes for the most capable node, incentivising node operators to build the fastest most capable machines in order to attract the most hash. Hash power can be switched instantaneously off or on, or even moved between nodes so ensuring it is being used profitably is a top priority.
- In this way, the network stays healthy by incentivising node operators to invest in the best and most capable machinery to run the network.

07 Abstract Assessment No.2

08 Cooperation in the network

- As long as a majority of CPU power is controlled by nodes that are not cooperating to attack the network, they’ll generate the longest chain and outpace attackers. - Satoshi Nakamoto, Bitcoin Whitepaper
- One of the ways it is possible to attack the integrity of the ledger is for a node operator to include in a block a transaction that double spends coins that were previously spent, or to otherwise include a transaction in a block that is invalid per network rules. Honest nodes will recognise this rule violation and refuse to build new blocks that follow this invalid block. Instead they will build new blocks on top of it’s valid predecessor and create an honest competing chain. So long as the honest node controls more than half of the hash power the honest chain will become longer and signal to all network participants to ignore the invalid block.
- As the attacker’s block is no longer recognised by the rest of the network the reward for creating the invalid blocks is also not recognised. As such the attempt becomes a significant cost to the attacking node, discouraging attempts at dishonest behaviour by making it very risky.
- In this way, nodes use hashpower to enforce network rules. A node who publishes blocks that violate the rules waste the energy investment used to generate the valid proof of work, and will

lose the pool of hash which voted for it. In this way, proof of work incentivises honest behaviour, creating a system where all nodes compete in a cooperative way to enforce the established network rules.

09 Network structure

- The network itself requires minimal structure. - Satoshi Nakamoto, Bitcoin Whitepaper
- The network forms naturally through the incentive structure that drives participation in usage of the ledger, and the corporate activity that revolves around processing transactions from the ledger into blocks.
- Winning blocks can be a lucrative business activity for an efficient processor
- There is no central governance, and nodes simply need to adhere to the established rules in order to participate. As transaction fee revenue grows commensurate with network usage, individual node operators will invest in infrastructure that increases their capacity to validate all available transactions and gain access to a larger pool of fee revenue. This is not something that is governed by the network or any central party but rather driven by incentive to maximise profits by becoming more competitive with other nodes on the network.

10 Messaging between nodes

- Messages are broadcast on a best effort basis, and nodes can leave and rejoin the network at will, accepting the longest proof-of-work chain as proof of what happened while they were gone. - Satoshi Nakamoto, Bitcoin Whitepaper
- Messages that are broadcasted on the network are limited to new transactions and new block discovery announcements. - When a node receives a new transaction, it automatically broadcasts it to all the nodes with which it has a peer connection. By ensuring all other nodes have the transaction, the node reduces the time those nodes will need to validate a block found which includes this transaction, giving them the best chance of their block being validated quickly.
- Nodes can leave and re-join the network at any time and there is no central governance required for this to occur. When a node re-joins the network, they connect to other nodes and request the records of all transactions and blocks that have been seen on the network since they were disconnected. The nodes then validate the information and rejoin the competition from the most recently discovered block.

11 Abstract Video

12 Abstract Assessment No.3

Chapter 2: Introduction (Approx 45 mins)

00 Introduction read-through

Commerce on the Internet has come to rely almost exclusively on financial institutions serving as trusted third parties to process electronic payments. While the system works well enough for most transactions, it still suffers from the inherent weaknesses of the trust based model. Completely non-reversible transactions are not really possible, since financial institutions cannot avoid mediating disputes. The cost of mediation increases transaction costs, limiting the minimum practical transaction size and cutting off the possibility for small casual transactions, and there is a broader cost in the loss of ability to make non-reversible payments for nonreversible services. With the possibility of reversal, the need for trust spreads. Merchants must be wary of their customers, hassling them for more information than they would otherwise need. A certain percentage of fraud is accepted as unavoidable. These costs and payment uncertainties can be avoided in person by using physical currency, but no mechanism exists to make payments over a communications channel without a trusted party. What is needed is an electronic payment system based on cryptographic proof instead of trust, allowing any two willing parties to transact directly with each other without the need for a trusted third party. Transactions that are computationally impractical to reverse would protect sellers from fraud, and routine escrow mechanisms could easily be implemented to protect buyers. In this paper, we propose a solution to the double-spending problem using a peer-to-peer distributed timestamp server to generate computational proof of the chronological order of transactions. The system is secure as long as honest nodes collectively control more CPU power than any cooperating group of attacker nodes.

01 Commerce on the internet

02 Non reversible transactions

- Completely non-reversible transactions are not really possible, since financial institutions cannot avoid mediating disputes. The cost of mediation increases transaction costs, limiting the minimum practical transaction size and cutting off the possibility for small casual transactions, and there is a broader cost in the loss of ability to make non-reversible payments for non-reversible services. - Satoshi Nakamoto, Bitcoin Whitepaper
- Because financial institutions are required to be part of each transaction that uses legacy payment systems, when there is mediation they are forced to intervene. This is a time consuming and costly process for merchants and represents a drain on commerce, where depending on industry, as many as 3% of all credit card transactions are contested, costing the merchants fees, the cost of the goods charged back, and time, stress and effort.
- In turn, these costs are passed onto the consumer who pay an invisible margin on top of all goods and services to cover not just the cost of their transaction, but the cost of mediating the transactions that malicious actors make using stolen cards or through fraudulent back charging. In these systems there is no ability for a merchant selling a non-returnable good or service to receive a non-reversible payment for that good or service which is a missing link in the chain of commerce.

03 Introduction Assessment No.1

04 Privacy in commerce

- With the possibility of reversal, the need for trust spreads. Merchants must be wary of their customers, hassling them for more information than they would otherwise need. - Satoshi Nakamoto, Bitcoin Whitepaper
- Due to the need to track customers to prosecute fraudulent behaviour, merchants who use legacy payment systems are forced to request details from customers that don’t relate to the nature of the commerce, and serve no purpose other than to back-stop the merchant’s liability in the case of fraudulent actions. Despite this, the cost of prosecuting fraudsters who abuse the payment system is prohibitive for smaller transactions, and merchants only recourse is to keep records of bad actors so that they can decline their business in future.
- This represents a huge problem for the privacy of good actors within the system, as their identity details often end up being stored in large merchant databases with their corresponding payment details. Merchants do not often spend the time or money needed to adequately secure this

information and breaches in customer privacy have created situations where thousands or millions of customer details have been leaked onto the dark web, causing financial and identity theft all over the world.

05 The paradigm of fraud acceptance

- A certain percentage of fraud is accepted as unavoidable. These costs and payment uncertainties can be avoided in person by using physical currency, but no mechanism exists to make payments over a communications channel without a trusted party. - Satoshi Nakamoto, Bitcoin Whitepaper
- Due to the way in which payment services by trusted third parties operate, for merchants operating both on-line and in physical locations it is almost impossible for them to avoid coming into contact with bad actors who use fraudulent practices to obtain goods without paying. Commonly this is done through the acquisition of compromised credit card numbers and details, or by the true owner of the credit card reversing the charges made by the merchant through their own financial institution.
- These problems can be mostly avoided by accepting physical currency however this is becoming less and less desirable to both consumers and merchants due to the overheads involved for both parties in handling banknotes and coins.
- There are no methods of transacting electronically available to users of legacy money systems that do not require the use of trusted third parties in the transaction.

06 What is needed…

- What is needed is an electronic payment system based on cryptographic proof instead of trust, allowing any two willing parties to transact directly with each other without the need for a trusted third party. - Satoshi Nakamoto, Bitcoin Whitepaper
- Due to the reasons outlined in the previous section, what is needed is a system that uses cryptographic knowledge proofs to allow the purchasing party (customer) to establish a firm basis of custody over the money being used in a transaction. Bitcoin achieves this by using digital signatures and a simple but fully featured scripting language.
- Bitcoin signatures are simple for the receiving party to validate and can be stored on the Bitcoin public ledger with efficiency and very low overheads. Because the sending party can establish control over the tokens themselves without using a third party to hold funds and manage the transfer, transactions are very fast and simple.
- By using Bitcoin, receivers can quickly and simply validate that funds were indeed controlled by the sending party, and that the transaction correctly allocates the correct amount to their own control without requiring additional validation by third parties.

07 Introduction Assessment No.2

08 Protecting sellers from fraud

- Transactions that are computationally impractical to reverse would protect sellers from fraud, and routine escrow mechanisms could easily be implemented to protect buyers. - Satoshi Nakamoto, Bitcoin Whitepaper
- Thanks to the practically irreversible nature of the Elliptic Curve Digital Signature Algorithm (ECDSA) sellers who receive payments over the Bitcoin network can simply and quickly verify the authenticity of the funds received without needing to revert to a trusted third party.
- Where the payment involves the delivery of goods or services, payments can be locked using simple scripting functions that require proof of delivery prior to the release of funds, with simple

conditional clauses allowing for funds to be returned to the payer in the event of non delivery. These features can be implemented using features native to the Bitcoin protocol drastically reducing the incentive or possibility to commit fraud.

09 Proposed solution

- In this paper, we propose a solution to the double-spending problem using a peer-to-peer distributed timestamp server to generate computational proof of the chronological order of transactions. - Satoshi Nakamoto, Bitcoin Whitepaper
- One of the core innovations of the Bitcoin system is the way in which it prevents users from taking digital coins which have been used in a transaction and double spending them to a different party as a means to commit fraud by snatching the allocated funds from the Merchant's wallet and re-allocating them back to their own wallet. Transactions are recorded as plaintext on the ledger and are readable by all parties.
- As transactions are created, network nodes assemble them into block templates against which they perform proof of work computations. When a valid proof of work solution is found, the block becomes a proof of existence timestamp for all of the transactions it includes whilst establishing which, of any pair of conflicting (double spending) transactions, is accepted as first- seen and valid. Nodes append transactions to a block template in an order that closely matches the chronological order in which they were receive.
- This means each valid block represents a consensus driven agreement on the order in which events were recorded by the network. Blocks are added in chronological order and as more work is added to the chain of blocks, this serves as proof that transactions in a given block were validated and accepted by the network participants collectively prior to the time indicated in the block header.

10 Security and honesty

- The system is secure as long as honest nodes collectively control more CPU power than any cooperating group of attacker nodes. - Satoshi Nakamoto, Bitcoin Whitepaper
- As long as there is a pool of nodes who are competing to collect and append new transactions to the longest chain of proof of work, the system's security is maintained. This system works against fraudulent actors both in the payments system, and at the node level by allowing honest systems to reject blocks which include transactions that double spend inputs they have already seen used in validated transactions or which violate the established rules of the network.
- This enforcement is achieved through accumulation of work by honest nodes within the system and creates a situation where attackers must overpower the network for an indefinite amount of time in order to maintain a chain of work that includes the fraudulent activity. In this way, the hashpower that performs the work on blocks acts as an enforcement system, allowing the honest actors within the network to collectively expend enough energy to outpace the attacking systems over time.
- Time in this scenario is open ended and attacking chains can emerge which retain an appearance of legitimacy and viability for extended periods of time. Thankfully, due to the high cost of performing proof of work, the dishonest nodes are forced to spend large sums of money to maintain the fraud. This expenditure is financially nonviable to maintain, eventually leading to the re-emergence of the honest chain as the legitimate record of activity on the ledger.

11 Introduction Video

12 Introduction Assessment No.3

Chapter 3: Transactions (Approx 45 mins)

00 Section read-through

We define an electronic coin as a chain of digital signatures. Each owner transfers the coin to the next by digitally signing a hash of the previousntransaction and the public key of the next owner and adding these to the end of the coin. A payee can verify the signatures to verify the chain of ownership.

- The problem of course is the payee can't verify that one of the owners did not double-spend the coin. A common solution is to introduce a trusted central authority, or mint, that checks every transaction for double spending. After each transaction, the coin must be returned to the mint to issue a new coin, and only coins issued directly from the mint are trusted not to be double-spent. The problem with this solution is that the fate of the entire money system depends on the company running the mint, with every transaction having to go through them, just like a bank.
- We need a way for the payee to know that the previous owners did not sign any earlier transactions. For our purposes, the earliest transaction is the one that counts, so we don't care about later attempts to double-spend. The only way to confirm the absence of a transaction is to be aware of all transactions. In the mint based model, the mint was aware of all transactions and decided which arrived first. To accomplish this without a trusted party, transactions must be publicly announced [1], and we need a system for participants to agree on a single history of the order in which they were received. The payee needs proof that at the time of each transaction, the majority of nodes agreed it was the first received.

01 Electronic Coins

- We define an electronic coin as a chain of digital signatures. - Satoshi Nakamoto, Bitcoin Whitepaper
- In Bitcoin, transactions are built by referencing unspent digital coins called 'Unspent Transaction Outputs' or "UTXOs" in transaction inputs. To be spent, the input must also include a solution to the script puzzle contained in the output which locks the satoshi tokens it holds. In almost all cases one part of this script puzzle requires a valid digital signature to be provided by the party that holds custody of the UTXO.
- The tokens (coins) are spent into the transaction and redistributed into new unspent transaction outputs, consuming the UTXOs reference by the inputs in the process. In this way, a coin's history can be mapped as a chain of valid script solutions or digital signatures. This chain leads all the way back to the blocks in which each of the Satoshi tokens being used in the transaction was first made accessible in the form of a coinbase reward.
- This chain forms a Directed Acyclic Graph or DAG. The combined DAGs that represent the history of all UTXOs that currently exist on the network is what we refer to as the Bitcoin ledger.

02 Spending a coin

- Each owner transfers the coin to the next by digitally signing a hash of the previous transaction and the public key of the next owner and adding these to the end of the coin. - Satoshi Nakamoto, Bitcoin Whitepaper
- Each UTXO is locked with a script commonly called a scriptPubKey. This scriptPubKey is a puzzle that defines a predicate. The spending party must provide a solution which allows the predicate to be solved with a 'True' or non-zero outcome.
- When spending a coin, the owner (or their proxy) first constructs a message that includes:

1. The identity of the coin or coins on the ledger (transaction ID and output index)

2. Valid solutions to each of the locking scripts written onto the ledger when the coin or coins were created
3. A new locking script for each of the outputs being generated in the transaction which tests a knowledge proof provided by the receiver to the spending party
4. Other details including a locktime parameter and protocol version.

- Depending on the spender's preference, some or all of this message is combined, and the resulting string hashed and used to generate an ECDSA signature using a keypair controlled by the spending party and specified by scriptPubKey contained in the input being spent. Once the transaction is complete, it is ready to be sent onto the network to be processed into a block.

03 Transactions Assessment No.1

04 Payee verification

- A payee can verify the signatures to verify the chain of ownership. - Satoshi Nakamoto, Bitcoin Whitepaper
- Each iteration of a coin contains a solution to the previously defined predicate. Typically, this includes a public key and a corresponding signature created using Elliptic Curve Digital Signature Algorithm (ECDSA) making it computationally trivial to validate the proof of ownership supplied with the given coin.
- This is a crucial element of the peer to peer nature of the system as it means that a merchant can accept payments directly from users and validate their ownership claim without the need of a trusted third party using only information that can be validated against the network's proof of work process.
- This makes it possible for merchants to broadcast transactions to the network on behalf of users whose wallets can be used as very simplified applications which hold and sign digital coins but do not have an interface to the network. This also makes it possible for Bitcoin to be used with devices as simple as a smart card

05 Existing solutions

- The problem of course is the payee can't verify that one of the owners did not double-spend the coin. A common solution is to introduce a trusted central authority, or mint, that checks every transaction for double spending. After each transaction, the coin must be returned to the mint to issue a new coin, and only coins issued directly from the mint are trusted not to be double-spent. The problem with this solution is that the fate of the entire money system depends on the company running the mint, with every transaction having to go through them, just like a bank. - Satoshi Nakamoto, Bitcoin Whitepaper
- When payments are made, the simple validation of the signature does not provide a means for the merchant to check that the coins being handed over have not been spent in a separate transaction which would invalidate this transaction. This is a recurring problem in digital currencies and prior to the introduction of Bitcoin a typical solution was for each transaction to be routed back to a central arbiter who is responsible for checking the validity of the transaction using a closed system.
- The inherent weakness of this style of system is that the centralised validation system is managed by a single company or entity which operates like a central clearing house through which all transactions must be routed. This creates problems around transparency and scalability as unlike Bitcoin where a competitive system of incentives drives the network to scale organically, the system is reliant on a single entity or team to manage the growth of the central system and for whom competitive pressures do not exist to keep transaction fees as low as economically feasible.

06 First Seen Rule

- We need a way for the payee to know that the previous owners did not sign any earlier transactions. For our purposes, the earliest transaction is the one that counts, so we don't care about later attempts to double-spend. - Satoshi Nakamoto, Bitcoin Whitepaper
- Bitcoin is a solution that solves the double spending problem without the need for a centralised decision making system through the use of the 'First Seen Rule'. This rule states that the first seen transaction that validly spends a given UTXO is the transaction which is accepted by the node.
- In a case where two conflicting transactions are introduced simultaneously onto the network which each seek to spend the same input coins to different output destinations, the one which reaches the network nodes which have been allocated the most hashpower stands to be the version of the transaction which is written into a block. The other transaction will never be accepted into a block and as such can never become a permanent part of the Ledger.
- Additional protections can be implemented such as 'Double Spend Notifications' which detect and announce double spend attempts to peers on the network including merchants handling the double spends. Because a double spend must occur within a very short window (less than 5 seconds) it is very simple for online and physical merchants to reject transactions that are double spent or even attempted.

07 Transactions Assessment No.2

08 Broadcasting Transactions

- The only way to confirm the absence of a transaction is to be aware of all transactions. - Satoshi Nakamoto, Bitcoin Whitepaper
- For a node that is seeking to build blocks and extend the ledger, the only way to be certain that each transaction they have received is the only one trying to spend a given UTXO or set of UTXOs is to be aware of all of the transactions taking place on the network in as close to real time as possible.
- If a node tries to create a block using a transaction that other nodes on the network recognise as a double spend, there is a very strong chance that they will lose their money, effectively wasting the cost of constructing their block and performing the proof of work.
- This creates an incentive for nodes to ensure that all other nodes are aware of the transactions they are working on incorporating into a block, as without this awareness, the chance of using a double spent input in a block is higher.
- This awareness has an added benefit in that it allows each node to ensure that all other nodes have every transaction they are working on, minimising the time it will take to validate any new blocks that are found.

09 Achieving Consensus

- In the mint-based model, the mint was aware of all transactions and decided which arrived first. To accomplish this without a trusted party, transactions must be publicly announced [1], and we need a system for participants to agree on a single history of the order in which they were received. - Satoshi Nakamoto, Bitcoin Whitepaper
- When the system is managed by a centralised party, the monitoring of which inputs are used in a transaction and their spent/unspent status is managed simply on a first seen basis.
- In Bitcoin, the system has no such means for centralised decision making so each node must operate as a self-contained mint.

- Public announcement of all transactions is critical to the sound operation of the system and provides the fastest means for participants in the competitive block building process to come to consensus on the order of events when a valid block is announced to the network. Without public announcement, every time a potential block is found, the discovering node must subsequently supply any missing transactions to the rest of the network which consumes time and resources, impacting its ability to operate as effectively and efficiently as possible.
- The first seen rule and its application to both transaction and block announcements is a core element of the incentives that drive the construction of the infrastructure that underpins the Bitcoin network.

10 Proof of acceptance

- The payee needs proof that at the time of each transaction, the majority of nodes agreed it was the first received. - Satoshi Nakamoto, Bitcoin Whitepaper
- For merchants to have confidence in funds received via the Bitcoin network, there must be a means for them to be able to query the nodes who validate the transaction to ensure that it is not double spending valid inputs. This can be achieved in a variety of ways including:

Querying a selection of nodes for a transaction output that meets the merchant’s requirements

1. By the merchant sending the transaction on the sender’s behalf, and receiving notification of the transaction’s validity from the nodes it uses
2. Through double-spend alarm systems that notify any parties to transactions that double spend an input
3. Without these systems, it becomes much harder for merchants to accept Bitcoin payments for goods or services without waiting for them to be confirmed in a valid block secured by proof of work.

11 Transactions Video

12 Transactions Assessment No.3

Chapter 4: Timestamp Server (Approx 15 mins)

00 Section read-through

- The solution we propose begins with a timestamp server. A timestamp server works by taking a hash of a block of items to be timestamped and widely publishing the hash, such as in a newspaper or Usenet post [2-5]. The timestamp proves that the data must have existed at the time, obviously, in order to get into the hash. Each timestamp includes the previous timestamp in its hash, forming a chain, with each additional timestamp reinforcing the ones before it.

01 Timestamped Hashes

- The solution we propose begins with a timestamp server. A timestamp server works by taking a hash of a block of items to be timestamped and widely publishing the hash, such as in a newspaper or Usenet post [2-5]. The timestamp proves that the data must have existed at the time, obviously, in order to get into the hash. - Satoshi Nakamoto, Bitcoin Whitepaper

- A Bitcoin block consists of an ordered set of transactions which each validly spend existing unspent transaction outputs into new transaction outputs. The network considers each transaction to be a separate item or event, and builds the blocks as such, using the hashed transaction message as an input into the block's own hash function.
- When a node finds a valid block, each transaction is published as part of that block, and through the hash of the transaction message can be provenly shown to have existed at the time the block was found. Blocks are broadcast across the whole Bitcoin network and either accepted or rejected by the rest of the nodes in the competition. These broadcast events can be considered akin to the publishing of a notice on a publicly available bulletin board or website.

02 A chain of timestamped hashes

- Each timestamp includes the previous timestamp in its hash, forming a chain, with each additional timestamp reinforcing the ones before it. - Satoshi Nakamoto, Bitcoin Whitepaper
- A block's header must include the hash of the block upon which it is built. This 'link' forms a single chain of valid blocks leading all the way back to the very first block (the Genesis block) ever created using the system. A single block can have multiple child blocks which are built upon it, however only one of those children can become part of the permanent 'longest chain' of proof of work, further incentivising a strong connection between nodes who always want their own valid blocks to become part of the chain.
- As blocks are added to the chain the cumulative proof of work built upon all the blocks preceding it is increased. In this way, as transactions age they become more secure through the proof of work applied to all blocks that come after.
- This chain of proof of work forms a separate and distinct DAG from the transaction ledger commonly referred to as the Timechain or Blockchain. The contents of this chain are an immutable record of what took place on the ledger and the order in which it took place. Any transaction that contradicts an event (e.g. double spends an input) that has been recorded in a block which forms part of the chain is considered invalid and can never become a part of the longest chain of proof of work.
- Transactions which appear in blocks that do not form part of the longest chain of proof of work and which either contradict transactions that do appear in the longest chain, or which never get included in the longest chain of proof of work are considered invalid and are effectively removed from the ledger.
- Transactions that are validly created (e.g. not double spent inputs, valid script) may still end up being excluded from the longest chain of proof of work due to not being correctly broadcast or not paying enough of a transaction fee to be committed to the ledger. These transactions can persist in local versions of the transaction DAG however can never become a permanent part of the ledger and when lost from local memory are effectively erased.

03 Timestamp Server Video

04 Timestamp Server Assessment No.1

Chapter 5: Proof of Work (Approx 60 mins)

00 Section read-through

- To implement a distributed timestamp server on a peer-to-peer basis, we will need to use a proof-of-work system similar to Adam Back's Hashcash [6], rather than newspaper or Usenet posts. The proof-of-work involves scanning for a value that when hashed, such as with SHA-256, the hash begins with a number of zero bits. The average work required is exponential in the number of zero bits required and can be verified by executing a single hash.

- For our timestamp network, we implement the proof-of-work by incrementing a nonce in the block until a value is found that gives the block's hash the required zero bits. Once the CPU effort has been expended to make it satisfy the proof-of-work, the block cannot be changed without redoing the work. As later blocks are chained after it, the work to change the block would include redoing all the blocks after it.
- The proof-of-work also solves the problem of determining representation in majority decision making. If the majority were based on one-IP-address-one-vote, it could be subverted by anyone able to allocate many IPs. Proof-of-work is essentially one-CPU-one-vote. The majority decision is represented by the longest chain, which has the greatest proof-of-work effort invested in it. If a majority of CPU power is controlled by honest nodes, the honest chain will grow the fastest and outpace any competing chains. To modify a past block, an attacker would have to redo the proof-of-work of the block and all blocks after it and then catch up with and surpass the work of the honest nodes. We will show later that the probability of a slower attacker catching up diminishes exponentially as subsequent blocks are added.
- To compensate for increasing hardware speed and varying interest in running nodes over time, the proof-of-work difficulty is determined by a moving average targeting an average number of blocks per hour. If they're generated too fast, the difficulty increases.

01 Hashcash

- To implement a distributed timestamp server on a peer-to-peer basis, we will need to use a proof-of-work system similar to Adam Back's Hashcash [6], rather than newspaper or Usenet posts. - Satoshi Nakamoto, Bitcoin Whitepaper
- The implementation of Bitcoin's distributed timestamp uses hash-based proof of work to give nodes the ability to demonstrate their willingness to both invest in network infrastructure and place operational expenditure at risk via the hashing competition.
- Hashpower can be dynamically deployed giving owners of hashing machinery the ability to vote for nodes they believe are most capable at gathering and timestamping events. Hashcash was originally conceived as an anti-spam measure for email inboxes and was intended as a means to set a price on sending an email to disincentivise the creation of the hundreds of billions of spam emails that are sent every day to user inboxes.
- In the context of Bitcoin, the proof of work is used as a signal from one node to another that they are a capable and dedicated player on the network whose block solutions deserve consideration for insertion into the chain. This is important because the cost of one node connecting to, receiving and validating a block from another node is not trivial. Famously, this solution solves the Byzantine Generals Problem* which has long stood in the way of the implementation of distributed computing networks that do not require a central governance system.
- Nakamoto Consensus is a Byzantine fault tolerant consensus algorithm that works in conjunction with proof of work (PoW) to govern the Bitcoin blockchain. Byzantine fault tolerance (BFT) is a condition where a distributed system can remain fault-tolerant in the presence of malicious actors and network imperfection.
- *Byzantine Generals Problem is a term used in computing to explain a situation where components of a system may fail if participants don't agree on a strategy to deal with the problem. The problem assumes that some of the participants are bad actors who may spread misinformation or are in some way unreliable.
- Imagining a war where many different armies must work together to conquer a common enemy in a situation where there are an odd number of armies, common consensus must be reached in order to successfully attack. But, certain generals of some armies choose to disagree, leading to a critical system failure.
- This failure is known as a Byzantine Fault, in computing this is when it is unclear whether a component in a network is working properly or not. In Bitcoin each node is considered a 'general'

who contributes to the consensus of a network.

02 Scanning random space

- The proof-of-work involves scanning for a value that when hashed, such as with SHA-256, the hash begins with a number of zero bits. The average work required is exponential in the number of zero bits required and can be verified by executing a single hash. - Satoshi Nakamoto, Bitcoin Whitepaper
- The proof-of-work process involves taking a block header which is arranged in a pre-defined format and using it plus a ‘nonce’ value (a nonce is an arbitrary number that can be used just once in cryptographic communication) as the message to be hashed. The node passes out block headers containing a hashed record of all the transactions the block includes plus a timestamp and a unique identifier to hash machines whose hashing power is being applied to solving its block.
- Critically, validation of the proof of work is as simple as hashing the block header including the winning nonce value. Nodes do this before verifying the block’s contents to ensure they are working on a block solution which has been solved by a capable node.

03 Proof of Work Assessment No.1

04 Nonce

- For our timestamp network, we implement the proof-of-work by incrementing a nonce in the block until a value is found that gives the block’s hash the required zero bits. - Satoshi Nakamoto, Bitcoin Whitepaper
- A Nonce is a ‘Number used ONCE’ and is the means by which the block header is iterated during the proof of work process. A section of the block header is changed by a small amount each time so that a new hash can be calculated from the number.
- A Hash is a one-way function which takes a given input and produces a deterministic output. Someone given the output cannot reveal the input, but anyone with the input can verify its authenticity if its hash is public.
- The hash machines take the block header and begin testing different message hashes by incrementing the 4-byte nonce value and re-hashing the message as many times as possible per second. A block is solved when a hash machine finds a nonce value that when combined with the block header creates a message that hashes to a value which is less than the difficulty target which is stored in the block header as a 4-byte floating point number.
- Thanks to optimisations in hardware, most hashing machines can cycle through the approx. 4.3 billion different values the 32-bit nonce can represent in a matter of seconds, so further ‘Extra Nonce’ fields are used within the coinbase transaction to increase the amount of hashing each hash machine can perform on a given block template by causing a change to the merkle root.

05 Immutable Work

- Once the CPU effort has been expended to make it satisfy the proof-of-work, the block cannot be changed without redoing the work. - Satoshi Nakamoto, Bitcoin Whitepaper
- The nature of proof of work is that it can only be attempted upon a fixed block of information. In this way, capable nodes will always ensure that power expended performing proof of work is used on blocks that are fully complete and do not waste energy. Once discovered, a block with a valid proof of work solution cannot be modified in any way without changing its message hash and invalidating the work.

- Nodes who waste hashpower will find that the owners of the hashpower soon migrate to other more capable nodes, reducing their ability to generate blocks and cutting their revenue generating ability. This incentivises all nodes to act with honesty and to create blocks that are both valid and economical.

06 Chained effort

- As later blocks are chained after it, the work to change the block would include redoing all the blocks after it. - Satoshi Nakamoto, Bitcoin Whitepaper
- Once a block has been created and accepted as valid by other nodes on the network, the network collectively begins trying to build the next block using the hash of this block as part of the new block header. Every block since the Genesis block is built upon a previous block in the chain, so overwriting information stored at a certain point in the chain would require the proof of work on the block containing the information being overwritten, and every block discovered since in order for the network to recognise the new version of the time chain as valid history.
- Anyone seeking to overwrite a transaction using this method must build the new proof of work chain and outpace the constantly lengthening chain-tip for their blocks to be considered as valid, making it computationally impractical to erase information that has been captured in a block.
- In this way, information contained in blocks that have an established quantity of proof of work on top of them are considered unchangeable, or immutable.

07 Proof of Work Assessment No.2

08 One CPU, one vote

- The proof-of-work also solves the problem of determining representation in majority decision making. If the majority were based on one-IP-address-one-vote, it could be subverted by anyone able to allocate many IPs. Proof-of-work is essentially one-CPU-one-vote. - Satoshi Nakamoto, Bitcoin Whitepaper
- The very nature of proof-of-work is such that it cannot be faked, misrepresented or otherwise falsified. Without a Central Processing Unit (CPU) to cycle through combinations of nonce and block header, a node cannot participate in the block building process. Because proof of work requires the accumulation of infrastructure and the expenditure of energy, there is an inherent need for any node that wishes to participate in the activity of block building to have CPU power available to it in order to continuously process new combinations of block header and nonce to solve proof of work.
- A system such as One-IP-address-one-vote or proof of stake can be gamed by accumulation of resources that aren't representative of an investment in the reliability and security of the network. This discourages investment in network infrastructure as any costs related to the construction of more capable hardware takes away from the node's accumulated holdings reducing the node's ability to monetize its position. In this way these methods of achieving consensus can be shown to create instability and reduced levels of investment in the network.
- It is only via proof of work that all network participants can be evaluated on an even footing as to their node's capability and suitability to create blocks and to receive votes from hash generating CPUs. This system incentivises a race to find the most efficient ways of processing block headers and has already led to advances in hardware driving the block difficulty rate up many orders of magnitude since the beginning of the network.

09 The majority decision

- The majority decision is represented by the longest chain, which has the greatest proof-of-work effort invested in it. - Satoshi Nakamoto, Bitcoin Whitepaper
- At any given moment, it is possible for anyone to look at the longest chain of blocks that has been produced to determine where the greatest amount of work has been applied by multiple unconnected parties. This enables users of the network who are not interested in participating in the creation of blocks to become simple observers of the chain. The accumulated proof-of-work of the whole network acts as a signal that highly invested parties have come to agreement as a means to determine the validity of any given block or transaction.
- From time to time, competing chain-tips will emerge, however in time the competitors in the network will always unite their effort back to a single point upon which the cumulative power of all nodes is applied as a foundation for the continuance of the chain.

10 The honest chain

- If a majority of CPU power is controlled by honest nodes, the honest chain will grow the fastest and outpace any competing chains. - Satoshi Nakamoto, Bitcoin Whitepaper
- As long as more than half of the CPU power used to perform proof of work is controlled by honest nodes, the honest chain will remain the longest chain and the most easy to follow chain by the users of the network. This is true even when a chain tip is extended by nodes who use their hashpower to accept a set of dishonest transactions, or to enforce rules that go against the will of the honest nodes in the system.
- Importantly, while a set of dishonest nodes is able to extend their chain faster than the cumulative effort of the honest nodes in the system, their chain will have the appearance of representing the longest chain of proof-of-work to casual users of the network. However, for the dishonest nodes to maintain their lead and retain the longest chain they must continuously perform proof-of-work at a rate equal or greater than that of the honest chain for an indefinite period.
- This strategy is extremely costly to pursue. This shows that the network incentivises honest behaviour by allowing honest nodes to continuously work without the need to aggregate resources to pursue a dishonest end.
-

11 Proof of Work Assessment No.3

12 Attacking the longest chain

- To modify a past block, an attacker would have to redo the proof-of-work of the block and all blocks after it and then catch up with and surpass the work of the honest nodes. We will show later that the probability of a slower attacker catching up diminishes exponentially as subsequent blocks are added. - Satoshi Nakamoto, Bitcoin Whitepaper
- Thanks to the cumulative nature of Bitcoin’s proof of work process, it can be shown that for an attacking node to overwrite a particular block or record which has already been built upon by the honest operators on the network, they would first have to create a competing block. This block includes or excludes the information targeted by the attack before creating a new chain of blocks that is longer than the existing chain of honest blocks being built upon by the network.
- In order to uphold the fraud, the dishonest nodes must now also perform enough proof of work to generate new blocks at a rate that exceeds the generation of the honest blocks for as long as they require to perpetrate their fraud. They must do so in a way that is completely public, exposing their dishonesty to the entire world and leaving a direct trail to their system for law enforcement to follow.
- It should be noted that as the requirement for proof of work grows with the size of the network as a whole, the operation of a node leaves a larger and larger footprint that becomes near

- impossible to hide. Thus, further disincentivizing dishonest behaviour to the ease of identifying the dishonest actor.
- In a subsequent section of the whitepaper, Satoshi uses mathematics that show the difficulty of creating and maintaining a competing dishonest chain, which we will cover in a later section of this module.

13 Controlling the block discovery rate

- To compensate for increasing hardware speed and varying interest in running nodes over time, the proof-of-work difficulty is determined by a moving average targeting an average number of blocks per hour. If they're generated too fast, the difficulty increases. - Satoshi Nakamoto, Bitcoin Whitepaper
- In order to allow for node operators who invest in more efficient hardware to help discover more valid blocks through the proof of work process, the network uses an algorithm that adjusts the network difficulty to maintain a steady rate of block discovery no matter how many CPU cycles are applied to the proof of work process.
- The original Bitcoin client updated the difficulty target every 2016 blocks to try and control the block discovery rate to a speed as close to 6 blocks per hour as possible, however the current difficulty adjustment algorithm changes the rate every block in an attempt to compensate for the dynamics of the multiple competing SHA256 chains that currently exist. The difficulty algorithm will be adjusted back to the original 2016 block adjustment rate in the near future.
- This results in an ideal average of 144 blocks per day and means that the nodes would update the difficulty target every 2 weeks in a case where the network hashrate remained relatively static. The reality is that the block discovery process is a randomised process and can result in blocks being discovered at constantly changing rates.

14 Proof of Work Video

15 Proof of Work Assessment No.4

Chapter 6: Network (Approx 45 mins)

00 Section read-through

The steps to run the network are as follows:

1. New transactions are broadcast to all nodes.
2. Each node collects new transactions into a block.
3. Each node works on finding a difficult proof-of-work for its block.
4. When a node finds a proof-of-work, it broadcasts the block to all nodes.
5. Nodes accept the block only if all transactions in it are valid and not already spent.
6. Nodes express their acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash.

- Nodes always consider the longest chain to be the correct one and will keep working on extending it. If two nodes broadcast different versions of the next block simultaneously, some nodes may receive one or the other first. In that case, they work on the first one they received, but save the other branch in case it becomes longer. The tie will be broken when the next proofof-work is found and one branch becomes longer; the nodes that were working on the other branch will then switch to the longer one.

- New transaction broadcasts do not necessarily need to reach all nodes. As long as they reach many nodes, they will get into a block before long. Block broadcasts are also tolerant of dropped messages. If a node does not receive a block, it will request it when it receives the next block and realizes it missed one.

01 Running the Network

- The steps to run the network are as follows:

1. New transactions are broadcast to all nodes.
2. Each node collects new transactions into a block.
3. Each node works on finding a difficult proof-of-work for its block.
4. When a node finds a proof-of-work, it broadcasts the block to all nodes.
5. Nodes accept the block only if all transactions in it are valid and not already spent.
6. Nodes express their acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash.

- Satoshi Nakamoto, Bitcoin Whitepaper

- These listed steps are the way by which nodes running on the network compete to secure the contents of the ledger by storing it in blocks. The steps are broken down as follows:

1. New transactions are broadcast to all nodes.

- When a transaction is ready for transmission to the network, the wallet or application that created it will usually broadcast that transaction to a set of network peers that it recognises. This recognition might be pre-configured in the system, or created through a discovery process, but importantly, the transaction must at some point reach at least one network node.
- If the node finds that the transaction is valid and meets the network's rules it immediately broadcasts it to all other nodes that it is aware of on the network. Each of these nodes also validates the transaction as well, and broadcasts it to the nodes that they know about. Thanks to the dense connectivity at the center of the Bitcoin network, propagation from the first node the transaction reaches to all other nodes is very fast, requiring less than 1 second for global awareness.
- This process is critical for a node to be a valid competitor in the block building process. A node that fails to broadcast transactions out to the rest of the network risks expending proof of work on a block which might be rejected due to the overhead needed for all the other nodes to download the withheld information. Making sure that all nodes have all information needed to validate a block is the best way to ensure success.

2. Each node collects new transactions into a block.

- As the node receives new transactions, it continuously adds the ones that meet its requirements to its block template. As the time since the last block was discovered grows, so does the block. There is an inherent requirement for nodes to be sufficiently capable of collecting all of the transactions happening in the world in real time into their block in order to compete.
- The incentive to do so is present in the coinbase reward which is a combination of the block subsidy and transaction fees paid by users for the timestamping service. While the subsidy is high, nodes can act profitably without needing to manage millions of transactions. However, when the subsidy reduces to a level that makes it unprofitable, it will become crucial for the nodes to manage a much larger workload as each fee-bearing transaction added to its candidate block increases the revenue available in the block reward.

3. Each node works on finding a difficult proof-of-work for its block.

- As the block template grows, the node constantly updates the hashers with new versions of the template that include the most possible transactions in order to maximise revenue. Hashers always work against the most up-to-date version of the block template to ensure their work will be as profitable as possible by including the most transactions. The node and the hashers are a team in this regard, working together to solve blocks as effectively and efficiently as possible.

4. When a node finds a proof-of-work, it broadcasts the block to all nodes.

- As soon as a hasher returns a valid proof of work to a node, it must broadcast the fully formed block to all nodes on the network as quickly as possible. If the node delays in the transmission, it is possible that a different node might find a competing block which will win the race to become recognised as the first seen block. This requirement, and the risk of being outcompeted by other nodes in the block announcement process is a major driver of the incentives that lead to node operating enterprises to build custom high bandwidth infrastructure to manage their interconnectivity and expand the capacity of the network as a whole.

5. Nodes accept the block only if all transactions in it are valid and not already spent.

- When a node receives a block announcement from another node, it must validate its content. This involves going through and checking each transaction spends valid coins which have not already been spent and have valid scripts. If the node has already received the transactions in the block, it can skip parts of the validation process however if there are transactions that it has not seen in the block, it must request those transactions from the block winning node and validate them.

- In the meantime, if another block is discovered that the node can validate sooner due to not having to request and download additional information, it will choose to build on the block it validates first. This creates a very strong incentive for nodes not to withhold transactions from the rest of the network.

6. Nodes express their acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash.

- By building on a received block, nodes signal that the block was the first seen valid block and contains a set of transactions that are all valid and spend unspent coins. This is the process by which the longest valid chain of proof of work is extended and represents the collective agreement of the majority of hashpower on the network.

02 Network Assessment No.1

03 The longest chain

- Nodes always consider the longest chain to be the correct one and will keep working on extending it. - Satoshi Nakamoto, Bitcoin Whitepaper
- A node will always try to build on top of the longest valid chain of proof of work, as any block discovered on a block that is not at the longest valid chain tip will never be built upon by competing nodes. If a block is not built upon, the node operator receives no reward for creating the block and the proof of work that was done to build it would be wasted.
- Because proof of work costs money, node operators are incentivised to always ensure that their nodes are working on the most valid chain tip, minimising the risk of money being wasted on a block that has either already been built upon or surpassed by a competing valid chain tip.

04 Simultaneous blocks

- If two nodes broadcast different versions of the next block simultaneously, some nodes may receive one or the other first. In that case, they work on the first one they received, but save the other branch in case it becomes longer. - Satoshi Nakamoto, Bitcoin Whitepaper
- Because nodes work simultaneously to solve proof of work on separate and distinct block templates, there is always a chance that when a node finds a block, a competing node will find a different valid block solution for their own template at or around the same time.
- Nodes will always choose to work on top of the block they ‘saw’ first, where ‘seen’ means ‘received and validated’ so there is an incentive for any node that discovers a block to broadcast it as fast as possible, and where possible to ensure that every other node on the network has copies of all of the transactions in its block. Without these transactions, the validation process takes a lot longer and the chance of other nodes choosing this block to build upon is reduced.
- When nodes see two or more competing blocks, they will keep copies of each so that when the next block is announced they can jump across to the longest chain tip without having to re-do the block validation. Nodes are not typically made aware of which blocks other nodes are working on so must choose a path based only on the valid blocks that it has received.

05 Network Assessment No.2

06 Breaking the tie

- The tie will be broken when the next proof-of-work is found, and one branch becomes longer; the nodes that were working on the other branch will then switch to the longer one. - Satoshi Nakamoto, Bitcoin Whitepaper
- As soon as a node finds a new valid block and one of the competing chaintips is extended, the rest of the nodes on the network will immediately begin working on top of the longest chaintip. It is important for all nodes that they have a means by which they can quickly change to the new block even if they were working on a different chain, which is why competing valid blocks are held and validated even if they were not the first seen.
- The longer a node takes to move to the longest chaintip, the higher their chances of a valid block they discover on the competing chaintip being ignored by the rest of the network. This is the incentive that pushes all nodes to have a hyper-awareness of new block discoveries.
- This process is usually minimally disruptive to users as the competing blocks almost always contain transactions from the same global set.

07 Missed messages

- New transaction broadcasts do not necessarily need to reach all nodes. As long as they reach many nodes, they will get into a block before long. Block broadcasts are also tolerant of dropped messages. If a node does not receive a block, it will request it when it receives the next block and realizes it missed one. - Satoshi Nakamoto, Bitcoin Whitepaper
- In the rare situation where a transaction has reached some but not all of the nodes on the network, it only takes until one of the nodes that received it to find a block that includes that transaction. This has no impact on the validity of the transaction, or the immediate usability of its outputs accepting that the timestamp applied to its place in the blockchain would necessarily be some time later than the ideal ‘as soon as possible’. Transactions can be missed or omitted due to many factors such as a particular node’s own policies or a break in network communications between competing parts of the network.
- Similarly, when a node misses a block announcement message, they will keep working on the chaintip they believe to be the longest. When another block is announced that builds upon the

missed block, the node can quickly validate the headers against its own historical record and request any missing information regarding the new chaintip.
- Nodes do not help other nodes, so missed messages are not repeated on the network. Nodes only broadcast information because they are incentivised to do so. Where other nodes miss out on receiving that information, it is upon the operators of those nodes to improve their infrastructure and connectivity to ensure it does not happen again. In this way, the network spontaneously builds robust and dense connections between nodes without the need for any centralised authority or design.

08 Network Video

09 Network Assessment No.3

Chapter 7: Incentive (Approx 45 mins)

00 Section read-through

- By convention, the first transaction in a block is a special transaction that starts a new coin owned by the creator of the block. This adds an incentive for nodes to support the network, and provides a way to initially distribute coins into circulation, since there is no central authority to issue them. The steady addition of a constant of amount of new coins is analogous to gold miners expending resources to add gold to circulation. In our case, it is CPU time and electricity that is expended.
- The incentive can also be funded with transaction fees. If the output value of a transaction is less than its input value, the difference is a transaction fee that is added to the incentive value of the block containing the transaction. Once a predetermined number of coins have entered circulation, the incentive can transition entirely to transaction fees and be completely inflation free.
- The incentive may help encourage nodes to stay honest. If a greedy attacker is able to assemble more CPU power than all the honest nodes, he would have to choose between using it to defraud people by stealing back his payments, or using it to generate new coins. He ought to find it more profitable to play by the rules, such rules that favour him with more new coins than everyone else combined, than to undermine the system and the validity of his own wealth.

01 The Coinbase Transaction

- By convention, the first transaction in a block is a special transaction that starts a new coin owned by the creator of the block. - Satoshi Nakamoto, Bitcoin Whitepaper
- A coinbase transaction is the first transaction in a block. It is a unique type of bitcoin transaction that can be created by a miner. The miners use it to collect the block reward for their work and any other transaction fees collected by the miner are also sent in this transaction.
- The first transaction in any block is called a ‘Coinbase transaction’. This transaction has by definition a single input which contains an arbitrary string of text up to 100 bytes long which nodes use to mark the number of the block in the chain and in some cases to identify themselves.
- The transaction can have multiple outputs, the value of which when combined must be equal to or less than the block reward, which includes a distribution of coins and the fees collected from all of the transactions that have been added to that block.
- If the node fails to include coins that are part of the new coin distribution or were given as transaction fees to the value of the coinbase transaction’s outputs, those coins are lost and

cannot be recovered.

02 Coin Distribution

- This adds an incentive for nodes to support the network and provides a way to initially distribute coins into circulation, since there is no central authority to issue. - Satoshi Nakamoto, Bitcoin Whitepaper
- The distribution of new coins to the operators of block winning nodes acts as a subsidy to bootstrap the build-out of network infrastructure, giving the network time to accumulate users and providing a simple way to distribute the coinage used on the network to the miners who are providing the service of building blocks for network users.
- Because there is no central authority, the distribution is moderated through *Nakamoto Consensus, with nodes validating other nodes' self awarding of coins in the coinbase transactions they construct as part of their blocks. Because of this, nodes are incentivised to behave honestly and to perform a highly accurate accounting of their subsidy payment and the fees contained in the block.
- All satoshi tokens used in transactions on the Bitcoin network are distributed through this subsidy, and all coins in circulation can be traced via the transaction DAG back to a coinbase transaction.
- *Nakamoto Consensus refers to the set of rules, in combination with the Proof of Work consensus model in the network, that govern the consensus mechanism and certifies its trustless nature. Nakamoto Consensus can be broken down into 4 main parts.

• Proof of Work (PoW) • Block Selection • Scarcity • Incentive Structure

- Nakamoto Consensus is a Byzantine fault tolerant consensus algorithm that works in conjunction with proof of work (PoW) to govern the Bitcoin blockchain. Byzantine fault tolerance (BFT) is a condition where a distributed system can remain fault-tolerant in the presence of malicious actors and network imperfection.

- The combination of components allows Bitcoin to become the distributed network for value transfer that it is. It operates with trustless consensus and will remain secure as long as the majority of power contributed to the mining process is in the hands of honest miners.

03 Incentive Assessment No.1

04 Mining analogy

- The steady addition of a constant amount of new coins is analogous to gold miners expending resources to add gold to circulation. In our case, it is CPU time and electricity that is expended. - Satoshi Nakamoto, Bitcoin Whitepaper
- As nodes extend the chain of proof of work and are awarded coins distributed from the initial issuance the quantity of coins in reserve is depleted. The analogy is akin to a group of miners digging out a stope to find gold to add to a circulating economy.
- In the case of Bitcoin, nodes perform the work of 'mining' by compiling transactions into block templates then digging through hash combinations looking for a particular combination that solves the block difficulty puzzle. This exercise is costly in terms of computing power and requires both energy and infrastructure so it is similar to real world mining. Hash providers are incentivised to find efficiencies such as lower cost energy and more efficient machinery.
- A good analogy is that in 1850, it was efficient to mine for gold with a shovel and a pan, thanks to the exceptionally high concentrations of gold in the available ore, however by 1900 sluice guns

and larger machinery was needed to remain competitive as most of the easy to find gold had already been pulled from the land.
- In today's mining industry, large players spend hundreds of millions of dollars purchasing and operating plants and machinery as the amount of available gold reduces over time. This doesn't mean one can't find gold with a pan and shovel, however the rewards for doing so are much less than they were back at the beginning of the gold rush.

05 Transaction fees

- The incentive can also be funded with transaction fees. If the output value of a transaction is less than its input value, the difference is a transaction fee that is added to the incentive value of the block containing the transaction. - Satoshi Nakamoto, Bitcoin Whitepaper
- Nodes can also augment their income with transaction fees paid by users each time they make a transaction. These fees are paid by users creating transactions in which the combined value of the inputs is more than the value of the outputs. These tiny differences in value are aggregated by the nodes as they construct their block templates and paid out to the node operator in the coinbase transaction.
- The more transactions that a node can fit into a block, the more earning potential it has through transaction fees. In this way, scaling the network can be seen as a way for node operators to improve their potential income.

06 The end of inflation

- Once a predetermined number of coins have entered circulation, the incentive can transition entirely to transaction fees and be completely inflation free. - Satoshi Nakamoto, Bitcoin Whitepaper

- Over time, the amount of coins distributed to node operators through the block subsidy reduces. The first blocks found each awarded the winning blocks 50 Bitcoins, however every 210,000 blocks, or approximately every four years, the amount is cut in half, eventually reaching 1 Satoshi per block and then going to zero in around 2140 or after 32 'halvening' events. From this moment there will be no further inflation in the number of Satoshis that enter the network.

- The reduction in the rate at which new Bitcoins enter the economy is an important aspect of the system of incentives that encourage network scaling as it incentivises competing enterprises to build a network that can accommodate larger numbers of transactions with a view to using the fees paid as a means to replace the subsidy income.

- This also creates an incentive for node operators to build usage through targeted funding and the creation of novel use cases as an investment in a highly used application can deliver a big long-term return in the form of additional transaction fees through the network.

07 Incentive Assessment No.2

08 Encouraging honesty

- The incentive may help encourage nodes to stay honest. - Satoshi Nakamoto, Bitcoin Whitepaper
- Nodes make their income through the coins they award themselves in the coinbase transactions of valid blocks. The network has a rule that prevents coinbase rewards from being spent until 100 blocks have been built on top of the block in which they were awarded, meaning that without the support of the majority of competing nodes building on top of the block that contains it, a

coinbase reward might never become a permanent part of the ledger, rendering the coins it contains unspendable.
- For a node to create a block with dishonest activity such as a double spent transaction output or extra coinbase rewards, the rest of the network would have to collectively support their dishonest behaviour, investing proof of work and building upon the dishonest actions taken.
- Node operators are diverse and have headquarters in many different nation states. They are largely enterprise level organisations who pay taxes and are beholden to the laws applicable in the country where they reside.
- Actively spending money on work to build upon a block containing knowingly fraudulent activity is highly risky for an individual node, and if perpetrated by a collective majority of network actors would represent an existential threat to the network and their livelihoods thus creating a powerful incentive for all nodes to protect the integrity of the system. Importantly though it would be highly visible that this was happening to the entire world.
- To conduct a double spend attack on the network, the attacking node must accumulate enough hash power to overrule more than half of the other nodes on the network, and must then pay to maintain that hash power for an indefinite period of time, while it tries to convince the remaining nodes on the network to support its illegal actions.
- Hash power is the The computational capability in a machine that is used for crypto mining. Hash power may also refer to the total computing power of a miner or mining pool.
- Hash rate is a measure of the computational power on a blockchain network. Hash rate is determined by how many guesses are made per second. The overall hash rate helps determine the security and mining difficulty of a blockchain network
- Maintaining the attack as a single malicious actor is tremendously expensive due to the cost of performing proof of work, yet the attacker must break the law in plain sight using the Bitcoin ledger in full view of the public and law enforcement.
- It is easy to see that it is far less risky and much more lucrative for a node controlling such a large quantity of hash power to participate as an honest actor securing the network to legitimately win honest rewards as income.

10 Incentive Video

11 Incentive Assessment No.3

Chapter 8: Reclaiming Disk Space (Approx 30 mins)

00 Section read-through

- Once the latest transaction in a coin is buried under enough blocks, the spent transactions before it can be discarded to save disk space. To facilitate this without breaking the block’s hash, transactions are hashed in a Merkle Tree, with only the root included in the block’s hash. Old blocks can then be compacted by stubbing off branches of the tree. The interior hashes do not need to be stored.
- A block header with no transactions would be about 80 bytes. If we suppose blocks are generated every 10 minutes, 80 bytes * 6 * 24 * 365 = 4.2MB per year. With computer systems typically selling with 2GB of RAM as of 2008, and Moore’s Law predicting current growth of 1.2GB per year, storage should not be a problem even if the block headers must be kept in memory

01 Spent transactions

- Once the latest transaction in a coin is buried under enough blocks, the spent transactions before it can be discarded to save disk space.
- Satoshi Nakamoto, Bitcoin Whitepaper
- When a UTXO is created in a transaction, the outputs that were spent as inputs to the transaction are consumed by the action, and cease to exist as live spendable tokens on the network. Once the transaction has been ‘mined’ into a block which the network has then expended work building upon, it can be said that the transaction record is immutable. This means that anyone with a copy of the transaction can prove that the transaction was created before the block timestamp.
- Once the spent outputs from old transactions, that are referred to in the new transaction’s inputs, have been made immutable, nodes are free to remove them from their copy of the block chain. They are not required for the process of creating new blocks and consume hard disk storage space. At this point, it becomes the responsibility of the people who made the transactions to keep their own copies of them. This can be managed through archive services, private storage and more.

02 The Merkle Tree

To facilitate this without breaking the block’s hash, transactions are hashed in a Merkle Tree, with only the root included in the block’s hash.

- Satoshi Nakamoto, Bitcoin Whitepaper
- In order for nodes to be able to remove individual transactions that have been placed in a block without impacting the integrity of the block hash, a data structure known as a Merkle Tree is used.
- Merkle Tree is a hash structure that can hold an unbounded number of data items in a particular order, and identify the set using a single ‘Merkle Root’ value.
- Imagine a Merkle root like an inverted family tree. We start with a long list of individuals, and each is partnered and generates one child who partners with the child of another pair. This leads down to a single person who is the subject of the family tree.
- The Merkle root is a single hash which represents the ordered list of all of the transaction ID’s processed by the node during the construction of their block. This Merkle root is embedded in the block header which forms the blockchain DAG. Because every transaction included in the block is effectively an input into the final hash that produces the Merkle root, changing any detail of any individual transaction, or even the order of the transactions would produce a different Merkle root. We can exploit this property to provide a mathematical proof that a transaction is part of a Merkle tree with a given root, but we do not need the data for the entire tree in order to do so. The required data for such a “proof of inclusion” is very compact.
- Merkle trees allow Bitcoin blocks to grow at an unbounded rate with minimal impact on the user experience.
- A Merkle tree enables a node to remove or ‘prune’ individual transactions from their record of a given block, and to retain only hashes of the transaction, or even hashes of the branch the transaction was in. With just a block header, a node can provably show that any transaction for which they have a corresponding hash and merkle path is contained within a block. This allows nodes to keep efficient, high speed systems at the forefront of the network, making the most of dynamic memory which is expensive to keep.

03 Reclaiming Disk Space Assessment No.1

04 Compacting blocks

- Old blocks can then be compacted by stubbing off branches of the tree. The interior hashes do not need to be stored. - Satoshi Nakamoto, Bitcoin Whitepaper

- Like the original generation parents in the family tree, the Merkle tree starts with a list of transactions which are hashed to form the base level. As long as the node has copies of all of the transactions, they can validate the whole structure.
- Transactions are hashed to form their transaction IDs, which are then hashed in pairs to create the hashes at the parent branch, effectively halving the number of data items at the next level in the tree. This is akin to the parents at the top each producing a unique child.
- Parent branches are hashed with their adjacent parent branch creating another level half the size and so on and so forth until the two deepest branches are combined to form the Root Hash of the Merkle tree.
- This is like each generation halving the number of people, and those people's pairing with each other resulting in a next generation half the size. This tree gets smaller and smaller until we reach the root of the tree, or the 'subject' of the family tree.
- It is this root hash that is used in the block header which is subjected to Proof of Work, and it is this value which must remain provably linked to the tree the node is storing.
- Thanks to this structure, nodes are able to streamline their operation by cutting back the number of spent transaction records they store and focusing mainly on storing live UTXOs as needed for the operation of the network. As transactions are removed, the parent hash is generated and stored so that the node can still establish the authenticity of the remaining records they have kept. As more transactions are removed, the stubs that are stored move higher into the branches of the tree, further reducing the size of the block record.
- Imagine that the subject decides information about one of the original generations is no longer needed, and they cut it from the tree. As long as they keep information about the children, we can find a path from the subject back to the original generation. If enough parents are discarded, the system can purge their children from the tree as well. Records of these middle generations are only needed to ensure that the link between the subject and the original parent generation are able to be proven.
- Nodes role in the operation of the network does not involve storing a full copy of the transaction records in every block forever. This information is not required to validate transactions or to build blocks. This realisation allows the role of storing complete copies of every block to be pushed out from the central core of the network to archiving systems that can create business models around selling access to old, spent transaction records.
- This also creates an incentive for users to take responsibility for their own data. Records can be kept locally or in cold storage, removing the need to access any kind of archive. This also allows users to attach additional details to transactions which may not be stored on the ledger.

05 Block Headers

- A block header with no transactions would be about 80 bytes. If we suppose blocks are generated every 10 minutes, 80 bytes * 6 * 24 * 365 = 4.2MB per year. With computer systems typically selling with 2GB of RAM as of 2008, and Moore's Law predicting current growth of 1.2GB per year, storage should not be a problem even if the block headers must be kept in memory. - Satoshi Nakamoto, Bitcoin Whitepaper
- The record of a block's existence is its header. This is like the birth certificate of the subject of the family tree. From the header, a node can see which block it builds upon, what time it was discovered and can validate the proof of work done by the node that discovered it. The contents of the merkle tree, which are all of the transactions that the block includes, are not part of the header and are not needed to prove the existence of a block whose transactions were validated by the network a long time ago.
- Using the equations above, it becomes easy to see that the amount of data needed to store a full record of the chain of proof of work and the live UTXO set, plus a full set of all transactions from recently mined blocks is a much more efficient way to manage an operating node rather than having to store and manage transactions which will never again be referenced in the generation of a new transaction or block.

- The list of block headers grows linearly at a rate of around 4.2MB per year while the capability of computing systems scales exponentially. This makes it possible for the system to scale to accommodate the needs of a global financial network and for end user wallets which only require this small subset of Bitcoin data to operate efficiently on consumer grade devices such as mobile phones or even embedded IoT systems.

06 Reclaiming Disk Space Video

07 Reclaiming Disk Space Assessment No.2

Chapter 9: Simplified Payment Verification (Approx 45 mins)

00 Section read-through

- It is possible to verify payments without running a full network node. A user only needs to keep a copy of the block headers of the longest proof-of-work chain, which he can get by querying network nodes until he’s convinced he has the longest chain, and obtain the Merkle branch linking the transaction to the block it’s timestamped in. He can’t check the transaction for himself, but by linking it to a place in the chain, he can see that a network node has accepted it, and blocks added after it further confirm the network has accepted it.
- As such, the verification is reliable as long as honest nodes control the network, but is more vulnerable if the network is overpowered by an attacker. While network nodes can verify transactions for themselves, the simplified method can be fooled by an attacker’s fabricated transactions for as long as the attacker can continue to overpower the network. One strategy to protect against this would be to accept alerts from network nodes when they detect an invalid block, prompting the user’s software to download the full block and alerted transactions to confirm the inconsistency. Businesses that receive frequent payments will probably still want to run their own nodes for more independent security and quicker verification.

01 Full network nodes

- It is possible to verify payments without running a full network node. - Satoshi Nakamoto, Bitcoin Whitepaper

- In the context of this statement, a ‘Full Network Node’ is considered a node which performs all of the tasks outlined in Section 5 of the whitepaper, which are:

1. New transactions are broadcast to all nodes.
2. Each node collects new transactions into a block.
3. Each node works on finding a difficult proof-of-work for its block.
4. When a node finds a proof-of-work, it broadcasts the block to all nodes.
5. Nodes accept the block only if all transactions in it are valid and not already spent.
6. Nodes express their acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash.

- Satoshi Nakamoto, Bitcoin Whitepaper

- It is clear from this that nodes are systems that are specifically designed to certify the records being timestamped on the ledger.

- Anything else that connects to the network is a peer of a different type, and those peers can verify transactions that took place on the network without needing a full copy of the ledger or the need to build blocks or vote with hash power. This capability comes about through Simplified Payment Verification, or SPV*.

- *Simple Payment Verification (SPV), is a system outlined in the Bitcoin Whitepaper that enables clients like wallets to verify that a transaction has been included in a block and therefore a payment has been made and accepted by the network.

- This is possible because the Merkle tree structure stores the transactions in each block. The structure of a Merkle tree means that we only need to know the merkle root/top hash and a small branch of the tree to verify if a transaction is part of the tree, that is, if it's been included into a Bitcoin block. This is done by taking the nodes that are in the path that connects the Merkle root with one of the bottom transactions and bundling them together to create a proof.

- Running a node requires downloading the entire blockchain, but a Bitcoin user can use SPV proofs and only needs to know the header of each block which includes the Merkle root, in order to verify any transaction in that block, so we only have to store 80 bytes per block, instead of the entire block required for nodes.

02 Merkle Branches

- A user only needs to keep a copy of the block headers of the longest proof-of-work chain, which he can get by querying network nodes until he's convinced he has the longest chain, and obtain the Merkle branch linking the transaction to the block it's time stamped in. - Satoshi Nakamoto, Bitcoin Whitepaper
- In a similar fashion to how nodes can store a pruned selection of transactions in their copy of the block chain, a user can store only a highly curated section of transactions using much the same technique, except instead of starting with the full copy of the block chain and selectively removing transactions, the user starts with a copy of the block header list which can be easily verified through Proof of Work, and selectively adds only the transactions which they directly are interested in.
- Going back to the analogy of a family tree, this is the ability of someone with the details of an original parent and the middle generations of the family tree can prove that they are linked to the subject. When handling these transactions, they can be a very small number as would be required by a user of a wallet or app, or a larger set comprising millions of transactions, but which is a minority subset of the overall ledger.
- Looking at our family tree example, each original parent needs only to keep the path back to the subject's identity. The family path can be requested from archive systems which keep the full family tree.
- Each transaction can be proven to exist within a real block on the block chain through the Merkle Branches which connect the hash of its transaction ID all the way back up to the Merkle Root which is contained within the block header. If the transaction can be proven to exist in a block, it can be shown that a node accepted it as a valid transaction written to the ledger and time stamped with other nodes agreeing by building further on top of the block the transaction was contained in.

03 Simplified Payment Verification Assessment No.1

04 Transaction acceptance

- He can’t check the transaction for himself, but by linking it to a place in the chain, he can see that a network node has accepted it, and blocks added after it further confirm the network has accepted it. - Satoshi Nakamoto, Bitcoin Whitepaper
- By being able to validate that the hash of the transaction itself (the TXID) can be probably shown to exist in a block that has been accepted and built upon by the network, a user can safely say that the transaction is valid.
- This allows a user with a set of block headers to build a verifiable history of all transactions that are relevant to their needs and provides a clear and simple means to demonstrate that proof to other peers on the network. This is important for any application that builds upon information that is stored on the ledger and allows users to ensure that the information is valid and exists in a block.

05 Verification during attack situations

- As such, the verification is reliable as long as honest nodes control the network, but is more vulnerable if the network is overpowered by an attacker. - Satoshi Nakamoto, Bitcoin Whitepaper
- If the network is overpowered by an attacker who is extending the longest chain of proof of work upon an invalid block (either a block that contains an invalid transaction or which doesn’t comply with some other aspect of the network’s rules) user wallets will be unable to determine that the longest chain is not the longest valid chain of proof of work. In this instance it would be possible for the attacker to present information about an invalid transaction that would imply that it had been accepted into the longest chain of proof of work and built upon by the majority of network CPUs.

06 Maintaining an attack

- While network nodes can verify transactions for themselves, the simplified method can be fooled by an attacker’s fabricated transactions for as long as the attacker can continue to overpower the network. - Satoshi Nakamoto, Bitcoin Whitepaper
- This strategy and the illusion of the invalid transactions appearing valid can only be maintained for as long as the attacking node can afford to maintain a majority of the network hash. As soon as the honest chain of blocks overtakes the dishonest chain, user systems will reject the invalid chain and jump across to the now longer valid chain, rendering the attacker’s transactions void and destroying the investment in the proof of work used to build the chain.
- A dishonest attack of this form is enormously costly and must be conducted in a way that is fully visible to the public, making it extremely risky for the operators of dishonest nodes to participate in such attacks on the network’s validity. This high economic cost of attacking the network is part of the security model of Bitcoin and serves to strengthen the system against dishonest players.

07 Simplified Payment Verification Assessment No.2

08 Invalid Block Relay System

- One strategy to protect against this would be to accept alerts from network nodes when they detect an invalid block, prompting the user’s software to download the full block and alert transactions to confirm the inconsistency. - Satoshi Nakamoto, Bitcoin Whitepape

- This proposed solution would be to set up a system to alert user wallets who requested the longest chain of proof of work from the network that the longest chain they could download was in-fact a rejected dishonest attack chain. This gives them details of the earliest block in the invalid chain, details of the detected invalid transactions, and the details of the longest valid chain including the valid versions of any double spends inserted into the malicious chain.
- This is a theoretical system to protect against a very theoretical but possible attack. User wallets using SPV to track valid transactions within a known set of keys can be tricked into following an invalid chain of proof of work, even for a short period of time.

09 Businesses running nodes

- Businesses that receive frequent payments will probably still want to run their own nodes for more independent security and quicker verification.
- Satoshi Nakamoto, Bitcoin Whitepaper
- A business that is making or receiving very large numbers of transactions may elect to operate their own node, participating in the process of receiving and validating all transactions on the network and vying for user hash power to be allocated to their node in order to win blocks.
- This would allow a business to follow the longest valid chain very closely and to have full visibility of any attempted attack on the network, and any double spends that might be directed at them or any other businesses for whom they monitor accounts.
- This opens up new business opportunities for major on-line payment processors to operate network systems and to enhance their own operational security at the same time.

10 Simplified Payment Verification Video

11 Simplified Payment Verification Assessment No.3

Chapter 10: Combining and Splitting Value (Approx 30 mins)

00 Section read-through

- Although it would be possible to handle coins individually, it would be unwieldy to make a separate transaction for every cent in a transfer. To allow value to be split and combined, transactions contain multiple inputs and outputs. Normally there will be either a single input from a larger previous transaction or multiple inputs combining smaller amounts, and at most two outputs: one for the payment, and one returning the change, if any, back to the sender.
- It should be noted that fan-out, where a transaction depends on several transactions, and those transactions depend on many more, is not a problem here. There is never the need to extract a complete standalone copy of a transaction’s history.

01 Dynamically sized coins

“Although it would be possible to handle coins individually, it would be unwieldy to make a separate transaction for every cent in a transfer.”

- Satoshi Nakamoto, Bitcoin Whitepaper

- Each time a bitcoin transaction takes place, one or more unspent transaction outputs are gathered into the transaction as inputs and spent into a new combination of outputs scripts. A Bitcoin transaction can handle as many or as few Satoshi tokens as are needed by the user and supported by network nodes.

- Without this ability to merge and split outputs via a single transaction, the requirement to sign a new output for each satoshi in the transaction would make it economically unfeasible to manage the ledger. This approach gives the necessary balance required to manage bitcoins on the ledger without creating an undue burden through breaking down the outputs into arbitrarily sized pieces.

02 Inputs and Outputs

- To allow value to be split and combined,transactions contain multiple inputs and outputs. - Satoshi Nakamoto, Bitcoin Whitepaper
- To facilitate the aggregation of many small coins into one large payment, transactions support the capability to use multiple coins as inputs. This means that a user who's wallet mainly receives micropayments can still use those small coins to make larger purchases without burdening the receiver. This same mechanism also allows for multiple users to participate in a transaction by separate signing an input or inputs into the transaction.
- Similarly, on the output side, a single transaction can pay out to one or many separate output scripts, providing a means for users to pay multiple parties in a single transaction. For example, a merchant who receives goods on consignment can have their customer pay the manufacturer directly at the point of sale, while also paying their cut, sales tax and any municipal surcharges that might apply. In this scenario, all parties to the transaction receive their payments instantaneously, removing the burden of quarterly accounting from the merchant.

03 Combining and Splitting Value Assessment No.1

04 A typical example

- Normally there will be either a single input from a larger previous transaction or multiple inputs combining smaller amounts, and at most two outputs: one for the payment, and one returning the change, if any, back to the sender. - Satoshi Nakamoto, Bitcoin Whitepaper
- In a typical Bitcoin transaction being conducted between two peers, the sending party will take either a single large coin or several smaller coins and spend them into one or two outputs. A payment into one output would mean that the aggregate value of all coins (minus processing fees) would correspond to the exact amount the receiver was expecting for the transaction.
- When a coin or combination of coins cannot be found in the user's wallet that makes an exact value as requested for the payment, the wallet creates a second output which is paid to another address from within the user's wallet. It sends the overflow amount which was over and above the required payment value back to the sending party as 'change'. This money is spent back into a script that the wallet knows how to spend and is immediately received back into the wallet's balance as spendable funds.

05 Fan-out

- It should be noted that fan-out, where a transaction depends on several transactions, and those transactions depend on many more, is not a problem here. There is never the need to extract a complete standalone copy of a transaction's history. - Satoshi Nakamoto, Bitcoin Whitepaper

- When a transaction is spending multiple inputs from multiple different transactions, the wallet does not necessarily need to know the full provenance of the input other than it's transaction ID and index number. The network nodes perform the task of making sure that the inputs being spent do indeed come from valid transactions which have been or are in the process of being timestamped into a block, and that they have valid history leading back to their issuance. This removes the burden of chasing down a transaction's history from the user.
- The full history is only needed if a user wants to validate a coins history for their own purposes outside of the needs of the network. This may be through an application layer requirement or simply curiosity, but it is not a fixed requirement of using the network.

06 Combining and Splitting Value Video

07 Combining and Splitting Value Assessment No.2

Chapter 11: Privacy (Approx 45 mins)

00 Section read-through

- The traditional banking model achieves a level of privacy by limiting access to information to the parties involved and the trusted third party. The necessity to announce all transactions publicly precludes this method, but privacy can still be maintained by breaking the flow of information in another place: by keeping public keys anonymous. The public can see that someone is sending an amount to someone else, but without information linking the transaction to anyone. This is similar to the level of information released by stock exchanges, where the time and size of individual trades, the "tape", is made public, but without telling who the parties were.
- As an additional firewall, a new key pair should be used for each transaction to keep them from being linked to a common owner. Some linking is still unavoidable with multi-input transactions, which necessarily reveal that their inputs were owned by the same owner. The risk is that if the owner of a key is revealed, linking could reveal other transactions that belonged to the same owner.

01 Traditional Models

- The traditional banking model achieves a level of privacy by limiting access to information to the parties involved and the trusted third party. - Satoshi Nakamoto, Bitcoin Whitepaper
- Traditional banking systems work by holding data about each of the bank's account holders on secure systems managed and maintained by the bank. These systems are used as a means to tie an account or source of funds to an individual or business with the bank acting as a trusted guardian of the information.
- While the systems are designed for exclusive use of the bank or institution who keeps the details, these identity systems are frequently built using legacy frameworks that have been adapted to on-line requirements which has led to many instances of customer data being accessed and used maliciously to perpetrate theft and fraud against bank account holders.

02 Privacy in Bitcoin

- The necessity to announce all transactions publicly precludes this method, but privacy can still be maintained by breaking the flow of information in another place: by keeping public keys anonymous. - Satoshi Nakamoto, Bitcoin Whitepaper

- Because most outputs in Bitcoin are locked with a script requiring a knowledge proof, it isn't possible to remove all information about the identity of the receiving party from the transaction. By virtue of the fact that information they provided to the payer is embedded in the transaction the transaction itself cannot be anonymous, so care must be taken by the receiver to retain their own privacy over the information in the script.
- Even when the transaction is spent and the knowledge proof needed is written to the ledger, the user can still remain private by not publishing their details onto the ledger as part of the transaction. Both parties to a transaction can make separate records to ensure that any requirement to re-visit the transaction can be executed with accurate and full knowledge of the nature of the transaction.

03 Privacy Assessment No.1

04 Public records

- The public can see that someone is sending an amount to someone else, but without information linking the transaction to anyone. - Satoshi Nakamoto, Bitcoin Whitepaper
- When both parties to a transaction utilise robust privacy measures to ensure that only they are privy to the details of the transaction, it becomes almost impossible for third parties who are outside the sphere of trust pertaining to the transaction to follow the details. This can be further augmented by breaking the payment into multiple outputs or even multiple transactions to further mask the details.
- In this way it becomes very easy for lawful and considerate users of Bitcoin to guard their privacy and maintain a secure online presence.

05 Stock Exchange Comparison

- This is similar to the level of information released by stock exchanges, where the time and size of individual trades, the "tape", is made public, but without telling who the parties were.
- Satoshi Nakamoto, Bitcoin Whitepaper
- In the same way that stock exchanges keep private records of who exchanged with who and release a cleaned set of records containing only the price and size of each trade, so the Bitcoin ledger maintains a private tape of all records of exchange.
- User wallets can build repositories of transaction data which can be kept in offline archives, or even stored on the ledger but importantly these records are completely separate to the actual transaction itself and could only be exposed through one of the users repositories being compromised.
- In this way, not only is the issue of large centralised repositories of data solved by each user holding their own records, the ownership of transaction records is also transferred to the user. This vastly reduces the risk of hacks that expose the records of large numbers of users as now the hackers must compromise a separate system for each individual user's records.

06 Key Re-use

- As an additional firewall, a new key pair should be used for each transaction to keep them from being linked to a common owner. - Satoshi Nakamoto, Bitcoin Whitepaper
- In situations where a user is receiving all of their funds into a re-used locking script, it becomes much easier to see what funds that user has received, and when they are spending them as a locking script is tied to a private key that would typically be held by a single person. In order to mitigate against this risk, the user can simply choose a new private key every time new funds are being received in order to separate the digital coins on the ledger.

- Most wallets are capable of doing this through techniques collectively known as Hierarchically Deterministic Keychains which allow the wallet to generate an effectively unlimited number of keypairs from a predetermined seed. This ostensibly allows the user to ‘recover’ their wallet through re-creation of the original seed in the event the wallet is lost or destroyed.

07 Privacy Assessment No.2

08 Linking inputs

- Some linking is still unavoidable with multi-input transactions, which necessarily reveal that their inputs were owned by the same owner. - Satoshi Nakamoto, Bitcoin Whitepaper
- When a user creates a transaction that aggregates multiple inputs to pay one larger output, the coins that are used in that transaction can be traced back as likely belonging to a single owner (although this is not guaranteed). The record on the ledger is not enough on its own to identify that user, however it is possible for anyone with specific knowledge of the transaction to show that the inputs used may have belonged to the spending party.
- Where possible wallets can avoid this by always spending a larger output than the payment itself, however this is not always possible and transactions that aggregate coins are required.

09 Linking the owner

- The risk is that if the owner of a key is revealed, linking could reveal other transactions that belonged to the same owner.
- Satoshi Nakamoto, Bitcoin Whitepaper
- If a user spends coins that it has received from third parties, or which were spent back to themselves as change, it becomes possible to trace some part of the chain of ownership back via the ledger. This risks exposing a user’s financial activities to malicious parties who have an understanding of how to analyse the public ledger.
- Mitigation strategies include the use of separate transactions in instances where the many inputs are each spent in completely separate transactions, thereby avoiding the possibility of linking the user to a group of coins.

10 Privacy Video

11 Privacy Assessment No.3

Chapter 12: Calculations (Approx 60 mins)

00 Section read-through

01 Attacking the chain

02 Things the attacker cannot achieve…

03 The only thing the attacker can achieve…

04 The Binomial Random Walk

05 The Gambler’s Ruin

06 Exponential odds

07 Waiting for confirmation…

08 Attack via proof of work

09 Vanishing probabilities

10 Calculations Video

11 Calculations Assessment No.1

Chapter 13: Conclusion (Approx 15 mins)

00 Section read-through

- We have proposed a system for electronic transactions without relying on trust. We started with the usual framework of coins made from digital signatures, which provides strong control of ownership, but is incomplete without a way to prevent double-spending. To solve this, we proposed a peer-to-peer network using proof-of-work to record a public history of transactions that quickly becomes computationally impractical for an attacker to change if honest nodes control a majority of CPU power. The network is robust in its unstructured simplicity. Nodes work all at once with little coordination. They do not need to be identified, since messages are not routed to any particular place and only need to be delivered on a best effort basis. Nodes can leave and rejoin the network at will, accepting the proof-of-work chain as proof of what happened while they were gone. They vote with their CPU power, expressing their acceptance of valid blocks by working on extending them and rejecting invalid blocks by refusing to work on them. Any needed rules and incentives can be enforced with this consensus mechanism.

01 Conclusion

We have proposed a system for electronic transactions without relying on trust. We started with the usual framework of coins made from digital signatures, which provides strong control of ownership, but is incomplete without a way to prevent double-spending.

- Satoshi Nakamoto, Bitcoin Whitepaper

To solve this, we proposed a peer-to-peer network using proof-of-work to record a public history of transactions that quickly becomes computationally impractical for an attacker to change if honest nodes control a majority of CPU power.

- Satoshi Nakamoto, Bitcoin Whitepaper

The network is robust in its unstructured simplicity. Nodes work all at once with little coordination. They do not need to be identified, since messages are not routed to any particular place and only need to be delivered on a best effort basis. .

- Satoshi Nakamoto, Bitcoin Whitepaper

Nodes can leave and rejoin the network at will, accepting the proof-of-work chain as proof of what happened while they were gone.

- Satoshi Nakamoto, Bitcoin Whitepaper

They vote with their CPU power, expressing their acceptance of valid blocks by working on extending them and rejecting invalid blocks by refusing to work on them. Any needed rules and incentives can be enforced with this consensus mechanism.

- Satoshi Nakamoto, Bitcoin Whitepaper

In this whitepaper, Satoshi has proposed a system that creates a system for managing control of custody via a global ledger which is stored as a series of time-stamped blocks that group valid transaction records together. The nodes who build the ledger compete in an honest competition where the highest performing nodes represent investments of several hundred million dollars at this point in time, this investment has a demonstrated history of exponential growth, and can be expected to grow much larger in the future.

They collaboratively construct the chain of blocks using a consensus model that requires nodes to be present and aware of the network at all times. The system requires constant participation which is managed by arranging nodes in a highly distributed yet densely connected small world network. Nodes can drop off the network and rejoin at will, and it remains robust.

Nodes follow the chain of blocks that they believe to be the honest and correct history of the network. When nodes see blocks that contain double spends, it is their prerogative to choose which chain they believe to be honest. This is typically the chain which contains the transaction that the node saw first, however nodes will jump forward if the rest of the network agrees that the other transaction was first.

Nodes are incentivised also to enforce the rules of the network using hashpower. This is a process that happens between nodes which have adequate hash to discover new blocks and can be used to manage all disputes on how the network should function.

02 Conclusion Video

What is Proof of Work

Proof of work is used to protect the integrity of new data on a decentralised network used by cryptocurrencies and other Defi apps.

What is Blockchain Concensus Mechanism

What does it mean for a blockchain to be in concensus?. Blockchain concensus is the process through which the network's peers come to an understanding on the current state of data and to verify transactions and maintain the security of the underlying blockchain.

Types of Concensus Mechanism

- Proof-of-Work
- Proof-of-Stake
- Proof-of-Importance
- Proof-of-Capacity
- Proof-of-Elasped Time
- Proof-of-Burn
- Proof-of-Authority

What is Proof-of-Work?

Proof-of-Work is the process that enables the centralized blockchain network to reach concensus or agreement, on issues like account balances and the chronological order of transactions. Proof of Work is considered the most dependable and most safe of all other consensus mechanism.

> In Cryptocurrency, Miners and Proof-of-Work guarantee Transparent and accurate transactions.

Proof-of-Work and Mining

Workings Of Proof-of-Work

Step 1: New transactions are grouped together (The transactions are then combined into a block).

Step 2: Crytominers compete to (be the first to) process the new block (by solving a difficult mathematical problem). A miner earns the priviledge to execute the block transactions by providing evidence they have performed the computing work(Known as a hash)

Step 3: Only one miner can add a new block. Who ever gets to process the block is chosen at Random to some extent. The winner gets a reward (Brand new crypto coin).

Example of Proof-of-Work

In Proof-of-Work, a computer must randomly perform a Hashing operations until it generates an output with the required minimum number of leading zeros to provide Proof-of-Work.