finxter

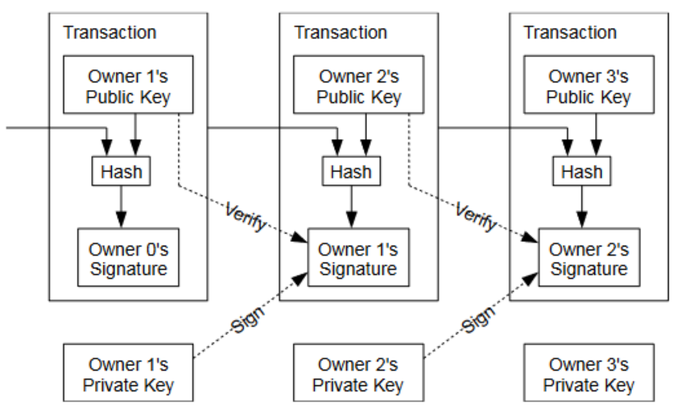
Bitcoin Whitepaper Cheat Sheet

💡 ***Download PDF: https://blog.finxter.com/bitcoin-whitepaper-cheat-sheet***

Abstract: A purely peer-to-peer (P2P) version of electronic cash for direct payments without centralized financial institutions based on digital signatures. We propose a solution to the double-spending problem using a P2P network using timestamped transactions in an ongoing chain of hash-based proof-of-work (PoW). This block-chain forms a record that cannot be changed without redoing the PoW. 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 not cooperating to attack the network, they'll generate the longest chain and outpace attackers. The network requires minimal structure. Messages are broadcast on a best effort basis. Nodes leave and join the network at will, accepting the longest PoW chain to learn about the current state of the chain.

1. Introduction **Weaknesses eCommerce:** (1) Relies on financial institutions as trusted 3rd parties for ePayments. (2) Non-reversible native transactions impossible leading to costly mediation. (3) Trust-based system increase scrutiny & fraud. **Bitcoin fixes this**: An electronic payment system based on cryptographic proof for direct transactions without trusted third party. Bitcoin uses a peer-to-peer distributed timestamp server to establish the chronological order of transactions. The system's security relies on honest nodes collectively having more CPU power than any group of attacker nodes.

2. Transactions An electronic coin is a *chain of digital signatures*, transferred through digital signing of transaction hashes & public keys of next owners.
Problem: Double spending of coins! **Fiat System**: Central authority checks every transaction. **Bitcoin**: Publicly announce all transactions o no previous owner signed earlier transactions w/o central authority. Create a system ensuring global majority consensus of transaction order.



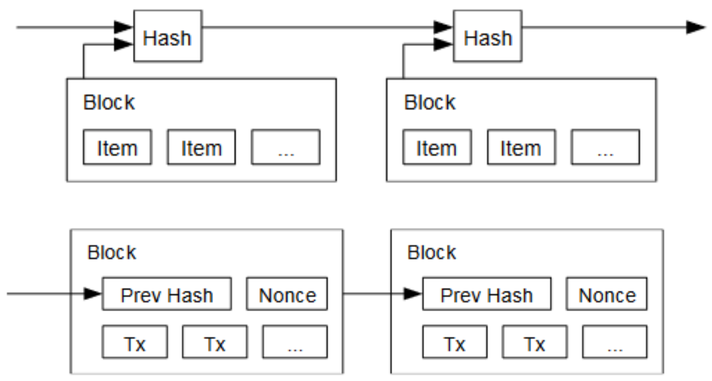
3. Timestamp Server Bitcoin's solution begins with a timestamp server that takes a hash of a block of items to be timestamped. We publish the hash. The timestamp proves the data must have existed at the time to get into the hash. Each timestamp includes the previous timestamp in its hash, forming a chain, with each additional timestamp reinforcing the ones before it.

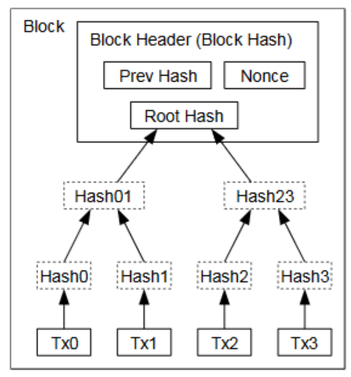
4. Proof of Work (PoW) Bitcoin uses a PoW system to create a P2P timestamp server. Collectively, miners find a value that starts with several zero bits when hashed with SHA-256. The *PoW difficulty* is the number of zero bits. PoW is achieved by incrementing a nonce in the block until a value gives the block's hash the required zero bits. The work needed is exponential in the zero bits required: Each additional zero doubles the average work. Altering the block would require redoing the work for all subsequent blocks. PoW also helps in majority decision-making (*"one-CPU-one-vote"*) so that the heaviest chain with most CPU effort determines the majority decision. If honest nodes control most CPU power, their chain will grow fastest. To change a past block, an attacker must redo the PoW for that and all following blocks to surpass the honest nodes' work. The chances of a slower attacker catching up decrease exponentially as new blocks are added. PoW difficulty adjusts targeting average six blocks per hour, compensating for varying node hardware speeds and mining interest. If blocks are generated too quickly, difficulty increases.

6. Network – Steps to Run (1) Broadcast new transactions to all nodes. (2) Each node gathers new transactions into a block. (3) Nodes work to find a difficult proof-of-work for their block. (4) When a node finds a proof-of-work, it broadcasts the block. (5) Nodes only accept block if all its transactions are valid and unspent. (6) Nodes show block acceptance by creating the next block in the chain, using the accepted block's hash as the previous hash.
Notes: Nodes consider the heaviest chain as correct & work to extend it. If two nodes broadcast different next blocks, receiving nodes work on the first but save the other in case its chain becomes heavier. The tie breaks when the next PoW is found and one branch grows heavier. Nodes then switch to the heavier branch. The Bitcoin protocol tolerates dropped messages because new transaction broadcasts don't need to reach all nodes to get into a block quickly. Nodes request missed blocks when they receive the next block.

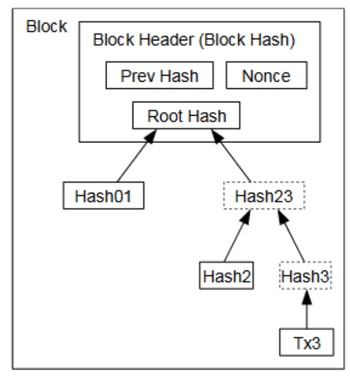
6. Incentive The first block "coinbase" transaction creates a new coin & sends it to block creator incentivizing miners to secure the network and circulating coins fairly w/o central authority. Plus, transaction fees will fully replace coinbase mining incentive once 21M BTC have been mined. No inflation! This incentivizes honesty among nodes. If an attacker amasses more CPU power than all honest nodes, they can either defraud others by reversing payments - or generate BTC revenue fairly which is likely to be more profitable while securing their own wealth.

7. Reclaiming Disk Space Once the latest transaction in a coin is buried under enough blocks, discard spent transactions before it to save disk space. To do this without breaking the block hash, use a Merkle Tree with only the root included in the block's hash. Old blocks can then be compacted by removing branches of the tree.



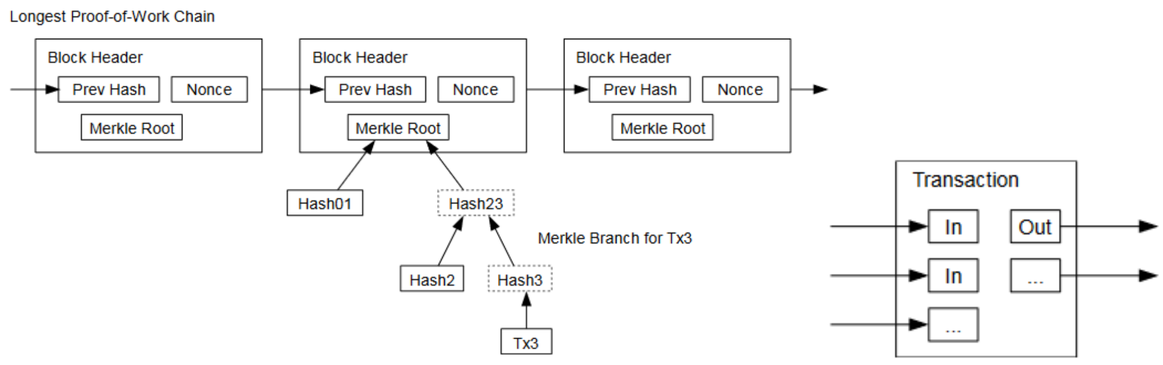


Transactions Hashed in a Merkle Tree



After Pruning Tx0-2 from the Block

8. Simplified Payment Verification You can verify your payments w/o a full network node by only keeping a copy of the block headers from the longest PoW chain obtained by querying network nodes for the chain and Merkle branch that connects the transaction to its timestamped block. The transaction's hash placement in the chain shows it has been accepted by a network node, with subsequent blocks confirming this acceptance. The verification method works if honest nodes dominate the network. Businesses or individuals might still prefer running their own nodes for enhanced security and faster verification.



9. Combining and Splitting Value Bitcoin allows to split and combine value by allowing transactions one or multiple in- and outputs.

10. Privacy The traditional banking model achieves privacy by limiting information access. Bitcoin makes all transactions public, but privacy can still be maintained by *keeping public keys anonymous*. You can see that X sends BTC to Y, but without being able to link (X,Y) to anyone. Tip: use a new key pair for each transaction to prevent them from being linked to common owner. Some linking will always be possible because multi-input transactions reveal that their inputs were owned by the same owner.

11. Calculations Assuming p>q, probability of attacker catching up drops exponentially as the no. blocks increases. Without early luck and with the odds against him, the attacker's chances become vanishingly small as he falls further behind.

$$q_z = \begin{cases} 1 & \text{if } p \le q \\ (q/p)^z & \text{if } p > q \end{cases}$$

p = probability an honest node finds the next block
q = probability the attacker finds the next block
q_z = probability the attacker will ever catch up from z blocks behind