Bitcoin Abbreviated: A P2P Electronic Cash System

Adapted by Justin Huang

1 Introduction

Double-Spending Problem -> Potential flaw in digital spending where same payment is used elsewhere

Internet payments often rely on financial institutions serving as trusted third parties to process electronic payments.

Bitcoin is an electronic payment system based on **cryptographic proof** and **computational work** instead of trust, removing the need for a third party.

The computational work involved is the foundation of **blockchain**, a peer-to-peer timestamp server used to generate computational proof of the chronological order of transactions.

2 Transactions

Electronic coin -> Chain of **digital signatures**, where coins are transferred by signing the **hash** of the previous transaction and public key of owner.

To solve the double-spending problem, we need to know that the previous owner of the coin didn't sign any other transactions. In Bitcoin, the earliest transaction is the only one that counts. In order to confirm lack of other transactions with the same coin, we need to know all transactions.

Without a trusted third party, this is accomplished by **publicly-announced transactions**, and a system where participants agree on a **single history of the order in which transactions are received**.

In other words, the majority of nodes must agree that the time of the transaction is the first time the transaction was received.

3 Timestamp Server

The solution lies in a timestamp server—the **blockchain**.

The blockchain works by taking a hash of a **block** of items to be timestamped and then widely publishing the hash. Each block includes the previous block in its hash to form a chain, thus the name of a blockchain.

4 Proof-Of-Work

Proof-of-work involves searching for a value that, when hashed with the block, creates a hash starting with a certain number of zero bits. This value is known as the **nonce**.

In the blockchain, proof-of-work is performed by incrementing the nonce while searching for the required number of zero bits. The required number of zero bits indicates the **target**. Once the

requirement is satisfied (the hash of the nonce and block meet the target), the block is added to the blockchain. This process is colloquially referred to as **mining**.

Proof-of-work is the system that verifies the authenticity of a blockchain through computational work, as mentioned previously. Once the block is added to the blockchain, it cannot be changed without redoing the work. When other blocks are added to the chain, redoing one block in the middle requires redoing the work for all of the blocks after it, since the previous block's hash is included in the block's header.

Computational work also verifies the correct blockchain to follow. The longest chain has the most computational work invested in it, which verifies its authenticity and replaces any competing chains.

Because of this, if an attacker wishes to modify a past block, they would have to redo the work for that block and all subsequent blocks, *and* surpass the work of all other honest nodes mining blocks. So long as the attacker does not control the majority of computational work in the blockchain, they cannot change the blockchain. We refer to this situation as the **51% attack**.

The target (and subsequently **difficulty** of proof-of-work) increases as block generation increases, to compensate for increasing hardware speed and number of nodes.

5 Network

Nodes are miners working on creating blocks for Bitcoin. The network is maintained as follows:

1. New transactions are broadcast to all nodes running Bitcoin
2. Each node collects new transactions into a block
3. Each node mines the block through proof-of-work
4. When a node satisfies the proof-of-work, the block is added to the blockchain for all nodes
5. The block is accepted only if all transactions are valid and not spent
6. Nodes "accept" the block by working on creating the next block while using the new block as the previous hash

If two nodes braodcast different blocks at the same time, the other nodes will work on a blockchain used with the node they receive first, but save the other block as a separate branch. Whichever branch of the chain grows faster first is used as the correct blockchain. This is known as **consensus**.

6 Incentive

The first transaction in a block is a "fee" or "miner's reward"—a new coin given to the creator of the block. This gives nodes incentive to mine blocks and maintain the Bitcoin network.

The incentive also helps encourage honest nodes. If an attacker has more than 50% of computational power in the network, they may find it more profitable to mine rather than to steal payments.

7 Reclaiming Disk Space

Transactions in a block are hashed into a **Merkle Tree** with only the **root** included in the block's header to save space.

A **Merkle Tree** is a binary tree where every leaf is the hash of a transaction, and every non-leaf is the hash of its children (hash of hashes).

8 Simplified Payment Verification

To verify payments, a user only needs block headers, not the entire blockchain. By first checking that they have the longest chain, the user finds the Merkle branch linked to the timestamped transaction, which would confirm that the transaction has been accepted.

If an attacker were to modify the chain, they could get away with it as long as they outpace the rest of the honest nodes on the network.

Businesses should run their own nodes for security and easy payment verification.

9 Combining and Splitting Value

Every **transaction** actually contains multiple inputs and outputs. There can only be a maximum of two outputs: one for the payment, and one returning any change to the sender. However, multiple inputs can be combined.

10 Privacy

Although blocks are announced publicly, so long as public keys stay anonymous, privacy can be maintained. As an additional measure, a new key pair should be generated for each transaction. Bitcoin uses the Elliptic Curve Digital Signature Algorithm (ECDSA) to generate key pairs.

11 Calculations

See the real white paper for this section.

12 Conclusion

Bitcoin is a system for electronic transactions without relying on trust. We start with the usual framework of digital signatures, then apply the proof-of-work system to prevent double-spending. So long as honest nodes control the majority of CPU power, attacker will not be able to modify the blockchain. Bitcoin, at heart, modifies the warm, intagible notion of trust with the cold pragmatism of computational work.