Bitcoin Blockchain

Zichao Yang

Zhongnan University of Economics & Law

Date: November 8, 2024

Introduction

In a centralized ledger system, the ledger tracks which coins belongs to who.

In a decentralized system like bitcoin, how to make sure that the owner does not spend his same bitcoin twice?

This is called the *double-spending problem*. Then how can we solve the problem?

Introduction

Solution 1: we could order transactions one at a time, nodes on the network should agree on where transaction n should be before we move on to $n+1$. This method would cause a lot of transmission overhead.

Solution 2: we could order large batches of transactions, maybe once per day. This method would reduce overhead but your transaction may have to wait for a whole day before it gets confirmed.

Solution 3: Bitcoin settles batches of transactions every 10 minutes. These batches of transactions are what we call **blocks**.

In this lecture we'll see what is a Bitcoin blocks, the **blockchain** built on blocks, and the consensus mechanism **proof-of-work** that prevents double spending.

Block Components

Block: A block is defined as a storage unit which contain confirmed transaction data.

Block header: A block header is part of a block, and it contains three sets of block metadata:

- the location of last block
- metadata related to mining
- merkle tree root

Block Components

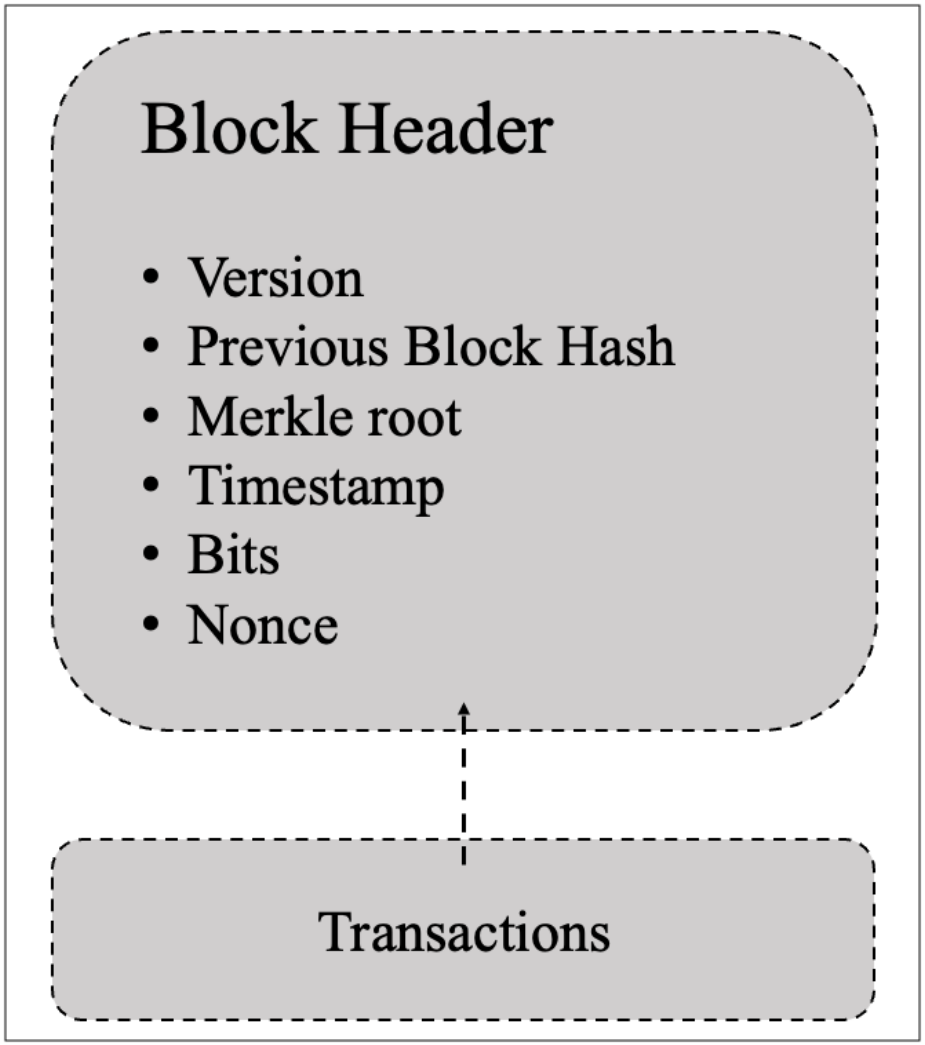


Figure 1: The structure of a bitcoin block

Block Components: Version

The version field identifies the version of the protocol that was used to generate the block.

BIP0009 introduced a signaling mechanism for activating soft forks within the Bitcoin network through changes in the block version field.

- **Threshold for Activation**: BIP0009 allows each soft fork to specify an activation threshold (e.g., 95% of blocks within a specific time period). If the threshold is met, the change becomes active, allowing a gradual, miner-supported upgrade.
- **Multiple Proposals**: By using specific bits for each soft fork proposal, BIP0009 allows multiple upgrades to be signaled and activated independently, without affecting other network operations.

Block Components: Previous Block Hash

All blocks have to point to a previous block. This is why the data structure is called a **blockchain**.

Blocks link back all the way to the very first block, or the *genesis block*. And in the genesis block, the previous block hash is filled with 0.

Source: Antonopoulos, Andreas M., and David A. Harding. *Mastering bitcoin*. O'Reilly Media, Inc., 2023.

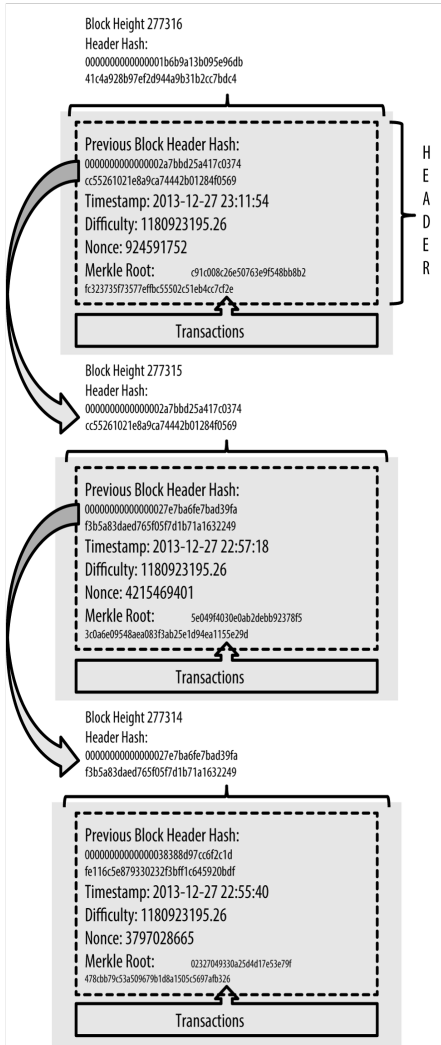


Figure 2: The Blockchain

Block Components: Merkle Root

The Merkle root encodes all the ordered transactions in a 32-byte hash.

In a *merkle tree* the transactions are arranged in pairs and the hash values (`SHA256d`) of each pair are calculated.

Whenever a merkle tree has an odd number of elements, the last element is duplicated and the hash value of the two identical elements computed.

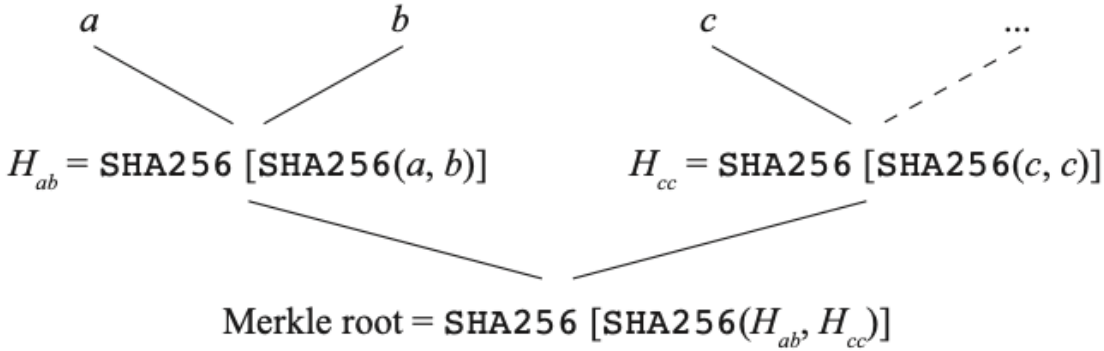


Figure 3: Merkle Tree

Source: Schar, Fabian, and Aleksander Berentsen. *Bitcoin, Blockchain, and cryptoassets: a comprehensive introduction*. MIT press, 2020.

Block Components: Merkle Root

The merkle root's purpose is to guarantee the integrity of the transactions included in the block.

If only a single transaction in the tree is altered, the merkle root will assume a totally different value.

Merkle tree also allows efficient verification of specific transactions. With N transactions, a maximum of $2\log_2(N)$ computational steps are needed to verify whether a transaction is an element of the merkle tree.

Block Components: Timestamp

To prevent manipulation, Bitcoin imposes constraints on the block timestamp:

- It must be greater than the median of the timestamps of the previous 11 blocks.
- It cannot be more than two hours in the future relative to the network's current adjusted time, known as the "network time".

Timestamp serves two purposes:

- validate timestamp-based locktimes on transactions
- calculate a new bits/target/difficulty every 2016 blocks.

Timestamp filed is 4 bytes long, so we will run out of timestamp after $2^{4*8} - 1$ seconds, roughly 136 years after 1970. Need to fix this issue before 2106.

Block Components: Bits & Nonce

Bits is a field that encodes the proof-of-work necessary in this block. This field contains two different numbers: **exponent** and **coefficient**.

Based on these two numbers, we can calculate a new number called **target**, which is used in the **proof-of-work** consensus mechanism.

$$\textbf{Target} = \textbf{coefficient} \times 256^{\textbf{exponent}-3}$$

Some books simply treat *bits* and *target* these two concepts interchangeable.

Nonce stands for "number used only once," or n-once. This number is what is changed by miners when looking for proof-of-work.

Identify a Block in the Blockchain

Block height identifies the position of a given block within the chain. The first block (i.e., genesis block) is assigned the height of 0. It serves as the id of a block.

Block header hash (a.k.a., **block hash**): can also serve as the id of a block. *Note*: block N's hash is not included in block N, but in block $N+1$.

Block depth refers to a block's distance from the most recent block. The block depth is important for security analyses.

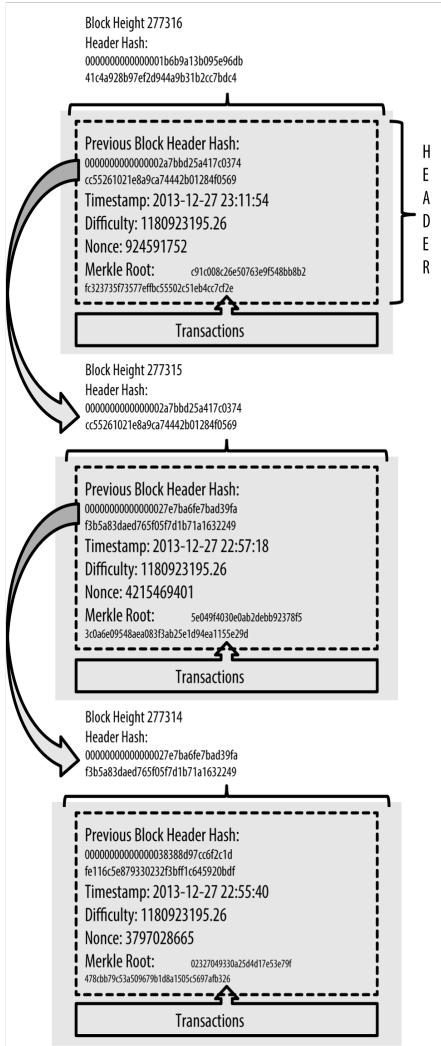
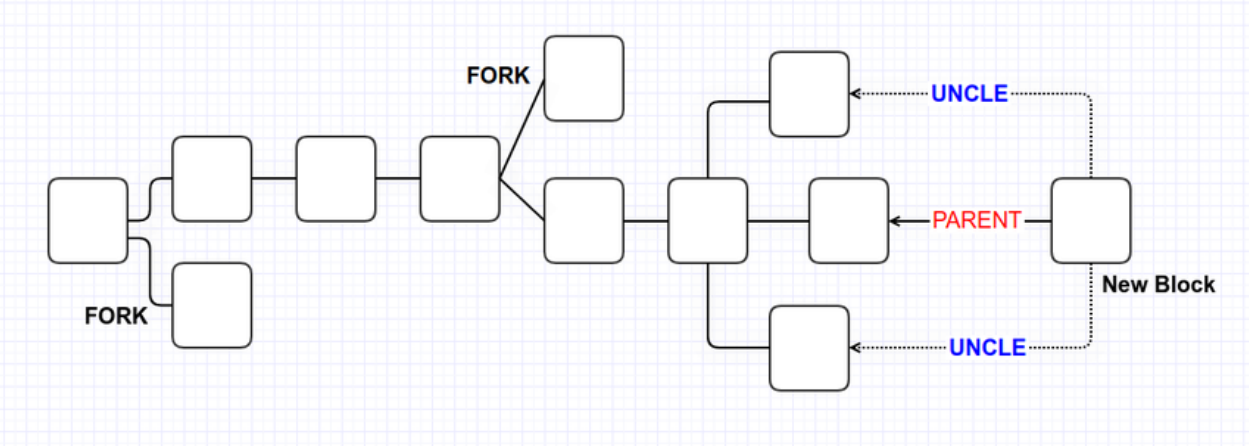


Figure 4: The Blockchain

Extend the Chain

To extend the blockchain, miners create new candidate blocks. They may use any subset of transactions from their mempool, compute the merkle root, and assemble new candidate blocks.

Now think about the following two problems:
Q1: which block should be linked?

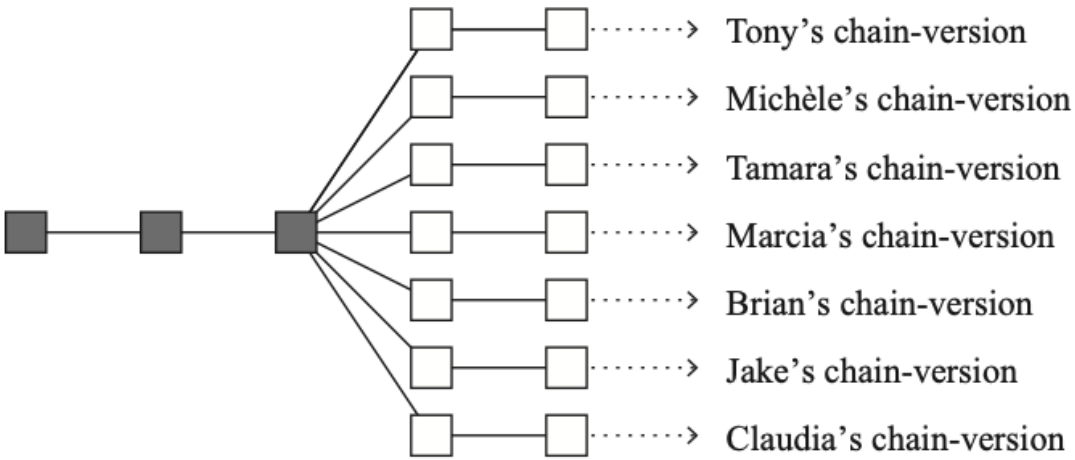


Source: Goncharov, Sergey, and Andrey Nechesov. *Axiomatization of Blockchain Theory.* Mathematics 11.13 (2023): 2966.

Extend the Chain

A1: the **longest chain**: the longest chain is the legitimate one, new block will be linked to the longest chain.

Q2: what if many miners generate the new block at the same time?



Source: Schar, Fabian, and Aleksander Berentsen. *Bitcoin, Blockchain, and cryptoassets: a comprehensive introduction*. MIT press, 2020.

A2: We will discuss it in the next section...

Proof-of-Work

Proof-of-work is a consensus algorithm that secures a blockchain network by requiring miners to solve computationally intensive puzzles.

Proof-of-work was originally developed to counteract spam email messages and DoS attacks.

The fundamental idea is that if an activity can be executed too quickly, it should be slowed down.

This idea is achieved by requiring the activity to present the solution to a time-consuming mathematical problem, which can be only solved by using a trial-and-error method.

Proof-of-Work

The proof-of-work process has several effects:

- It solves the problem that blocks are generated too quickly.
- It ensures that the chain cannot be altered and recreated without incurring costs.
- It induces the nodes to behave honestly, as the costly hashing power that they invest would otherwise be wasted.

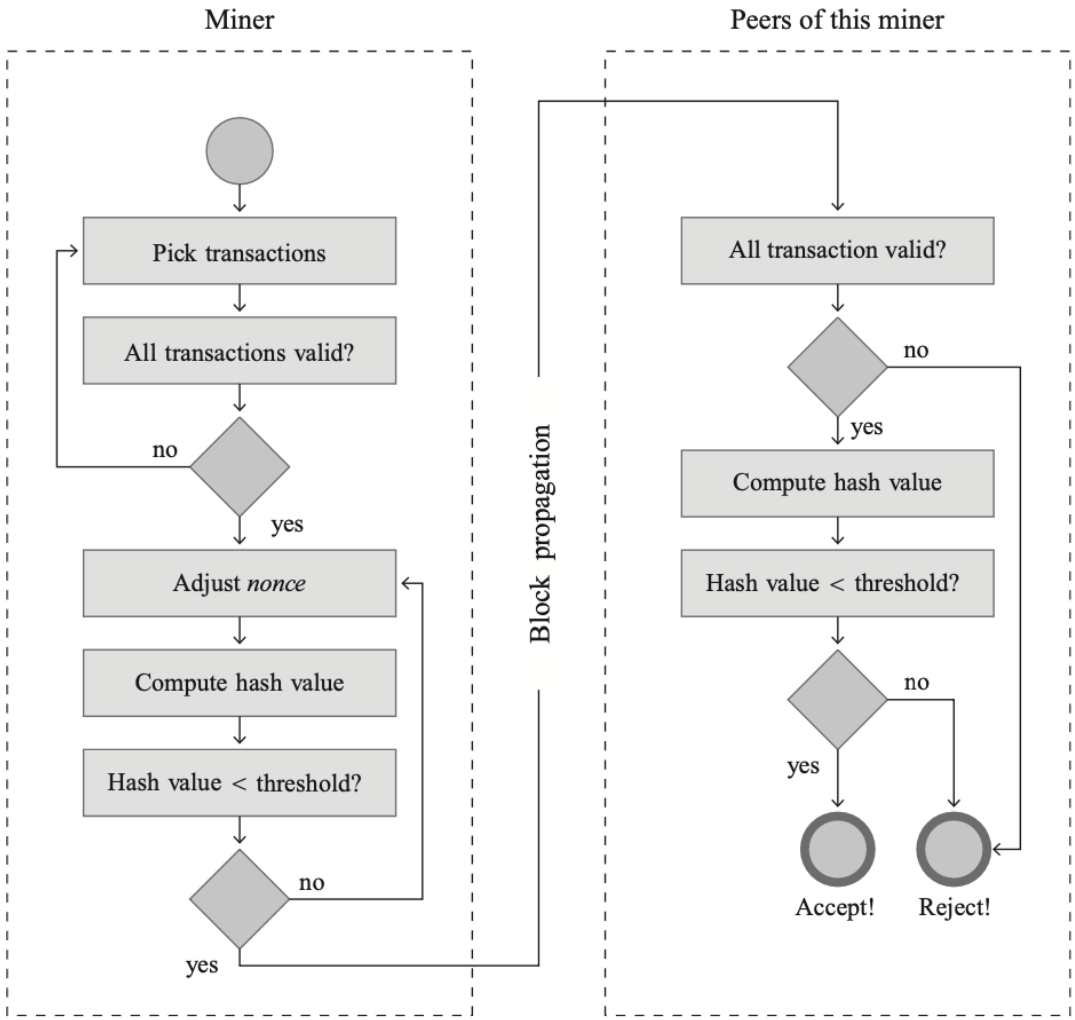
What is mining?

Block header hash:
0000000000000000000ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f

Mining: The action of hashing the block header repeatedly. In each time, the miner change the value of **nonce** a little bit until the block header hash satisfies the **target** set by the **Proof-of-Work** algorithm.

Example: Block 570,072

What is mining?



Source: Schar, Fabian, and Aleksander Berentsen. *Bitcoin, Blockchain, and cryptoassets: a comprehensive introduction*. MIT press, 2020.

Why mining works?

Difficult to find the nonce, but easy to verify the finding

Makes the blockchain almost immutable (with mining competition)

Auto-adjustable, generates blocks, on average, every 10 minutes

Q: how does the mining action make the blockchian almost immutable? Can it prevent 51% attach?

More on the nonce

Since the nonce field only has 4 bytes (32 bits), there are only 2^{32} possible nonces that a miner can try.

However, modern ASIC equipment can calculate way more than 2^{32} different hashes per second, so what else can miners change?

- change the coinbase transaction
- change the Merkle root
- minor Timestamp adjustments
- modify the version field

More on Target and Difficulty

Target: We have mentioned that the **Target** is calculated based on **Bits**. Target sets a number that must be greater or equal to the block hash in the new block. Otherwise, the new block is not valid.

Difficulty: Intuitively, you can think of the difficulty as related to the number of leading zeros in the hash target. The more leading zeros required in the hash, the harder it is to find a hash that meets the target, thus increasing the difficulty.

However, if you want to calculate the exact difficulty value, which is more precise than simply counting zeros, you can calculate it based on the Target value:

$$\text{Difficulty} = 0xffff \times 256^{0x1d-3}/\text{Target}$$

Difficulty Adjustment

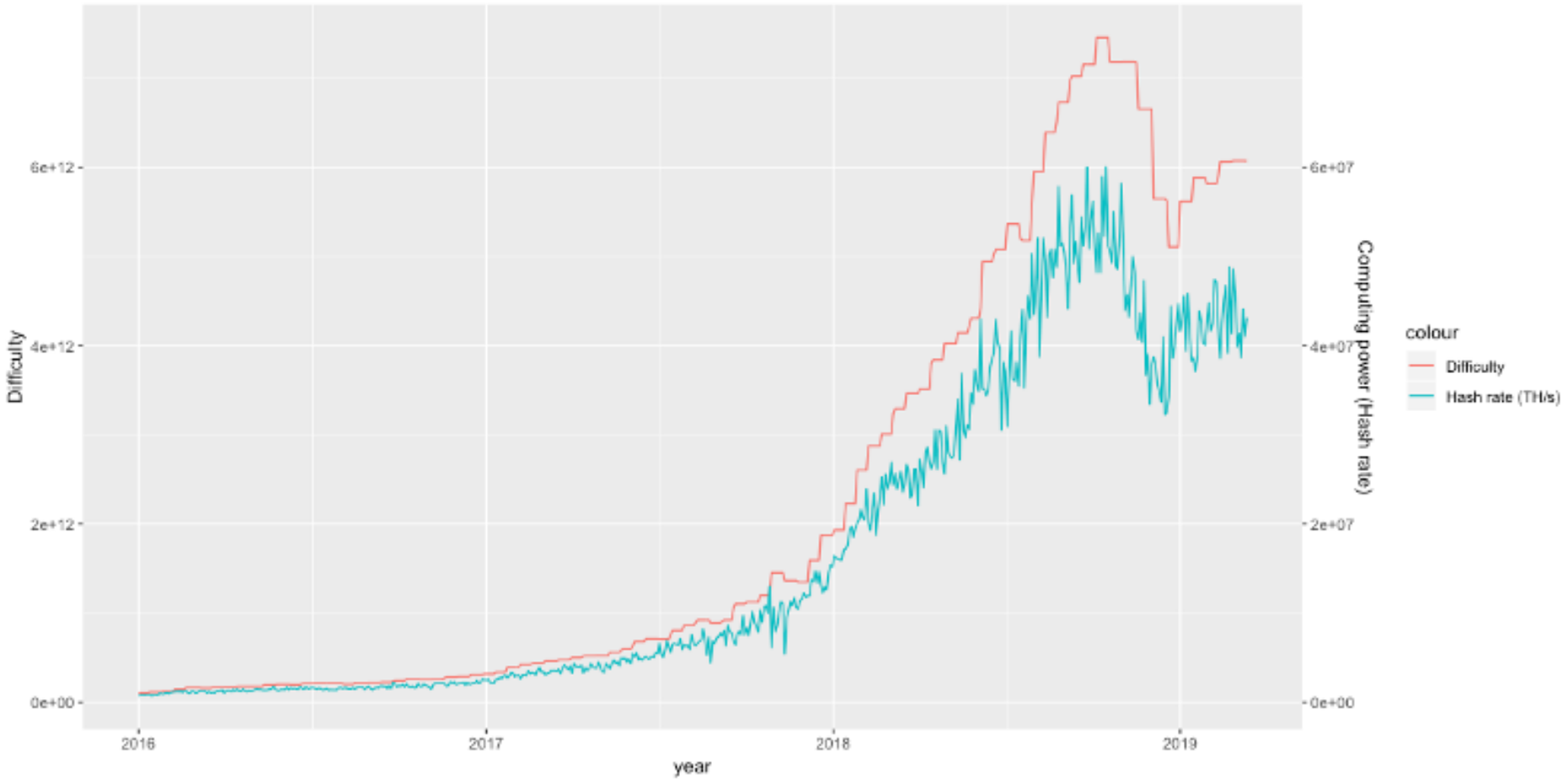
Every 2016 blocks, all bitcoin nodes retarget the Proof-of-Work:

- If actually time $>$ 20,160 minutes, reduce the difficulty
- If actually time $<$ 20,160 minutes, increase the difficulty

Difficulty adjustment mechanism makes sure that block are mined, on average, every 10 minutes.

Difficulty adjustment is asymmetrical.

Asymmetrical adjustment of difficulty



Mining Pools & Mempools

Mining Pool: Miners pool their hashing power together to mine a block and share the Coinbase reward.

Q: why does mining pool exist?

Mempools: A mempool is where all unconfirmed transactions stay in the bitcoin network.

Mempools are maintained by miners themselves.

The Mempool size indicates how congested the bitcoin network is.

The procedure of transaction confirmation