---

title: Proof-of-work (PoW) description: An explanation of the proof-of-work consensus protocol and its role in Ethereum. lang: en

---

The Ethereum network began by using a consensus mechanism that involved**Proof-of-work (PoW)**. This allowed the nodes of the Ethereum network to agree on the state of all information recorded on the Ethereum blockchain and prevented certain kinds of economic attacks. However, Ethereum switched off proof-of-work in 2022 and started using proof-of-stake instead.

Proof-of-work has now been deprecated. Ethereum no longer uses proof-of-work as part of its consensus mechanism. Instead, it uses proof-of-stake. Read more on proof-of-stake and staking.

Prerequisites {#prerequisites}

To better understand this page, we recommend you first read up ontransactions, blocks, and consensus mechanisms.

What is Proof-of-work (PoW)? {#what-is-pow}

Nakamoto consensus, which utilizes proof-of-work, is the mechanism that once allowed the decentralized Ethereum network to come to consensus (i.e. all nodes agree) on things like account balances and the order of transactions. This prevented users from "double spending" their coins and ensured that the Ethereum chain was tremendously difficult to attack or manipulate. These security properties now come from proof-of-stake instead using the consensus mechanism known as Gasper.

Proof-of-work and mining {#pow-and-mining}

Proof-of-work is the underlying algorithm that sets the difficulty and rules for the work miners do on proof-of-work blockchains. Mining is the "work" itself. It's the act of adding valid blocks to the chain. This is important because the chain's length helps the network follow the correct fork of the blockchain. The more "work" done, the longer the chain, and the higher the block number, the more certain the network can be of the current state of things.

More on mining

How did Ethereum's proof-of-work work? {#how-it-works}

Ethereum transactions are processed into blocks. In the now-deprecated proof-of-work Ethereum, each block contained:

- block difficulty – for example: 3,324,092,183,262,715
- mixHash – for example: `0x44bca881b07a6a09f83b130798072441705d9a665c5ac8bdf2f39a3cdf3bee29`
- nonce – for example: `0xd3ee432b4fb3d26b`

This block data was directly related to proof-of-work.

The work in proof-of-work {#the-work}

The proof-of-work protocol, Ethash, required miners to go through an intense race of trial and error to find the nonce for a block. Only blocks with a valid nonce could be added to the chain.

When racing to create a block, a miner repeatedly put a dataset, that could only be obtained by downloading and running the full chain (as a miner does), through a mathematical function. The dataset was used to generate a mixHash below a target that is dictated by the block difficulty. The best way to do this is through trial and error.

The difficulty determined the target for the hash. The lower the target, the smaller the set of valid hashes. Once generated, this was incredibly easy for other miners and clients to verify. Even if one transaction were to change, the hash would be completely different, signalling fraud.

Hashing makes fraud easy to spot. But proof-of-work as a process was also a big deterrent to attacking the chain.

Proof-of-work and security {#security}

Miners were incentivized to do this work on the main Ethereum chain. There was little incentive for a subset of miners to start their own chain—it undermines the system. Blockchains rely on having a single state as a source of truth.

The objective of proof-of-work was to extend the chain. The longest chain was most believable as the valid one because it had the most computational work done to generate it. Within Ethereum's PoW system, it was nearly impossible to create new blocks that erase transactions, create fake ones, or maintain a second chain. That's because a malicious miner would have needed to always solve the block nonce faster than everyone else.

To consistently create malicious yet valid blocks, a malicious miner would have needed over 51% of the network mining power to beat everyone else. That amount of "work" requires a lot of expensive computing power and the energy spent might even have outweighed the gains made in an attack.

Proof-of-work economics {#economics}

Proof-of-work was also responsible for issuing new currency into the system and incentivizing miners to do the work.

Since the Constantinople upgrade, miners who successfully create a block were rewarded with two freshly minted ETH and part of the transaction fees. Ommer blocks also compensated 1.75 ETH. Ommer blocks were valid blocks created by a miner practically at the same time as another miner created the canonical block, which was ultimately determined by which chain was built on top of first. Ommer blocks usually happened due to network latency.

Finality {#finality}

A transaction has "finality" on Ethereum when it's part of a block that can't change.

Because miners worked in a decentralized way, two valid blocks could be mined at the same time. This creates a temporary fork. Eventually, one of these chains became the accepted chain after subsequent blocks were mined and added to it, making it longer.

To complicate things further, transactions rejected on the temporary fork may not have been included in the accepted chain. This means it could get reversed. So finality refers to the time you should wait before considering a transaction irreversible. Under the previous proof-of-work Ethereum, the more blocks were mined on top of a specific block N, the higher confidence that the transactions in N were successful and would not be reverted. Now, with proof-of-stake, finalization is an explicit, rather than probabilistic, property of a block.

Proof-of-work energy-usage {#energy}

A major criticism of proof-of-work is the amount of energy output required to keep the network safe. To maintain security and decentralization, Ethereum on proof-of-work consumed large amounts of energy. Shortly before switching to proof-of-stake, Ethereum miners were collectively consuming about 70 TWh/yr (about the same as the Czech Republic - according to digiconomist on 18-July-2022).

Pros and cons {#pros-and-cons}

| Pros | Cons | | ---------- | ---------- | | Proof-of-work is neutral. You don't need ETH to get started and block rewards allow you to go from 0ETH to a positive balance. With proof-of-stake you need ETH to start with. | Proof-of-work uses up so much energy that it's bad for the environment. | | Proof-of-work is a tried and tested consensus mechanism that has kept Bitcoin and Ethereum secure and decentralized for many years. | If you want to mine, you need such specialized equipment that it's a big investment to start. | | Compared to proof-of-stake it's relatively easy to implement. | Due to increasing computation needed, mining pools could potentially dominate the mining game, leading to centralization and security risks. |

Compared to proof-of-stake {#compared-to-pos}

At a high level, proof-of-stake has the same end goal as proof-of-work: to help the decentralized network reach consensus securely. But it has some differences in process and personnel:

- Proof-of-stake switches out the importance of computational power for staked ETH.
- Proof-of-stake replaces miners with validators. Validators stake their ETH to activate the ability to create new blocks.
- Validators don't compete to create blocks, instead they are chosen at random by an algorithm.
- Finality is clearer: at certain checkpoints, if 2/3 validators agree on the state of the block it is considered final. Validators must bet their entire stake on this, so if they try to collude down the line, they'll lose their entire stake.

More on proof-of-stake

More of a visual learner? {#visual-learner}

Further Reading {#further-reading}

- Majority attack
- On settlement finality

Videos {#videos}

- A technical explanation of proof-of-work protocols

Related Topics {#related-topics}

- Mining
- Proof-of-stake