---

title: Mining algorithms description: A detailed look at the algorithms used for Ethereum mining. lang: en

---

Proof-of-work is no longer underlying Ethereum's consensus mechanism, meaning mining has been switched off. Instead, Ethereum is secured by validators who stake ETH. You can start staking your ETH today. Read more on The Merge, proof-of-stake, and staking. This page is for historical interest only.

Ethereum mining used an algorithm known as Ethash. The fundamental idea of the algorithm is that a miner tries to find a nonce input using brute force computation so that the resulting hash is smaller than a threshold determined by the calculated difficulty. This difficulty level can be dynamically adjusted, allowing block production to happen at a regular interval.

Prerequisites {#prerequisites}

To better understand this page, we recommend you first read up onproof-of-work consensus and mining.

Dagger Hashimoto {#dagger-hashimoto}

Dagger Hashimoto was a precursor research algorithm for Ethereum mining that Ethash superseded. It was an amalgamation of two different algorithms: Dagger and Hashimoto. It was only ever a research implementation and was superseded by Ethash by the time Ethereum Mainnet launched.

Dagger involves the generation of a Directed Acyclic Graph, random slices of which get hashed together. The core principle is that each nonce only requires a small portion of a large total data tree. Recomputing the subtree for each nonce is prohibitive for mining - hence the need to store the tree - but okay for a single nonce's worth of verification. Dagger was designed to be an alternative to existing algorithms like Scrypt, which are memory-hard but difficult to verify when their memory-hardness increases to genuinely secure levels. However, Dagger was vulnerable to shared memory hardware acceleration and dropped in favor of other avenues of research.

Hashimoto is an algorithm that adds ASIC-resistance by being I/O bound (i.e. memory reads are the limiting factor in the mining process). The theory is that RAM is more available than computation; billions of dollars worth of research have already investigated optimizing RAM for different use cases, which often involve near-random access patterns (hence "random access memory"). As a result, existing RAM is likely to be moderately close to optimal for evaluating the algorithm. Hashimoto uses the blockchain as a source of data, simultaneously satisfying (1) and (3) above.

Dagger-Hashimoto used amended versions of the Dagger and Hashimoto algorithms. The difference between Dagger Hashimoto and Hashimoto is that, instead of using the blockchain as a data source, Dagger Hashimoto uses a custom-generated data set, which updates based on block data every N blocks. The data set is generated using the Dagger algorithm, allowing for efficiently calculating a subset specific to every nonce for the light client verification algorithm. The difference between Dagger Hashimoto and Dagger is that, unlike in the original Dagger, the dataset used to query the block is semi-permanent, only being updated at occasional intervals (e.g. once per week). This means that the portion of the effort of generating the dataset is close to zero, so Sergio Lerner's arguments regarding shared memory speedups become negligible.

More on Dagger-Hashimoto.

Ethash {#ethash}

Ethash was the mining algorithm that was actually used on the real Ethereum Mainnet under the now deprecated proof-of-work architecture. Ethash was effectively a new name given to a specific version of Dagger-Hashimoto after the algorithm got significantly updated, whilst still inheriting the fundamental principles of its predecessor. Ethereum Mainnet only ever used Ethash - Dagger Hashimoto was an R&D version of the mining algorithm that was superseded before mining started on Ethereum mainnet.

More on Ethash.

Further reading {#further-reading}

Know of a community resource that helped you? Edit this page and add it!