Comparing Major Mining Algorithms

Author	josh
Date	2019-10-17 20:00:38

Overview

As it stands, all of the top cryptocurrencies (Bitcoin Cash, Ethereum, Litecoin, and Bitcoin) use proof-of-work mining to secure their networks. With proof-of-work, special nodes on the network called miners use their computing power to try and solve a mathematical problem. This problem is designed so that a miner has to do a bunch of guessing to get the answer, but anyone else can verify that answer very quickly.

The general idea across proof-of-work variations is the same: miners have to guess a bunch to get an answer, essentially proving that they've done some amount of work. And the collective amount of computing power on the network makes pulling off fraud impossibly hard. However, there are some different variations of these proof-of-work algorithms used in different cryptocurrencies. Let's take a high-level look at these variations and how they achieve the same goal in different ways.

Mining Algorithm Variations

Bitcoin Cash/Bitcoin Mining with SHA-256

The original mining algorithm used in a cryptocurrency is the fairly straightforward SHA-256 used by Bitcoin. This mining algorithm solves a simple problem: given some block data, add a random number called a "nonce", and run that through the SHA-256 hashing algorithm. This *one way* cryptographic hash outputs a very large number (256 bits if you're curious), and that number has to be less than a *difficulty target number* for the problem to be "solved" with that nonce. With a simple toy algorithm (8 bits) - a solution might look something like this:

```
0 0 1 0 0 0 0 0 - Difficulty target value

1 0 0 1 1 1 1 1 - Guess #1 is not valid - greater than target
...
...
...
...
0 0 0 0 1 0 1 0 - Guess N is a valid hash - less than target
```

This algorithm is a clean and simple one. Guess a number, hash the data, and hope the resulting *block hash* is less than the *difficulty target*.

However, a disadvantage of this algorithm is in the equipment needed to contribute to mining on the network. SHA-256 mining is a hard *computing* problem, but that's all its limited by. As the Bitcoin network has adjusted the difficulty target over time, profitable mining has become limited to specialized computing devices called ASICS - Application Specific Integrated

Circuits. It's not profitable or feasible for a single user like you or I to mine Bitcoin on a single device like a PC anymore - it's the world of specialized companies and mining pools. This can be considered a problem of centralization, as less everyday users can participate in this part of securing the network.

Litecoin Mining with Scrypt

One of the first major forks of the Bitcoin codebase resulted in the popular currency Litecoin, which made changes to the mining algorithm in an attempt to solve this problem of a high barrier of entry for mining. Litecoin uses an alternative hashing algorithm called *Scrypt* in place of SHA-256. Scrypt is actually considered a key-derivation function rather than a pure hashing function, although the end goal is roughly the same: a one-way function that takes some data and outputs some bits that are the same every time for a particular input.

The different with a key-derivation function or specialized hashing algorithm like this is that they're designed to be more computationally difficult that algorithms like SHA-256. Scrypt is *memory hard*, meaning that the algorithm is more limited by the available memory in the system than by the computing power.

For key derivation, this is great because it's hard to do brute force attacks on a database of keys - in other words, it's hard to guess what the original password was. For our mining algorithm, it's great because ASICs don't really give miners an advantage. This makes mining easier for folks that only have access to devices like GPUs, and prevents some of the mining centralization and barrier to entry that's seen with SHA-256 mining.

Ethereum Mining with Ethash

Ethereum mining follows a similar model to Litecoin - it was designed to prevent mining centralization. However, Ethereum goes further than simply using a memory-hard key derivation function or something of that nature. Ethereum uses its own memory-hard algorithm for mining called *Ethash*, custom designed by its creators.

Ethash is based on an algorithm called *Dagger-Hashimoto* used to make mining a memory-hard problem. Every N blocks or so, a large dataset is generated using the block data as a "seed". The *Dagger* part of the algorithm was designed by Ethereum's creator Vitalik Buterin to make mining memory hard, but make verifying the answer relatively easy for non-mining nodes on the network. The *Hashimoto* part was designed by Thaddeus Dryja to make a memory-hard hashing problem. Combining these concepts into *Ethash* makes a mining algorithm that's less prone to requiring specialized hardware over time than SHA-256 mining.

Mining Variations - Same Idea, Different Requirements

The overall problem in proof-of-work mining is the same across currencies and algorithm variations - a mining node must expend resources to guess a bunch and find an answer to the problem. However, there are a variety of ways in which problem those miners solve can be constructed.

Some variations like SHA-256 are simple, but prone to centralization and specialized hardware requirements over time. Other like Litecoin and Ethereum take a different approach, desiring to make mining a more equitable process across the network at the expense of some complexity.

Regardless of the approach, proof-of-work mining allows lots of individuals to come together and create a peer-to-peer network of money, without the need to trust any one central "clearing house" to process transactions. Proof-of-work mining makes pulling off fraud a difficult or impossible endeavor, so that these currencies remain globally decentralized and secure.