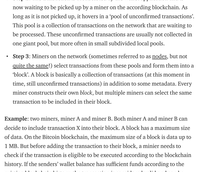
1.1K

★ Member preview

Jimi S. Follow
Areas of interest: Financial technology, biotechnology, blockchain, durable energy solutions, traditional stock markets and other financial markets.
May 3 · 7 min read

# Blockchain: how mining works and transactions are processed in seven steps

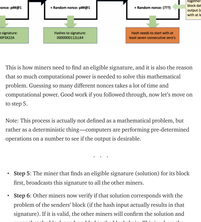
Have you ever wondered how the mining process on a blockchain works, or how your transaction gets confirmed and is added the blockchain? Well, so have I. And since I couldn't find any clear step by step explanation of this process, I decided to dig into it and write a guide myself. Here is how a blockchain transaction is processed from your wallet into the blockchain, in seven steps.




- **Step 1**: A user signs off on a transaction from their wallet application, attempting to send a certain crypto or token from them to someone else.
- **Step 2**: The transaction is broadcasted by the wallet application and is now waiting to be picked up by a miner on the according blockchain. As long as it is not picked up, it hovers in a 'pool of unconfirmed transactions'. This pool is a collection of transactions on the network that are waiting to be processed. These unconfirmed transactions are usually not collected in one giant pool, but more often in small subdivided local pools.
- **Step 3**: Miners on the network (sometimes referred to as nodes, but not quite the same!) select transactions from these pools and form them into a 'block'. A block is basically a collection of transactions (at this moment in time, still unconfirmed transactions) in addition to some metadata. Every miner constructs their own block, but multiple miners can select the same transaction to be included in their block.

**Example:** two miners, miner A and miner B. Both miner A and miner B can decide to include transaction X into their block. A block has a certain size of data. On the Bitcoin blockchain, the maximum size of a block is data up to 1 MB. But before adding the transaction to their block, a miner needs to check if the transaction is eligible to be executed according to the blockchain history. If the senders' wallet balance has sufficient funds according to the existing blockchain history, the transaction is considered valid and can be added to the block. Miners will usually prioritise transactions that have a high transaction fee set, because this gives them a higher reward.

- **Step 4**: By selecting transactions and adding them to their block, miners create a block of transactions. To add this block of transactions to the blockchain (to have all other nodes and miners register the transactions), the block first needs a signature. This signature is created by solving a very complex mathematical problem that is unique to each block of transactions. Each block has a different mathematical problem, meaning each miner will work on a different problem that is unique to the block they built, but all of these problems are equally hard to solve. In order to solve this mathematical problem, a lot of computational power is needed (and thus a lot of electricity). This is the process referred to as **mining**. If you want to know more about how this works exactly, please continue reading below, otherwise skip to step 5.



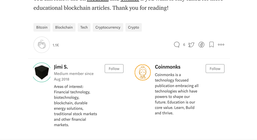
### Mining aka hashing (Proof of Work algorithm)

The mathematical problem every miner is facing when trying to add a block to the blockchain is to find a **hash output (aka signature)** for their data in its block, that starts with a certain amount of consecutive zero's. That sounds complicated, right? But it is really not that hard. Let me try to explain this to you in a simple way.

Before we proceed, it is important to know what a *hash function* is. No worries, I will not go into too much technical detail. A hash function is simply put a mathematical problem that is very **hard to solve**, but where the answer is very **easy** to verify.

A hash function takes an input string of numbers and letters (literally any string of random letters, numbers and/or symbols), and turns it into a new 32 *digit string* existing out of **random** letters and numbers. This 32 digit string is the **hash output**. If any number or letter in the input string is changed, the hash output will also change randomly. However, the same string of input will **always** give the same string of output.

Now consider the data inside a block to be the **hash input** (a string of data). When this input is hashed, it gives a **hash output** (32 digit string). A rule of the Bitcoin blockchain is that this output string needs to start with a consecutive amount of zero's in order to be eligible as a block signature. This is what every miner is looking for when trying to add a block to the blockchain; an output string that starts with a certain amount of zero's. But what if the data string of the block doesn't hash into an output string that starts with that amount of zero's? Well, this is why miners repeatedly change a part of the data inside their block called the **nonce**. Because the nonce changes all the time, the input data for the hash function also changes, leading to *different hash outputs*. Eventually, the miner hopes to find an input string (string of block data and the nonce) that hashes to an eligible output string (that starts with an amount of zero's). The example below uses seven zero's, but this amount of zero's readily depends on the block difficulty on a blockchain. Don't click that if your not ready for it:




This is how miners need to find an eligible signature, and it is also the reason that so much computational power is needed to solve this mathematical problem. Guessing so many different nonces takes a lot of time and computational power. Good work if you followed through, now let's move on to step 5.

Note: This process is actually not defined as a mathematical problem, but rather as a deterministic thing—computers are performing pre-determined operations on a number to see if the output is desirable.

• • •

- **Step 5**: The miner that finds an eligible signature (solution) for its block first, broadcasts this signature to all the other miners.
- **Step 6**: Other miners now verify if that solution corresponds with the problem of the senders' block (if the hash input actually results in that signature). If it is valid, the other miners will confirm the solution and agree that the block can be added to the blockchain. This is where the definition 'proof of work' comes from. The miner that finds a solution sends his 'proof of work', aka the solution, to the other miners, and they in their turn verify if the solution is legitimate. If it is, then other miners will agree and 'consensus' on the blockchain is reached. The block can now be added to the blockchain, and is broadcasted to all other nodes on the network along with its signature. The other nodes will accept the block and save it to their transaction data as long as the transactions inside the block corresponded correctly with the current wallet balances (transaction history) at that point in time.
- **Step 7**: If the majority of the miners reaches consensus, the block gets added to the blockchain. Every time another block gets added on top of this block, it counts as another '*confirmation*' for the block beneath it. For example, if my transaction is included in block 502, and the blockchain is 507 blocks long, it means my transaction has 5 confirmations (507–502). This is also what Etherscan is referring to when showing you your transaction details. The more confirmations your transaction has, the harder it is for attackers to alter it. When a new block is added to the blockchain, all miners will have to start over again at step three by forming a new block of transactions. Miners cannot continue (well, they can, but that is quite irrelevant) mining aka solving the problem of the block they were working on because of two reasons.

One: it may contain transactions that have been confirmed by the last block that was added to the blockchain and therefore some of these transactions may now be invalid, making the block invalid as a whole, and two: every block needs to add the hash output of the *last block* that was added to the blockchain into *their metadata*. This is what makes it a blockchain. If a miner keeps mining the block they were already working on, other miners will notice that the hash output does not correspond with that of the latest added block on the blockchain, and will therefore reject the block.

• • •

Was this article helpful? Help others find it by applauding or sharing. You can read any of my other short blockchain articles of:

**Beginner 1: How blockchain works in 7 steps**

**Beginner 2: How mining works and how transactions are processed**

**Beginner 3: How a hacker performs a 51% attack**

**Beginner 4: Nodes and masternodes**

**Beginner 5: Mining difficulty and block time**

You can follow me on Medium and Twitter if you want to stay tuned for more educational blockchain articles. Thank you for reading!

Bitcoin Blockchain Tech Cryptocurrency Crypto

1.1K 4

Jimi S. Follow
Medium member since Aug 2018
Areas of interest: Financial technology, biotechnology, blockchain, durable energy solutions, traditional stock markets and other financial markets.

Coinmonks Follow
Coinmonks is a Medium-based publication embracing all technologies which have powers to shape our future. Education is our main priority. Build and thrive.



Responses

Write a response…

Conversation with Jimi S.

Sumit V
May 29 · 1 min read

Wonderfully explained. That's what i was looking out for. After hunting almost a week, how transactions are validated, i stumble upon your post and got answers.

Request, if you also explain in detail about how security is been maintained while carrying out transaction using public, private key and digitally signed…

Read more…

2 responses

Jimi S.
May 31 · 1 min read

Hi Sumit,

Thank you, glad to be of help. Unfortunately I won't be able to answer your further questions, simply because I do not know the answers. The only thing I do know though, is that most nodes run a full copy of the blockchain (you need to download it when running a node), which allows them to check if transactions are…

Read more…

1

Conversation with Jimi S.

Massimo Franceschet
Jun 15

Clear and instructive. However, what prevents a miner to modify a transaction in the block he is mining? For instance, if there exists a transaction 'send from A to B a value of 1', where A is the miner, what prevents the miner to replace (and digitally sign) it with 'send from A to B a value of 0.1'?

1 response

Jimi S.
Jun 16

Thanks. The signature would no longer match the data of the transaction, and thus it would require a new signature from the according wallet. For that, access to private key would be needed, and miner doesn't have that.

Conversation with Jimi S.

Yahel
Aug 20

Miners on the network (often referred to as nodes)

A miner is not a node

1 response

Jimi S.
Aug 20

Correct, a miner needs to run a node, but it is technically not a node. I corrected this. Thanks for pointing out.

Show all responses