Note: I in no way own any of the knowledge I have written in this document. This is just my one stop reference when I need it. I have made this as a part of learning blockchain technology. I am following the blockchain at Berkeley's course as a base along with many other references wherever I get stuck. Thanks to all the people in the references who've helped me learn blockchain.

Types of users in bitcoin:

Not every client is a miner
What if I don't have a powerful computer?

Not every client has the entire blockchain (160+ GB)
What if I just want to send bitcoins with my phone?

Not every client is directly connected to the network
What if I don't need to make regular transactions?

Not every client has a wallet
What if I have a separate wallet client?

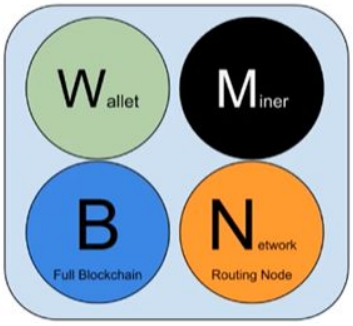


Image source: Mastering Bitcoin

Wallets:

Doesn't have the entire blockchain downloaded, and doesn't need fancy functionality. Just a wallet to help manage keys, and this type of user can send and receive bitcoins with the rest of the network. List transactions for you. Bitcoin wallets don't actually store bitcoins.

To secure our **identity**, we need to secure our **private key**

How do we manage all of our keys? With wallets!

Hot wallets are connected to the internet and cold storage are not. Coinbase.com is an example of web based hot wallet.

Cold storage websites ask to download the entire website for maximum security, people use an old instead of smart printer to avoid any cache, they use a completely new computer for a new operating system. There are also hardware wallets who are completely away from the internet. There are also brain wallets immune to any kind of loss. Memorize the phrases, whose hash would give us the private key. While it's easy to guess phrases (dictionary attack), people should use different and not easy to guess phrases.

Key stretching:

Hash your brain wallet a LARGE number of times. Hacking a brain wallet that has been key stretched is exponentially harder to brute force than a brain wallet that is only hashed once. Choosing a wallet brings a tradeoff between convenience and security. Eg: coinbase.com stores all our wallets on the cloud and never tells us our private key (less secure), other wallets don't store them on cloud, you are responsible for your private key.

Bitcoin ATMs:

You scan your QR code for bitcoin address, put the cash in it, and you get your bitcoin.

Wallet Mechanics:

Simple Payment Verification(SPV):

How come when we use our wallet software, we don't have to download the entire bitcoin blockchain? We can't afford to download the entire blockchain on our phones. SPV! It's a method of verifying if our particular transactions are included in a block by only having to download the blockheaders of each block, instead of the entire block, which includes all the transaction. In the bitcoin smartphone wallet apps, we don't have the entire blockchain stored, but a wallet to keep track of our keys and a network routing component, that allows us to connect to the bitcoin p2p protocol. All of this is done without a full blockchain, thus clients that run SPV are called lightweight or thin clients.

In SPV, since we have the block headers, we can run a merkle proof of inclusion, to verify if the transaction is valid, just that we have to assume that the block headers we get are not fake otherwise someone might double spend us. As long as we connect to many different nodes, then the probability of each one being controlled by one malicious entity is significantly lowered. And as long as we are connected to a variety of nodes, in the long term, the chain we see will be honest: a core assumption that we make in Bitcoin to begin with.

MultiSignature:

When, 3 people share the wallet, the point of failure is distributed. If I and my family create a 4 of 5 multisig, even by any chance if I lose my private key, I can still claim my funds because we have the other 4 signatures, which are required in the transaction. Some companies also do this. They provide a wallet/exchange for 2-of-3 or anything like that multisig services. In a 2-of-3, we can keep two of our keys and the trusted third party holds the third. The exchange can't use our funds, because they only have 1 sig, while it requires 2. However if we lose one of our keys, the company can provide us one we have kept with them.

KEY GENERATION PRACTICES

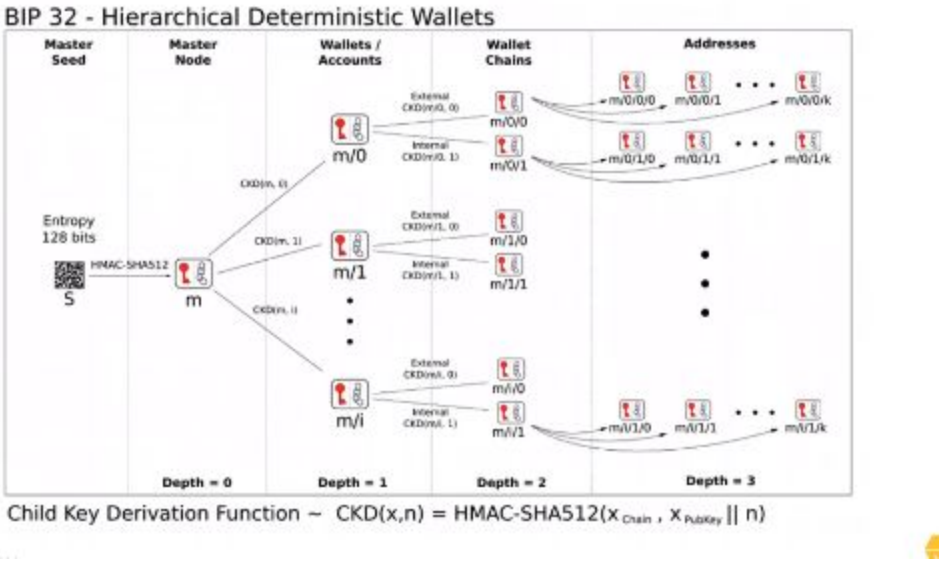
- Best practice is to never reuse pseudonyms
- Why?
  - Someone should not be able to determine how much bitcoin you own
  - Compromising one key is independent of the other ones
  - Keys are computationally easy to generate anyways
- Wallet software will handle this

JBOK Wallets (Just a Bunch Of Keys):

- **JBOK (Just a Bunch Of Keys)**
  - New backup required for every new key pair
  - Or, generate a bunch of keys when first started
  - Not too convenient because you have to store every key pair

HD (Hierarchical Deterministic) Wallets:

You could store a master seed, and for every transaction append something to the master seed and hash it in order to get the key for transaction. Thus wallets won't have to save many private keys. Eg, if for third transaction, you append 3 to your master seed and hash it and so on. The master seed is the parent key and the derived ones are child, which further on can be used again to derive for more transactions. Exchanges use these as they have to deal with a lot of users.



However, in this organisation, president with the master key would have the most power. Everyone's key originates in some way from the president's master key, so the president would be able to spend from everyone's wallet.

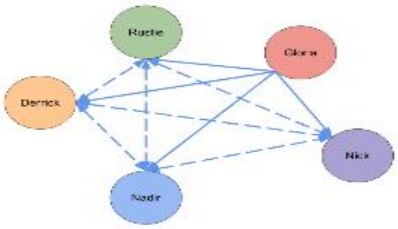
Mining:

A full-fledged Bitcoin miner must:

0. **Download** the entire Bitcoin blockchain
1. **Verify** incoming transactions
2. **Create** a block
3. **Find** a valid nonce
4. **Broadcast** your block
5. **Profit**!

STEP 0: DOWNLOAD THE BLOCKCHAIN

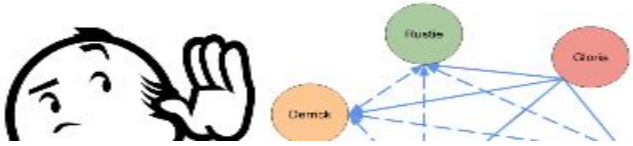
- Get blocks from your peers
- Download the entire blockchain
  - Start from the genesis block
- Stay up to date



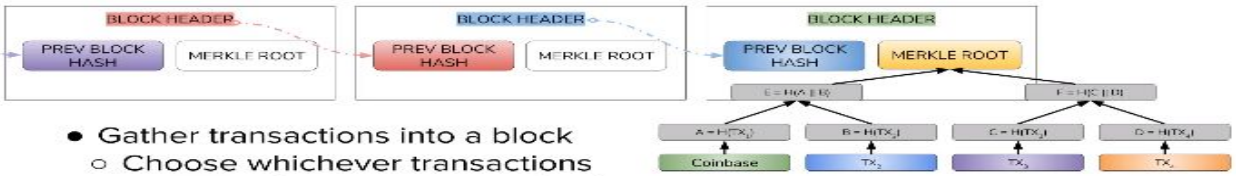
Step 1: Verifying Transactions

As transactions come to the miner, they will store those in a mempool, where “mem” means “memory” and “pool” means supply (referring to supply of transactions). It’s where all the pending transactions live before they make their way into a block.

- Listen to the Bitcoin network for transactions
- Unconfirmed (pending) transactions sit in the **mempool** for a miner to include it in a block
- Verify incoming transactions by running the unlocking script (remember P2PKH and P2SH?)

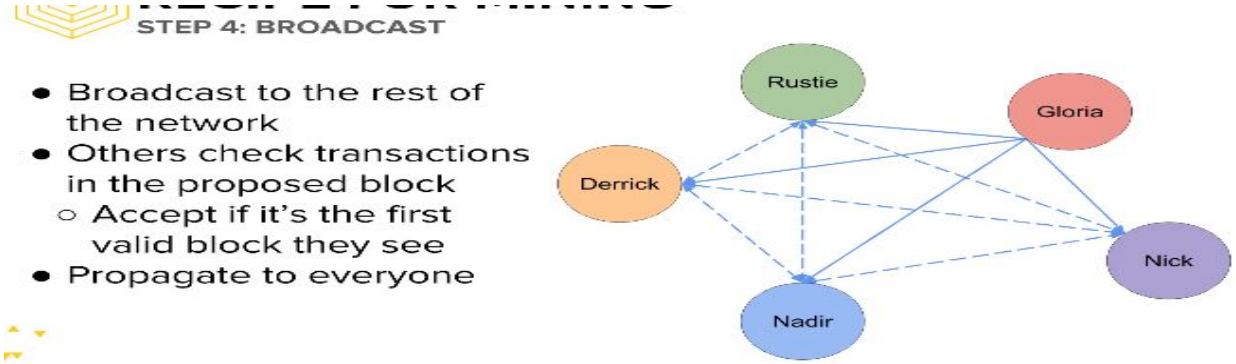


STEP 2: CREATE A BLOCK



- Gather transactions into a block
  - Choose whichever transactions you want (most transaction fees)
- Get previous block hash and other necessary metadata

I did not understand how we change and use the coinbase nonce. *Copying exactly what it says in **step 3**, to understand it in future, also see the pseudocode that is above:*
Keep in mind that the nonce in the header is a reasonably small number: only a 32 bit number. This means that a powerful device can run through all the nonce possibilities within a second. To make it such that we don't reach a dead end with our mining puzzle, we need to change our puzzle if the header nonce has no viable options. To do this, we change the coinbase nonce, a field within the coinbase transaction. By changing this nonce, we change the Merkle Root, and therefore, we're able to explore more options and hopefully find the answer to the mining puzzle. Our mining puzzle is completely different. We go through these loops until finally we find a valid nonce.

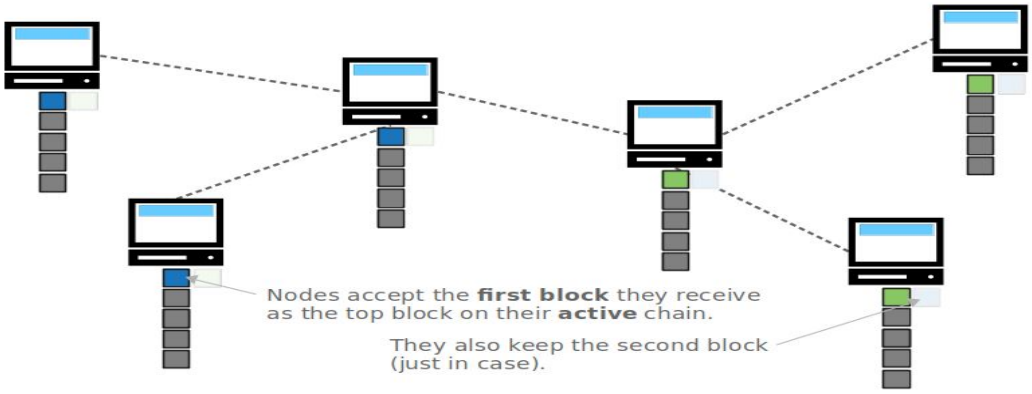


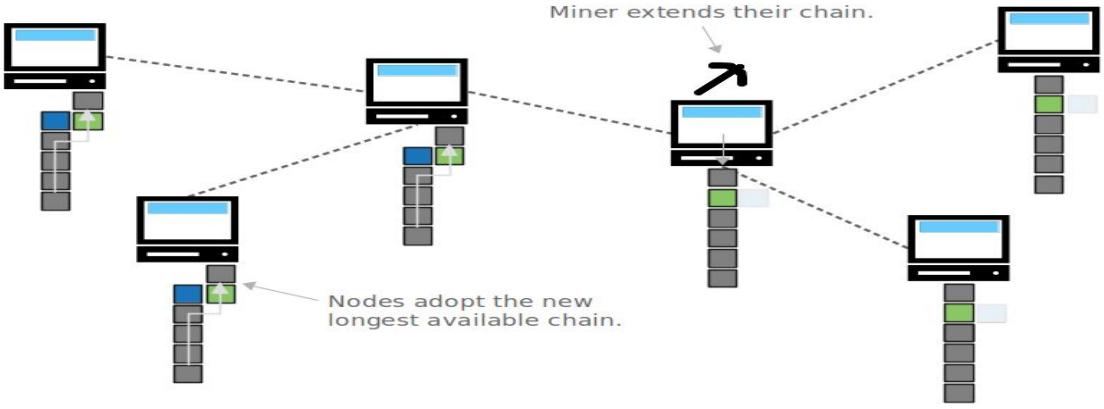
After finding the nonce, the typical miner broadcasts it as soon as possible so that other miners are aware that a block was found. Those miners will validate the block for themselves before accepting it into their own chain, and then broadcast the block once more. By broadcasting the block first, other miners will abandon their previous blocks and start to mine on this new longest chain.
Step 5: Profit

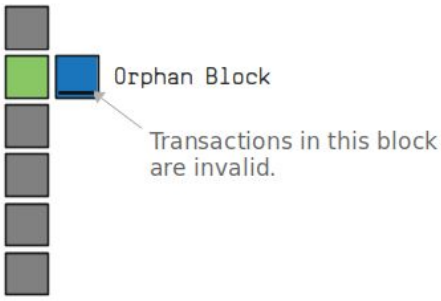
A situation in which our block is not in the longest chain is where our block is competing with another block that was submitted to the network at the same time. In which case, miners will likely choose randomly between the two, meaning that it's once again luck. If your block is orphaned, or in a fork that's not the longest chain, then you get no profit. We won't actually ever

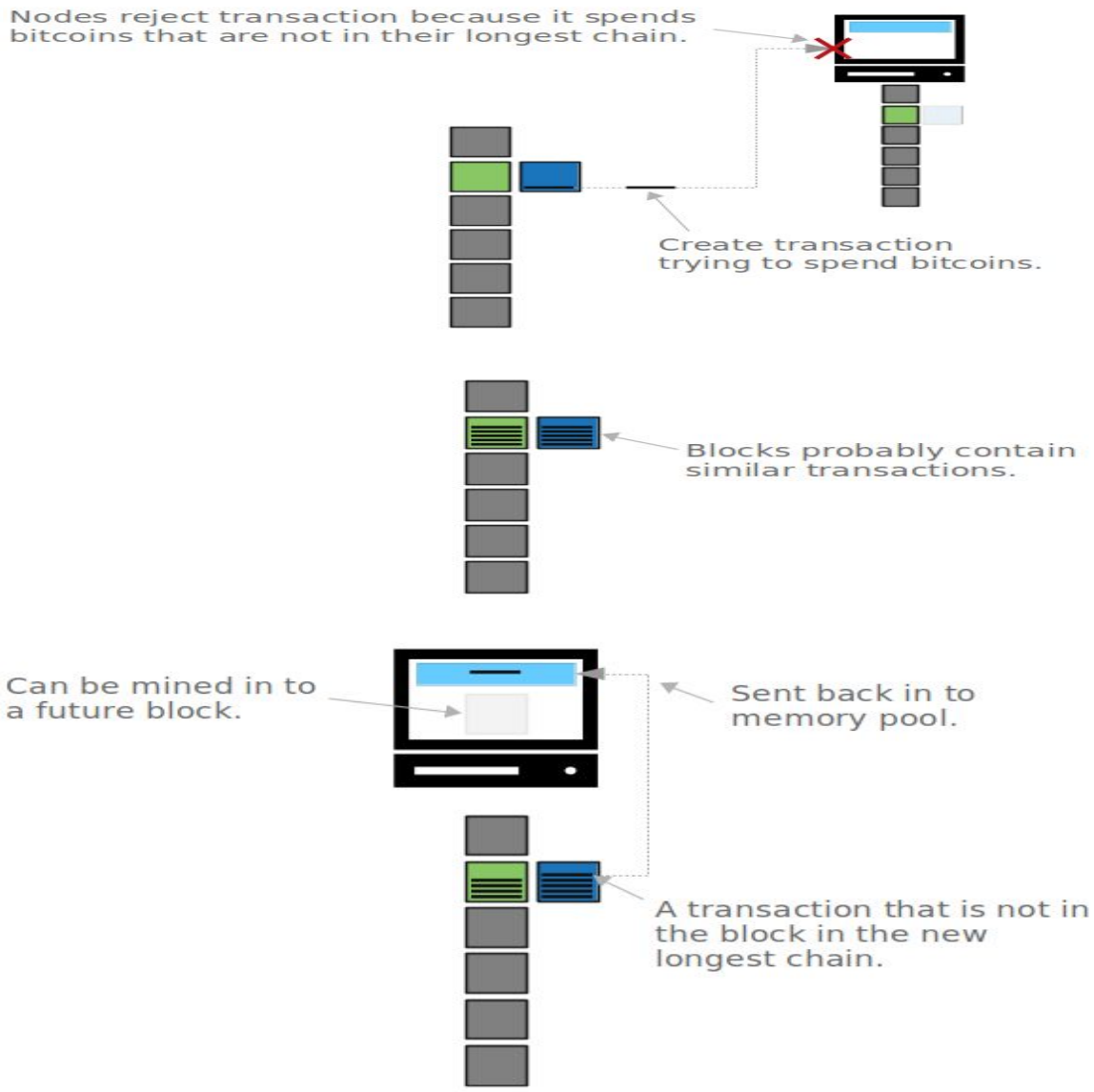
be 100% certain that our block is in the longest chain. We just assume that the probability of a fork happening gets lower over time.

https://bitcoin.stackexchange.com/questions/67097/so-i-mined-a-block-but-why-would-other-nodes-accept-my-mined-block
https://learnmeabitcoin.com/guide/chain-reorganisation









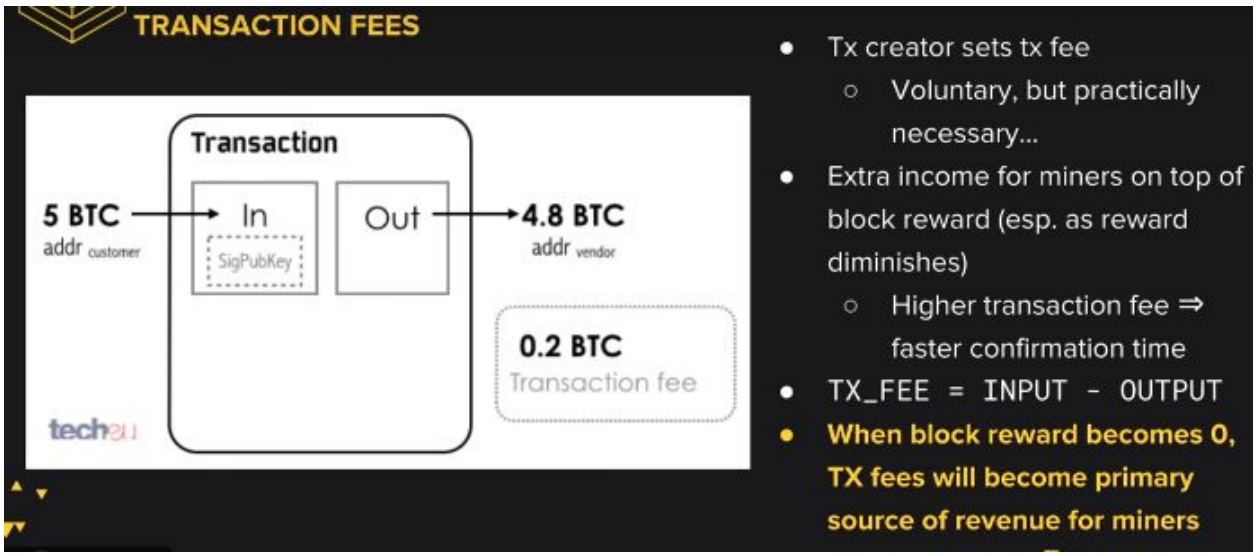
Mining Incentives:

```
MINING_REVENUE = BLOCK_REWARD + TX_FEES
MINING_COST = FIXED_COSTS + VARIABLE_COSTS

if MINING_REVENUE > MINING_COST:
 miner.get_profit()
```

- Miner receives BTC for every confirmed block
  - Currently 12.5 per block
- Miner includes special transaction to self
  - Incentive (profit!) for honest behavior
- Halves every 210,000 blocks
  - Finite # of BTC
- BTC supply cap: 21,000,000

The reward that miner receives is the reason for minting bitcoins and rewarding honest people in the network. Incentive halves after almost 4 years, started with 50btc per block. By using a geometric series, the supply cap is calculated to be 21 million bitcoins. One caveat about this: Not all 21 million bitcoins will be liquid. For example, approximately 1 million belongs to Satoshi Nakamoto, and it is possible that those bitcoins will never move. In addition, some private keys have been lost, and some bitcoins have been burned, all decreasing the usable supply cap.



	hashes / second	time to block (years)
CPU	20 million	7,620,101
GPU	200 million	762,010
FPGA	1 billion	152,357
ASIC	14 trillion	10.88

FPGA : Field Programmable Gate Arrays

- **F**ield **P**rogrammable **G**ate **A**rrays
  - Developing Bitcoin-specific hardware without losing all customizability
- Trade-off between dedicated SHA-256 and general purpose hardware
  - If Bitcoin fails, SHA-256 specific hardware is worthless
  - But if Bitcoin thrives, specialized hardware generates higher **PROFIT**!

ASIC : Application Specific Integrated Circuit

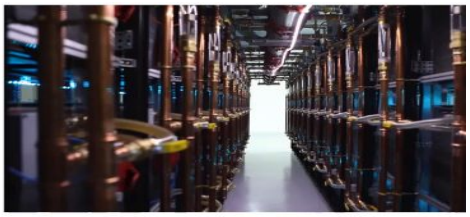
- **A**pplication-**S**pecific **I**ntegrated **C**ircuit
  - Does nothing but SHA-256 -- but does it better than anything else
- Huge variety with various tradeoffs
  - Lower base cost vs lower electricity usage
  - Compact device vs higher hashrate
- Manufacturing ASICs takes large upfront capital, inducing production centralization
- Antminer S9 (14 TH/s): $3000

OPERATING COSTS

- Energy consumed in mining:
  - **Embodied energy,** to produce your hardware
  - **Electricity**, to power your hardware
  - **Cooling**, to maintain your hardware
- Infrastructure
  - Warehouses
  - Personnel
- All energy converted to heat -- is this not wasteful?
  - The “data furnace” approach: using mining equipment to generate heat
    - Unless a high percentage of the network stops mining during the heat, leading to miners dropping out for days on end, or even a whole summer!

Mining farm:

CHINESE ASIC MINING FARM



Source: https://www.theregister.co.uk/2014/08/12/chinese_bitcoin_farms_from_scifi_to_scuzzy/

MINING POOLS

Mining pools allow individual miners to combine, or 'pool', their computational power together

- Reduces variance in mining rewards
- Run by **pool managers** or **pool operators**
- Pool manager usually takes a cut of the mining rewards

MINING SHARES

Miners in a pool submit **shares** ('near-valid' blocks) to the pool manager

- Producing shares implies computational power being expended
- Pool operator pays for valid shares
  - Rewards distributed proportional to # of shares submitted
- Valid blocks are shares as well
  - Individual who finds valid block is not awarded any extra coins

FAQ: Why can't someone submit shares in a pool and keep the reward of the valid block for themselves?

- The valid block is based on the Merkle root given by the pool operator.
- Pool public key → Coinbase tx → Merkle Root

Mining pool schemes:

Pay-per-share

Pool pays out **at every share submitted.** By default will be proportional to work done by individuals

1. More beneficial for **miners**
2. Individual miners have no risk from reward variance
   a. Pool takes on the risk completely
3. Problem: No incentive for individuals to actually submit valid blocks
   a. Individuals are paid regardless

Proportional

Pool pays out **when blocks are found,** proportional to the work individuals have submitted for this block

1. More beneficial for the **pool**
2. Individual miners still bear some risk in variance proportional to size of the pool
   a. Not a problem if pool is sufficiently large
3. Lower risk for pool operators - only pay out when reward is found
   a. Individuals thus incentivized to submit valid blocks

Pros

- Allows individual miners to participate
- Easy to upgrade software changes

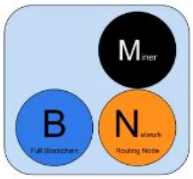
Cons

- Pool manager must be trusted
- Centralized
- Enables a multitude of attacks

The community exhibits backlash against large mining pools. Eg: GHash.io took 50% of the mining power in 2014 and thus was backlashed. However, there is no guarantee someone in the mining pool has 20% of the mining power and is also submitting it's hashpower to other pools, thus gaining more. This is called as Laundering Hashes.
More mining power leads to centralization!!

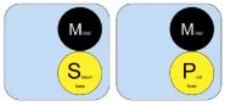
Reference Client (Bitcoin Core)

Contains a Wallet, Miner, full Blockchain database, and Network routing node on the bitcoin P2P network.



Solo Miner

Contains a mining function with a full copy of the blockchain and a bitcoin P2P network routing node.



Mining Nodes

Contain a mining function, without a blockchain, with the Stratum protocol node (S) or other pool (P) mining protocol node.

Stratum mining protocol is an unofficial standard for submitting hashpower to a pool.
getwork, getblocktemplate, and Stratum

getwork is a remote procedure call method that a client can call to get a block header from a mining pool operator. However, a caller of this method does not receive any information about the block or the transactions within it. A malicious server could easily send this user a block header hash that contains invalid transactions, and the user would mine on this invalid block.

getwork has been replaced by getblocktemplate, which allows miners to create their own blocks instead of having to trust somebody else (e.g. a pool operator) to provide them a valid block to mine on. Only a block template is provided with specific configuration parameters, with block data that the client can modify if needed. This way, the control is given back to the miner, because they can now choose the transactions to be included in the block they are mining on. However, miners are incentivized to include pool specific data, such as a coinbase transaction to the pool operator in order to make valid shares and get rewards from the mining pool.

Although getblocktemplate does provide good functionality, it requires a lot of bandwidth due to the transferring of the entire block. Because of this scalability issue, some pools now use Stratum, which is another protocol for mining. Similar to getblocktemplate, the server sends the client a block template. However, in Stratum only the block header and first transaction are included, instead of the entire block.

That’s it for now! Will continue later.