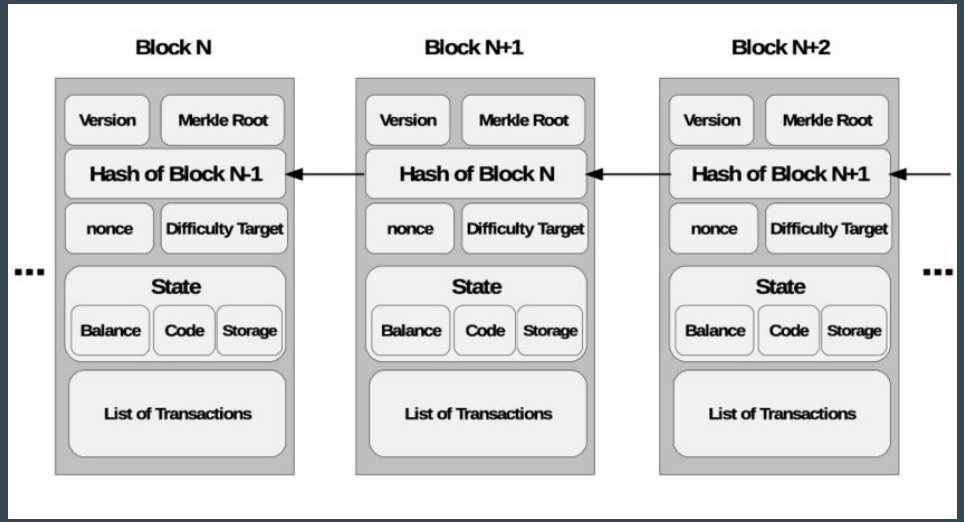
CMSC 398F
Week #3
Blockchain Structure, Mining, Transactions

Announcements

- Quiz 1 scores are out
- Project #1 will be released next week
- Join the class Piazza!
  - piazza.com/umd/fall2022/cmsc398f

Blockchain Structure Review: Components of a Block

- Block Header:
  - Previous Block Hash
  - Root hash of Merkle tree
  - Nonce
  - Other Metadata: Timestamp, the goal of the current difficulty
- Block Body
  - List of Transactions



“Bitcoin mining is the process by which new bitcoins are entered into circulation. It is also the way the network confirms new transactions and is a critical component of the blockchain ledger's maintenance and development.”

Mining

- When a Bitcoin transaction occurs, it is grouped together in a mathematically protected “block” with other transactions that have happened in the same time frame.
- These blocks cannot be added to the Blockchain until they are verified by miners.
- Miners are computers on the Bitcoin network, who use their computational power validate blocks

Mining

- Say we have a block of transactions that needs to be verified. How is this actually done?
- Recall that the Nonce is a variable value added to each block.
- The math problem stipulates that the first miner to produce a hash with a certain amount of leading 0s will be the winner of that block and be able to add it to the network.
- Miners continuously change the Nonce until SHA256 hash function results in a hash with a certain amount of leading 0s

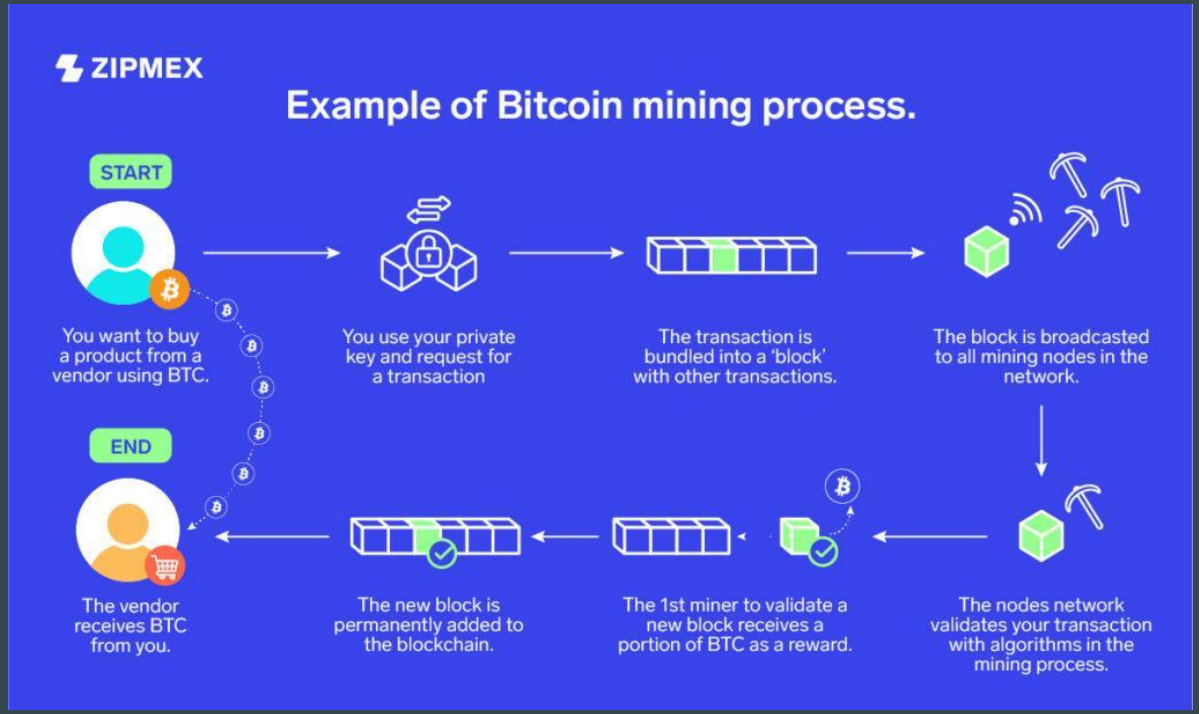
Components of a Block: Nonce

- 32-bit integer that is included in the block
- The nonce will hash together with the previous block hash and merkle root hash to create the hash for the entire block
- A block hash is valid if it is less than a certain target, which usually means starting with a certain number of 0-bits (i.e., the hash must look like 0x0000000023FB23..., not 0x12FD23A123...)
  - Small exceptions to this
- If the block has the correct hash, we call that nonce the “winning” nonce

Example

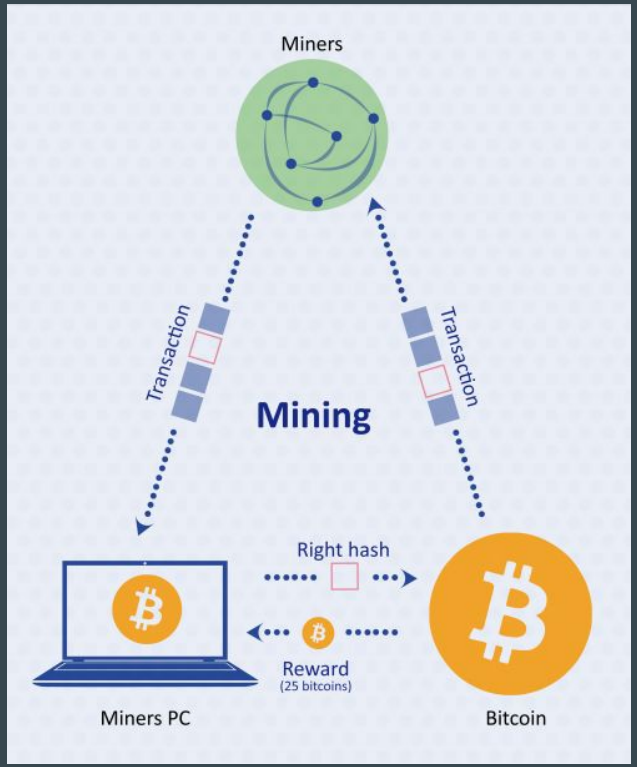
- Say that Om sends Soham 10 BTC
  - It gets added to a block, along with other transactions, which is hashed by the BTC network into "9f86d081884c7d659a2feaa0c55ad015a3bf4f1b2b0b822cd15d6c15b0f00a08"
- The crypto network says: find the Nonce that when appended to the block, produces a hash with 10 or less leading 0s.
- Computers on the network work hard to find this hash. Once it is found, the block is validated and gets added to the network
- If the Nonce cannot be found, the block cannot be validated
- Once the correct hash is found, the transaction and the hash are permanently stored in the blockchain, and if anyone tries to change the information in the block, the hashes will mismatch.

Mining: Bringing it all together



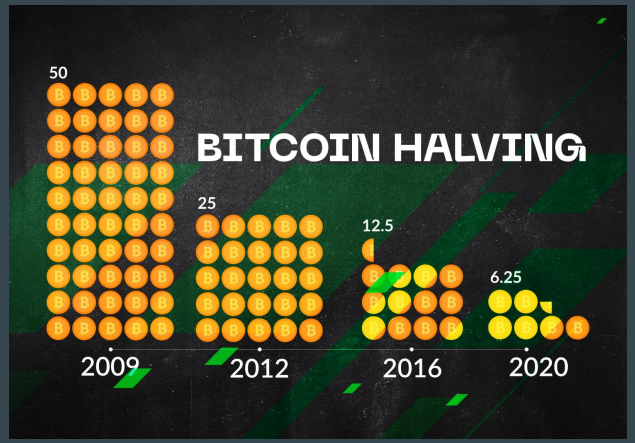
Coinbase Transactions

- A coinbase transaction is the first transaction in a block. It is a unique type of bitcoin transaction that can be created by a miner, and is used to reward the miner
- When BTC first came out, the reward for mining one block was 50 BTC.
- Currently, the mining award is sitting at 6.25 BTC.
- Why does this happen? Halving.



Bitcoin Halving

- One of the most pivotal events on Bitcoin's blockchain is the halving, when the supply of new bitcoins is cut in half.
- Currently, miners are rewarded 6.25 BTC
  - In 2024, this will fall to 3.125 BTC
- Each halving reduces the rate of inflation, thereby creating upwards pressure on the Bitcoin price.
- Occurs around every 4 years
- The total number of BTC in circulation will be capped at 21 million.
- As of June 2022, there are 19.07 BTC in circulation



Why coinbase transactions?

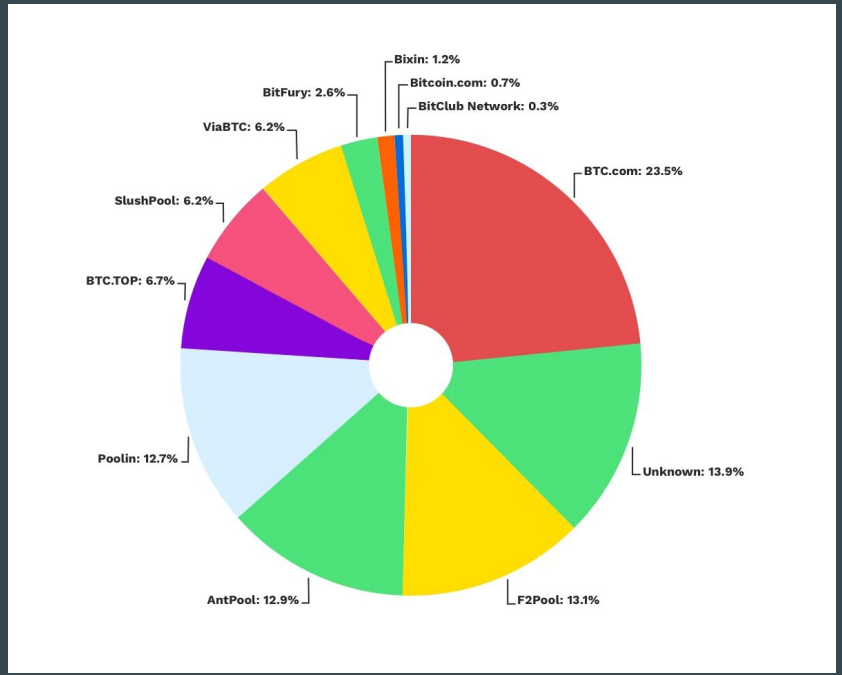
- The purpose of a coinbase transaction is to provide miners with an incentive so that they keep supporting the network. If miners are not mining, then no new transactions will be added to the blockchain.
- Supply and Demand:
  - Help increase the supply of BTC in circulation
  - Cover up for lost BTC
  - Coinbase transaction sets a limit for the number of BTC (21 Million)

Difficulty in Bitcoin

- More and more people mine because they want to win the reward
- As more people mine, and add their computational resources to the network, the Nonce will be found faster and faster
- The Bitcoin protocol has an explicit **goal** to add a block once every 10 minutes.
- Setting difficulty is accomplished by establishing a "target" for the hash: the lower the target, the smaller the set of valid hashes, and the harder it is to generate one. In practice, this means a hash that starts with a very long string of zeros.
- Every 2016 blocks, Bitcoin measures how long it took to solve the last 2016 blocks and adjusts the difficulty (leading 0s)
- Other cryptocurrency networks also define a target rate for blocks to be added, and adjust the difficulty accordingly

Mining pools

- A mining pool is the pooling of resources by miners, who share their processing power over a network, to split the reward equally.
- Block reward is usually distributed among miners based on the amount of hashing power they provide to the collective pool
- Why?
- In addition to creating new coins, pools simultaneously work to keep bitcoin's network functioning
- Mining is only profitable if you make more in bitcoin than you spend on electricity per month



Proof-Of-Work

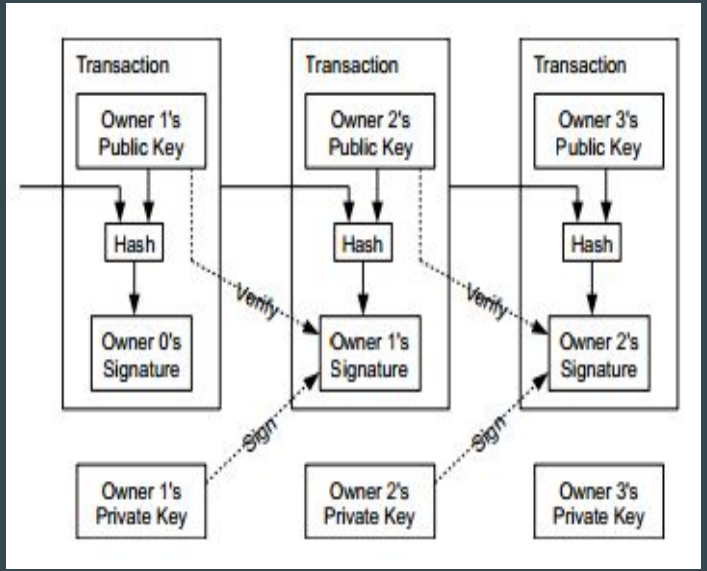
- Proof of work describes the process that allows the bitcoin network to remain robust by making the process of mining, or recording transactions, difficult.
- This entire process of mining and validating blocks, so that new blocks can be appended is known as proof-of-work.

Faucets

- A crypto faucet lets users earn small crypto rewards by completing simple tasks.
- The earliest crypto faucet may be a bitcoin faucet created in 2010 by the then-lead developer of the Bitcoin network named Gavin Andresen. It gave 5 BTC for free to each user who completed a simple captcha.
- Crypto faucets are designed to provide users with free cryptocurrency to start learning about digital assets and eventually use them.
- Examples: Coinbase, Bitcoinker

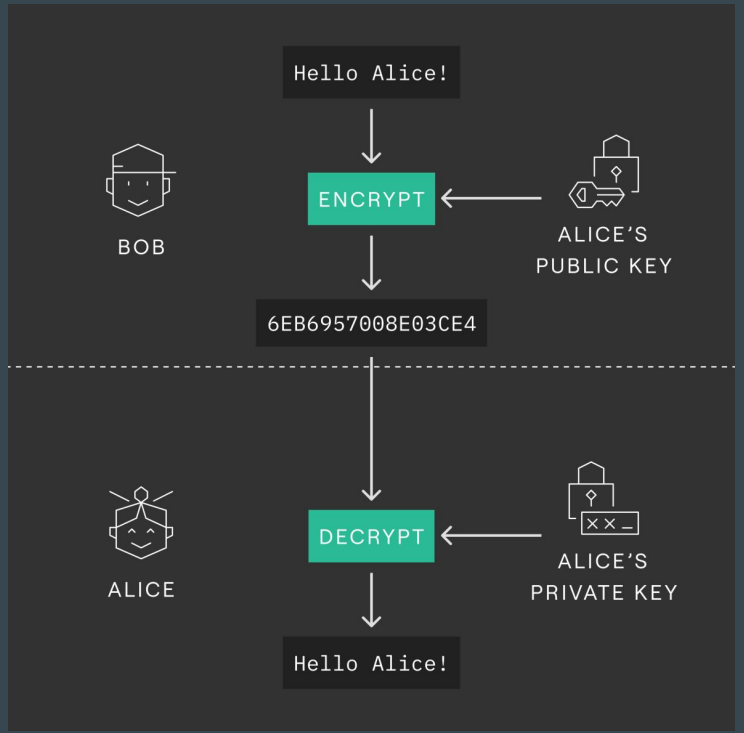
Transactions: Bitcoin

- A Bitcoin address is a string of letters and numbers that represents a destination on the Bitcoin network, nothing more.
- Private keys are important because private keys allow you to send bitcoin from your address to another
- Your bitcoin address is a hash of your public key, which itself is part of an ECDSA (more on it later) pair with your private key
- Each person can create their own private keys in whatever method they choose



Transactions: Public and Private Keys

- Your private and public keys are a 256-bit integers, and your address is a 160-bit hash of your public key
- Private key (256 bits) → Public key (256 bits) → Hash of public key (256 bits) → Address (160 bits)
- Public keys are derived from private keys via Elliptic Curve Digital Signature Algorithm (ECDSA)
- Cannot derive a private key given just a public key
- Public keys are then hashed using SHA-256, and then run through the RIPEMD-160 hashing algorithm to make a 160-bit address
- Private keys allow you to create a digital signature that proves that you want to pay your bitcoin to someone else

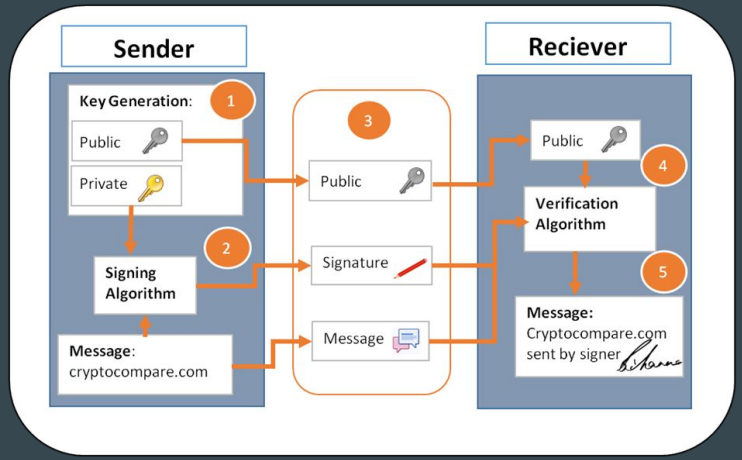


Transactions: Digital Signatures

- A signature is a hash comprised of several inputs
- Normally, a (simplified) function signature is called like:
- signature = sig(private_key, message, public_key)
- A signed message m would look something like:
- (m, sig(private_key, m, public_key))
- Verifying a signature requires the message, the signature, and the public key of the key that signed the message
- is_valid = verify(m, sig, public_key)
- This signed message is broadcast to the network when someone wants to spend bitcoin from one address to another

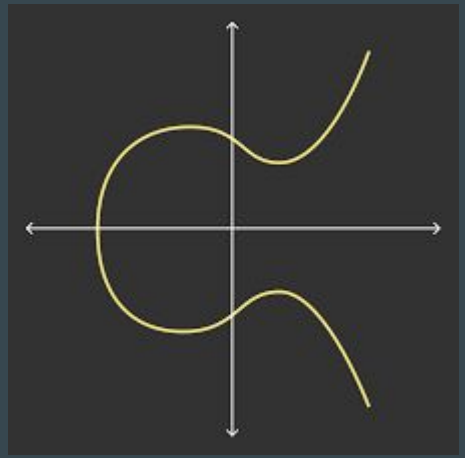
Validating Transactions

- If Alice wants to send Bob 1 bitcoin, she must sign a transaction spending 1 bitcoin of inputs with her private key and send it to nodes on the network. The miners, who know her public key, will then check the conditions of the transaction and validate the signature.



ECDSA

- Bitcoin's current signature scheme is known as the Elliptic Curve Digital Signature Algorithm (ECDSA).
- An elliptic curve is a finite group of points on a curve where some operations are easy to perform in one direction but difficult in the other direction.
- The ECDSA algorithm relies on this to generate signatures that are difficult to forge and easy to verify.
- More about ECDSA in CMSC456!!



Why have a digital signature?

- Owners of Bitcoin can create a transaction and ‘sign’ it using their private key
- This signature can then be verified mathematically by using the owner’s public key via ECDSA
- It keeps transactions secure as an owner can only sign transactions with their private key

How is bitcoin transferred?

- When transactions are created, they are posted to the network
- Nodes have to verify the transaction by checking several things
  - The signature over the transaction input has to be valid
  - The amount of bitcoin being sent must be less than or equal to the amount of bitcoin the user has
  - The bitcoin being spent has not already been spent in another transaction
- Nodes pass verified transactions on to miners so that they may be included in the next valid block
- In general, transactions are not considered to be final until they have a certain number of confirmations
  - Usually somewhere between 2 and 6

What exactly goes into a bitcoin transaction?

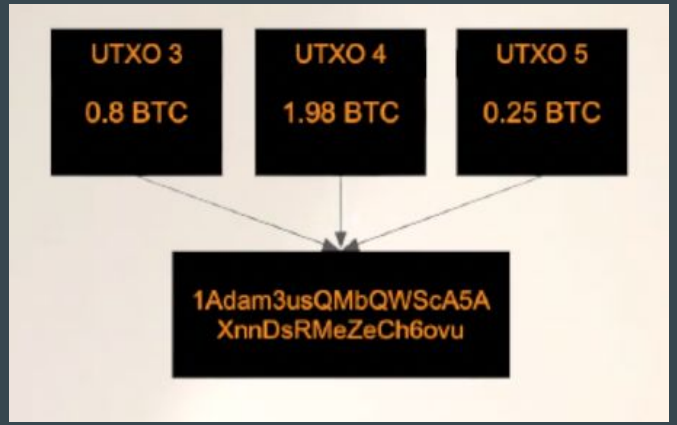
- Transactions are comprised of inputs and outputs
- Recall that transactions are just chains of digital signatures
  - This means that in order to claim bitcoin, you must point to a previous transaction that has the digital signature that sent the bitcoin to you
  - What does this claim look like?
  - In terms of inputs and outputs, the input to your transaction (which is the bitcoin you are trying to spend) must be the output of a previous transaction
  - If you've been sent bitcoin, but have not spent it yet, that output is called an **unspent transaction output (UTXO)**

Transactions

- We know how banks keep track of accounts and the identities of the people who own these accounts
- How do we do this in Bitcoin and other cryptocurrencies? How do we know who owns what?
- Also, how are transactions between users actually handled in Bitcoin?

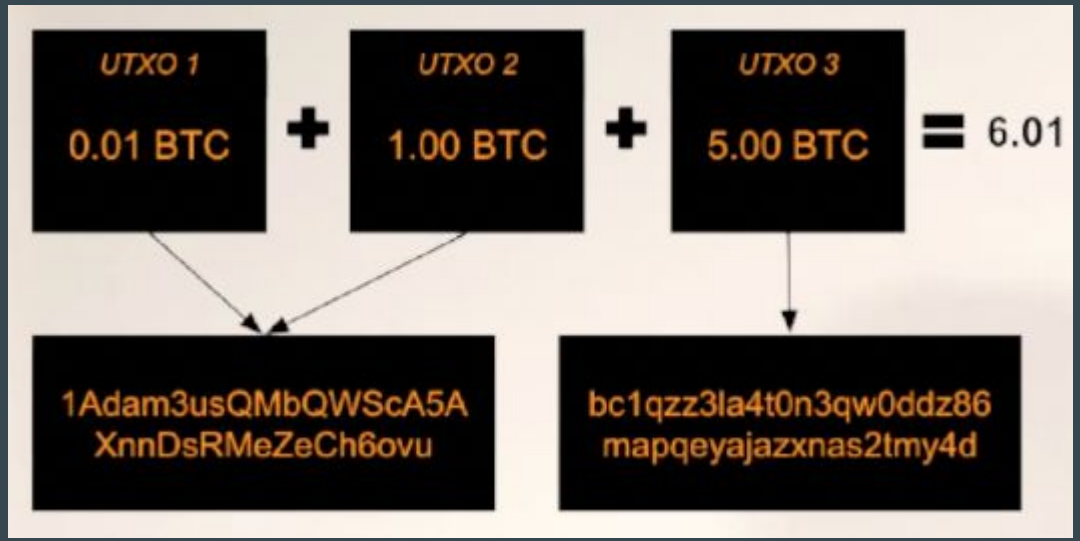
UTXOs

- Contrary to popular belief, Bitcoin is not a system of accounts and balances
- Instead, the network keeps track of these things called UTXOs
  - Stands for ‘Unspent Transaction Output’
- It is simply an **amount of bitcoin that is assigned to a Bitcoin address**
  - Can be any amount of BTC, as long as it is 1 Satoshi
    - 1 BTC == 100 million Satoshis
- Ensure that users are not spending more than they can



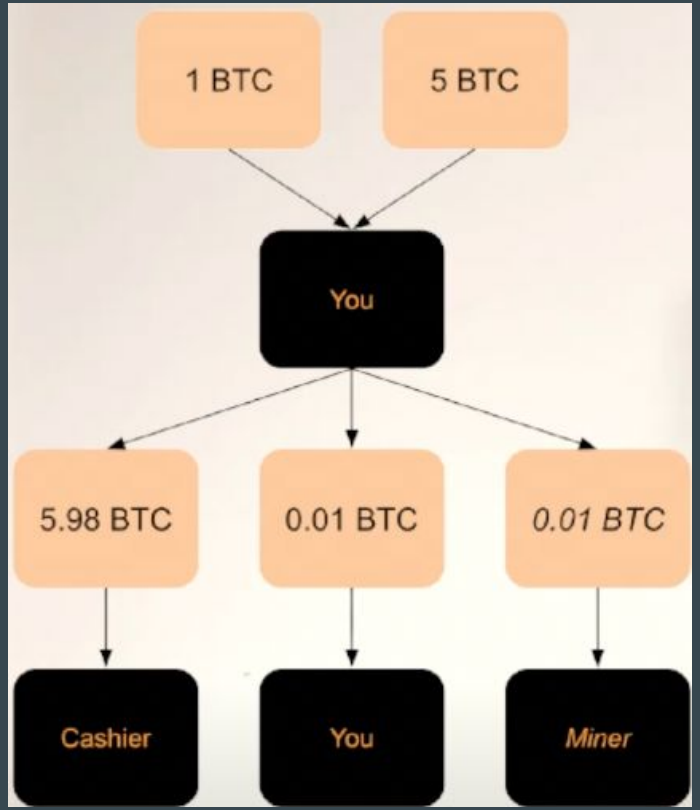
An Example

- Say you have 6.01 BTC in your wallet, that is represented as 3 UTXOs.
  - UTXO 1 is .01 BTC
  - UTXO 2 is 1 BTC
  - UTXO 3 is 5 BTC



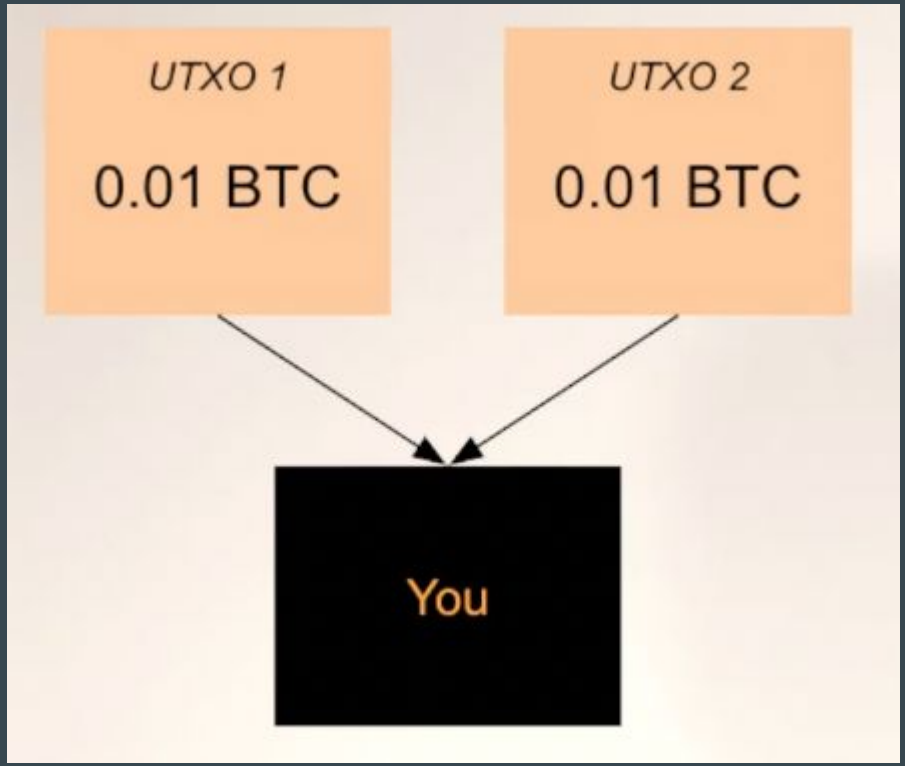
An Example

- Say you want to pay your friend 5.98 BTC using your 1 BTC UTXO and 5 BTC UTXO
  - These are known as inputs to a transaction
- Your friend will receive a new UTXO that has 5.98 BTC
- Another UTXO worth .01 BTC is created, and sent back to yourself
- Another UTXO worth .01 BTC is given to the miner (mining fee)
- At this point, the two UTXOs used as inputs are deleted and cannot be used again
- The new UTXOs created as outputs are unspent, and will be used in future transactions



An Example

- We are left with two UTXOs
  - One UTXO was given back to us as ‘change’. (Output of previous transaction)
  - The other UTXO was ours to begin with (was never an input)



UTXOs

- Think of UTXO's as Piggy Banks
  - Every time a transaction is made to us, we put all that money into a UTXO, or piggy bank.
  - When we want to spend money, we break open that piggy bank (UTXO), spend whatever we want to (and send it as a UTXO), and then put the rest of the money (change) into another piggy bank (UTXO) and give it back to ourselves.
- Complexity for transaction verification drastically improves
  - Question changes from "Is this user trying to spend more money than they can?" to "Does this UTXO have enough funds for the current transaction?"

UTXOs: Recap

1. UTXO: amount of Bitcoin assigned to an address
2. Your balance is the sum of all of your UTXOs
3. UTXOs are inputs to transactions
4. New UTXOs with new values are created as outputs
5. UTXO can be any size bigger than 100 millionth of a BTC

UTXOs

- By adding together the entire BTCs UTXO set, we can calculate the amount of Bitcoin currently in circulation
- In other words, we can independently verify and audit Bitcoins money supply.

```
umbrel@umbrel:~ $ ~/umbrel/bin/bitcoin-cli gettxoutsetinfo
{
  "height": 716884,
  "bestblock": "00000000000000000000ad8639af353cdcb5958fd150ebd3ba9afcd24371362fe",
  "txouts": 78075591,
  "bogosize": 5838694906,
  "hash_serialized_2": "6146964c6f3aee7b694b148909328586989183b8d06a14d96d957de4e52b3469",
  "total_amount": 18917825.04143106,
  "transactions": 47204957,
  "disk_size": 4750590682
}
```

What happens once a transaction is included in the block?

- Usually merchants wait for confirmation before releasing goods/services
- Recall the concept of UTXOs
  - Once a new block is created, the UTXOs that were claimed will now be spent
- All UTXOs are kept track of by nodes in the **UTXO pool**
  - When blocks are published, nodes update their UTXO pools to remove the inputs that are being spent and add the outputs that were created by the block
  - This will maintain the accuracy of the total amount of bitcoin a user currently has

Other commonly used signature schemes in blockchain

- BLS Signatures (ETH2)
  - The current Ethereum chain also uses ECDSA
- Schnorr Signatures (Bitcoin)
  - Much more elegant than ECDSA
- EdDSA