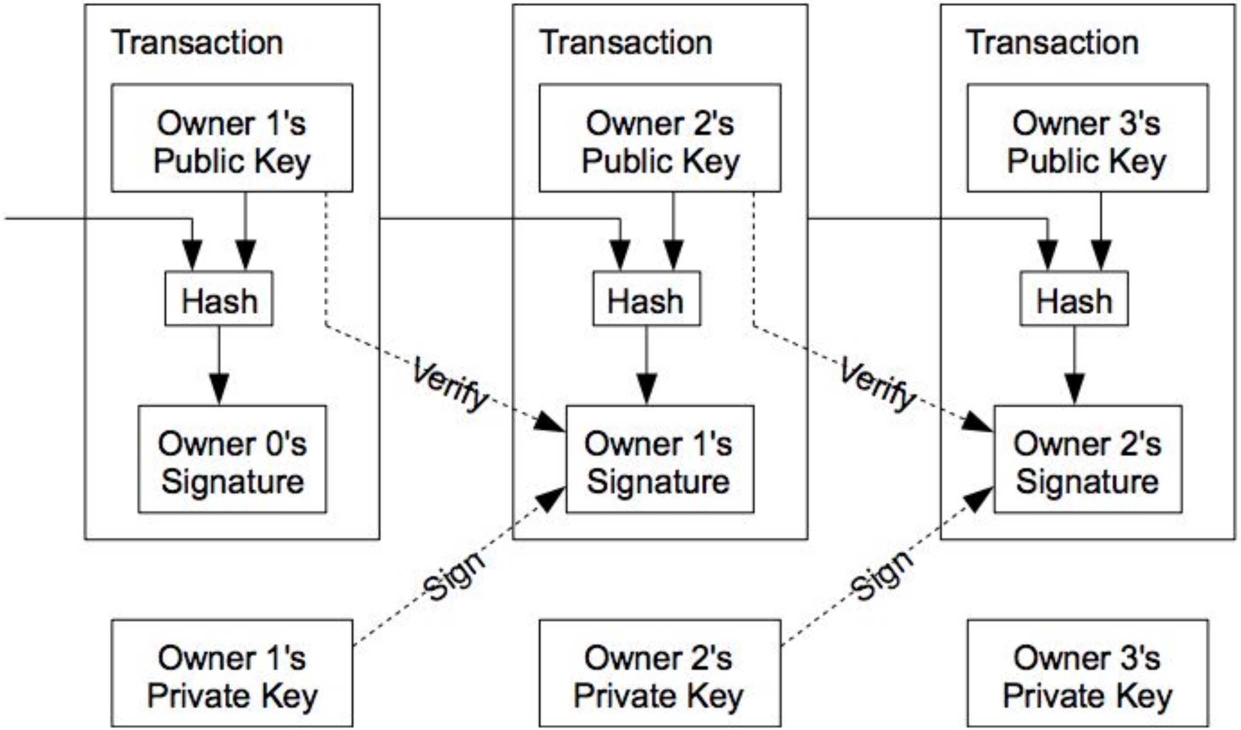
Crypto News?

Transactions

What is a bitcoin?



What is a bitcoin?

- A bitcoin is a chain of digital signatures, with the last address on that chain being the current owner of that amount
- This chain of digital signatures is append-only
  - Where have we seen this before?
- All bitcoin is mined into existence

What does it mean to own bitcoin?

- In order to own bitcoin, you must own a bitcoin address
  - What does owning an address entail?
  - Private vs. Public keys
- Private keys are important because private keys are what allows you to send bitcoin from your address to another
  - Your private key effectively works as a password
  - What happens if you lose your “password” in bitcoin? Can you recover it?
- Your bitcoin address is a hash of your public key, which itself is part of an ECDSA pair with your private key
- Each person can create their own private keys in whatever method they choose
  - Which method is most effective for security?

How are addresses generated?

- 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

What is a digital signature?

- A signature is a hash comprised of several inputs
- Normally, a (simplified) function signature is called like so:
  signature = sig(private_key, message, public_key)
- A signed message m would look something like so:
  (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

Why have a digital signature?

- Owners of Bitcoin can create a transaction and ‘sign’ it using their private key
    - More on this in a few slides
- This signature can then be verified mathematically by using the owner’s public key via ECDSA
- On a high level, this allows for owners of bitcoin to relinquish their coins to another address
- It keeps transactions secure as an owner can only sign transactions with their private key
    - How does quantum computing affect this?

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 for most merchants

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)**

What exactly goes into a bitcoin transaction?

- Bitcoin script
  - Bitcoin comes with a scripting language that allows nodes to run short programs to verify ownership of bitcoin before allowing the transaction to go through
  - Stack-based, not Turing complete (why?)
- For this course, we will be diving into two types of Bitcoin transaction
  - Pay-to-Pubkey-Hash
  - Pay-to-Script-Hash-Multisig

Pay-to-Pubkey-Hash (P2PKH)

- Each input to a transaction contains information about the previous output it wants to spend, plus a short script consisting of <sig>, <pubKey>
- Each output consists of a value, plus another short script that looks like the following: OP_DUP, OP_HASH160, <hash of pubKey>, OP_EQUALVERIFY, OP_CHECKSIG
- The node verifying this transaction will concatenate the input script and the previous output script and attempt to run it
  - If it runs to completion with no errors, then it’s valid; if an error is thrown, the transaction is invalid

Pay-to-Script-Hash (P2SH) Multisig

- Sometimes we may want multiple entities to own the same money
  - We don't want any one person to just be able to spend it at will (say, company funds for example)
- Instead of paying directly to an address hash, we can actually pay to the hash of *another* Bitcoin script
- This is called a Pay-to-Script-Hash script
  - We won't actually go over the script itself here because it's fairly complex and beyond the scope of this class
- In order to allow multiple entities to own the same funds, we create a **multisig** address

Pay-to-Script-Hash (P2SH) Multisig

- A multisig address is a bitcoin script that requires m-of-n (say, 3 out of 5) signatures from m-of-n public keys to spend the funds
- When the entities want to spend the money, they'll need to use the script, along with m signatures (in the order of the public keys in the script)
- This prevents bad actors from stealing all the money out of the address without a majority of owners agreeing to move the funds

What happens once a transaction is included in a 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 what is called a **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

Summary

- Bitcoins are chains of digital signatures; the latest address on the chain owns the coins
- Addresses are hashes of public keys which are derived from private keys
- Transactions are made of input and output scripts, and inputs must be digitally signed by the person who wants to spend the bitcoin

Questions?