Bitcoin

Table of Contents:

Background Noise: Youtube

Hash Functions

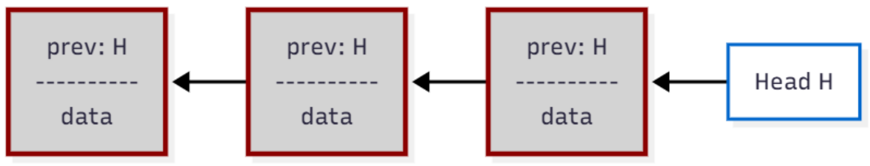
Properties of hash functions:

1. Collision-Resistance: every hash is unique
2. One-Way: if you know the output you cannot find the input
3. Puzzle-friendliness: the only possible way to find a x of H(x) is trying random values for eternity

Hash Pointers?

It's basically for storing data and dealing with hashed values, pointers are used to verify that the hash has not changed, and to find related info and stuff.

How hashes are used in blockchains to secure integrity:



This can be used as an integrity safety net:
If one block gets modified then the data's block (prev H) and the pointer H (arrow) before it wont be qual.

Why have a head H (genesis block)?

To have the ultimate integrity test, you can modify all the blocks you want but you cannot modify the first Head H. That's why it's called the genesis block.

> Remember, blockchains are simply data structures!

What's better than a linked list structure? A BINARY TREE!!

Merkle tree is a blockchain structure were the blocks are connected in a binary tree pattern. So, of course it has a better search time with *log(n)*.

Digital Signatures

Why use them?

> AUTHENTICITY

Digital Signatures are used in blockchains to add a layer of authenticity on top of the integrity, each arrow is signed with the owner's name. So, if someone signs a blockchain that has 100 blocks, in theory he agrees and respects all the signatures before him.

What counts as a signature?

Well... in a perfect world you would use complex math and supercomputers to have super cool authentic signatures. But alas we cant do that, we need something fast and light. That is why secret keys (PK) can be used as signatures, hashed SK is even better!. So when people want to verify your signature all they need is your public key (PK) which is available for all.

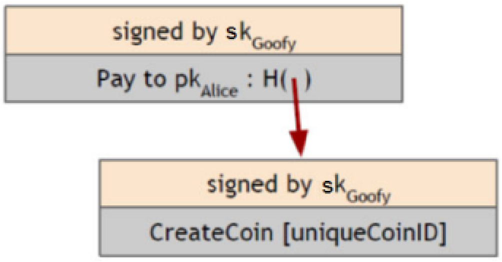
Bitcoin

Buying a Coin:

Unlike real money, crypto does not "transfer" from one hand to another, no it's created for the person who's buying it from scratch.

Here Goofy is selling the coin to Alice.
STEPS:

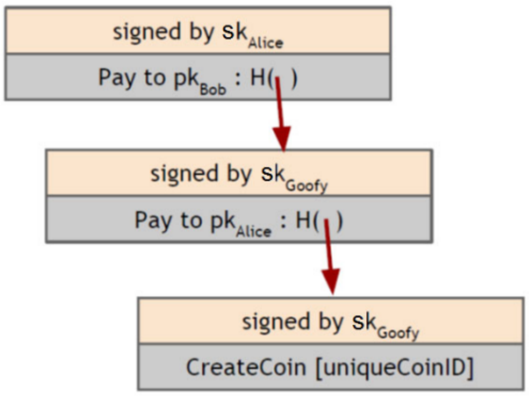
1. Goofy creates a section in his block to announce his statement "Pay to PK(Alice)".
2. Use his H pointer to point to the newly CREATED coin for Alice.
3. Signs the newly CREATED block with his (goofy) SECRET KEY (SK).



Transferring an existing coin (SELLING):

If Bob wants to buy the coin from Alice directly, she can transfer ownership to him without creating new coins. What happens is:

1. Alice creates a block in the blockchain (not a coin).
2. Alice signs the block with her SK.
3. Alice adds that Bob paid her for the coin and confirms with his PK.
4. She points the block to the rest of the blockchain with her H pointer.



Double-Spending

This is a big problem, the prev structure had an issue...
What if the coin was sold (transferred) to two people at the same time!

This issue can be solved like this (ScoorgeCoin Example / Centralized Approach):

- Do not transfer owner ship immediately! wait before appending blocks to blockchain.
- The blocks are not signed by individuals, it's signed by a trusted authority AFTER it checks that there is no duplicates or Double-spending.

> Another problem raises... this creates MONOPOLY where the power is in one hand only ;(

```
There is more to this.. but honestly i got bored and highly doubt its important :D trust me
vru
```

Decentralized Blockchain (Bitcoin?)

Decentralized means it runs on P2P network, where there is no centralized authority to keep track of node identities. That is why a *Sybil Attack* could happen, where an adversary can control copies of a node which may cause traffic or ruin trust in the network.

Decentralized blockchains are built on *distributed consensus protocol*, which states that there are n nodes working together and each has an input value. Some of these nodes can get malicious or faulty.

Distributed consensus protocol properties:

- All honest nodes must agree on a value.
- The value must be generated by a honest node.

> In context of Bitcoin 'value' could be a new block, transactions order or a ledger

But how would transactions be proposed in a consensus network?

- You could do it by having time intervals, every 10 mins a new person proposes their transaction pool

Problems in 100% P2P approach (above example):

- Malicious nodes that ruin trust

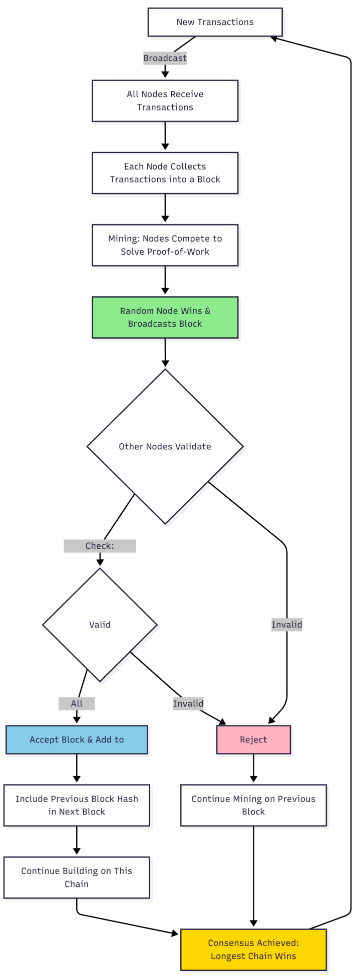
- Poor connectivity if the network sucks
- There is no centralized clock to order the events

BUT there is a way to make this work!

1. Incentives: Bitcoin is MONEY and you can bribe people :D soo have a reward system to reward honest nodes (no more traitor nodes)
2. Add randomness to block appending instead of depending on time intervals or central clocks (use *tickets ID* to decide since nodes dont have ID).
3. Consensus in the long run: Even if there's temporary disagreement (two blocks found simultaneously), after a few more blocks are added, everyone converges on the same chain (the longest/heaviest chain wins).

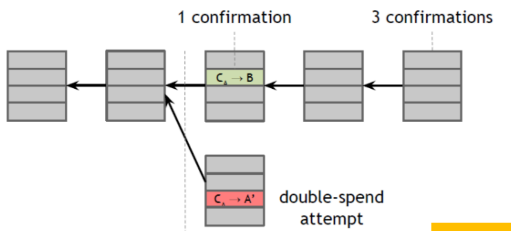
Basically this is how the consensus algorithm would work (**BITCOIN MINING**):

```
1. New transactions are broadcast to all nodes

2. Each node collects new transactions into a block

3. In each round a random node gets to broadcast its block

4. Other nodes accept the block only if all transactions in it are valid (unspent, valid signatures)

5. Nodes express their acceptance of the block by including its hash in the next block they create
```


New Transactions
Broadcast
All Nodes Receive Transactions
Each Node Collects Transactions into a Block
Mining: Nodes Compete to Solve Proof-of-Work
Random Node Wins & Broadcasts Block
Other Nodes Validate
Check:
Invalid
Valid
All
Invalid
Accept Block & Add to
Reject
Include Previous Block Hash in Next Block
Continue Mining on Previous Block
Continue Building on This Chain
Consensus Achieved: Longest Chain Wins

How this architecture prevents Double-Spending?

Lets say Alice only has 50$, she sends transaction of 50$ to Bob, then immediately she sends 50$ to her other account. The blockchain will pick one of them but the money wont move out of Alice account until at least there is 6 blocks (6 confirmations) after the transaction was appended to the chain.



Incentives and proof of work

Incentives is basically rewarding the hard working miner for their work.
Their rewards can be:

1. Block reward: if their block gets picked in the looooooong chain
2. Transaction Fees: input money < output money. BUT this is voluntary, the person sending the transaction is the one paying not the coin creator.

Proof-Of-work (PoW): is when miners do crazy complex math problems to generate the perfect block. By crazy math i mean they need to try a gazzilion nonce values till they satisfy this equation:

```
H(nonce ‖ prev_hash ‖ tx ‖ … ‖ tx) < Target
```

Proof-Of-work limits these problems:

1. How to pick random node
2. How to prevent greed where everyone wants a piece of the pie
3. How to prevent Sybil attacks

PoW Properties:

1. Difficult to compute
2. Parameterizable cost: Target changes every two weeks so everyone needs to redo calculations
3. Trivial to verity: verifying blocks should be easy and fast