5 min break

Discussion

1. What is a blockchain?
2. Describe the 51% attack
3. What did you think of the readings?

Blockchain at Michigan

W2: Consensus

2022

Topics

1. Double Spending Problem
2. Various consensus mechanisms

Double Spending Problem

What prevents me from spending the same money twice?

From creating money out of thin air?

Double Spending Problem

What even is money?

1. Bartering
2. Precious metals
3. Paper money (gold standard)
4. Fiat currency
5. Digital money

Double Spending Problem

Trends:
- Less “real”
- More third parties + trust

“Real” things are easy to verify.

“Not-real” things are not → trust needed.

What even is money?

1. Bartering
2. Precious metals
3. Paper money (gold standard)
4. Fiat currency
5. Digital money

Double Spending Problem

Trends:
- Less “real”
- More third parties + trust

“Real” things are easy to verify.

“Not-real” things are not → trust needed.

What even is money?

1. Bartering
2. Precious metals
3. Paper money (gold standard)
4. Fiat currency
5. Digital money

Could we have money that is less “real”/digital yet doesn’t require trust?

Double Spending Problem

How is digital money validated currently?

Trusted centralized third parties (banks)

Double spending problem is significant in

decentralized, digital systems.

The **main innovation of Bitcoin** was that it solved the double spending problem in a decentralized, digital context.

Questions?

Proof of Work

Proof of Work

How does paper money prevent forgery?

1. UV Ink
    a. Requires forger to do **work**
2. Unique serial number
    a. Element of **randomness**

Proof of Work

HashCash (1997), by Adam Back

Used "work" to stop spam emails.

Made your computer "work" before it could send an email.

Spammers had to do a lot of "work" to send out spam emails.

Proof of Work

HashCash (1997), by Adam Back

Used “work” to stop spam emails.

Made your computer “work” before it could send an email.

Spammers had to do a lot of “work” to send out spam emails.

Work changes incentives.

Proof of Work

What is "work"?

Specifically, how can we make **computers** work?

Proof of Work

What is “work”?

Specifically, how can we make **computers** work?

- Lots of computations
- “Pointless” by design

Proof of Work

What is “work”?

Specifically, how can we make **computers** work?

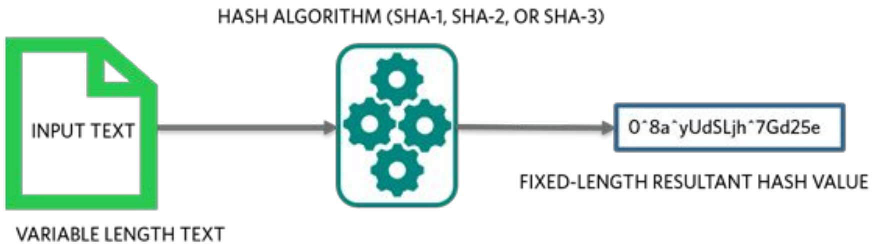
- Lots of computations
- “Pointless” by design

(If the work is “pointful”, people could devise clever algorithms to do the work more efficiently)

Proof of Work

Hashing

A clever mathematical algorithm that takes some text and outputs a **random** number.



(Reference)

Proof of Work

Hashing

Demo:

https://andersbrownworth.com/blockchain/blockchain

Proof of Work

Hashing

A clever mathematical algorithm that takes some text and outputs a **random** number.

Why is **randomness** important?

Proof of Work

In other words:

Difficult to find a nonce that creates a hash with 4 leading 0s.

But easy to check that a hash has 4 leading 0s.

Hashing

A clever mathematical algorithm that takes some text and outputs a **random** number.

Another important point:

Easy to check, yet **difficult to create**.

Proof of Work

So having the right nonce acts as **proof**.

Proof of what?

Proof of **work**!

Hashing

A clever mathematical algorithm that takes some text and outputs a **random** number.

Another important point:

Easy to check, yet **difficult to create**.

Proof of Work

Summary

We force nodes to **work** to contribute to the blockchain. This disincentivizes evil actors.

In particular, this work involves **hashing** a block repeatedly, changing the **nonce** each time, until a particular hash is found.

Trying to find this nonce is called **mining**.

Proof of Work

Math is cool!

Bitcoin has an underlying algorithm that tracks how many hashes are made per second.

It self adjusts the **difficulty** of the problem so that a new block is **mined** every ~10 mins.

Summary

We force nodes to **work** to contribute to the blockchain. This disincentivizes evil actors.

In particular, this work involves **hashing** a block repeatedly, changing the **nonce** each time, until a particular hash is found.

Trying to find this nonce is called **mining**.

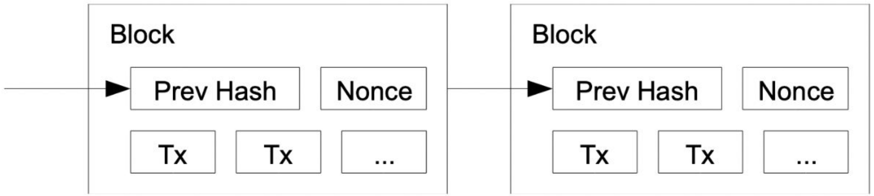
Proof of Work

In Satoshi's own words:

4. Proof-of-Work

To implement a distributed timestamp server on a peer-to-peer basis, we will need to use a proof-of-work system similar to Adam Back's Hashcash [6], rather than newspaper or Usenet posts. The proof-of-work involves scanning for a value that when hashed, such as with SHA-256, the hash begins with a number of zero bits. The average work required is exponential in the number of zero bits required and can be verified by executing a single hash.

For our timestamp network, we implement the proof-of-work by incrementing a nonce in the block until a value is found that gives the block's hash the required zero bits. Once the CPU effort has been expended to make it satisfy the proof-of-work, the block cannot be changed without redoing the work. As later blocks are chained after it, the work to change the block would include redoing all the blocks after it.



Questions?

Let’s look at other consensus algorithms!

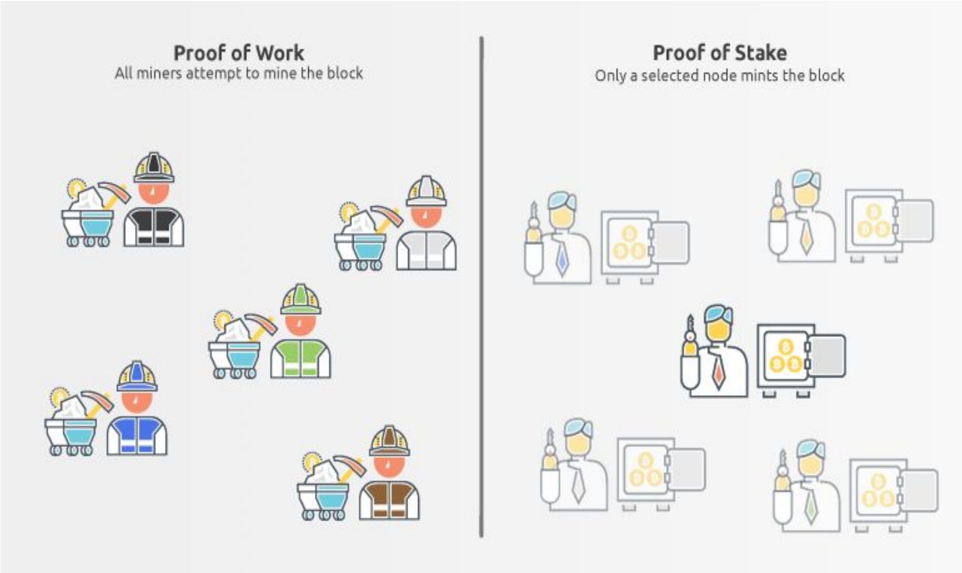
Proof of Stake: How does it work?

Instead of **Miners**, there are **Validators**

Blocks are not **Mined**, they are **Minted**

Validators are have a probability to be chosen proportional to their **stake**.

This is done through a **pseudo-random** selection



Question Time:
What benefits might PoS offer over PoW?

Proof of Stake: How are decisions made?

Finality Conditions:

The rules that determine when the given hash can be considered finalized.

Slashing Conditions:

The rules that determine if a validator has misbehaved (leading to their stake being "slashed")

If `MESSAGES` contains messages of the form `["COMMIT", HASH1, view]` and `["COMMIT", HASH2, view]` for the same `view` but differing `HASH1` and `HASH2` signed by the same validator, then that validator is slashed.

If `MESSAGES` contains a message of the form `["COMMIT", HASH, view1]`, then UNLESS either view1 = -1 or there also exist messages of the form `["PREPARE", HASH, view1, view2]` for some specific `view2`, where `view2 < view1`, signed by 2/3 of all validators, then the validator that made the COMMIT is slashed.

Proof of Stake: Security

Biasing Attacks

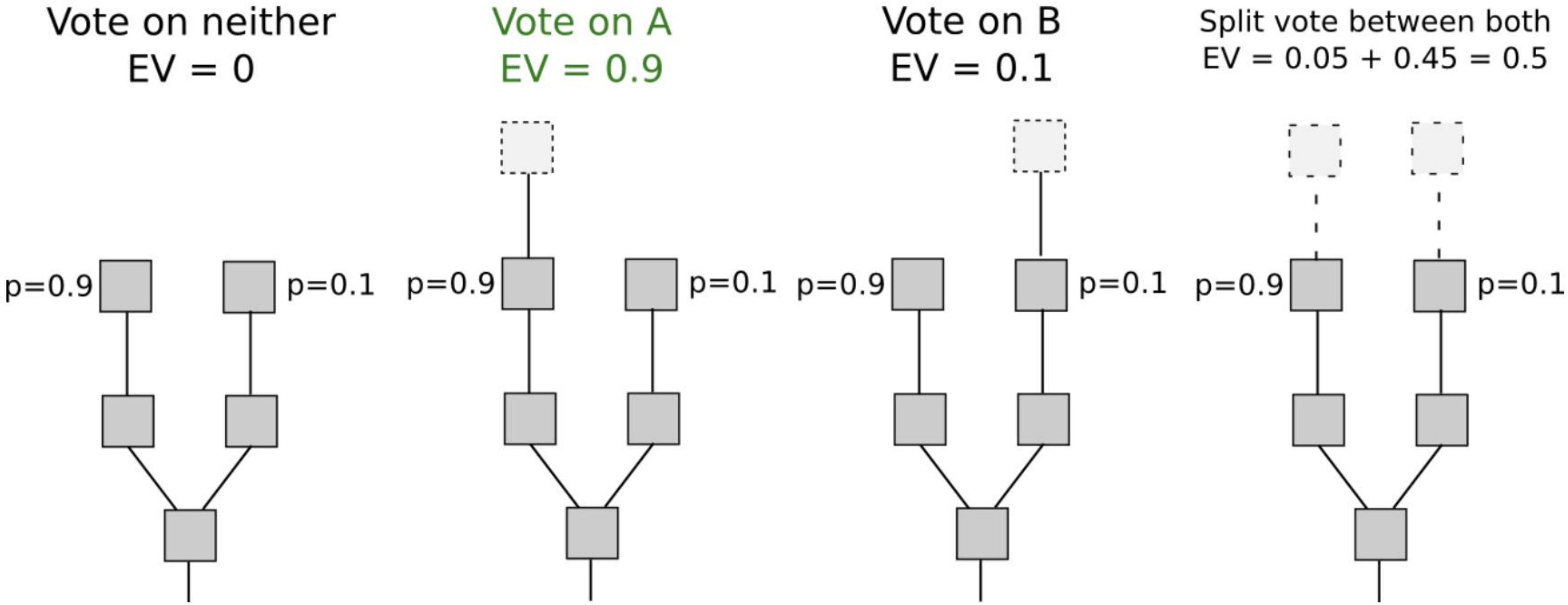
An adversary may try to bias the random selection process in their favor (recall, the stakeholders run this random selection process).

Nothing-at-Stake Attack

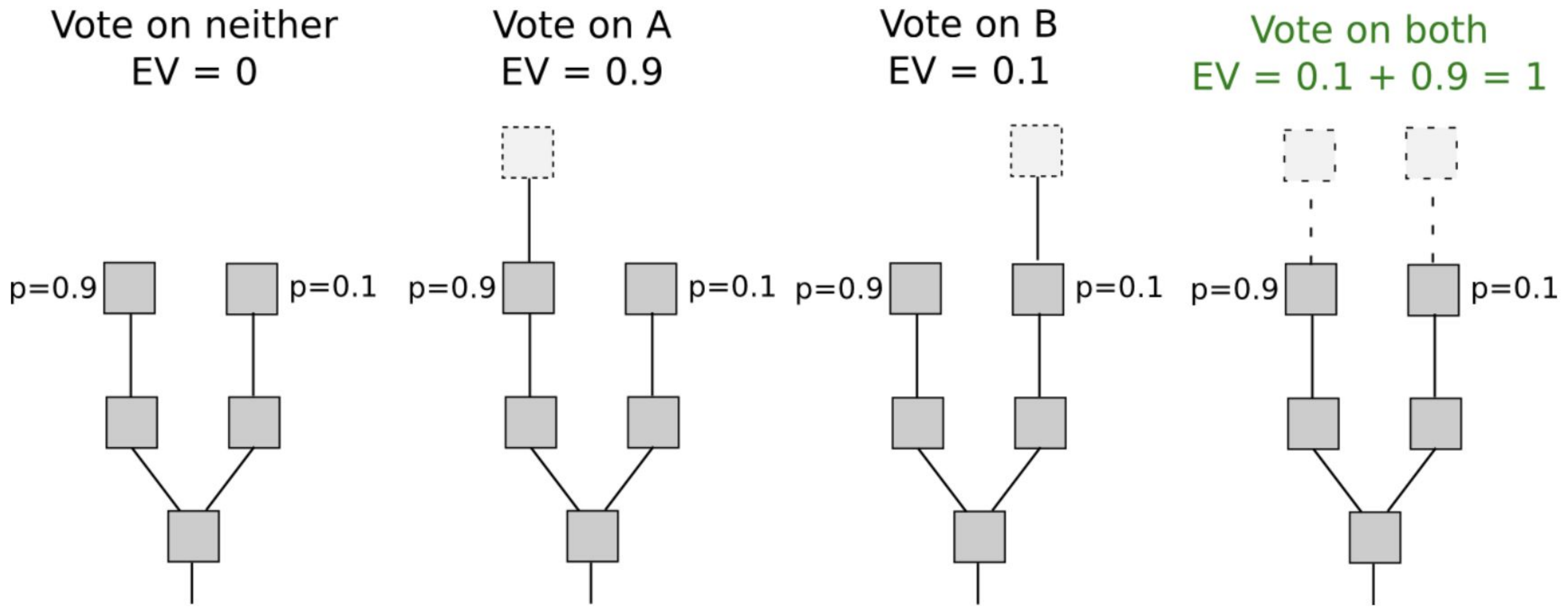
A malicious majority of stakeholders can take any point in time in that system and replay the whole ledger without expending significant compute power.

Question Time:
Does PoW have these sorts of problems? Why/why not?

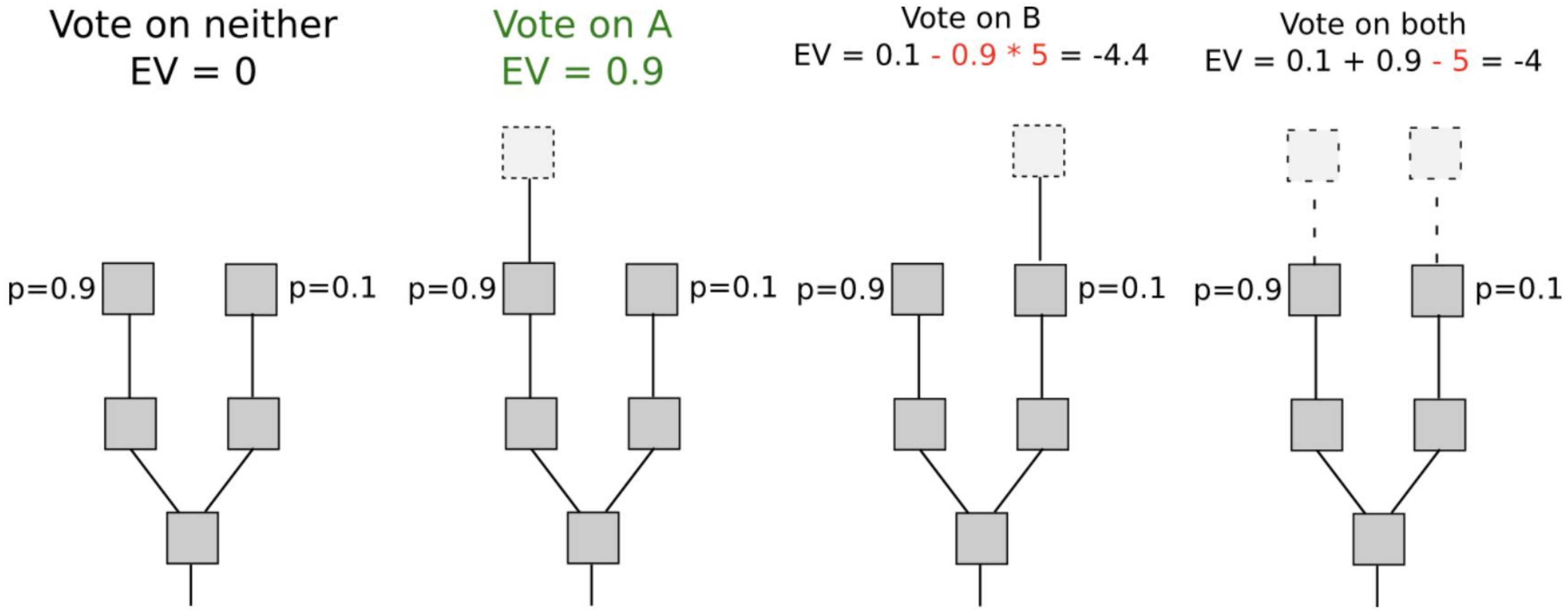
BTC, EV analysis on mining on two forks.



Proof of Stake: Nothing at Stake



Proof of Stake: Stake Something!



Proof of Stake: Solutions To These Problems

depending on the implementation, between 33-50% of validators can interfere in the operation

Biasing Attacks

Don't have the hash/signature of blocks be the primary determinant of randomness.

Using **secret-sharing** or a **threshold signature scheme.**

Threshold signature schemes require a minimum # of participants to compute a valid signature.

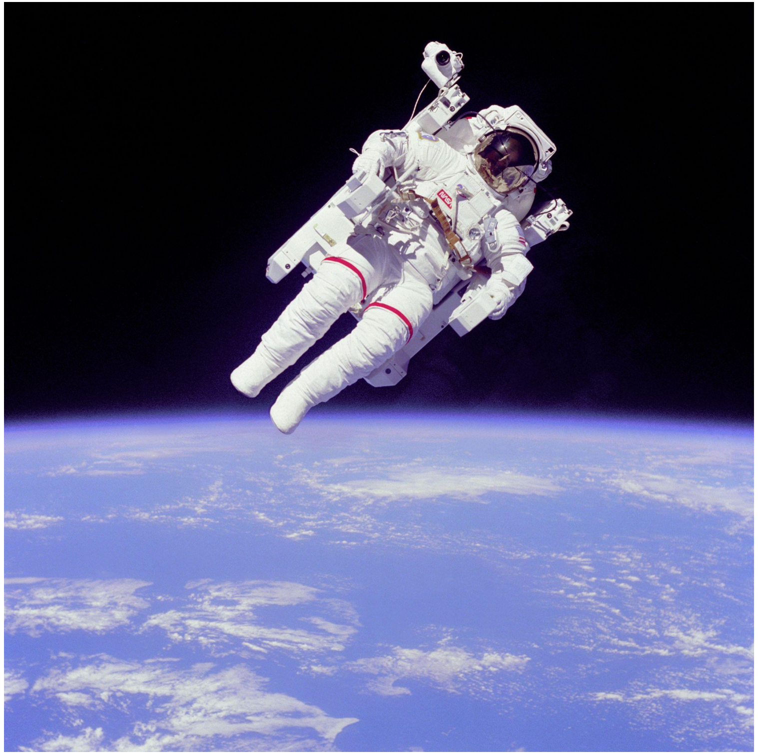
Nothing-at-stake

Have participants put something at stake!

Questions?

Proof of Space

- Similar to PoW in that it requires hardware to solve a problem
- However, instead of computational power, free storage space is used
- Data is sent to the verifier, who then has to prove that it was stored, and the space for it was reserved on the machine
- Theoretically much better for the environment than PoW



Proof of Importance

- Conceptually similar to PoS
- Besides the overall amount of coins owned, an importance coefficient is also used to prioritize validators
- Importance is calculated through multiple factors, like, for example, the number of transactions made

Proof of Authority/Proof of Reputation

- New blocks are validated by approved accounts
- Incentives to verify transactions truthfully are social in nature
- Identities associated with accounts are recorded on the blockchain: in both consensus algorithms validators stand to lose their reputation by acting against the consensus

Proof of Burn

- Under PoB the distributed ledger contains a “burner” address
- Validators gain their status by committing coins to this “burner” account
- Depending on the implementation of the system, chance to be selected to validate can depend on the number of currency burned, for example

Proof of History

- Validity of new blocks is proven through the time of their creation
- Under PoH there exists a record with transaction history that remembers not just their contents, but also the exact specific moment in time when they occurred
- Used by Solana

Proof of Activity

- A mixed approach that utilizes principles from PoW and PoS
- New blocks are mined like in PoW
- Unlike PoW, mined blocks do not contain transactions
- Then validators are chosen based on the amount of coins owned by each user (similarly to PoS)
- Both the miner and the validators receive a reward

Questions?

Readings:

1. Double Spending, by Binance Academy
2. Byzantine Fault Tolerance, by Binance Academy
3. Proof of Work, by Binance Academy

*There will be a discussion next session