Key ideas: Concepts

- Proof of Work (PoW)
  => Cryptopuzzle - originally invented for spam email

- Blockchain (Dist. Ledger)
  => Ordering on operations (txns)
  Read/Write Shared State

- P2P + Byzantine threat model
  Arbitrary peer Behavior

- Eventually Consistency?
  If you wait long enough
  then everyone will observe same state
  Blockchain
Intercepted: Man in the middle

Alice \( \xrightarrow{m} \) Bob

"I give 1 BTC to Bob" msg not signed: if Alice has a BTC

Charlie \( \downarrow \) Double Spending

Alice \( \xrightarrow{\text{sign}_A(m)} \) Bob \( \downarrow \) Bob can check that msg is really from Alice

\* MIM: at most can Replay the msg
\* Double Spending still a problem

Charlie \( \xrightarrow{4'} \)

3'. \( S_A(m) \) 

1. get (1 BTC)
2. \( \text{0xFF} \)
3. \( S_A(m) \) 
4. Checks if \( \text{0xFF} \) valid + Alice has a BTC to give
5. Ack

BANK

Central entity: provides accounting
Trusted
Provides uniqueness guarantees
Bank → Distributed P2P Context
"Make everyone the Bank" ⇒ Bank is public/transparent
⇒ all peers in the system track the ledger of txns

P2P network ~ Bank

A → B

× double spending] PoW
× Concurrency] Blockchain
× Incentives] Reward P2P peers
× Trust] Assumptions about majority of non-malicious

C → tx2

(tx2(A → C)

(tx1(A → B)

"txn committed" if majority of P2P netw. know about it

"A has 1 BTC to give"

Sybils

Any two majorities overlap

Requires to know the # of nodes in system

⇒ Easy to Join

⇒ Easy to create "Sybils" by 1 person

⇒ Sybil Attack
Proof of Work (PoW)

1. Make validation of txns in the network "difficult" (Why? A: Sybil's)
   ⇒ You need real physical resources (CPU cycles for computing PoW)

2. Incentives for nodes to compute PoW
   - Reward for solving a PoW ⇒ # of BTC
   - Scales with amount of CPU cycles

3. Transactions come with a fee that is given to a node that "validates" it using PoW

   △ M1 Check txi valid (consistency check)
   △ M2 Solve a cryptopuzzle (PoW)

   \[ h = \text{sha-256} \text{ hashing fn.} \]

   Find a nonce value s.t.
   \[ h(\text{Block}) \leq \text{target value} \]

   i.e., \[ h(\text{Block}) = 0x00...00 \text{ SAF42...} \]

   Key Conditions for PoW
   1. Difficult to find nonce
   2. Easy to verify the nonce
Mining generates reward to miner (in BTC form)
⇒ Race between miners to mine blocks ⇒ Mining pools
   for cooperation

Miners have to balance
# of txns in a block
with the fact that other
miners are already
mining

Select some
# of txns
(Bound on block size)

BTC reward is generated
until 2140

After 2140
Mining is incentivized
using only tx fees

Missing: Ordering of txns
(txn₁ ≤ txn₂)

Blockchain
(actually a tree)

To mine a block,
a miner has to
pre-commit to
where the block
will go in
Blockchain
Miners - Work along the longest chain (that they know)

Miners - Keep track of all forks (the entire tree)

In short term "longest chain" is unclear
But... in long term "longest chain" is stable

Race cond. in mining
Network latency
Network connectivity

⇒ txn is not "confirmed" unless
   1. txn is on longest chain
   2. Must have 5 blocks that follow it
   "6 confirmations"

Implications:
1. Blocks are immutable: "ledger" ⇒ Append Only
2. Difficult to create a fork
   + Convince network to follow it
   ⇒ Requires maj. 1/3 CPU power

Need to Mine a longer chain than network
txn_A \& txn_B conflict: "double spend"
Bitcoin Overview

1. Flooding Txns
2. Mining process to
   2. Generate blocks
   3. Flooding Blocks (that include txns)

The End