(F) Proof of work: ① Make computationally costly for network users to validate txns

② Reward them for not helping to validate txns (incentive)

(1) Check txn valid
(2) Solve crypto puzzle (proof of work)

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

Find nonce \( n \) s.t. \( h(\text{block}) \leq \text{target} \)

or, think of \( n \) as \# of leading zeros = target

i.e., get \( h(\text{block}) = \text{000...000000000} \)

size of nonce \( \leq \text{target} \)

Exploration from [16x16x16...]

After 2140 txns fee as incentive specified by

Reward (generating BT)

Until ~2140

Chance 1/2^n of finding nonce random

Real finding nonce = compute power \( \Rightarrow \) Sybils ineffective!

(G) Missing: ordering of txns (\( \text{txn}_1 \leq \text{txn}_2 \))?

Include hash of previous block

But, can create a fork in this Chain

Rule: (1) Only work to extend longest fork
(2) Keep track of all forks

txn not "confirmed"

Unless (3) It's part of longest chain
(4) It has 5 blocks follow it

("6 confirmations")

Key: Difficult to create a fork + reorganize all 7 txns.

(Note: Requires controlling >50% of comp.)
Key ideas:
- Proof of work
- Blockchain
- P2P transactions ledger

Alice $\rightarrow$ Bob

"I'm giving Bob 1 Bitcoin"

**Key Challenges:**
- Double spending
- Proof of work
- Trust in the network
- Incentives

(A) Formable $\Rightarrow$ Sign the message with private key from Alice
difficult to forge

(B) Replay attack $\Rightarrow$ sign and replay with Alice's identity
B could claim 10 Bitcoin + Bank central

(C) Centralized
Rely all trust $\Rightarrow$ Make everyone the Bank
in any entity

(D) Have everyone keep track
F. who owns which BT.
i.e., shared ledger with all trans.

(E) $\Rightarrow$ Proof of work
Big problem: A could still rock puppet = sybil attack / double spend by installing worth 10 BTC.
More details:
* Multiple inputs/outputs / All inputs "spent", w/ change given to outputs
* "just ledger of transactions"

Following:
Chain of transactions

Going:
1. Generate block: no inputs, 50 BTC output
2. Coinbase tx: reward to a miner

* Merkle tree data structure compact representation
* Network - join/leave protocol
* BT Scripting language: each tx have script
A Hard Fork: Non-Updated Nodes Reject the New Rules, Diverging the Chain.