Bitcoin

- Decentralized (No governments) peer-to-p
- Virtual
- Deflating (limited supply) 21 mil
- Universal

↓ Legality? Time lag

Hash

$x \longrightarrow H(x)$ fixed size

256

Collision free $x \neq y$ $H(x) = H(y)$ $x \neq y$

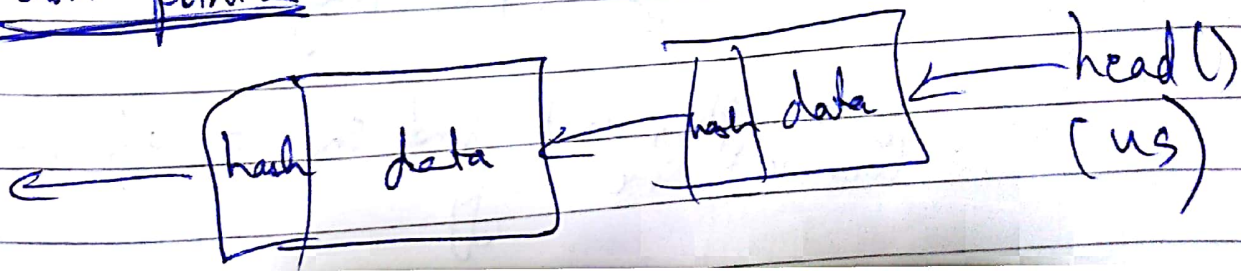
Hiding : given $H(x)$, cant find x → $H(r \| x)$

puzzle friendly → Hashing to a range is infeasible (truly random)

App : Commitment
Seal $H(msg \| key)$
give msg, seal with key in env
given Hom (H), cant find msg || key : Hide
no 2 msgs. : Commit / Bind

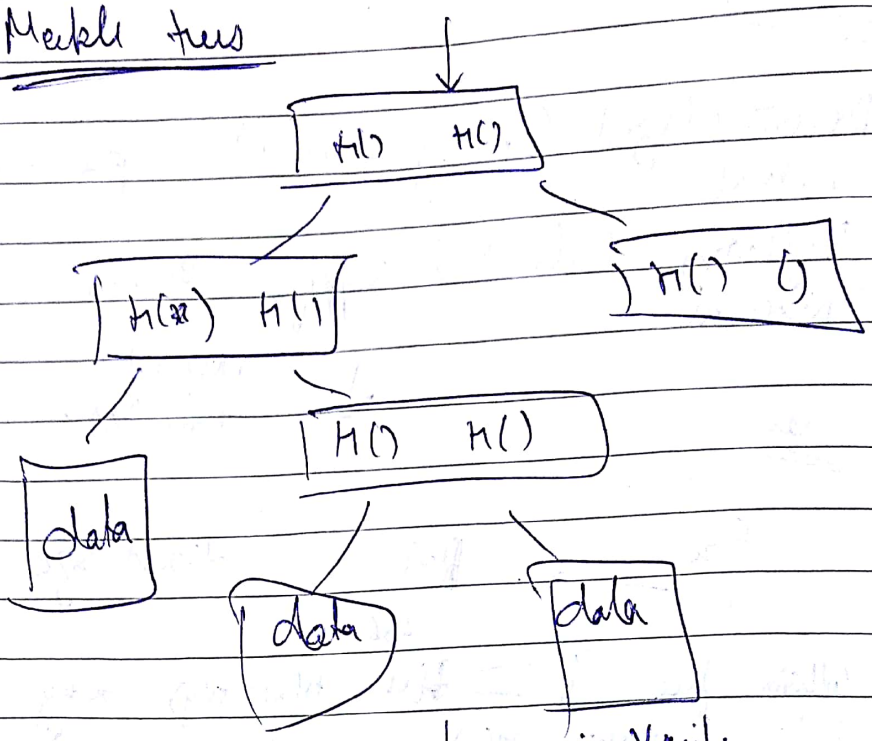
Message digest

Hash pointers



Date Page

Merkle tree



log n : verify
log n : time.

Signatures

RSA, DSS

gen : (sk, pk)
secret / signing key → public key (identity)

sig : sign (sk, msg)
isvalid = verify (pk, msg, sig)

- given pk, can't find sk = unforgeability
- consistent i.e. verify

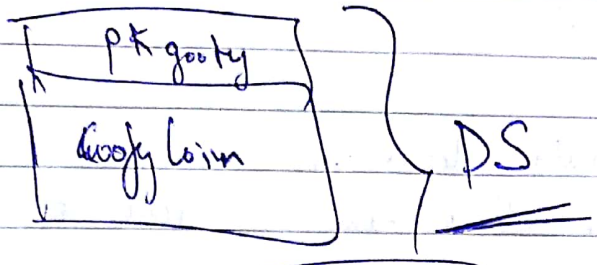
you are making statements on behalf of pk through sk.

verify (pk, msg, sign(sk, msg)) == true

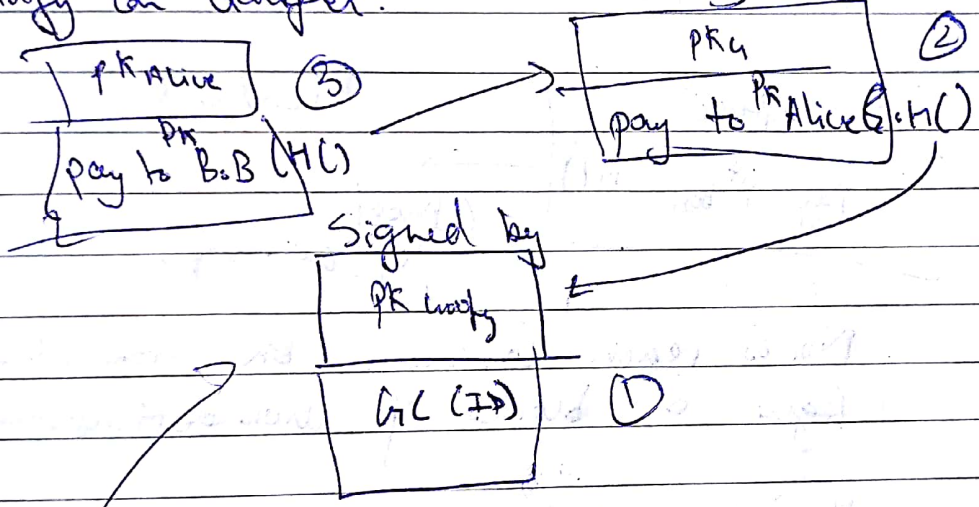
SK_A — Alice — 5BTC → — Bob — SK_B

PK_A — PK_B

1) Coofy wants

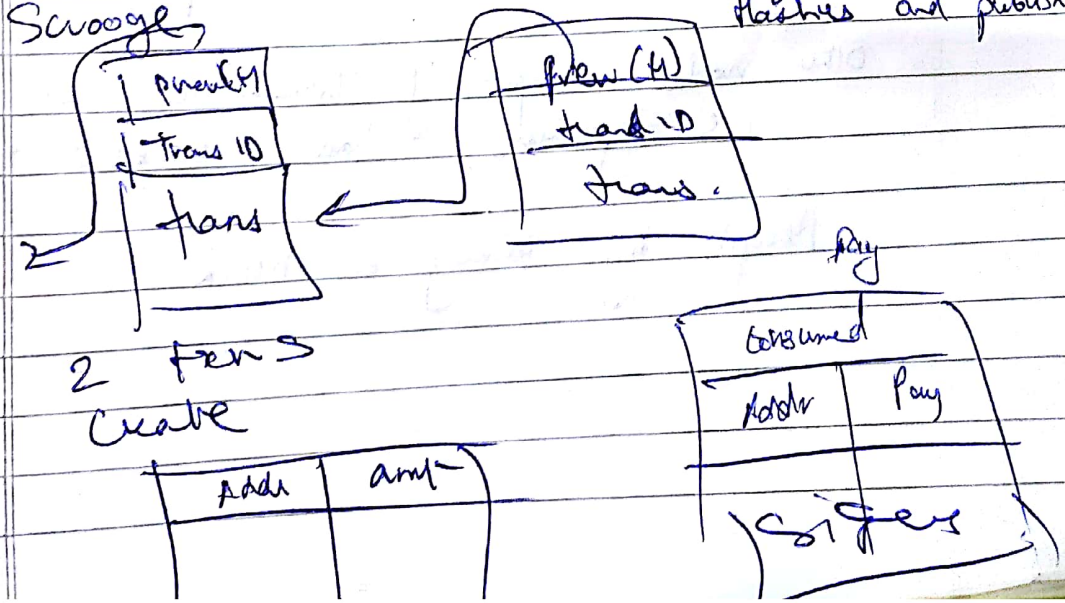


2) Coofy can transfer.



Double spend

Scrooge signs the hashes and publishes

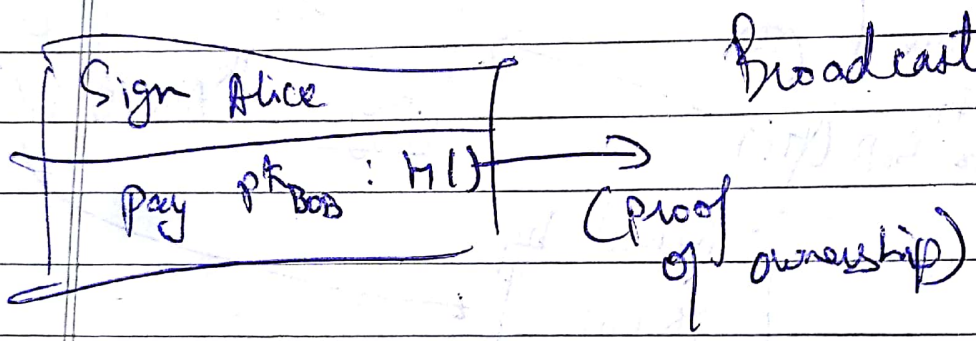


Decentralization

→ Maintaining the ledger.
→ Validity
→ Creation

Distributed consensus.
Record all or none : Posts on FB going to 30 servers.

peer to peer.



Nodes reach consensus on transactions and add Agree on blocks of transactions.

How? Implicit

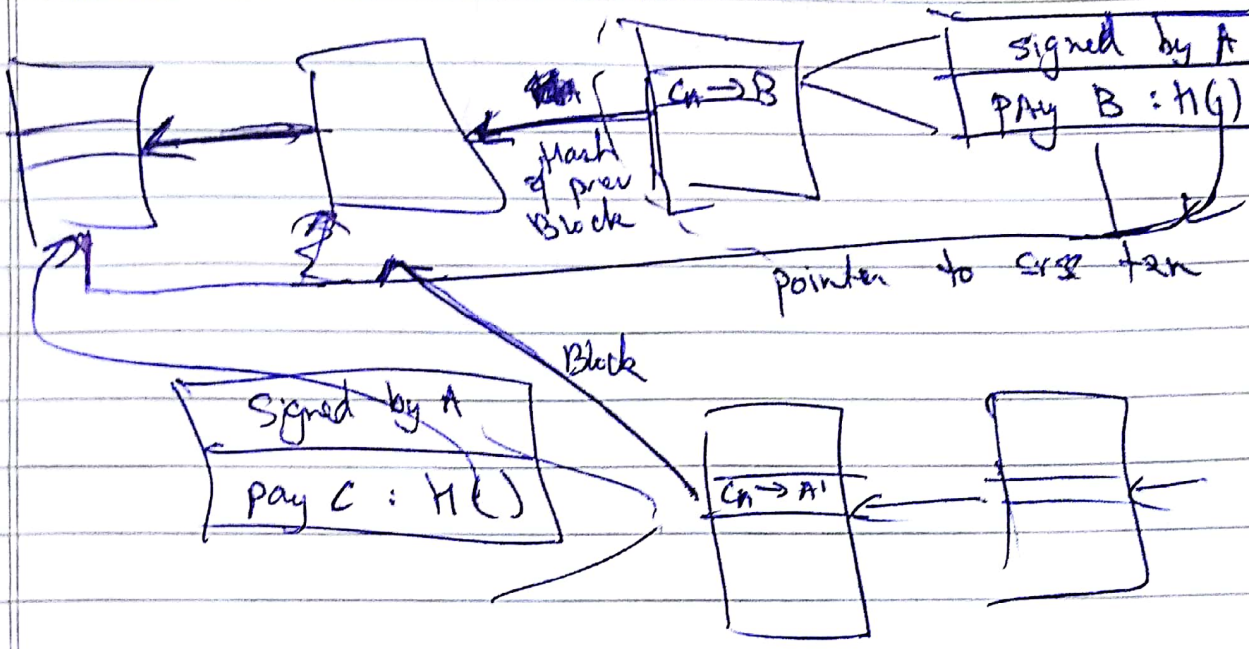
Each node collects broadcasts
Each round, random node proposes block.

Other nodes accept if transactions are valid.
i.e Signatures, hash, unspent txns.

Accept by mining on block

why does it work?

→ Stealing : transfer requires signatures (unforgeable)
→ DDoS : wait till next block.



~~How~~ Extend longest chain

Rounds : Mining

~~Coinbase~~ Coin mint to yourself.

Incentive for honesty? Block has to end up on long term chain.

transaction fee.

Rounds. Who proposes? Voting based on computing power (PoW) / competing.

H(nonce || prev hash || tx || tx || ...)

↑ brute force