Satoshi Nakamoto.

$USD → bitcoin

① Blockchain { immutable & tamper evident dB. } Data Structure
ADD only / append only.

Bitcoin { Accessible, Anonymous network ; no central registry }

② proof-of-work. / mining

[get rid of Bank]

Identity : users pick random private key —math fn→ public key 2^{160} grains of sand 2^{60}

Transaction : UTXO unspent Transaction output. ⇒ piggy bank Analogy
↳ can be only used once [balance goes into new a/c]

Record Keep : have everyone keep records (block-chain)
avoid [✗] double spend attack.
{Sybil attack} → multiple users to change vote. [n/w maj.]

Consensus { proof of work → ICPU } (computation power as voting power)

pseudonymous
Decentralized
Immutable
} Trustless

Ethereum – peer to peer smart Contracts

Blockchain Enterprise (banks) – distributed ledger
→ Hyperledger project

good wallet {parity}

Secure hash algo.

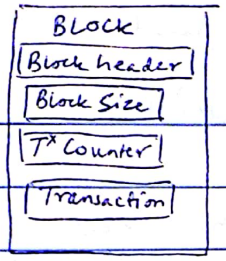
Cryptographic hash fingerprints are standardized randomness

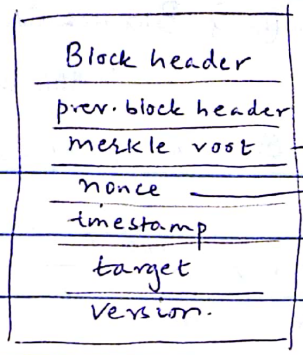
SHA-256 (NSA) ↓ built for security

input (x) —fn→ pseudo-random output.
same i/p → same o/p

① pre-image resistance
② second pre-image resistance
③ Collision resistance

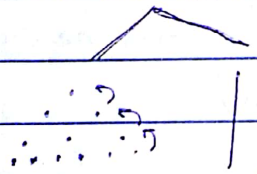
Avalanche effect (no pattern)





— Chain.
— summary of T^x
— proof of work.

Merkle Root (top of Tree) — specific version of binary tree



start with all T^x and get hash each one
then hash each pair together and continue until we have 1 root

$$\text{difficulty} \propto \frac{1}{\text{time to mine}}$$

elliptic curve + ECDSA
secp256k1

$y^2 = x^3 + ax + b$

ECDSA elliptic curve digital sign. algo.

Digital Signature Schemes (DSS)
- identify message origin
- non-repudiation (original sender should not be able to backtrack)
- message integrity (message can't be modified)

k (private key) —ECDSA→ K (public key) —hashing function→ A (bitcoin address)

SHA256
RIPEMD160
Base58Encode

{only forward}
{No backwards}
trapdoor
[chord tangent process]

Bitcoin Sender specify locking script
recipient provide unlocking script

P2SH

proof of burn → ethereum

→ User Experiences

↳ mining
routing
have full blockchain
key management

[wallet] secure private keys (store/access)

hot (internet connected) — AirBitz

Cold (no internet) — [brain wallet + key stretching]

Simple payment verification (SPV) — lightweight/thin clients
get block headers & run merkle proof of inclusion

Assumption : Chain you connect is honest

multisig → cryptographically sharing key b/w participants
have more points of failure < m of n >

JBOK Just a bunch of keys

HD Hierarchical Deterministic (start random seed)

[mining]

-get full blockchain, verify, create block, proof work nonce (solves partial preimage hash), broadcast, check profit

CPU → GPU → FPGA → ASIC