# Blockchain Fundamentals: A Quantitative Deep Dive

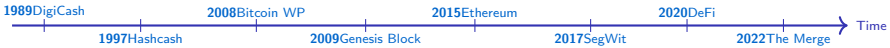
## Standalone Technical Lecture

Prof. Dr. Joerg Osterrieder

University Lecture Series

March 5, 2026

# Historical Timeline of Blockchain




## Foundations

- **1989**: DigiCash – blind signatures (Chaum)
- **1997**: Hashcash – proof-of-work for spam
- **2008**: Bitcoin whitepaper (Satoshi Nakamoto)
- **2009**: Genesis block mined, first BTC tx

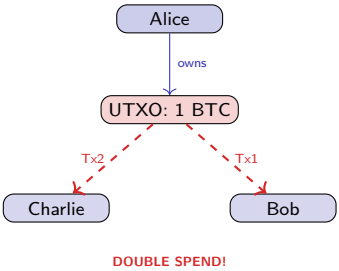
## Modern Era

- **2015**: Ethereum – Turing-complete contracts
- **2017**: SegWit fixes transaction malleability
- **2020**: DeFi summer, TVL exceeds $10B
- **2022**: Ethereum transitions to Proof of Stake

35

**years: from theoretical digital cash to global financial infrastructure**

# The Double-Spending Problem




## Formal Invariant

Let $\mathcal{U}$ be the UTXO set. The double-spend invariant requires:

$$\forall\, u \in \mathcal{U} : \sum_{T:\, u \in \mathrm{inputs}(T)} \mathbf{1}[\mathrm{conf}(T)] \leq 1$$

Each output may be consumed **at most once**.

## Why Difficult Without Central Authority?

- Network latency: conflicting txs arrive in different order at different nodes
- Sybil nodes selectively relay or suppress messages
- No global clock for deterministic ordering

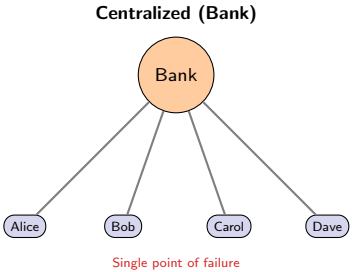
## Solution

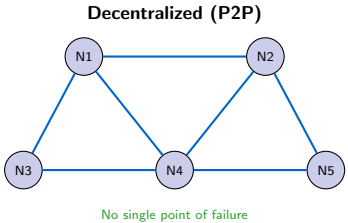
Nakamoto consensus: longest-chain rule with cumulative PoW defines canonical ordering.

**Bitco**

**solves double-spending without any trusted third party for the first time**

# Trust Models: Centralized vs. Decentralized




## Centralized

Fast, simple, regulated. But: censorable, single failure point, requires trust.

## Decentralized

Trustless, Byzantine fault tolerant, censorship-resistant. But: slower, complex.

**Block**

**replaces institutional trust with cryptographic and game-theoretic guarantees**

# Lecture Roadmap

## 8 Sections

1. **Introduction & Motivation** ✓
2. **Block Anatomy** – structure, UTXO, accounts
3. **Hash Functions** – SHA-256, properties
4. **Merkle Trees** – SPV proofs, tries
5. **Proof of Work** – mining, difficulty, energy
6. **Proof of Stake** – validators, Casper FFG
7. **Network** – P2P, propagation, forks
8. **Security & Future** – attacks, L2, quantum

## Learning Objectives

- Master blockchain data structures
- Derive mining difficulty mathematics
- Compare PoW vs. PoS security models
- Analyze network-level attack vectors
- Quantify decentralization with metrics

## Central Question

How do mutually distrusting parties reach consensus on a shared ledger history?

**Prere**

**probability theory, basic cryptography, graph theory**

# Learning Objectives

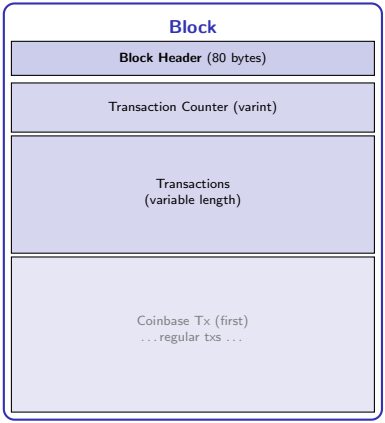
By the end of this lecture, you will be able to:

1. **Explain** the double-spending problem and how blockchain provides a trustless solution
2. **Describe** the internal structure of a block (header, body, Merkle root, hash pointers)
3. **Compare** Proof of Work and Proof of Stake consensus mechanisms quantitatively
4. **Analyze** the security guarantees of hash-chain immutability and collision resistance
5. **Evaluate** trade-offs in the blockchain trilemma (scalability, security, decentralization)

**Bloom**

**taxonomy levels: Remember → Understand → Apply → Analyze → Evaluate → Create**

# Block Data Structure




## Key Fields & Sizes

| Field | Size | Notes |
|---|---|---|
| Magic bytes | 4 B | 0xD9B4BEF9 |
| Block size | 4 B | bytes following |
| Header | 80 B | fixed |
| Tx count | 1–9 B | varint |
| Txns | var | coinbase first |

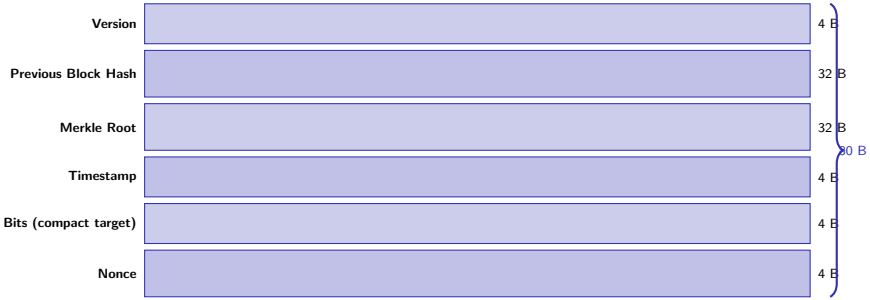
## Block Size Limits

- Bitcoin: 1 MB legacy / 4 MB weight
- Ethereum: target 15M gas / max 30M
- Larger blocks: more TPS, higher orphan rate

**Bitco**

**average block ≈1.5 MB; ≈2500 transactions; mined every 10 minutes**

# Block Header: 80 Bytes




## Field Roles

- **prevHash**: creates chain linkage
- **merkleRoot**: commits to all transactions
- **bits**: compact form of target $T$
- **nonce**: miner iterates this field

## Compact Target Decoding

$$T = \texttt{coeff} \times 256^{(\texttt{exp}-3)}$$

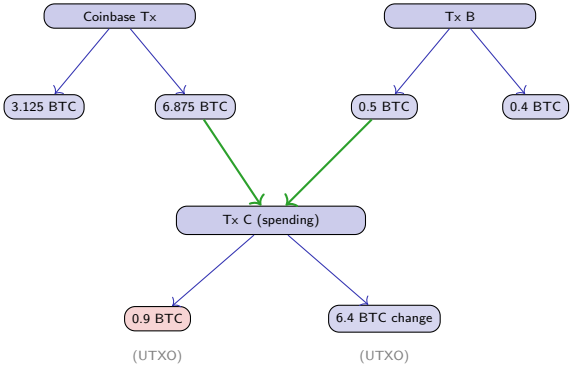
Mining succeeds when:

$$\text{SHA256}^2(\text{header}) < T$$

**Head**

**hashed twice (double-SHA256); nonce space $2^{32} = 4$B gives $\sim$4 GH per nonce cycle**

# The UTXO Model




## UTXO Properties

- Each output is atomic (no partial spend)
- Once spent, removed from UTXO set
- New outputs added when tx confirmed
- UTXO set fits in RAM (≈4 GB)

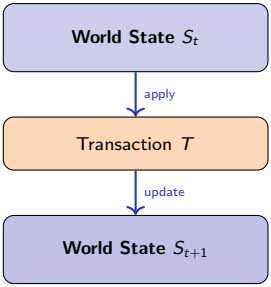
## Conservation Law

$$\sum_{\text{in}} v_i \geq \sum_{\text{out}} v_j$$

Difference = miner fee (implicitly claimed by coinbase)

**UTXO set: ≈4.5M entries (2024); stored in LevelDB on full nodes**

# Ethereum Account Model




$S_{t+1} = \Upsilon(S_t, T)$

## Account State Fields

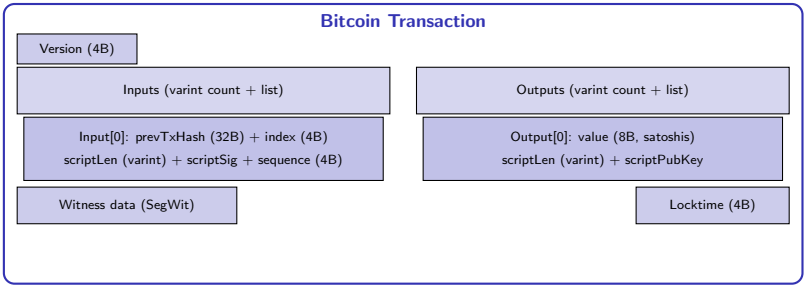
| Field | Meaning |
|---|---|
| nonce | tx count (EOA) / deploys (contract) |
| balance | ETH in Wei |
| codeHash | keccak256(bytecode) |
| storageRoot | Patricia trie root |

## EOA vs. Contract Account

- **EOA**: codeHash = keccak256(""), no storage; controlled by private key
- **Contract**: deployed bytecode; triggered by messages; has persistent storage

**State transition $\Upsilon$ defined in Ethereum Yellow Paper; gas limits computational cost**

# Bitcoin Transaction Anatomy




## Script Execution

Stack-based. P2PKH scriptPubKey:
`OP_DUP OP_HASH160 <hash> OP_EQUALVERIFY OP_CHECKSIG`

## Transaction ID

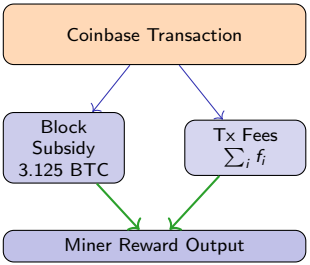
$$\texttt{txid} = \text{SHA256}(\text{SHA256}(\text{raw bytes}))$$

SegWit: `wtxid` includes witness. Segwit discount: witness bytes cost 0.25 weight units.

**Typic**

**P2PKH tx ≈250 bytes; SegWit P2WPKH ≈141 vBytes (virtual bytes)**

# Coinbase Transaction & Bitcoin Supply




## Halving Schedule

$$R(n) = \frac{50}{2^n} \text{ BTC}, \quad n = \left\lfloor \frac{h}{210{,}000} \right\rfloor$$

| Halving | Height | Reward |
|---|---|---|
| 0 | 0 | 50 BTC |
| 1 | 210,000 | 25 BTC |
| 3 | 630,000 | 6.25 BTC |
| 4 | 840,000 | 3.125 BTC |

## Total Supply (Geometric Series)

$$\sum_{n=0}^{\infty} 210{,}000 \cdot \frac{50}{2^n} = 210{,}000 \cdot 100 = 21{,}000{,}000 \text{ BTC}$$

**Last**

**Bitcoin mined ≈2140; thereafter miners earn transaction fees only**

# Cryptographic Hash Function Properties

## Definition

A hash function $H : \{0,1\}^* \to \{0,1\}^n$ must satisfy three hardness properties:

### 1. Preimage Resistance

Given $y$, hard to find $x$ s.t. $H(x) = y$.

Security: $\mathcal{O}(2^n)$ work.

*"One-way function"*

### 2. Second Preimage

Given $x$, hard to find $x' \neq x$ s.t. $H(x') = H(x)$.

Security: $\mathcal{O}(2^n)$ work.

*"Weak collision"*

### 3. Collision Resistance

Hard to find any $x \neq x'$ s.t. $H(x) = H(x')$.

Security: $\mathcal{O}(2^{n/2})$ via birthday.

*"Strong collision"*

## SHA-256 Parameters

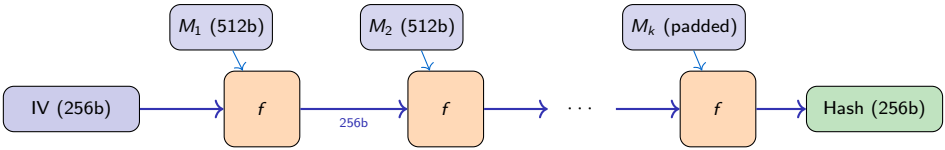
- Output: 256 bits = 32 bytes
- Block size: 512 bits
- Rounds: 64
- State: 8 × 32-bit words (256 bits)

## Bitcoin Usage

- Block hash: $\text{SHA256}^2(\text{header})$
- TXID: $\text{SHA256}^2(\text{raw tx})$
- P2PKH: $\text{RIPEMD160}(\text{SHA256}(\text{pubkey}))$
- Merkle nodes: $\text{SHA256}^2(\text{left} \| \text{right})$

**SHA-256: NIST standard (2001); no practical collision known; 128-bit post-quantum security**

# SHA-256: Merkle–Damgård Construction




## Compression Function $f$

Each call processes 512-bit block with 256-bit state. 64 rounds using message schedule $W_t$:

$$W_t = \sigma_1(W_{t-2}) + W_{t-7} + \sigma_0(W_{t-15}) + W_{t-16}$$

where $\sigma_0, \sigma_1$ are bitwise rotate/shift operations.

## Padding Rule

Append bit 1, then zeros, then 64-bit message length, to reach multiple of 512 bits:

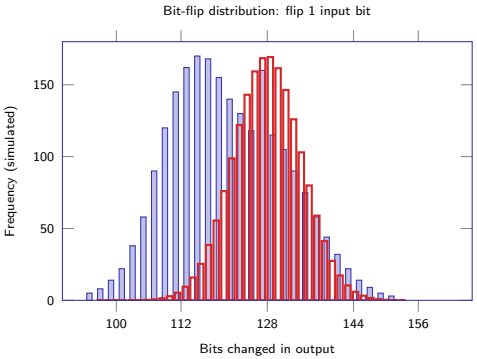
$$|M| + 1 + k + 64 \equiv 0 \pmod{512}$$

Length encoding prevents length-extension issues (partially).

**Bitco**

**uses SHA256$^2$: double-hash prevents length-extension attacks on Merkle trees**

# Avalanche Effect




## Avalanche Property

Flipping 1 input bit flips $\approx n/2$ output bits:

$$E[\Delta \text{ bits}] = \frac{n}{2} = 128$$

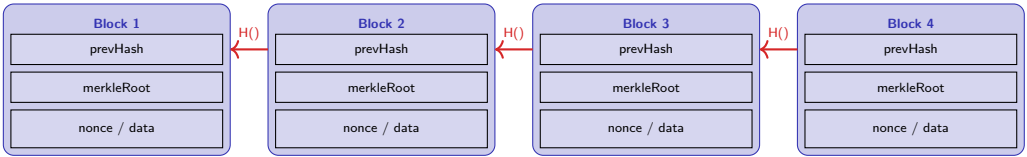
Variance $\approx n/4$; distribution $\approx \mathcal{N}(n/2, n/4)$.

## Implications for PoW

- Incrementing nonce produces completely different hash
- No shortcut: must try each nonce independently
- $P(\text{valid hash}) = T/2^{256}$ per attempt
- Mining is memoryless Bernoulli process

**Strict avalanche criterion (SAC): flip any single input bit $\Rightarrow$ each output bit flips with prob 1/2**

# Hash Pointers: Building the Chain




Tamper block 2 $\Rightarrow$ block 3's prevHash invalid
$\Rightarrow$ cascade invalidation to tip

## Tamper Detection

If adversary modifies block $k$:

1. Block $k$'s hash changes
2. Block $k+1$'s prevHash no longer matches
3. All subsequent blocks invalid
4. Must re-mine from block $k$ onward

## Immutability Cost

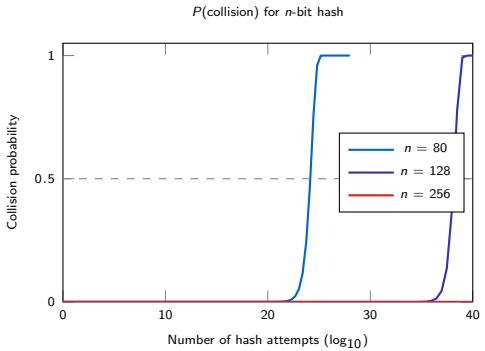
Cost to rewrite history from block $k$:

$$\text{Work} = \sum_{i=k}^{\text{tip}} D_i \cdot 2^{32} \text{ hashes}$$

With current difficulty $D \approx 10^{13}$, rewriting 6 blocks requires $\approx 6 \times 10^{13} \times 2^{32}$ hashes.

**Hash pointers provide tamper-evidence; PoW makes tampering computationally infeasible**

# Birthday Paradox & Collision Security




## Birthday Bound

For $n$-bit hash, expect first collision after $\approx 2^{n/2}$ attempts:

$$P(\text{collision after } k) \approx 1 - e^{-k^2/2^{n+1}}$$

At $k = 2^{n/2}$: $P \approx 1 - e^{-1/2} \approx 0.39$.

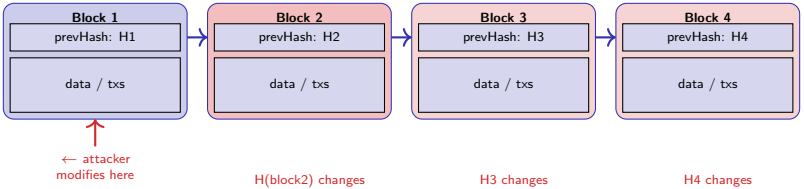
## SHA-256 Security

- Preimage: $2^{256}$ operations
- Collision: $2^{128}$ operations
- $2^{128} \approx 3.4 \times 10^{38}$: infeasible classically
- Quantum (Grover): $2^{64}$ – still large

**Birth**

**attack on 256-bit hash requires $2^{128}$ hashes; Bitcoin network does $\sim 10^{20}$ hashes/year**

# Cascade Invalidation: Tamper Proof by Design




All blocks from modified point onward must be re-mined

## Re-mining Cost

To rewrite from block $k$ to current height $N$:

$$\text{Expected hashes} = \sum_{i=k}^{N} D_i \times 2^{32}$$

$\approx (N - k) \times D \times 2^{32}$ if difficulty constant.

## 6-Confirmation Rule

After 6 confirmations, probability honest chain extended faster than attacker chain (with $q < 0.3$):

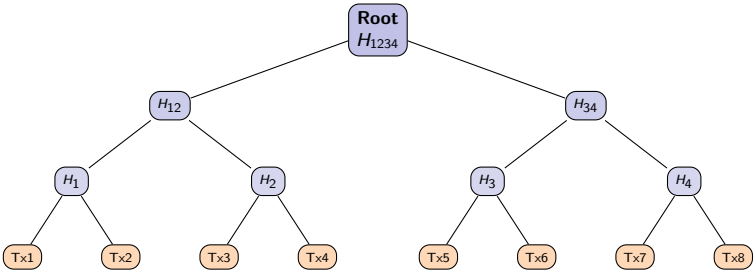
$$P(\text{success}) < 0.001$$

Merchants wait 6 blocks for high-value payments.

**Immu**

**is probabilistic: each additional block exponentially increases rewrite cost**

# Merkle Tree: Binary Hash Tree



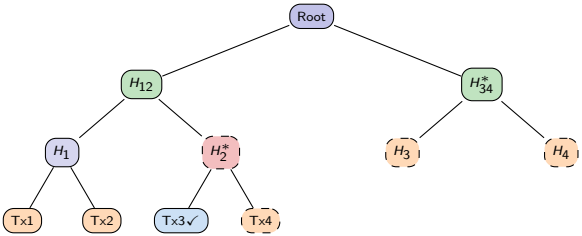

## Construction Algorithm

$H_i = \text{SHA256}^2(\text{Tx}_i)$; $H_{ij} = \text{SHA256}^2(H_i \| H_j)$; Root$= \text{SHA256}^2(H_{12} \| H_{34})$. For odd count: last leaf duplicated. Root stored in block header.

**Merk**

**root: 32-byte commitment to all transactions; any tx change $\Rightarrow$ root changes**

# Merkle Proof & SPV Clients




Prove Tx3 included: need $H^*_2$, $H_{12}$, $H^*_{34}$

## Proof Size

For $n$ transactions, proof requires $\lceil \log_2 n \rceil$ hashes:

| $n$ txs | Proof hashes | Size |
|---|---|---|
| 16 | 4 | 128 B |
| 1,000 | 10 | 320 B |
| 1M | 20 | 640 B |
| 1B | 30 | 960 B |

## SPV Client

Downloads headers only (80 B/block). Requests Merkle proof for specific tx. Verifies without full blockchain (<1 GB vs >500 GB).

**SPV:**

**Simplified Payment Verification (Bitcoin whitepaper, Section 8); enables lightweight wallets**

# Merkle Proof Verification

## Pseudocode

```
def verify_merkle_proof(tx, proof, root):
    """
    tx    : transaction bytes
    proof : list of (hash, side) pairs
            side in {'left', 'right'}
    root  : claimed Merkle root
    """
    current = sha256d(tx)
    for (sibling, side) in proof:
        if side == 'left':
            current = sha256d(sibling + current)
        else:
            current = sha256d(current + sibling)
    return current == root

# Example: prove Tx3 in 8-tx block
proof = [
    (H_Tx4, 'right'),    # H2 = H(Tx3||Tx4)
    (H_12,  'left'),     # H_1234 = H(H12||H34)
    (H_34,  'right'),
]
assert verify_merkle_proof(Tx3, proof, root)
```

## Complexity Analysis

| Operation | Cost |
|---|---|
| Build tree | $\mathcal{O}(n)$ hashes |
| Proof size | $\mathcal{O}(\log n)$ hashes |
| Verification | $\mathcal{O}(\log n)$ hashes |
| Storage (full) | $\mathcal{O}(n)$ hashes |

## Security

An adversary cannot forge a valid proof for a transaction not in the tree (preimage resistance + collision resistance of SHA-256).

## CVE Note

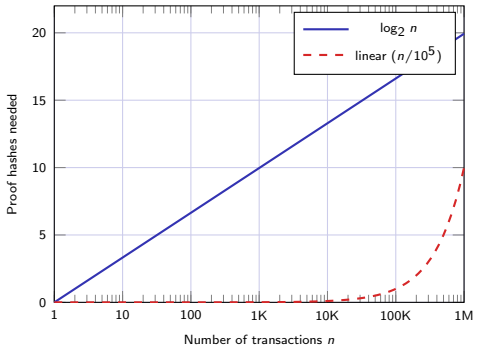
Duplicate-leaf vulnerability: duplicate internal nodes can cause false inclusion proofs in some implementations.

Bitco

**Core: merkle.cpp; OpenSSL used until 2015; now native SHA-256 implementation**

# Merkle Tree Efficiency

Proof size: $\lceil \log_2 n \rceil$ vs $n$




## Why Logarithmic Matters

- 1 million transactions in block
- Linear proof: 1,000,000 hashes = 32 MB
- Merkle proof: 20 hashes = 640 bytes
- Speedup: $\times 50,000$

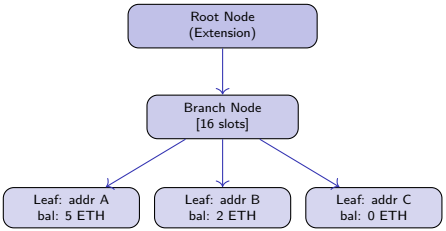
## Applications Beyond Bitcoin

- Git: content-addressed object store
- Certificate transparency logs
- IPFS / Filecoin storage proofs
- zk-SNARKs: Merkle membership proofs
- Ethereum state trie (Patricia Merkle)

**Merk**

**trees: Ralph Merkle (1979); fundamental data structure for authenticated data**

# Patricia Merkle Trie (Ethereum)




Each node: keccak256(RLP(node))

## Three Node Types

- **Extension**: shared prefix compression
- **Branch**: 16-slot array (hex nibble)
- **Leaf**: terminal key-value pair

## Four Ethereum Tries

| Trie | Keys / Values |
|---|---|
| State | address → account |
| Storage | slot → value (per contract) |
| Transaction | index → tx |
| Receipt | index → receipt |

**Patri**

**Merkle Trie: combines Patricia trie (space efficiency) + Merkle (authentication)**

# Proof of Work: Mathematical Definition

## Mining Condition

$$\text{SHA256}^2(\underbrace{\text{version}\|\text{prevHash}\|\text{merkleRoot}\|\text{time}\|\text{bits}}_{\text{fixed for mining round}}\|\underbrace{\text{nonce}}_{\text{varied}}) < T$$

## Target $T$ and Difficulty $D$

$$T = \frac{T_{\max}}{D}, \qquad T_{\max} = 2^{224}$$

$$D \approx \frac{2^{224}}{T}$$

Current Bitcoin difficulty: $D \approx 8.7 \times 10^{13}$ (2024).

Expected hashes per block:

$$E[\text{hashes}] = D \times 2^{32} \approx 3.7 \times 10^{23}$$

## Probabilistic Interpretation

Each hash attempt is an independent Bernoulli trial:

$$p = P(\text{valid hash}) = \frac{T}{2^{256}}$$

$$p \approx \frac{1}{D \times 2^{32}}$$

Number of attempts to find valid hash:

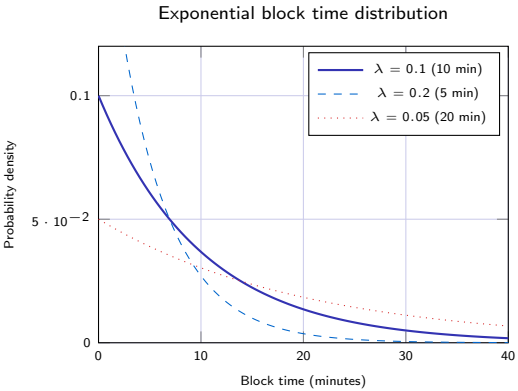
$$N \sim \text{Geometric}(p)$$

$$E[N] = \frac{1}{p} = D \times 2^{32}$$

PoW

**computationally expensive to produce, trivially cheap to verify; asymmetric by design**

# Mining: Geometric Distribution




## Memoryless Property

Block inter-arrival time: $T \sim \text{Exp}(\lambda)$

$$P(T > t) = e^{-\lambda t}$$

$$\lambda = \frac{H}{D \times 2^{32}}$$

where $H$ = network hash rate.

**Memoryless**: mining difficulty unchanged regardless of how long since last block.

## Multiple Miners

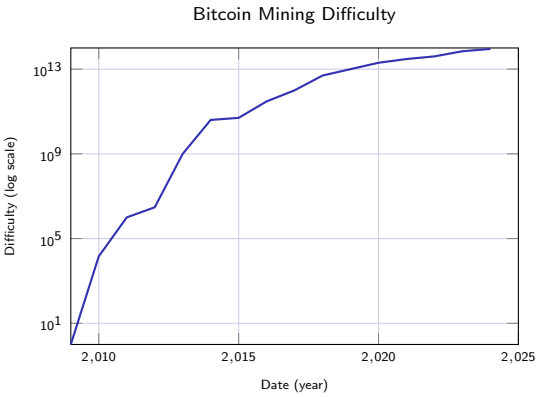
With $n$ miners each with hash rate $h_i$:

$$P(\text{miner } i \text{ wins}) = \frac{h_i}{\sum_j h_j}$$

Mining is a race: proportional to hash power.

**targets $\lambda = 1/600$ (one block per 10 minutes); adjusted every 2016 blocks**

# Difficulty Adjustment Algorithm




## Adjustment Formula

Every 2016 blocks (≈2 weeks):

$$D_{\text{new}} = D_{\text{old}} \times \frac{t_{\text{actual}}}{t_{\text{target}}}$$

where $t_{\text{target}} = 2016 \times 600 = 1{,}209{,}600$ s.

**Clamped:** factor bounded by $[1/4, 4]$ to prevent extreme swings.

## Equilibrium Condition

At steady state:

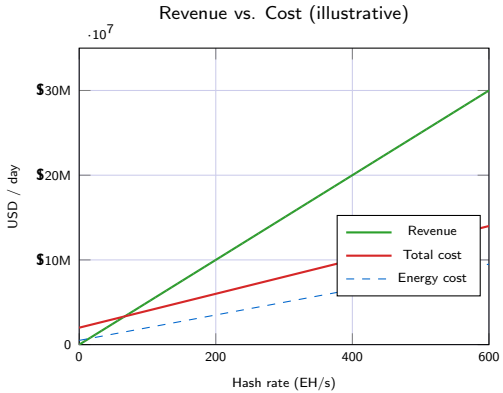
$$H = \frac{D \times 2^{32}}{600} \text{ H/s}$$

With $D = 8.7 \times 10^{13}$: $H \approx 600$ EH/s.

Diffic

**retarget: Bitcoin's self-regulating mechanism; hash rate tripled in 2023 yet blocks stay at 10 min**

# Mining Economics




## Revenue Formula

$$\text{Rev}_i = \frac{h_i}{H}\left(R_{\text{sub}} + \sum_j f_j\right) \cdot P_{\text{BTC}}$$

where $h_i$ = miner hash rate, $H$ = total, $R_{\text{sub}}$ = block subsidy, $f_j$ = fees.

## Break-Even Condition

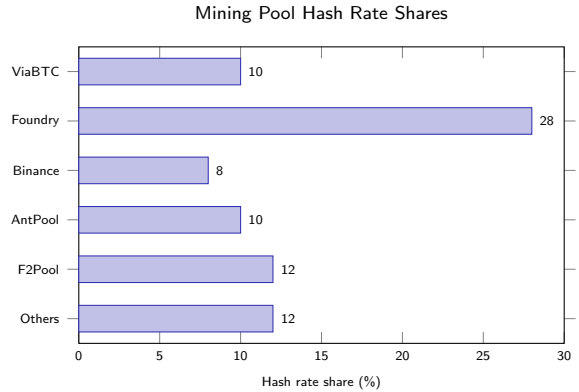
$$\frac{h_i}{H} \cdot R \cdot P > c_{\text{elec}} \cdot h_i \cdot \frac{E}{\text{TH}}$$

$c_{\text{elec}}$: electricity cost (\$/kWh)
$E/\text{TH}$: energy efficiency (J/TH)
**Efficient miners survive; inefficient exit.**

**Bitco**

**mining: ∼\$20B revenue/year (2024); dominated by ASIC hardware (Antminer S21: 200 TH/s, 17.5 J/TH)**

# Global Hash Rate Distribution (2024)




## Pool Economics

- Individual variance $\rightarrow$ pools reduce variance
- PPS (Pay Per Share): fixed rate per hash
- PPLNS: pay per last N shares
- Solo mining: $\sigma^2 \propto 1/h_i$

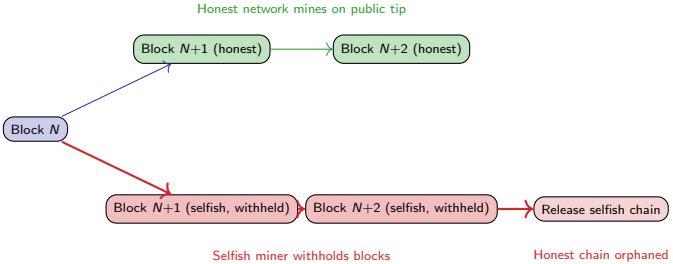
## Concentration Risk

Top 3 pools often $> 50\%$ of hash rate.
Theoretical 51% attack possible if colluding.

**Nakamoto coefficient**: min pools needed to exceed 50% = 2–3 (Bitcoin, 2024).

**centralization: geographic (60% USA post-China ban) and pool-level concentration**

# Selfish Mining Attack




## Selfish Mining Threshold

A pool with fraction $\alpha$ of hash rate can profitably selfish-mine when:

$$\alpha > \frac{1-\gamma}{3-2\gamma}$$

where $\gamma$ = fraction of honest miners who adopt the selfish chain when published simultaneously. Default: $\gamma = 0$ gives $\alpha > 1/3$.

## Revenue Advantage

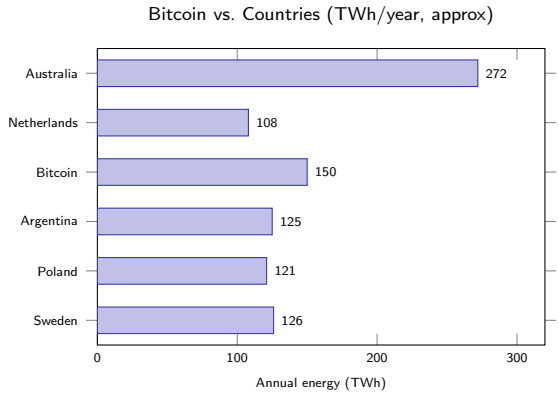
Relative revenue of selfish miner:

$$R_{\text{selfish}} = \frac{\alpha(1-\alpha)^2(4\alpha + \gamma(1-2\alpha)) - \alpha^3}{1-\alpha(1+(2-\alpha)\alpha)}$$

Exceeds $\alpha$ for $\alpha > 1/3$ (worst case).

**Eyal & Sirer (2014): selfish mining profitable below 50%; mitigated by FIFO propagation**

# Bitcoin Energy Consumption




## Energy Model

$$\text{Energy} = H \cdot \frac{E_{\text{eff}}}{\text{TH/s}} \cdot \frac{1}{10^{12}} \text{ TW}$$

At $H = 600$ EH/s, $E_{\text{eff}} = 25$ J/TH:

$$\approx 600 \times 10^{18} \times 25/(10^{12}) \approx 15 \text{ GW}$$

$15 \text{ GW} \times 8760 \text{ h} \approx 131 \text{ TWh/yr}$

## PoS Comparison

Ethereum post-Merge: ≈0.01 TWh/yr (99.95% reduction). Trade-off: different security model (economic vs. energy).

**Bitco**

**energy ≈ 0.4% global electricity; % of renewables disputed (est. 25–75%)**

## Validator Selection Probability

$$P(\text{validator } i \text{ selected}) = \frac{s_i}{\sum_{j=1}^{n} s_j}$$

where $s_i$ is the stake of validator $i$. Selection is proportional to economic stake.

## Ethereum 2.0 Parameters

| Parameter | Value |
|---|---|
| Min validator stake | 32 ETH |
| Max effective balance | 32 ETH |
| Epoch length | 32 slots |
| Slot duration | 12 seconds |
| Committee size | $\geq$128 validators |
| Finality | 2 epochs ($\approx$12.8 min) |

## PoS Security Model

Attack requires controlling $> 1/3$ of staked ETH.

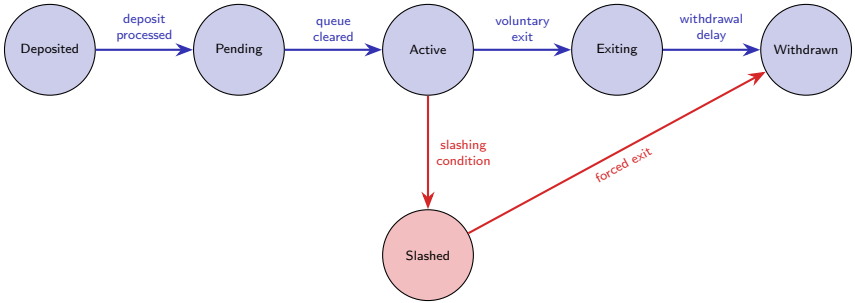
Total staked: $\approx$ 30M ETH (2024).
Attack cost: $\frac{1}{3} \times 30\text{M ETH} \times \$3000 = \$30\text{B}$

Additionally: slashing destroys attacker's stake, making attack economically self-destructive.

Ether

**PoS: $\sim$900,000 active validators (2024); $\approx$28% of all ETH staked**

# Validator Lifecycle Automaton




## Queue Mechanics

Entry and exit queues limit churn: max $\lfloor n_{\text{validators}}/65536 \rfloor$ per epoch. At 900K validators: $\approx$13 entries/exits per epoch.

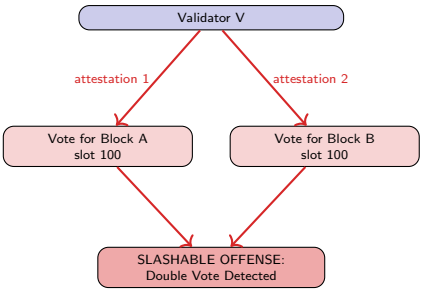
## Withdrawal Delay

After exit request: $\sim$256 epochs ($\approx$27 hours) before ETH unlocks. Prevents coordinated exit attacks.

**Valid**

**lifecycle: designed to maintain stability; sudden exits would threaten finality**

# Slashing Conditions




Penalty: 1/32 of balance + correlation penalty

## Two Slashing Offenses

**1. Double voting (equivocation):**
Sign two different blocks at the same slot.

$$h_1 = h_2, \quad B_1 \neq B_2$$

**2. Surround voting:**
Vote with attestation that surrounds or is surrounded by a prior vote.

$$(e_1 < e_2 < e_3 < e_4) \wedge (e_1, e_4) \text{ surrounds } (e_2, e_3)$$

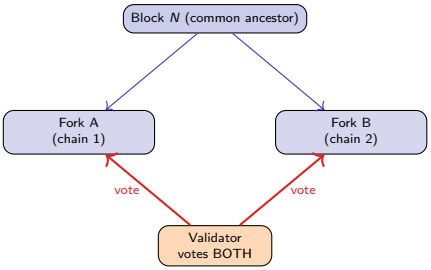
## Penalty Severity

Correlation penalty: if many validators slashed simultaneously, penalty scales up to 100% of stake (prevents coordinated attacks).

**Slash**

**game-theoretic deterrent; attacker loses stake while honest validators lose at most minor inactivity penalty**

# Nothing-at-Stake Problem




Rational: no cost to vote on both!

(without slashing mechanism)

## The Problem

In naive PoS: validator has no reason *not* to vote on all forks. Cost = 0; expected reward = doubled.

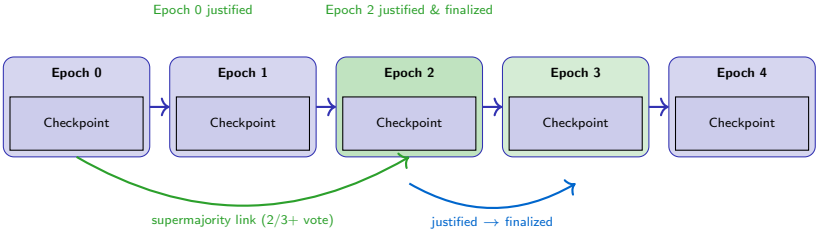
## Solutions

- **Slashing**: double-voting destroys stake (economic deterrent)
- **Long-range attacks**: stake grinding mitigated by checkpointing
- **Finality gadgets** (Casper): make certain chain history permanent
- **Weak subjectivity**: new nodes accept recent finalized checkpoint

**Noth**

**at-stake solved by slashing; long-range attacks mitigated by weak subjectivity checkpoints**

# Casper FFG: Finality Gadget




## Justification & Finalization

A checkpoint $C$ is **justified** if $\geq 2/3$ of validators vote for a supermajority link from a justified ancestor to $C$.

$C$ is **finalized** if $C$ is justified and the next checkpoint is also justified via a direct link from $C$.

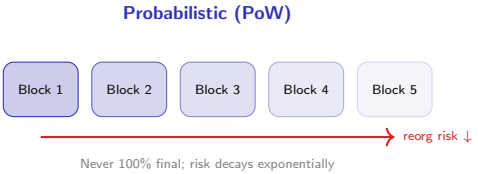
## Safety Theorem

Under Casper FFG: two conflicting checkpoints *cannot both be finalized* unless $\geq 1/3$ of validators are slashed.

$$\text{If } B_1 \perp B_2 \text{ both finalized} \Rightarrow \geq \frac{1}{3} \text{ stake slashed}$$

**FFG (Buterin & Griffith, 2017): first formal finality gadget with accountable safety**

# Block Finality: Probabilistic vs. Deterministic




## Probabilistic Finality (Bitcoin)

- Confidence grows with each confirmation
- After $z$ blocks: $P(\text{reorg}) = (q/(1-q))^z$
- 6 confirmations: $\sim$60 min, $P < 0.001$
- Never mathematically "final"

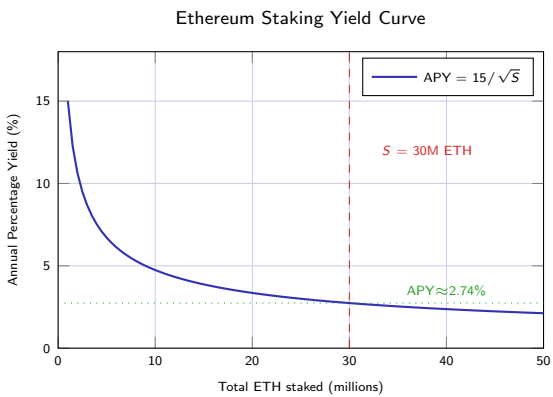
## Deterministic Finality (Ethereum)

- Casper FFG: finalized after 2 epochs ($\sim$13 min)
- Reversal requires slashing $\geq 1/3$ of total stake
- At 30M ETH staked: attack cost $>$ \$10B
- Tendermint (Cosmos): single-slot finality ($\sim$6s)

**Finali**

**type determines confirmation time: PoW chains need minutes; BFT chains can finalize in seconds**

# Staking Economics: Yield Curve




## Issuance Model

Base reward per validator:

$$r = \frac{B_{\text{eff}} \times c}{\sqrt{\sum s_i}}$$

where $c$ is the base reward factor (64 in Ethereum), $B_{\text{eff}}$ = effective balance.

Total issuance $\propto \sqrt{N_{\text{validators}}}$, but per-validator yield $\propto 1/\sqrt{N}$.

## Nash Equilibrium

Staking equilibrium where marginal validator is indifferent between staking and alternatives. Current: $\sim$3–4% APY.

Ether

**issuance: $\approx$600K ETH/year at 30M staked; deflationary post-EIP-1559 if burn $>$ issuance**

# Proof of Work vs. Proof of Stake

| Property | Proof of Work | Proof of Stake |
|---|---|---|
| Security resource | Computation (hash rate) | Capital (staked tokens) |
| Attack cost | $> 50\%$ hash rate | $> 33\%$ staked value |
| Energy use | $\sim$130 TWh/yr (BTC) | $\sim$0.01 TWh/yr (ETH) |
| Finality | Probabilistic | Deterministic (Casper) |
| Hardware req. | ASIC / GPU | Standard computer |
| Centralization | Mining pools (geographic) | Stake concentration |
| Bootstrapping | CPU mining egalitarian | Rich-get-richer tendency |
| Reorg defense | Accumulated PoW | Slashing mechanism |
| Quantum risk | Moderate (Grover $\times 2$) | ECDSA key exposure |
| Proven security | 15+ years (BTC) | 2+ years (ETH) |

## PoW Advantage

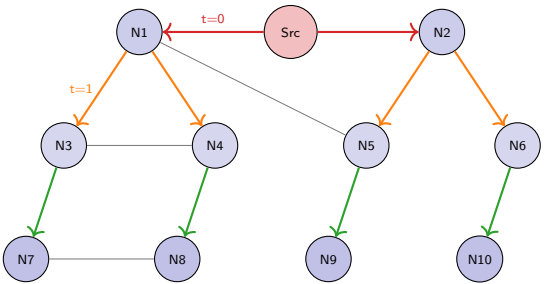
Longest-chain rule is simple; security doesn't depend on protocol complexity; no slashing; no stake nothing-at-stake.

## PoS Advantage

Orders of magnitude less energy; fast finality; no ASIC arms race; stake as bond enables slashing deterrent.

**No consensus on "better": PoW = energy security model; PoS = economic security model**

# P2P Network Topology & Gossip Propagation




Gossip: each node forwards to $k$ peers

## Gossip Protocol

Each node maintains ∼8–125 peer connections. New block/tx: node relays to all peers except sender ("flooding").

Propagation time with $k$ connections per node:

$$t_{\text{prop}} \approx \frac{\log N}{\log k} \cdot t_{\text{hop}}$$

For $N = 10{,}000$ nodes, $k = 8$, $t_{\text{hop}} = 100$ms: $t \approx 1.3$ s.

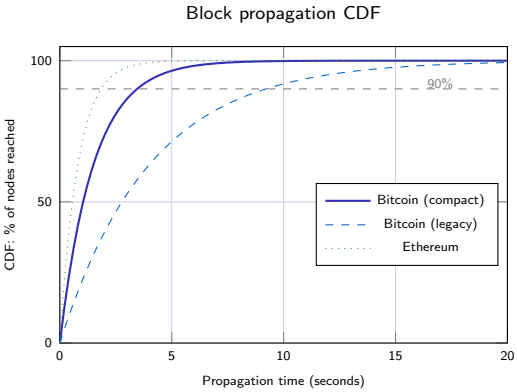
## Compact Block Relay

Bitcoin: send header + short tx IDs; receiver reconstructs from mempool. Reduces bandwidth by ∼98%.

**Bitco**

**P2P: ∼15,000 reachable full nodes; Ethereum: ∼7,000 nodes (2024)**

# Block Propagation Time Distribution




## Stale Block Rate

Probability block becomes stale (orphaned):

$$\rho \approx 1 - e^{-\lambda t_{\text{prop}}}$$

For $t_{\text{prop}} = 2\text{s}$, $\lambda = 1/600$: $\rho \approx 0.33\%$

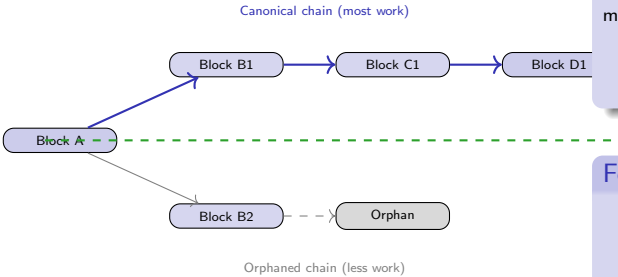
**GHOST protocol** (Ethereum pre-merge): include "uncle" blocks to reduce waste.

## FIBRE Network

Fast Internet Bitcoin Relay Engine: dedicated relay network reduces median propagation to $<$100ms for mining pools.

**Slow propagation $\Rightarrow$ higher orphan rate $\Rightarrow$ large miners advantaged (pool centralization pressure)**

# Fork Types & Resolution




## Nakamoto Longest Chain Rule

When two valid chains exist, nodes always extend the chain with most accumulated proof-of-work (not longest by block count).

$$\text{canonical} = \arg\max_{\text{chain}} \sum_{\text{blocks}} D_i$$

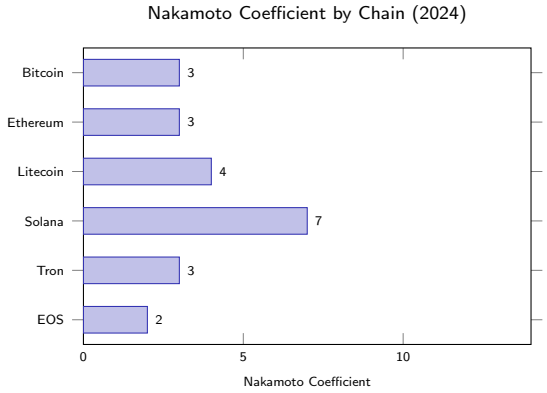
## Fork Types

- **Accidental**: natural propagation delay; resolved in 1–2 blocks
- **Soft fork**: tightens rules (backward compatible)
- **Hard fork**: loosens or changes rules (not backward compatible)
- **Contentious hard fork**: chain split (e.g., BCH, ETC)

**Orph**

**rate: ∼0.3% Bitcoin; Ethereum had ∼5% uncle rate pre-merge (12s blocks)**

# Nakamoto Coefficient: Measuring Decentralization




## Definition

**Nakamoto coefficient** $\mathcal{N}$: minimum number of entities that must collude to compromise consensus.

$$\mathcal{N} = \min k : \sum_{i=1}^{k} s_i > \frac{1}{2} \sum_{i=1}^{n} s_i$$

Higher $\mathcal{N}$ = more decentralized.

## Dimensions Measured

- Mining pools / validator sets
- Exchange custody
- Client software diversity
- Geography of nodes
- Developer concentration

**Naka**

**coefficient: single-number proxy for decentralization; true security requires multidimensional analysis**

# Quantifying Decentralization

## Gini Coefficient

Measures stake/hash rate inequality:

$$G = \frac{\sum_{i=1}^{n} \sum_{j=1}^{n} |s_i - s_j|}{2n \sum_{i=1}^{n} s_i}$$

$G = 0$: perfect equality; $G = 1$: one entity holds all.

Bitcoin mining pools: $G \approx 0.7$
Ethereum validators: $G \approx 0.6$

## Shannon Entropy

$$H = -\sum_{i=1}^{n} p_i \log_2 p_i$$

where $p_i = s_i / \sum_j s_j$.
Max entropy: $\log_2 n$ (all equal shares).

## Herfindahl–Hirschman Index

$$\text{HHI} = \sum_{i=1}^{n} \left( \frac{s_i}{\sum_j s_j} \right)^2 \times 10{,}000$$

HHI $< 1500$: competitive market.
HHI $> 2500$: highly concentrated.

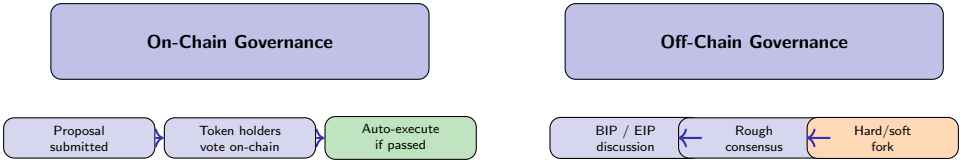
**Example**: 3 pools with 33% each:
HHI $= 3 \times (33)^2/100 = 3267$ (concentrated)

## Limitation

All metrics capture a single dimension. Real decentralization requires: geographic, client, ownership, and governance analysis.

**is multidimensional; no single metric captures all aspects of blockchain security**

# Blockchain Governance Models




## On-Chain (Tezos, Polkadot)

- Protocol upgrades via token voting
- Automatic enactment after threshold
- DAOs: Decentralized Autonomous Organizations
- Risk: plutocracy (rich dominate votes)

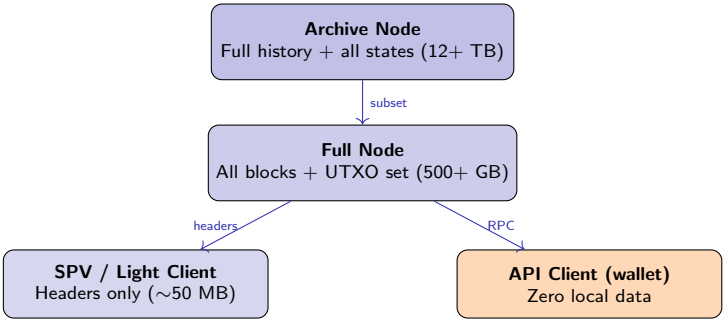
## Off-Chain (Bitcoin, Ethereum)

- BIPs/EIPs: formal improvement proposals
- Core devs, miners, users signal support
- Contentious changes → hard fork (e.g., BCH 2017)
- Slower but more conservative; "code is not law"

**Gove**

**determines how blockchains evolve; no perfect model – trade-off between speed and decentralization**

# Node Types: Full Nodes vs. Light Clients




## Full Node Security Model

Validates every transaction and block independently. Never trusts anyone – verifies all rules. Provides strongest security but requires >500 GB storage and ∼days to sync.

## Light Client Security Model

Trusts longest chain of headers (SPV). Requests Merkle proofs for specific transactions. Cannot detect invalid blocks unless > 50% honest miners. Suitable for mobile wallets.

**Light clients trust miners; full nodes trust no one. Bitcoin: ∼15K full nodes; Ethereum: ∼7K**

# 51% Attack: Quantitative Analysis

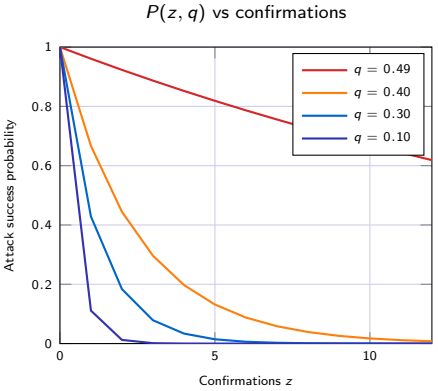
## Nakamoto's Formula

Let $q$ = attacker hash fraction, $z$ = confirmations.
Probability attacker catches up from $z$ blocks behind:

$$P(z,q) = \begin{cases} 1 & \text{if } q \geq 0.5 \\ \left(\frac{q}{1-q}\right)^z & \text{if } q < 0.5 \end{cases}$$

## Numerical Values

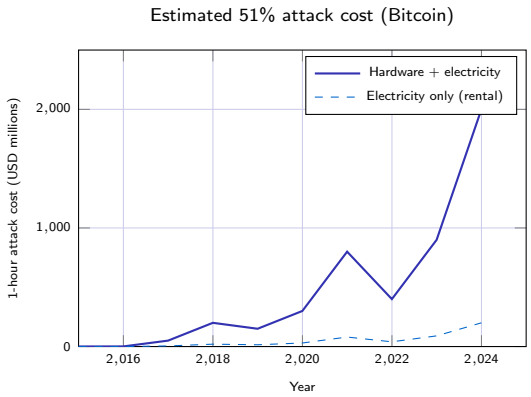
| $q$ | $z=1$ | $z=3$ | $z=6$ | $z=10$ |
|---|---|---|---|---|
| 0.10 | 0.012 | 0.002 | $< 10^{-4}$ | $10^{-7}$ |
| 0.30 | 0.184 | 0.062 | 0.012 | 0.002 |
| 0.40 | 0.444 | 0.296 | 0.175 | 0.085 |
| 0.49 | 0.960 | 0.885 | 0.784 | 0.620 |




6

**confirmations: standard for Bitcoin; $q = 0.3$: $P \approx 1.2\%$ after 6 blocks – acceptable risk for most merchants**

# 51% Attack Cost Over Time




## Attack Cost Formula

$$\text{Cost}_{\text{hw}} = \frac{H}{2} \cdot \frac{P_{\text{ASIC}}}{\eta_{\text{ASIC}}}$$

$$\text{Cost}_{\text{elec}} = \frac{H}{2} \cdot E_{\text{eff}} \cdot c_{\text{elec}}$$

$H/2$: need 50% of current hash rate
$P_{\text{ASIC}}$: ASIC price per TH/s
$E_{\text{eff}}$: J/TH efficiency

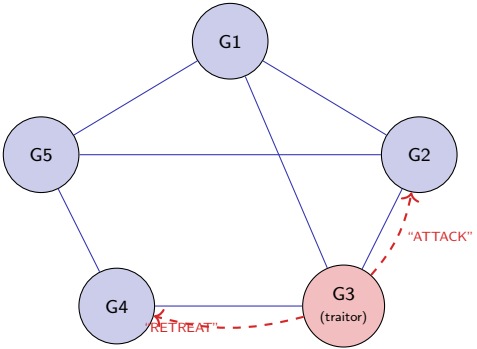
## Small Chain Attacks

Crypto51.app tracks rental costs. ETC attacked in 2020: 4 attacks, cost ∼$200K each; stolen ∼$5M. Altcoin defense: merge mining, finality checkpoints.

Bitco

**51% attack: $>$ $2B hardware + ongoing electricity; economically irrational (destroys attacker's own BTC holdings)**

# Byzantine Fault Tolerance




G3 sends conflicting messages

## Byzantine Generals Problem

$n$ generals must agree on attack/retreat. Up to $f$ are traitors sending conflicting messages.

## BFT Bound

Agreement is possible if and only if:

$$n > 3f \iff f < \frac{n}{3}$$

Requires $> 2/3$ honest parties. Classic BFT (PBFT): $\mathcal{O}(n^2)$ messages.

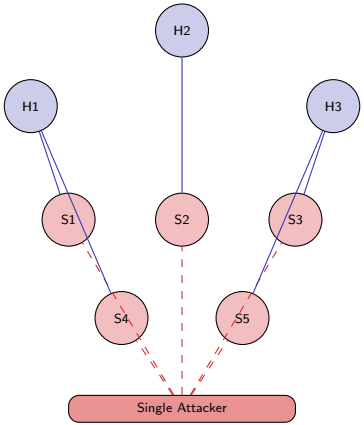
## Blockchain Approach

PoW/PoS replaces message complexity with economic cost. Bitcoin: probabilistic BFT. Casper: deterministic BFT with $f < n/3$.

**Lamp**

**Shostak, Pease (1982); blockchain provides practical BFT without quadratic message complexity**

# Sybil Attack




All "S" nodes are the same attacker

## Sybil Attack

Attacker creates many fake identities to gain disproportionate influence in peer selection, voting, or reputation systems.

## Countermeasures in Blockchain

- **PoW**: each identity costs computational work
- **PoS**: each identity costs staked capital
- **Eclipse attacks**: Sybil nodes surround victim, filter their view of network
- **Defense**: diverse peer selection, authenticated peer IDs

## Eclipse Attack

Attacker controls all of victim's connections $\Rightarrow$ can feed false chain, double-spend against victim.

**Sybil**

**resistance: core property PoW/PoS provide; Bitcoin randomizes peer selection to resist eclipse**

# Quantum Computing Threats

## Grover's Algorithm

Searches unsorted database of $N$ items in $\mathcal{O}(\sqrt{N})$.

Impact on hash-based PoW:

$$\text{Classical: } \mathcal{O}(2^{256}) \quad \rightarrow \quad \text{Quantum: } \mathcal{O}(2^{128})$$

SHA-256 still safe: $2^{128}$ operations beyond current/near-term quantum computers (need $\sim 10^6$ logical qubits).

## Vulnerable Bitcoin UTXOs

| Type | Risk |
|---|---|
| P2PK | High (pubkey in UTXO) |
| Reused P2PKH | High (pubkey seen) |
| Fresh P2PKH | Medium (hash only) |
| P2WPKH | Low (hash until spend) |

$\sim$25% of BTC supply in vulnerable addresses.

## Shor's Algorithm

Factors integers in polynomial time $\mathcal{O}((\log N)^3)$.

**Breaks ECDSA** (secp256k1):

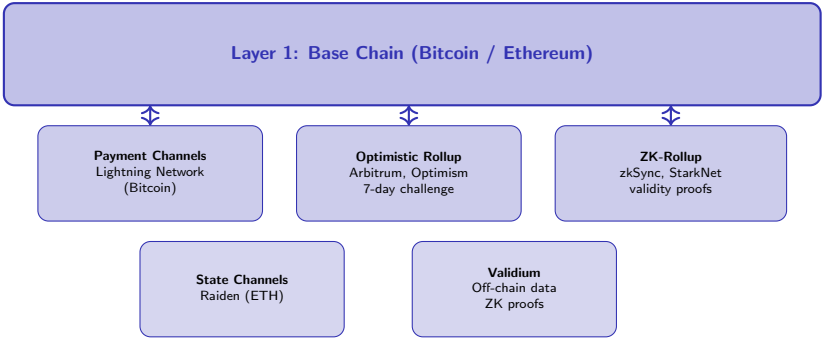
- Recovers private key from public key
- All coins with exposed public keys at risk
- P2PKH: public key only exposed on spend
- P2PK and reused addresses: exposed!

## Post-Quantum Candidates

- NIST PQC: CRYSTALS-Dilithium (signatures)
- Hash-based: SPHINCS+ (stateless)
- Timeline: migration needed before quantum threat materializes ($\sim$2030–2040?)

**NIST finalized first PQC standards (2024); blockchain migration will require soft/hard fork**

# Layer 2 Scaling Solutions




## Scalability Trilemma

Blockchains struggle to simultaneously achieve: **security** + **decentralization** + **scalability**. L2 offloads computation while inheriting L1 security.

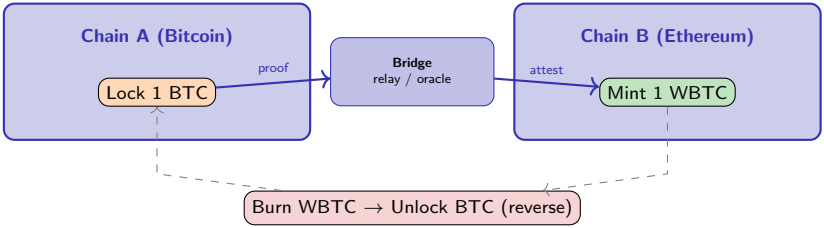
## Throughput Comparison

| | |
|---|---|
| Bitcoin L1: | ∼7 TPS |
| Ethereum L1: | ∼15 TPS |
| Lightning: | ∼millions TPS |
| ZK-Rollup: | ∼2000 TPS |

**Light**

**Network: ∼5,000 BTC capacity (2024); ZK-rollups: fastest growing L2 category**

# Cross-Chain Bridges




## Bridge Types

- **Custodial**: centralized (WBTC, BitGo)
- **Optimistic**: fraud proofs, 7-day delay
- **ZK**: validity proofs, instant finality
- **Hash Time Lock**: atomic swaps (no bridge needed for 1:1 swaps)

## Bridge Risks

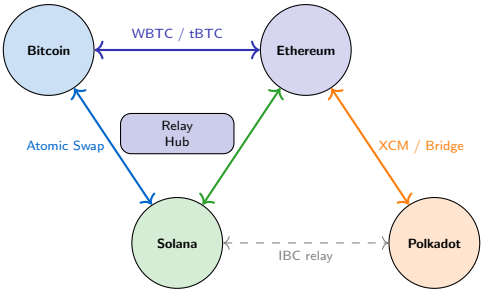
Bridges hold $billions in TVL and are prime hack targets:

- Ronin: $625M (2022)
- Poly Network: $611M (2021)
- Wormhole: $320M (2022)

**Root cause**: oracle manipulation, key compromise, logic bugs.

**Total bridge hacks (2021–2024): $> \$2$B; bridges are the weakest link in multi-chain ecosystems**

# Blockchain Interoperability




## Interoperability Mechanisms

- **Atomic Swaps**: HTLC-based trustless exchange across chains
- **Wrapped Tokens**: custodial (WBTC) or decentralized (tBTC)
- **Relay Chains**: Polkadot parachains, Cosmos IBC
- **Message Passing**: LayerZero, Axelar, Chainlink CCIP

## Atomic Swap Protocol (HTLC)

1. Alice creates hash lock $h = H(s)$ on Chain A
2. Bob locks funds with same $h$ on Chain B
3. Alice reveals secret $s$ to claim on Chain B
4. Bob uses $s$ to claim on Chain A
5. Timeout: refund if incomplete

**Interc**

**is the "TCP/IP moment" for blockchains – enabling composability across isolated networks**

# Smart Contract Platforms Comparison

| Feature | Ethereum | Solana | Cardano | Polkadot |
|---|---|---|---|---|
| **Consensus** | PoS (Casper) | PoH + Tower BFT | Ouroboros PoS | NPoS (GRANDPA) |
| **TPS (L1)** | ∼15–30 | ∼4,000+ | ∼250 | ∼1,000 |
| **Finality** | ∼12 min | ∼0.4 s | ∼20 s | ∼12–60 s |
| **VM / Lang** | EVM / Solidity | SVM / Rust | Plutus / Haskell | Wasm / Rust |
| **Fee model** | EIP-1559 (dynamic) | Fixed low fees | Predictable | Weight-based |
| **Smart contract** | Account-based | Account-based | eUTXO-based | Pallet-based |
| **Validator count** | ∼900,000+ | ∼1,900 | ∼3,200 | ∼300 |
| **Scaling** | L2 rollups | Firedancer | Hydra (L2) | Parachains |
| **TVL (2024)** | ∼$60B | ∼$5B | ∼$500M | ∼$300M |

## Design Philosophy Tradeoffs

- **Ethereum**: maximum decentralization, L2-centric roadmap
- **Solana**: raw throughput via hardware requirements
- **Cardano**: formal verification, peer-reviewed research
- **Polkadot**: shared security via relay chain

## Blockchain Trilemma Position

- All platforms trade off **decentralization**, **security**, and **scalability**
- Ethereum favors decentralization + security
- Solana favors scalability + security
- No platform achieves all three at L1

**Platf**

**choice depends on application requirements; Ethereum dominates by TVL and developer ecosystem (2024)**

# Real-World Blockchain Applications in Finance

## Central Bank Digital Currencies (CBDCs)

- **Digital Yuan (e-CNY)**: live since 2022, 260M+ users
- **Digital Euro**: ECB piloting (2024–2025)
- Wholesale vs. retail CBDC designs
- Programmable money: conditional payments, expiry

## Tokenization of Real-World Assets (RWA)

- **Securities**: tokenized bonds (EIB, Goldman Sachs)
- **Real estate**: fractional ownership via NFTs
- **Commodities**: Paxos Gold (PAXG), tokenized carbon credits
- BlackRock BUIDL fund: $500M+ tokenized treasuries

## Decentralized Finance (DeFi)

- **DEXs**: Uniswap, Curve – $billions daily volume
- **Lending**: Aave, Compound – algorithmic interest rates
- **Stablecoins**: USDC, DAI – $150B+ market cap
- Total DeFi TVL: ∼$90B (2024)

## Institutional Adoption

- Bitcoin ETFs: $50B+ AUM (approved Jan 2024)
- JPMorgan Onyx: blockchain-based payments
- SWIFT: CBDC interoperability experiments
- BIS Innovation Hub: Project mBridge (multi-CBDC)

**Block**

**in finance: from fringe experiment to institutional infrastructure; RWA tokenization projected at $16T by 2030**

# Summary: Key Formulas Reference Sheet

## Block & Consensus

| | |
|---|---|
| Mining condition | $H(\text{hdr}\|\|\text{nonce}) < T$ |
| Difficulty | $D = T_{\max}/T$ |
| Expected hashes | $E[N] = D \cdot 2^{32}$ |
| Block reward | $R(n) = 50/2^n$ BTC |
| Supply cap | $\sum 210K \cdot R(n) = 21M$ |
| Difficulty adj. | $D_{\text{new}} = D_{\text{old}} \cdot t_{\text{act}}/t_{\text{tgt}}$ |

## Security & Network

| | |
|---|---|
| 51% attack | $P(z,q) = (q/(1-q))^z$ |
| BFT bound | $n > 3f$ |
| Nakamoto coeff. | $\min k : \sum_{i=1}^{k} s_i > 0.5$ |
| Gini | $G = \frac{\sum \lvert s_i - s_j \rvert}{2n \sum s_i}$ |
| Entropy | $H = -\sum p_i \log_2 p_i$ |
| Propagation | $t \approx \frac{\log N}{\log k} \cdot t_{\text{hop}}$ |

## Hash & Merkle

| | |
|---|---|
| Birthday bound | $k \approx 2^{n/2}$ for 50% collision |
| Proof size | $\lceil \log_2 n \rceil$ hashes |
| PoS selection | $P(i) = s_i / \sum s_j$ |

## Core Insight

Blockchain: replace trust in institutions with cryptographic proofs + game-theoretic incentives + decentralized consensus.

**All formulas derived in-lecture; see Bitcoin whitepaper (Nakamoto 2008) and Ethereum Yellow Paper (Wood 2014)**