

Lesson 16: Proof of Work

Module 2: Blockchain Fundamentals

Digital Finance

The Double-Spending Problem

Digital Money Challenge:

- Digital files are copyable
- How to prevent spending same coin twice?
- Traditional solution: Central authority (bank)

Decentralized Challenge:

- No central ledger
- Network latency
- Conflicting transactions
- Malicious actors

`charts/lesson_16/double_spending_scenario.pdf`

Consensus Problem: Agreeing on Transaction Order

`charts/lesson_16/consensus_problem.pdf`

Core Idea:

- Make block creation expensive
- Require computational work
- Probability-based selection
- Longest chain wins

Properties:

- Sybil resistance (one CPU = one vote)
- Objective chain selection rule
- Economic security
- No coordination needed

`charts/lesson_16/pow_concept.pdf`

Mining Goal: Find nonce such that block hash is below target

$$\text{SHA256}(\text{Block Header}) < \text{Target}$$

Block Header Contains:

- Previous block hash
- Merkle root (transaction summary)
- Timestamp
- Difficulty target
- **Nonce** (number to vary)

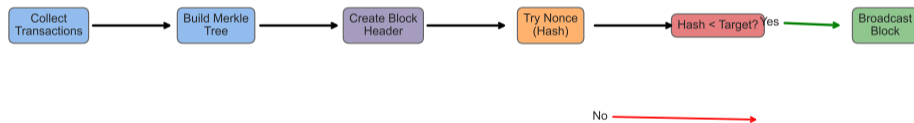
Example:

Target: 000000000000000000000000f1a2b3c4d...

Hash attempt 1: 8a3f2e1d9c... (too high)

Hash attempt 2: 000000000000000000000000a1b2c3d... (success!)

Bitcoin Mining Process



Steps:

- 1 Collect transactions from mempool
- 2 Build Merkle tree, create block header
- 3 Try different nonces until a valid hash is found

Difficulty Target: Controlling Block Time

Target Representation:

$$\text{Target} = \text{coefficient} \times 2^{8(\text{exponent}-3)}$$

Difficulty:

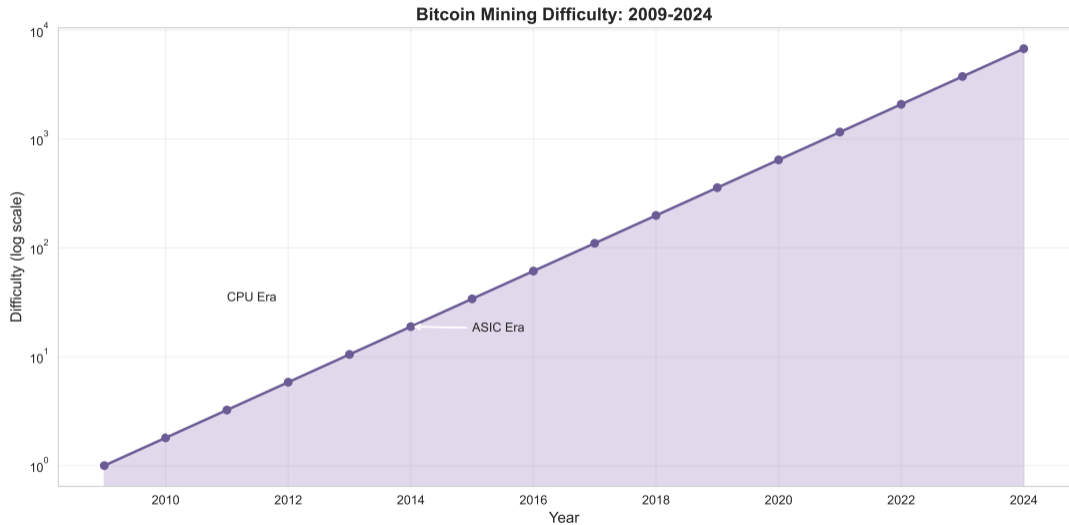
$$\text{Difficulty} = \frac{\text{Max Target}}{\text{Current Target}}$$

Bitcoin:

- Target block time: 10 minutes
- Adjusts every 2016 blocks (2 weeks)
- Difficulty \propto total hashrate

[charts/lesson_16/difficulty_adjustment.pdf](#)

Difficulty Over Time: Bitcoin Example



Observations:

- Exponential growth from 2009 to 2024

Probability of Success per Hash:

$$P(\text{success}) = \frac{\text{Target}}{2^{256}}$$

Expected Number of Hashes:

$$E[\text{hashes}] = \frac{2^{256}}{\text{Target}} = \text{Difficulty} \times 2^{32}$$

Expected Time to Find Block:

$$T = \frac{\text{Difficulty} \times 2^{32}}{\text{Hashrate}}$$

Example: Difficulty = 50 trillion, Hashrate = 100 TH/s

$$T = \frac{50 \times 10^{12} \times 2^{32}}{100 \times 10^{12}} \approx 2147 \text{ seconds} \approx 36 \text{ minutes}$$

Mining Difficulty vs Hashrate

`charts/lesson_16/hashrate_vs_difficulty.pdf`

Attack Scenario:

- Attacker controls $>50\%$ hashrate
- Can create longest chain
- Rewrite transaction history
- Double-spend attack

Limitations:

- Cannot steal others' coins
- Cannot create coins from nothing
- Cannot change protocol rules

`charts/lesson_16/51_percent_attack.pdf`

`charts/lesson_16/confirmation_depth.pdf`

Revenue:

- Block reward: 3.125 BTC (as of 2024, halves every 4 years)
- Transaction fees: Variable (0.1–2 BTC per block)

Costs:

- Hardware (ASICs): \$3,000–\$15,000 per unit
- Electricity: 3–6 cents per kWh (industrial rates)
- Cooling, maintenance, facility

Profitability Equation:

$$\text{Profit} = (\text{Block Reward} + \text{Fees}) \times \text{BTC Price} - \text{Electricity Cost}$$

Break-even: Electricity cost \approx 40–60% of revenue at scale

`charts/lesson_16/mining_hardware_evolution.pdf`

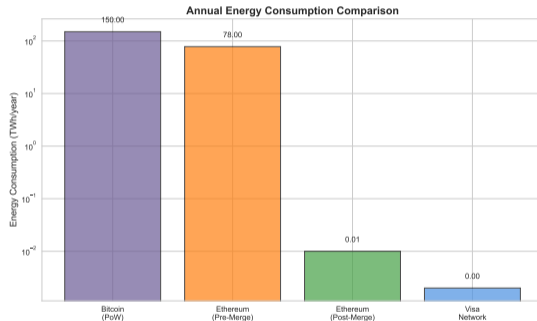
Energy Consumption: The Elephant in the Room

Bitcoin Network (2024):

- Total hashrate: ~ 600 EH/s
- Power consumption: ~ 150 TWh/year
- Comparable to Argentina or Netherlands

Per Transaction:

- ~ 700 kWh per transaction
- vs Visa: ~ 0.001 kWh
- But: Bitcoin = settlement layer



`charts/lesson_16/mining_energy_sources.pdf`

Criticisms:

- Massive carbon footprint
- E-waste from obsolete ASICs
- Inefficient compared to databases
- Competes with useful computing

Counterarguments:

- Energy = security (makes attacks expensive)
- Incentivizes renewable buildout
- Banking system also energy-intensive
- Enables censorship-resistant money

Trade-off: Security vs energy efficiency (Proof of Stake addresses this)

`charts/lesson_16/mining_pool_distribution.pdf`

Strategy:

- ① Miner finds block, keeps secret
- ② Continues mining on private chain
- ③ Reveals when ahead by 2+ blocks
- ④ Honest chain orphaned

Result:

- Unfair revenue (more than hashrate share)
- Effective with $>25\%$ hashrate
- Wastes other miners' work

`charts/lesson_16/selfish_mining.pdf`

| Mechanism | Selection | Pros | Cons |
|--------------------|---------------------|--------------------------------|--------------------------------|
| Proof of Work | Computational power | Proven security, decentralized | Energy intensive |
| Proof of Stake | Staked capital | Energy efficient | Rich get richer, slashing risk |
| Proof of Authority | Approved validators | Fast, low energy | Centralized, permissioned |
| Proof of Space | Disk storage | Lower energy than PoW | New, unproven security |

Note: Ethereum switched from PoW to PoS in 2022 (The Merge)

- **Double-Spending Problem:** Solved by probabilistic consensus via PoW
- **Mining:** Find nonce making block hash $<$ target (SHA256 puzzle)
- **Difficulty:** Auto-adjusts to maintain constant block time (10 min for Bitcoin)
- **Security:** 51% attack possible but expensive; confirmation depth increases safety
- **Economics:** Revenue (block reward + fees) vs costs (hardware + electricity)
- **Energy Debate:** ~ 150 TWh/year, trade-off between security and efficiency

Next Lesson: Proof of Stake – energy-efficient alternative consensus