Lesson 16: Proof of Work
Module 2: Blockchain Fundamentals

Digital Finance

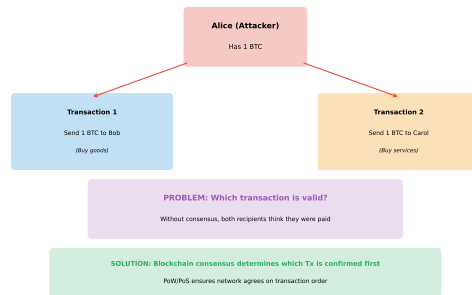# The Double-Spending Problem

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
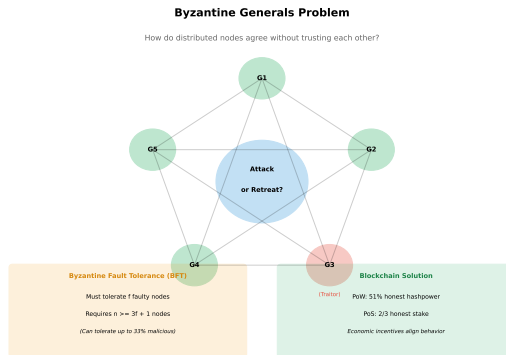- Malicious actors

The Double Spending Problem

Why digital cash needs blockchain consensus

Alice (Attacker)

Has 1 BTC

Transaction 1

Send 1 BTC to Bob

(Buy goods)

Transaction 2

Send 1 BTC to Carol

(Buy services)

PROBLEM: Which transaction is valid?

Without consensus, both recipients think they were paid

SOLUTION: Blockchain consensus determines which Tx is confirmed first

PoW/PoS ensures network agrees on transaction order

Source: Nakamoto, Bitcoin Whitepaper (2008), "Commerce on the Internet"

Key concepts from this slide inform practical applications in finance.

Byzantine Generals Problem

How do distributed nodes agree without trusting each other?

Byzantine Fault Tolerance (BFT)

Must tolerate f faulty nodes

Requires n >= 3f + 1 nodes

(Can tolerate up to 33% malicious)

Blockchain Solution

PoW: 51% honest hashpower

PoS: 2/3 honest stake

Economic incentives align behavior

Source: Lamport et al., "The Byzantine Generals Problem" (1982)

Key Questions:

- Which transaction came first?
- Who decides the canonical order?
- How to prevent censorship or manipulation?
- How to incentivize honest behavior?
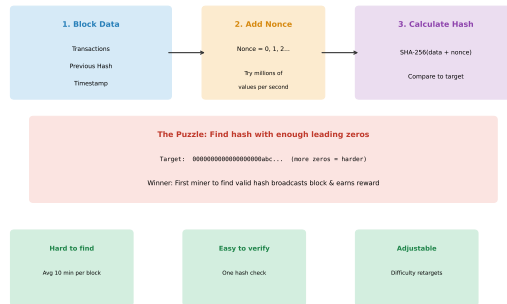
Proof of Work: The Solution

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

Proof of Work: The Mining Process

1. Block Data	2. Add Nonce	3. Calculate Hash
Transactions	Nonce = 0, 1, 2...	SHA-256(data + nonce)
Previous Hash	Try millions of	Compare to target
Timestamp	values per second	

The Puzzle: Find hash with enough leading zeros

Target: 0000000000000000abc... (more zeros = harder)

Winner: First miner to find valid hash broadcasts block & earns reward

Hard to find	Easy to verify	Adjustable
Avg 10 min per block	One hash check	Difficulty retargets

Source: Nakamoto, Bitcoin Whitepaper (2008), Section 4

Proof-of-Work provides security through computational cost but has energy concerns.

Hash Puzzle: Finding the Nonce

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

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

Block Header Contains:

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

Example:

Target: $0000000000000000000f1a2b3c4d\dots$

Hash attempt 1: $8a3f2e1d9c\dots$ (too high)

Hash attempt 2: $0000000000000000000a1b2c3d\dots$ (success!)

Mining Process

Input → Process A → Process B → Process C → Output

[SYNTHETIC DATA]

Steps:

1. Collect transactions from mempool
2. Build Merkle tree, create block header
3. Try different nonces (brute force search)
4. Hash until target reached
5. Broadcast valid block to network

Understanding the process flow is key to identifying optimization opportunities.

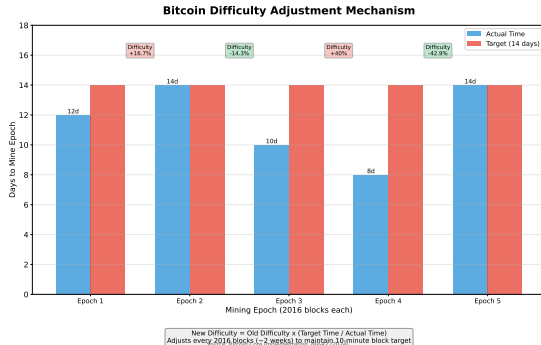# Difficulty Target: Controlling Block Time

Target Representation:

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

Difficulty:

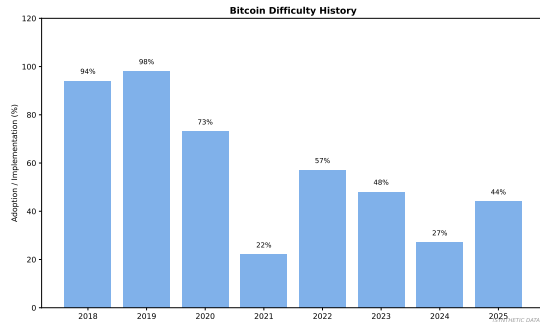$$\text{Difficulty} = \frac{\text{Max Target}}{\text{Current Target}}$$

Bitcoin:

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

Bitcoin Difficulty Adjustment Mechanism

New Difficulty = Old Difficulty x (Target Time / Actual Time)
Adjusts every 2016 blocks (~2 weeks) to maintain 10-minute block target

---

Key concepts from this slide inform practical applications in finance.

Difficulty Over Time: Bitcoin Example

Bitcoin Difficulty History

Observations:

- Exponential growth from 2009 to 2024
- Difficulty in 2024: $\sim 10^{13}$ times harder than 2009
- Hashrate: From CPU mining to specialized ASICs

Case studies provide concrete evidence of technology impact and adoption patterns.

Mining Probability and Expected Time

Probability of Success per Hash:

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

Expected Number of Hashes:

$$E[\text{hashes}] = \frac{2^{256}}{\text{Target}} = \text{Difficulty} \times 2^{32}$$
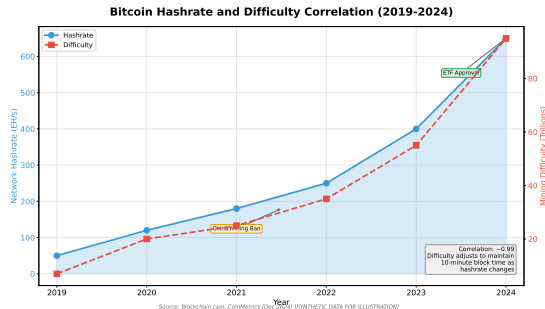
Expected Time to Find Block:

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

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

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

Key concepts from this slide inform practical applications in finance.

Mining Difficulty vs Hashrate

Bitcoin Hashrate and Difficulty Correlation (2019-2024)

Relationship:

- Hashrate increases → blocks found faster
- Difficulty adjusts upward → restores 10-min average
- Self-regulating system maintains predictable issuance

Comparative analysis helps identify the right tool for specific requirements.

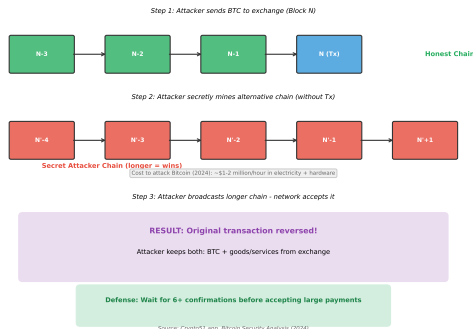# Blockchain Security: The 51% Attack

Attack Scenario:

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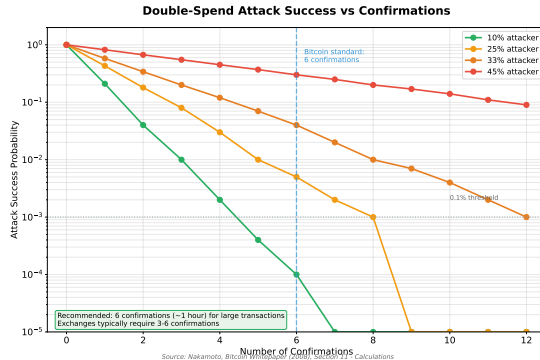- Double-spend attack

Limitations:

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

51% Attack: Double Spending via Chain Reorg

Step 1: Attacker sends BTC to exchange (Block N)

| N-3 | → | N-2 | → | N-1 | → | N (Tx) | **Honest Chain** |

Step 2: Attacker secretly mines alternative chain (without Tx)

| N'-4 | → | N'-3 | → | N'-2 | → | N'-1 | → | N'+1 |

Secret Attacker Chain (longer = wins)
Cost to attack Bitcoin (2024): ~$1-2 million/hour in electricity + hardware

Step 3: Attacker broadcasts longer chain - network accepts it

RESULT: Original transaction reversed!
Attacker keeps both: BTC + goods/services from exchange

Defense: Wait for 6+ confirmations before accepting large payments

Source: Crypto51.app, Bitcoin Security Analysis (2024)

Security analysis identifies vulnerabilities and helps design robust systems.

Double-Spend Attack Success vs Confirmations

Probability of Reversal:

$$P(\text{reorg after } z \text{ blocks}) \approx \left(\frac{q}{p}\right)^z$$

where p = honest hashrate fraction, q = attacker hashrate fraction

Bitcoin Standard: 6 confirmations ($\sim$1 hour) for high-value transactions

Security analysis identifies vulnerabilities and helps design robust systems.

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)
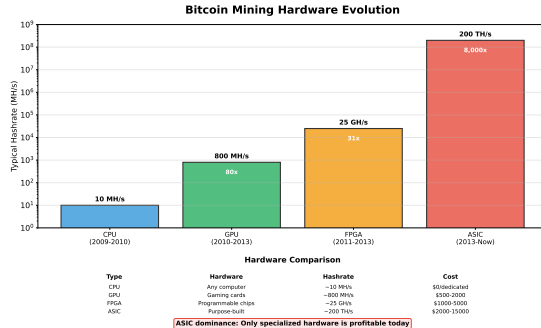- Cooling, maintenance, facility

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

Key concepts from this slide inform practical applications in finance.

Bitcoin Mining Hardware Evolution

Hardware Comparison

Type	Hardware	Hashrate	Cost
CPU	Any computer	~10 MH/s	$0/dedicated
GPU	Gaming cards	~800 MH/s	$500-2000
FPGA	Programmable chips	~25 GH/s	$1000-5000
ASIC	Purpose-built	~200 TH/s	$2000-15000

ASIC dominance: Only specialized hardware is profitable today

Source: CoinDesk Research, Bitmain Product Specifications (2024)

- **2009–2010:** CPU mining (1–10 MH/s)
- **2010–2012:** GPU mining (100–500 MH/s)
- **2013+:** ASIC mining (1–200+ TH/s)
- **Modern ASICs:** Antminer S19 Pro (110 TH/s, 3250W)

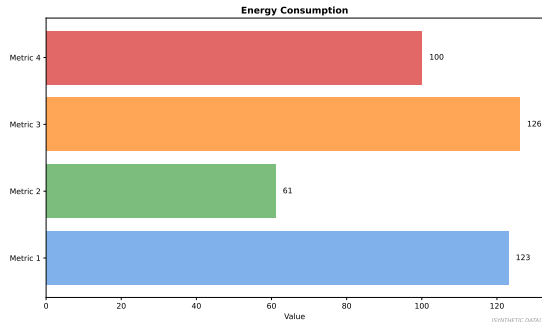**Understanding history helps predict future developments in the technology.**

Energy Consumption: The Elephant in the Room

Bitcoin Network (2024):

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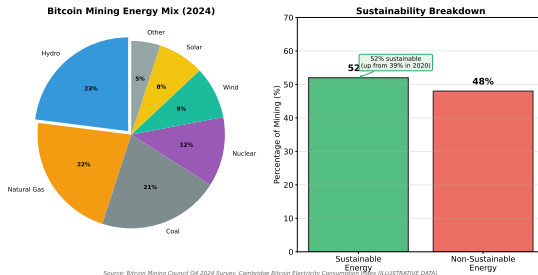- Comparable to Argentina or Netherlands

Per Transaction:

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

Key concepts from this slide inform practical applications in finance.

Bitcoin Mining Energy Mix (2024) / **Sustainability Breakdown**

52% sustainable (up from 39% in 2020)

Source: Bitcoin Mining Council Q4 2024 Survey, Cambridge Bitcoin Electricity Consumption Index [ILLUSTRATIVE DATA]

Estimates (2024):

- Renewable energy: 40–60% (hydroelectric, wind, solar)
- Natural gas: 20–30%
- Coal: 10–20%
- Nuclear: 5–10%

Trend: Miners seek cheap, stranded renewable energy (e.g., flare gas, curtailed hydro)

Key concepts from this slide inform practical applications in finance.

Environmental Criticisms and Counterarguments

Criticisms:

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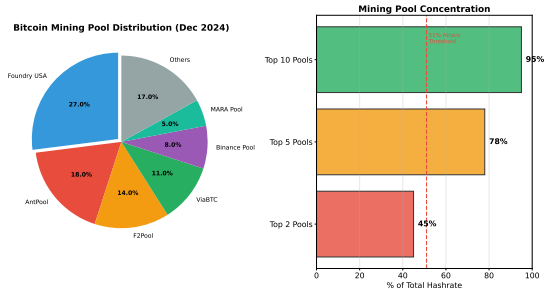- Competes with useful computing

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)

Key concepts from this slide inform practical applications in finance.

Bitcoin Mining Pool Distribution (Dec 2024)

Mining Pool Concentration

Source: BTC.com Pool Stats (December 2024) (SYNTHETIC) Centralization Risk: Top 2 pools control 45% of hashrate

Concerns:

- Top 5 pools control >70% hashrate
- Geographic concentration (China historically dominant, now US/Kazakhstan)
- Pool operators could censor transactions

Mitigation: Miners can switch pools, Stratum V2 protocol improves decentralization

Risk management is essential for financial stability and profitability.

Selfish Mining Attack

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

Security analysis identifies vulnerabilities and helps design robust systems.

Mechanism	Selection	Pros	Cons
Proof of Work	Computational power	Proven security	Energy intensive
Proof of Stake	Staked capital	Energy efficient	Rich get richer
Proof of Authority	Approved validators	Fast, low energy	Centralized
Proof of Space	Disk storage	Lower energy	Unproven security

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

Proof-of-Work provides security through computational cost but has energy concerns.

Summary

- **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