Proof of Work Consensus

BSc Blockchain, Crypto Economy & NFTs

Course Instructor

Module A: Blockchain Foundations

Learning Objectives

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

- Explain the proof-of-work consensus mechanism
- Describe the mining process and nonce searching
- Understand difficulty adjustment and its purpose
- Calculate mining profitability and hash rate economics
- Recognize the security guarantees and vulnerabilities of PoW
- Evaluate the 51% attack threat model
- Discuss energy consumption and environmental impact

The Byzantine Generals Problem

Distributed Consensus Challenge:

Byzantine generals surround a city and must coordinate attack or retreat:

- Generals communicate via messengers
- Some generals may be traitors (malicious)
- Traitors send conflicting messages
- How to reach consensus despite traitors?

Blockchain Analogy:

- Nodes = generals
- Transaction ordering = attack/retreat decision
- Malicious nodes = traitors
- Network delays and partitions = unreliable messengers

Proof-of-Work Solution:

- Make message creation costly (computational work)
- Honest majority by hash power (not node count)
- Longest chain rule resolves conflicts
- Economic incentives align honest behavior

What is Proof of Work?

Core Concept:

- Find a nonce such that hash of block header meets difficulty target
- Target: hash must be below a specific value (equivalently, N leading zeros)
- No shortcut: must try nonces randomly until one works
- Verification is instant: anyone can check hash validity

Mathematical Formulation:

$$\text{SHA-256}(\text{SHA-256}(\text{BlockHeader})) < \text{Target}$$

Key Properties:

1. **Asymmetry:** Hard to find, easy to verify
2. **Probabilistic:** Expected time to find solution, no guarantee
3. **Adjustable difficulty:** Target changes to maintain block time
4. **Progress-free:** Past attempts do not help future attempts

Analogy:

- Rolling dice until you get 10 sixes in a row
- Each roll is independent
- Expected number of attempts: 6^{10} (very large)
- Verification: just look at the result

Bitcoin Block Header Structure

Block Header (80 bytes):

1. **Version** (4 bytes): Protocol version
2. **Previous Block Hash** (32 bytes): Hash of previous block
3. **Merkle Root** (32 bytes): Root of transaction Merkle tree
4. **Timestamp** (4 bytes): Current time (Unix epoch)
5. **Difficulty Target** (4 bytes): Compact representation of target
6. **Nonce** (4 bytes): Random value to vary hash

Mining Process:

1. Construct block with transactions
2. Compute Merkle root
3. Set timestamp and difficulty
4. Try nonce = 0, compute hash
5. If hash ¡ target: success (broadcast block)
6. If hash $\geq$ target: increment nonce, repeat

Nonce Space Exhaustion:

- 4 bytes = 2^{32} = 4.3 billion possible nonces
- Modern ASICs exceed this in milliseconds
- Solution: modify coinbase transaction (extra nonce), recompute Merkle root

Merkle Trees: Efficient Transaction Commitment

Purpose:

- Commit to all transactions in block with single hash (32 bytes)
- Enable efficient transaction verification (SPV clients)
- Modify single transaction -¿ Merkle root changes

Construction:

1. Hash each transaction: $H(tx_1), H(tx_2), \ldots, H(tx_n)$
2. Pair hashes and hash again: $H(H(tx_1) \| H(tx_2))$
3. Repeat until single root hash remains

Properties:

- Tree height: $\log_2(n)$ for n transactions
- Proof size: $\log_2(n)$ hashes to prove transaction inclusion
- Example: 1000 transactions -¿ 10 hashes (320 bytes proof)

Extra Nonce Trick:

- Miners modify coinbase transaction (includes extra nonce field)
- Recompute Merkle root (different root for each extra nonce)
- Expands search space beyond 2^{32} nonces

Difficulty Target and Adjustment

Difficulty Target:

- 256-bit number representing maximum valid hash
- Lower target = harder mining (fewer valid hashes)
- Difficulty = how hard current target is relative to maximum

Target Representation:

- Compact format: 4 bytes (exponent-mantissa encoding)
- Example: `0x1b0404cb` $= 0x0404cb \times 2^{8\times(0x1b-3)}$
- Full 256-bit target reconstructed during validation

Difficulty Adjustment (Every 2016 Blocks):

$$\text{New Target} = \text{Old Target} \times \frac{\text{Actual Time}}{\text{Expected Time}}$$

- Expected time: 2016 blocks $\times$ 10 minutes = 20,160 minutes (2 weeks)
- Actual time: measured from timestamps
- Clamped to prevent extreme changes: $[T/4, T \times 4]$

Purpose:

- Maintain 10 minute average block time
- Adapt to changing total hash rate
- Self-stabilizing system

Hash Rate and Mining Economics

Hash Rate:

- Number of hashes computed per second
- Units: H/s (hashes), KH/s, MH/s, GH/s, TH/s, PH/s, EH/s
- Bitcoin network (2024): 500 EH/s (500 quintillion hashes per second)

Mining Probability:

$$P(\text{find block in 10 min}) = \frac{\text{Your Hash Rate}}{\text{Network Hash Rate}}$$

Example:

- Miner hash rate: 100 TH/s
- Network hash rate: 500 EH/s = 500,000,000 TH/s
- Probability: $\frac{100}{500{,}000{,}000} = 0.0000002 = 0.00002\%$
- Expected blocks per year: $0.0000002 \times 52,560 \approx 0.01$ blocks
- Expected time to find block: 100 years

Solution: Mining Pools

- Aggregate hash rate from many miners
- Share block rewards proportionally
- Reduce payout variance

Mining Profitability

Revenue:

$$\text{Daily Revenue} = \frac{\text{Your Hash Rate}}{\text{Network Hash Rate}} \times 144 \text{ blocks/day} \times (\text{Block Reward} + \text{Avg Fees})$$

Costs:

- **Hardware:** ASIC miner cost (e.g., Antminer S19 Pro: $2000-5000)
- **Electricity:** power consumption × electricity rate
- **Cooling:** additional power for air conditioning
- **Maintenance:** repairs, facility costs

Example Calculation (Antminer S19 Pro):

- Hash rate: 110 TH/s
- Power consumption: 3250 W = 78 kWh/day
- Electricity cost: $0.05/kWh × 78 = $3.90/day
- Revenue (BTC = $40,000): $\frac{110}{500{,}000{,}000} \times 144 \times 6.25 \times 40,000 \approx \$7.92/day$
- Profit: $7.92 - $3.90 = $4.02/day
- Payback period (hardware cost $3000): $\frac{3000}{4.02} \approx 746$ days (2 years)

Risk Factors:

- BTC price volatility
- Difficulty increases (hash rate growth)
- Hardware obsolescence
- Electricity price changes

ASIC Mining Hardware Evolution

CPU Mining (2009-2010):

- Early Bitcoin mining on personal computers
- Hash rate: 1-10 MH/s per CPU
- Quickly became unprofitable

GPU Mining (2010-2013):

- Graphics cards (NVIDIA, AMD)
- Hash rate: 100-1000 MH/s per GPU
- Parallel processing advantage

FPGA Mining (2011-2013):

- Field-Programmable Gate Arrays
- Hash rate: 100-1000 MH/s
- More efficient than GPUs

ASIC Mining (2013-Present):

- Application-Specific Integrated Circuits
- Designed solely for SHA-256 hashing
- Hash rate: 1-200 TH/s (2024 models)
- 1000x more efficient than GPUs
- Dominates Bitcoin mining

Implications:

Mining Pools

Why Pools Exist:

- Solo mining: high variance (might wait years for block)
- Pooled mining: steady income (proportional to hash rate)
- Risk mitigation for small miners

Pool Operation:

1. Pool coordinator distributes mining tasks (shares)
2. Miners submit partial solutions (lower difficulty)
3. Pool tracks contribution of each miner
4. When pool finds block, reward distributed proportionally
5. Pool takes fee (1-3%)

Payout Schemes:

- **PPS (Pay-Per-Share):** fixed payment per share (lowest variance)
- **PPLNS (Pay-Per-Last-N-Shares):** share revenue from recent blocks
- **FPPS (Full PPS):** PPS + transaction fees

Centralization Concern:

- Top 5 pools control 70% of hash rate
- Pools do not own hardware (miners can switch pools)
- Risk: pool operator could censor transactions
- Mitigation: decentralized pool protocols (P2Pool, Stratum V2)

Block Reward Components:

$$\text{Total Reward} = \text{Block Subsidy} + \text{Transaction Fees}$$

Block Subsidy (New Bitcoins):

- Initial reward (2009): 50 BTC per block
- Halves every 210,000 blocks (4 years)
- Current (2024): 6.25 BTC
- Next halving (2024): 3.125 BTC
- Asymptotic limit: 21 million BTC

Halving Timeline:

Period	Reward	Cumulative Supply
2009-2012	50 BTC	10.5M BTC
2012-2016	25 BTC	15.75M BTC
2016-2020	12.5 BTC	18.375M BTC
2020-2024	6.25 BTC	19.6875M BTC
2024-2028	3.125 BTC	20.34375M BTC

Implication:

- Transaction fees must eventually sustain mining
- Security model shifts over time

Transaction Fees as Mining Incentive

Current State (2024):

- Block subsidy: 6.25 BTC ($250,000 at $40,000/BTC)
- Transaction fees: 0.1-1 BTC per block ($4,000-40,000)
- Fees: 2-15% of total reward

Future Scenario (2140):

- Block subsidy: 0 BTC (last bitcoin mined)
- Transaction fees: 100% of mining revenue
- Security depends entirely on fee market

Challenges:

- Will fees be sufficient to secure the network?
- Fee volatility: low during quiet periods, high during congestion
- Miner revenue stability concerns

Potential Solutions:

- Layer 2 solutions (Lightning) move small transactions off-chain
- Base layer becomes settlement layer (high-value transactions)
- Higher fee-per-transaction compensates for lower transaction count
- Debate ongoing in Bitcoin community

The 51% Attack

Threat Model:

- Attacker controls ¿ 50% of network hash rate
- Can mine blocks faster than honest miners
- Longest chain rule allows attacker to dominate

What Attacker CAN Do:

- **Double-spend:** reverse own transactions
  - Send transaction to merchant (gets product)
  - Mine secret chain without transaction
  - Broadcast longer chain (reverses payment)
- **Censor transactions:** refuse to include specific transactions
- **Block other miners:** prevent competitors from earning rewards

What Attacker CANNOT Do:

- Steal bitcoins from others (requires private keys)
- Create bitcoins out of thin air (violates consensus rules)
- Change transaction history beyond attack start (infeasible to rewrite years of blocks)

51% Attack Economics

Cost of Attack:

$$\text{Cost} = \text{Hash Rate} \times \text{Duration} \times \text{Electricity Cost} + \text{Hardware Cost}$$

Example (Bitcoin):

- Network hash rate: 500 EH/s
- 51% attack: need 255 EH/s
- Hardware: $\frac{255{,}000{,}000 \text{ TH/s}}{110 \text{ TH/s}} \approx 2.3$ million Antminer S19 Pro
- Hardware cost: $2.3\text{M} \times \$3000 = \$6.9$ billion
- Electricity (1 hour): $2.3\text{M} \times 3.25 \text{ kW} \times \$0.05/\text{kWh} = \$373{,}750$
- Total (1 week attack): $\$6.9\text{B} + \$62.6\text{M} =$ $7 billion

Consequences:

- Attack becomes public knowledge immediately
- Bitcoin price crashes (attacker's hardware becomes worthless)
- Community may hard fork to new algorithm (bricks attacker's ASICs)
- Rational attacker: cost ¿ benefit for major cryptocurrencies

Vulnerable Chains:

- Small PoW chains (low hash rate)
- Shared mining algorithms (rent hash power from NiceHash)
- Historical attacks: Bitcoin Gold, Ethereum Classic, Verge

Selfish Mining

Attack Strategy:

- Miner finds block but does not broadcast immediately
- Continues mining on top of secret block
- If honest miner finds block: race to propagate
- Attacker reveals secret chain if it is longer

Potential Profit:

- Attacker can earn ¿ fair share of rewards with ¡ 50% hash rate
- Theoretical threshold: 33% hash rate (with optimal strategy)
- Wastes honest miners' work (reduces network security)

Mitigation:

- Random block propagation delays
- Penalize late-arriving blocks
- Timestamp-based block acceptance rules
- Not observed in practice (rational miners prioritize short-term honesty)

Open Question:

- Selfish mining debate ongoing since 2013
- Real-world evidence limited
- Game theory suggests instability at certain hash rate thresholds

Energy Consumption and Environmental Impact

Bitcoin Energy Usage (2024):

- Estimated annual consumption: 150 TWh (terawatt-hours)
- Comparable to countries: Argentina, Netherlands
- Percentage of global electricity: 0.6%

Sources of Energy:

- Renewable energy: 40-60% (hydroelectric, solar, wind)
- Fossil fuels: 40-60% (coal, natural gas)
- Nuclear: 5-10%
- Geographic concentration: areas with cheap electricity (Iceland, China, Kazakhstan, USA)

Environmental Concerns:

- Carbon emissions from fossil fuel usage
- Electronic waste from obsolete mining hardware
- Water usage for cooling in some regions

Counterarguments:

- Incentivizes renewable energy development (monetizes stranded energy)
- Facilitates grid balancing (flexible load)
- Energy usage proportional to security value
- Traditional banking system also consumes significant energy

Proof-of-Work Alternatives: ASIC Resistance

Motivation:

- Prevent mining centralization
- Enable consumer hardware mining (GPUs, CPUs)
- Increase decentralization

ASIC-Resistant Algorithms:

- **Scrypt (Litecoin):** memory-hard hashing
  - Requires significant RAM
  - ASIC eventually developed (2014)
- **Ethash (Ethereum, pre-merge):** memory-hard with large DAG
  - GPU-friendly, ASIC-resistant initially
  - ASICs developed but less dominant than Bitcoin
- **RandomX (Monero):** CPU-optimized
  - Frequently updated to thwart ASICs
  - Best performance on general-purpose CPUs

Trade-offs:

- ASIC resistance -¿ lower security per watt
- Easier for botnets to attack (commodity hardware)
- Algorithm changes create hard fork risks
- Debate: specialization increases security investment

Key Takeaways

- Proof-of-work provides Sybil resistance via computational cost
- Mining searches for nonces to produce valid block hashes
- Difficulty adjusts every 2016 blocks to maintain 10-minute block time
- Mining profitability depends on hash rate, electricity cost, and BTC price
- 51% attacks are economically infeasible for large PoW chains
- Block rewards halve every 4 years, shifting incentives toward transaction fees
- Energy consumption is a significant concern but incentivizes renewable energy
- Mining centralization and pool dominance pose governance risks

Core Insight:
Proof-of-work converts energy into cryptographic security. The cost of attacking the network is proportional to the cumulative computational work invested by honest miners.

Discussion Questions

1. Why is proof-of-work described as "progress-free"?
2. How does difficulty adjustment make Bitcoin resilient to hash rate fluctuations?
3. What would happen if block rewards fell to zero but transaction fees remained low?
4. Is ASIC mining centralization a threat to Bitcoin's decentralization?
5. How does mining pool concentration differ from miner concentration?
6. Can proof-of-work be justified from an environmental perspective?
7. Why has no successful 51% attack occurred on Bitcoin?

Next Lesson Preview: L08 Lab - Wallet Setup

Lab activities:

- Install and configure MetaMask wallet
- Understand seed phrase security and backup
- Connect to Ethereum testnet (Sepolia or Goerli)
- Obtain testnet ETH from faucets
- Execute first testnet transaction
- Explore wallet features and settings
- Best practices for wallet security

Preparation:

- Install a modern web browser (Chrome, Firefox, Brave)
- Review public-private key concepts from Lesson 5
- Prepare a secure location for seed phrase backup