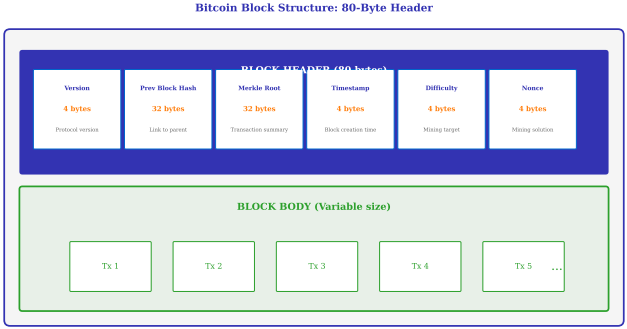
Lesson 14: Blocks and Cryptographic Hashing

Module 2: Blockchain and Cryptocurrencies

Digital Finance

December 13, 2025

Block Structure: Anatomy of a Bitcoin Block



The 80-byte header contains all essential metadata; body holds actual transactions.

Example: Bitcoin Block 800,000

Field	Value
Block Hash	000000000000000000002a7c4c1e48d76c5a37902165a270156b7a8d72728a054
Previous Hash	00000000000000000000012117ad9f72c1c0e42329492e8c18f9e945a5d0f1b9a4
Merkle Root	7e1c6b0f5e9c8d9e3a2f1b4c5d6e7f8a9b0c1d2e3f4a5b6c7d8e9f0a1b2c3d4
Timestamp	2023-07-13 20:42:05 UTC
Difficulty	53,911,173,001,054
Nonce	1,868,822,685
Transactions	3,285
Block Size	1,582,419 bytes (1.58 MB)
Block Reward	6.25 BTC
Total Fees	0.183 BTC
Height	800,000
Confirmations	50,000+ (as of Dec 2024)

Note: Hash starts with 19 leading zeros - probability of finding this: $1/2^{76}$

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

What is a Hash Function?

Hash Function: Mathematical algorithm that maps arbitrary data to fixed-size output

Properties:

1. **Deterministic:** Same input always produces same output
2. **Fast:** Quick to compute
3. **One-way:** Cannot reverse (pre-image resistance)
4. **Collision-resistant:** Hard to find two inputs with same hash
5. **Avalanche effect:** Tiny input change drastically changes output

Examples:

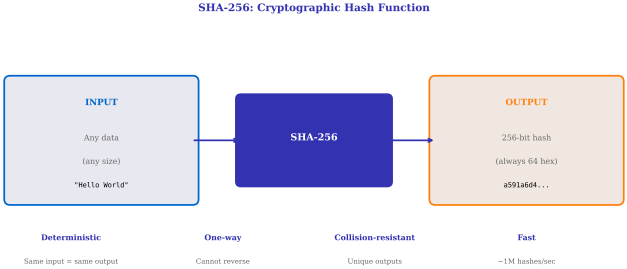
- MD5: 128 bits (broken, not secure)
- SHA-1: 160 bits (deprecated)
- SHA-256: 256 bits (Bitcoin)
- SHA-3: Variable (latest standard)
- BLAKE2: 256/512 bits (faster)

Output Size:

- SHA-256: 64 hex characters
- SHA-256: 2^{256} possible outputs
- More atoms in universe: $\approx 2^{265}$

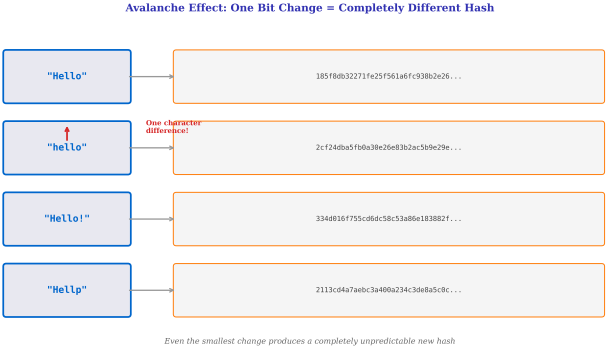
Clear definitions are essential for understanding complex technical concepts.

SHA-256: The Bitcoin Hash Function



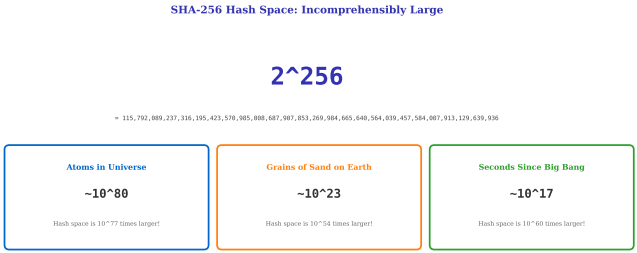
SHA-256 produces 256-bit fixed output regardless of input size - the backbone of Bitcoin security.

The Avalanche Effect: Demonstration



Even the smallest change produces a completely unpredictable new hash - no pattern correlation.

Hash Space and Collision Resistance



SHA-256 collisions are computationally infeasible - the space is incomprehensibly large.

Pre-image Resistance: The One-Way Property

Given a hash, can you find the original input?

Forward (Easy):

$$\text{Input} \xrightarrow{\text{SHA-256}} \text{Hash}$$

- Instant computation
- Example: `SHA256("Bitcoin")` = `b4056df6...`
- Deterministic and fast

Reverse (Impossible):

$$\text{Hash} \xrightarrow{???} \text{Input}$$

- No mathematical inverse
- Only option: Brute force
- Try all possible inputs
- Computationally infeasible

Example Attack: Find input for hash `000000000019d6689c085ae165831e93...`

- Average attempts needed: 2^{255} (half the hash space)
- At 1 quadrillion attempts/second: 10^{61} years
- Even with all computers on Earth: Still infeasible

Key concepts from this slide inform practical applications in finance.

Hash Pointers: Linking Blocks Together

Hash Pointer: Data structure that contains both location AND cryptographic hash of data

Regular Pointer:

- Points to memory address
- Can retrieve data
- No integrity check
- Data can be tampered

Example:

- Pointer: `0x7ffe5367e044`
- Data at address: `"Alice pays Bob 5 BTC"`
- If data changes: No detection

Hash Pointer:

- Points to data location
- Contains hash of data
- Can retrieve AND verify
- Tamper-evident

Example:

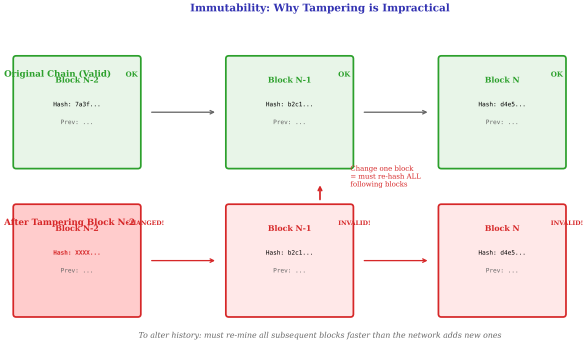
- Location: Block 100
- Hash: `00000000000080b66c911bd5ba14a74db...`
- If data changes: Hash mismatch detected

Blockchain Application: Each block header contains hash pointer to previous block

---

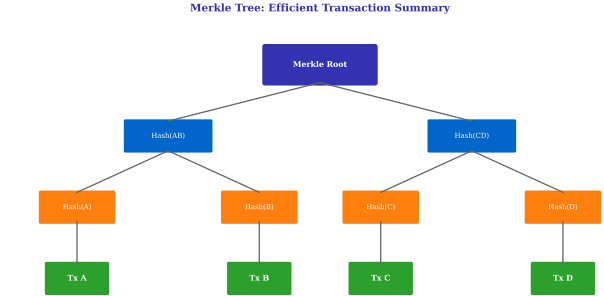
Cryptographic primitives provide the security foundation for blockchain systems.

Why Blockchain is Immutable



Changing one block requires re-hashing all subsequent blocks - impractical against the whole network.

Merkle Trees: Efficient Transaction Verification



Merkle root summarizes all transactions; changing any transaction changes the root.

Merkle Tree Example: 4 Transactions

Constructing Merkle Root:

$$\begin{aligned} \text{Transactions:} \quad & T_1 = \text{“Alice to Bob 1 BTC”} \\ & T_2 = \text{“Carol to Dave 2 BTC”} \\ & T_3 = \text{“Eve to Frank 0.5 BTC”} \\ & T_4 = \text{“Grace to Heidi 3 BTC”} \end{aligned}$$

Step 1 - Leaf Hashes:

$$\begin{aligned} H_1 &= \text{SHA256}(T_1) = \texttt{a3c8...} \\ H_2 &= \text{SHA256}(T_2) = \texttt{7f4b...} \\ H_3 &= \text{SHA256}(T_3) = \texttt{9e2d...} \\ H_4 &= \text{SHA256}(T_4) = \texttt{1c5a...} \end{aligned}$$

Step 2 - Parent Hashes:

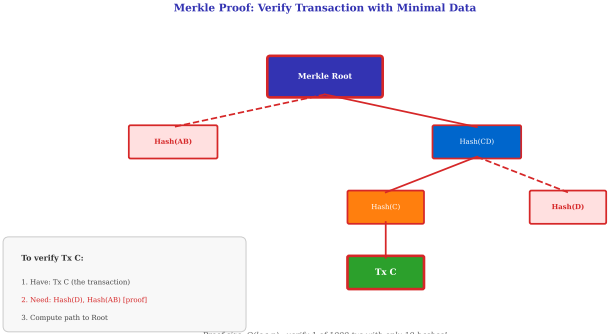
$$\begin{aligned} H_{12} &= \text{SHA256}(H_1 \| H_2) = \texttt{d8f3...} \\ H_{34} &= \text{SHA256}(H_3 \| H_4) = \texttt{6b9e...} \end{aligned}$$

Step 3 - Merkle Root:

$$\text{MerkleRoot} = \text{SHA256}(H_{12} \| H_{34}) = \texttt{4e7a...}$$

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

Merkle Proof: Verifying Transaction Inclusion



Proof size O(log n): verify 1 of 1000 transactions with only 10 hashes.

Difficulty Target and Leading Zeros

Finding a valid hash requires enormous computational effort - like finding a needle in a haystack.

Difficulty Adjustment Mechanism

Goal: Maintain 10 minute average block time despite changing network hash rate

Algorithm:

- Every 2,016 blocks (2 weeks)
- Calculate actual time taken vs. expected time (20,160 minutes)
- Adjust difficulty: $\text{NewDifficulty} = \text{OldDifficulty} \times \frac{\text{Expected Time}}{\text{Actual Time}}$
- Maximum adjustment: 4x increase or 1/4 decrease per period

Example (2023):

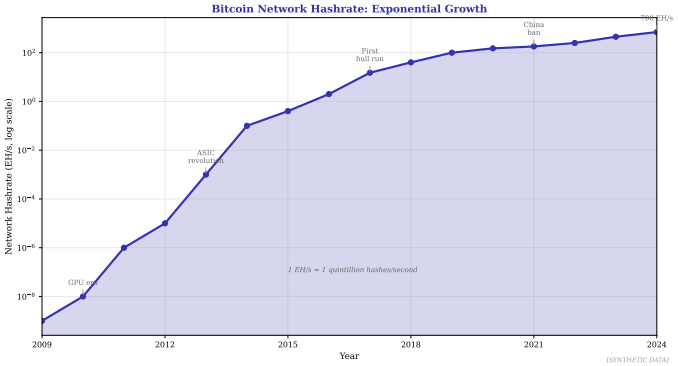
- Period: Blocks 780,000 - 782,015
- Expected: 20,160 minutes (14 days)
- Actual: 18,900 minutes (13.125 days) - blocks came faster
- Adjustment: Difficulty increased by 6.3%

Historical Trend:

- 2009: Difficulty = 1 (CPU mining)
- 2024: Difficulty = 73 trillion (ASIC mining)
- 10^{13} times harder than genesis block

Key concepts from this slide inform practical applications in finance.

Hash Rate: Network Computing Power



Network hashrate has grown exponentially: from CPUs to industrial ASIC farms.

Block Propagation and Orphan Blocks

Problem: Network latency - two miners find valid blocks simultaneously

Scenario:

1. Miner A finds Block 700,000 at 12:00:00
2. Miner B finds different Block 700,000 at 12:00:01
3. Both blocks are valid and propagate through network
4. Network temporarily splits (some nodes see A, others see B)
5. Miners start working on both competing chains
6. Miner C finds Block 700,001 building on Block A
7. Longest chain rule: Block A chain wins
8. Block B becomes "orphan block" - discarded
9. Miner B loses 6.25 BTC reward

Statistics:

- Orphan rate: 0.5% of blocks (rare but happens)
- Average propagation time: 1-2 seconds globally
- Why 10 minutes?: Minimize orphans while maintaining decentralization

Key concepts from this slide inform practical applications in finance.

Blockchain Data Structures Summary

Structure	Purpose	Properties
Hash Function	Fingerprint data	Deterministic, one-way, collision-resistant, avalanche effect
Hash Pointer	Link + verify	Points to data + contains hash for integrity
Merkle Tree	Efficient verification	$O(\log n)$ proof size, enables SPV
Block Header	Minimal representation	80 bytes contains all block metadata
Blockchain	Tamper-evident ledger	Chain of hash pointers, immutable
Difficulty Target	Mining puzzle	Adjusts to maintain 10-min block time

Key Insight: All blockchain security stems from cryptographic hash functions - if SHA-256 breaks, Bitcoin breaks

Security Analysis: What Could Go Wrong?

Hash Function Attacks:

- Pre-image: Find input from hash
- Collision: Find two inputs with same hash
- Length extension: Exploit hash construction

SHA-256 Status:

- Pre-image: 2^{256} security (safe)
- Collision: 2^{128} security (safe)
- Length extension: Not applicable to Bitcoin (double hashing)

Quantum Computing Threat:

- Grover's algorithm: $\sqrt{2^{256}} = 2^{128}$ speedup
- Still computationally infeasible
- Bigger threat: Public key crypto (covered next lesson)

Mitigation Strategies:

- SHA-3 ready as backup
- Post-quantum hash functions (SPHINCS+)
- Bitcoin can hard fork if needed

Current Assessment: No practical threat to SHA-256 in foreseeable future

Practical Exercise: Computing Block Hash

Simplified Example: Calculate hash for mini-block

Block Header (simplified):

- Previous Hash: `0000000000000000001`
- Merkle Root: `abcdef123456`
- Timestamp: `1702000000`
- Nonce: ?

Task: Find nonce such that hash starts with 4 zeros (`0000...`)

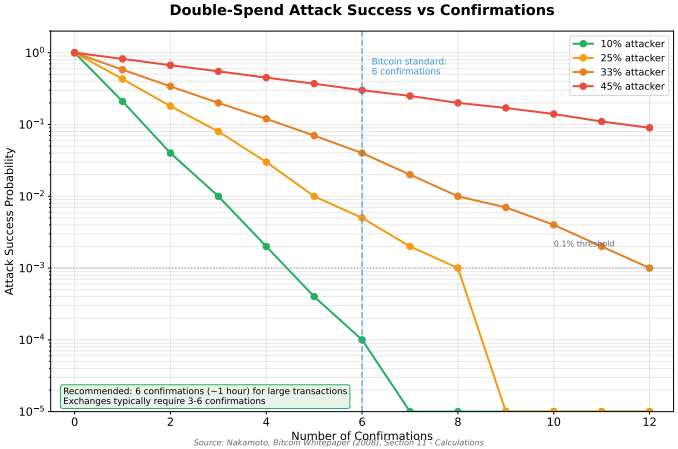
Brute Force Approach:

1. Concatenate header: `0000000000000000001|abcdef123456|1702000000|0`
2. Compute SHA-256: `7a3b2c1d...` (no leading zeros)
3. Increment nonce: Try 1, then 2, then 3...
4. Keep trying until hash starts with `0000`
5. Expected attempts: $2^{16} = 65,536$

Solution: Nonce `45,892` produces `0000f7a3b2c1d...`

Cryptographic primitives provide the security foundation for blockchain systems.

Block Finality: How Many Confirmations?



More confirmations = harder to reverse. 6 confirmations is the gold standard for Bitcoin.

Lesson 15: Public Key Cryptography and Signatures

What We'll Cover:

- Symmetric vs asymmetric encryption
- Elliptic Curve Cryptography (ECC) fundamentals
- Public/private key pairs in Bitcoin
- Digital signatures (ECDSA)
- Address generation and wallet structure
- Security best practices

Prepare:

- Review basic modular arithmetic ($a \bmod n$)
- Understand concept of mathematical groups
- Install Bitcoin wallet for hands-on key generation

Summary: Key Takeaways

1. **Block Structure:** 80-byte header + variable transaction body
2. **SHA-256:** One-way, collision-resistant hash with 2^{256} output space
3. **Avalanche Effect:** Tiny input change completely alters hash
4. **Hash Pointers:** Enable tamper-evident data structures
5. **Merkle Trees:** Efficient verification with $O(\log n)$ proof size
6. **Immutability:** Changing old block requires recalculating all subsequent blocks
7. **Difficulty:** Adjusts every 2016 blocks to maintain 10-min average
8. **Finality:** 6 confirmations (1 hour) considered irreversible

"Blockchain security is reducible to hash function security - SHA-256 is the foundation."