Blockchain Fundamentals

Lesson 2: Summary

Prof. Joerg Osterrieder

Spring 2026

What is a Blockchain?

Definition: A distributed, append-only data structure where blocks are linked using cryptographic hashes.

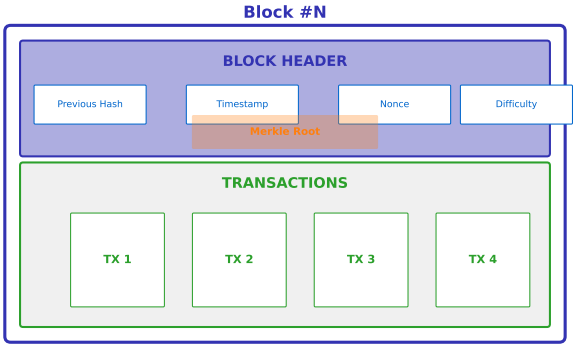
Key Properties:

- **Distributed**: Copies across many nodes
- **Append-only**: New data added, never removed
- **Cryptographically linked**: Hash chain
- **Tamper-evident**: Changes immediately detectable

Blockchain = distributed ledger + cryptographic guarantees

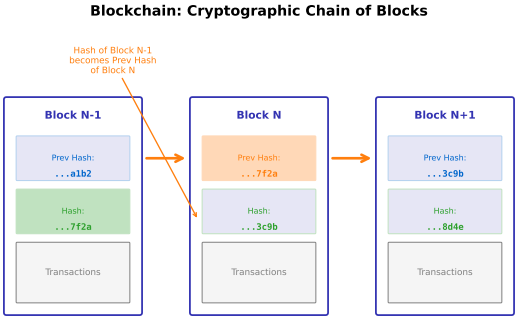
Block Structure

Anatomy of a Blockchain Block



Each block contains: Header (prev hash, timestamp, nonce) + Transactions

The Hash Chain



Each block contains the hash of the previous block, creating immutability

Why Tampering Fails

If you change Block 100:

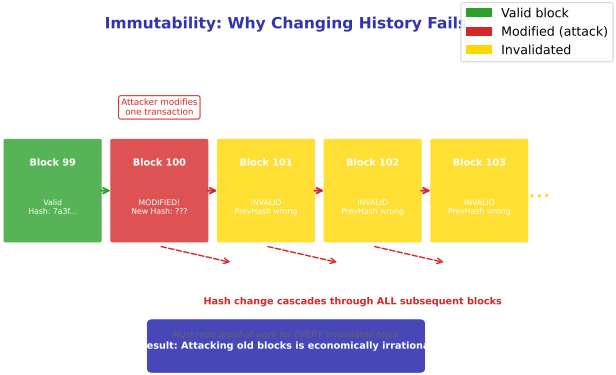
1. Block 100's hash changes (avalanche effect)
2. Block 101's "Previous Hash" no longer matches
3. Must redo Block 101's proof-of-work
4. Then Block 102, 103... all the way to current block

Result: Attacker needs more computing power than entire honest network.

Key Insight: The deeper a block, the more secure it becomes.

Economic cost of attack exceeds potential benefit

Visualizing Immutability



Modifying history requires redoing all subsequent proof-of-work

Merkle Trees

What: Binary tree of transaction hashes with single root in block header.

Why:

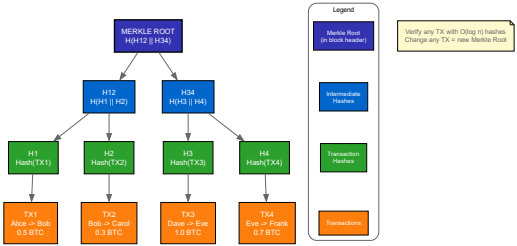
- Verify any transaction with $O(\log n)$ hashes
- Enable "light clients" without full blockchain
- Efficient proof that transaction is in block

How:

$$\text{Root} = H(H(H(TX_1)||H(TX_2))||H(H(TX_3)||H(TX_4)))$$

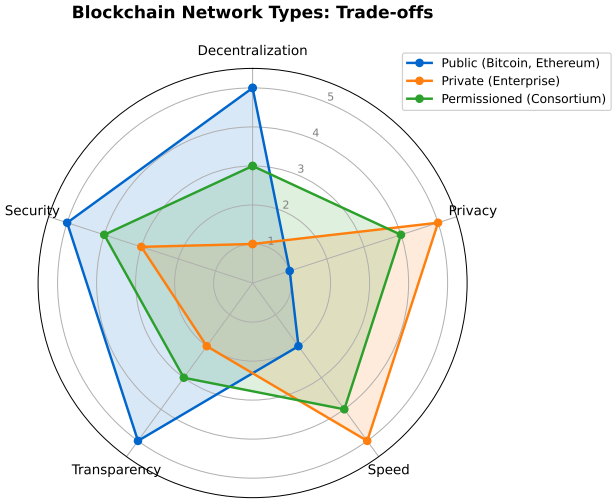
Merkle trees enable efficient verification on resource-constrained devices

Merkle Tree Structure



Changing any transaction changes the Merkle Root

Blockchain Types



Public (open), Private (single org), Permissioned (consortium)

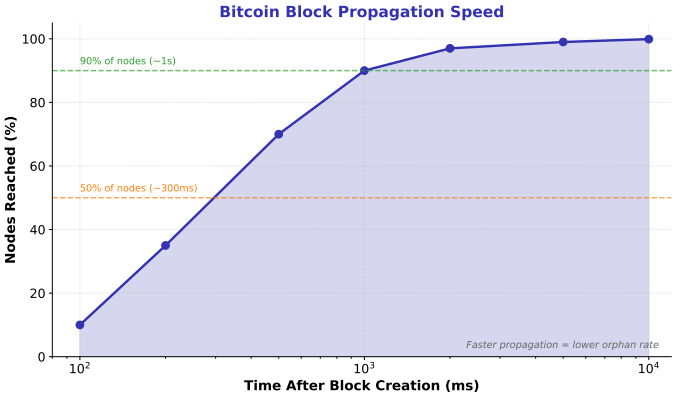
Public vs Private vs Permissioned

Feature	Public	Private	Permissioned
Access	Open	Restricted	Selected
Speed	Slow	Fast	Medium
Decentralization	High	Low	Medium
Privacy	Low	High	Medium
Examples	BTC, ETH	Hyperledger	Corda

Choose based on: Trust requirements, throughput needs, privacy constraints.

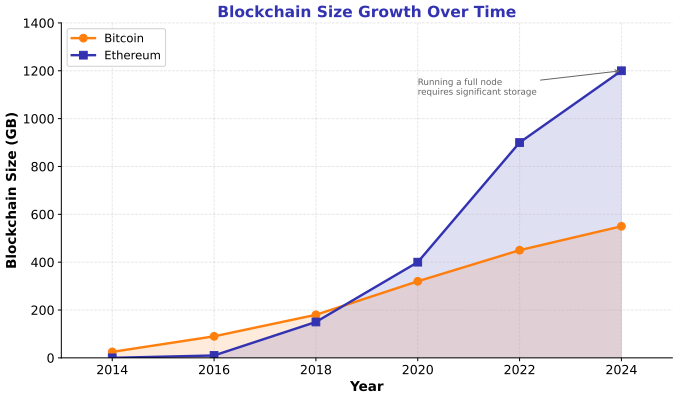
No single type is "best"—depends on use case

Block Propagation Timing



Propagation time affects fork probability and network security

Blockchain Size Growth



Full nodes require significant storage—drives centralization concerns

Key Takeaways

Remember These Points:

1. Blocks = Header (metadata) + Transactions (data)
2. Hash chain creates immutability (change one = redo all)
3. Merkle trees enable efficient verification
4. Three types: Public, Private, Permissioned

Core Insight:
Blockchain achieves immutability through cryptographic hashing + economic incentives.

Next: Cryptographic Foundations

Thank You

Questions?