THE GRAINGER COLLEGE OF ENGINEERING

CS 521

Technological Foundations of Blockchain and Cryptocurrency

Grigore Rosu

Topic 3 – Bitcoin

ILLINOIS

Early Cryptographic Digital Currencies … All Failed

- DigiCash (David Chaum) - 1989
- Mondex (National Westminster Bank) - 1993
- CyberCash (Lynch, Melton, Crocker & Wilson) - 1994
- E-gold (Gold & Silver Reserve) - 1996
- Hashcash (Adam Back) - 1997
- Bit Gold (Nick Szabo) - 1998
- B-Money (Wei Dai) - 1998
- Lucre (Ben Laurie) - 1999

Why did Early Digital Currencies Fail?

- Merchant adoption
- Centralization
- Double spending
- Consensus

Double Spend attack

- A simple attack:
  - When one person can use the same coin multiple times to buy things
- Easily solved with a centralized system
- Much harder when decentralized

The Riddle Remained

How to move value peer-to-peer
without any trusted central intermediary

Bitcoin: A Peer-to-Peer Electronic Cash System

From: Satoshi Nakamoto <satoshi <at> vistomail.com>

Subject: Bitcoin P2P e-cash paper

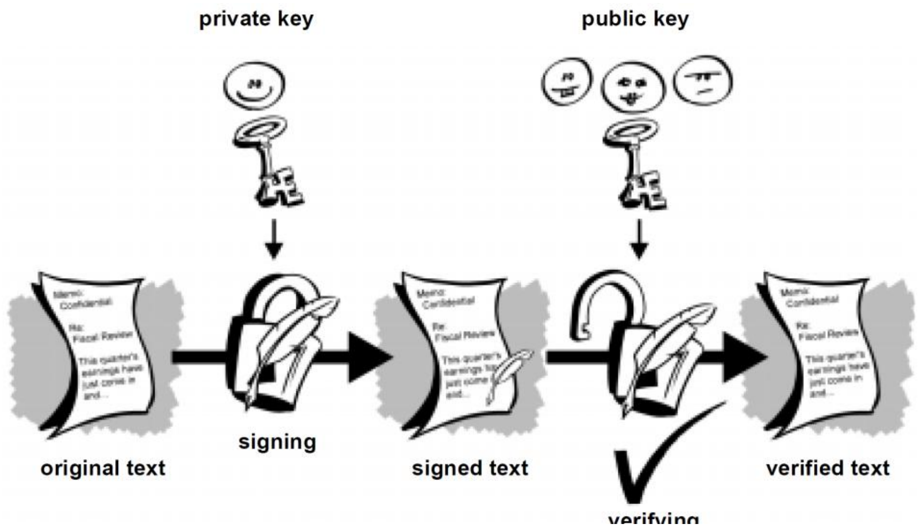
Newsgroups: gmane.comp.encryption.general

Date: Friday 31st October 2008 18:10:00 UTC

I've been working on a new electronic cash system that's fully peer-to-peer, with no trusted third party.

Bitcoin's Goal

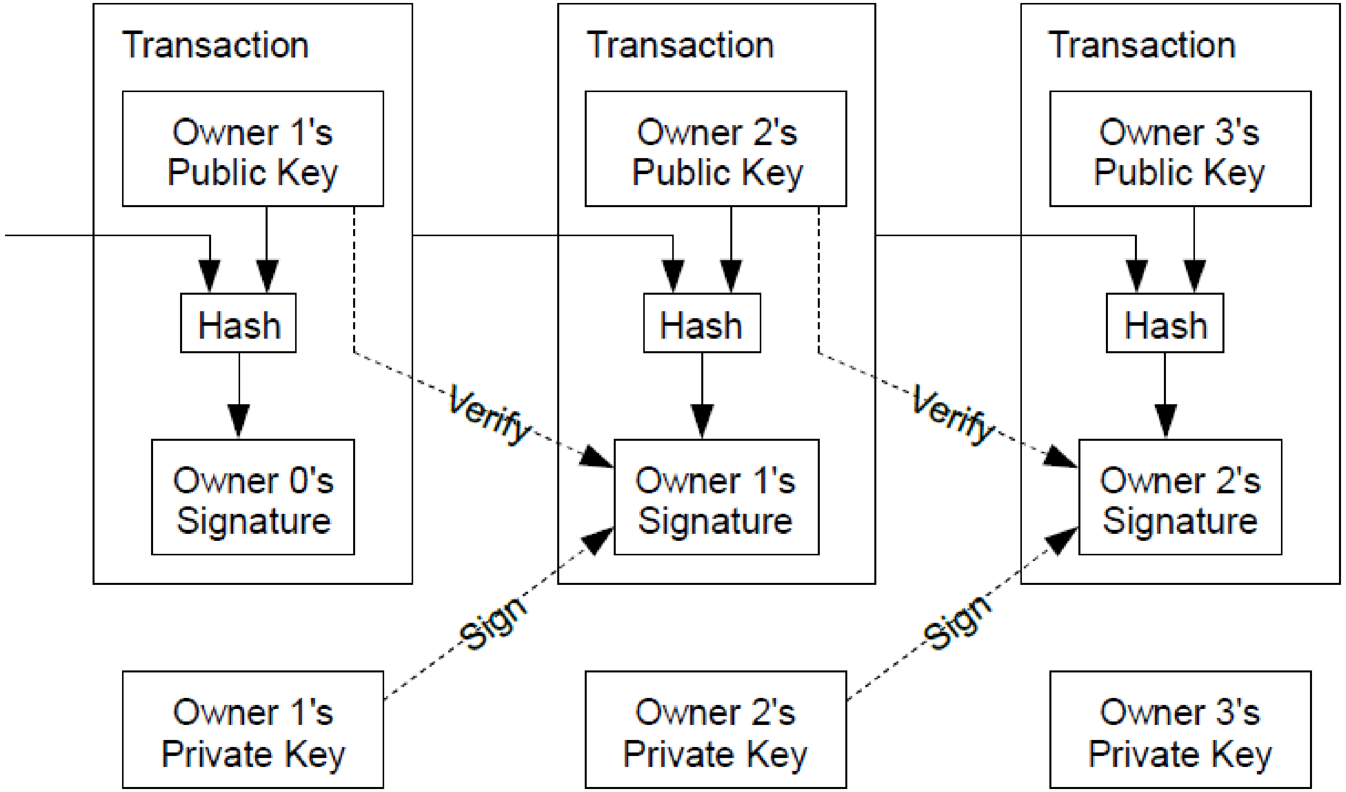
- Peer-to-peer digital money
  - No intermediaries (banks, centralized parties)
- Digital signatures were clearly the right direction



  - But the double-spending riddle could not be solved
  - Centralized / intermediary can solve double-spending, but defies the purpose

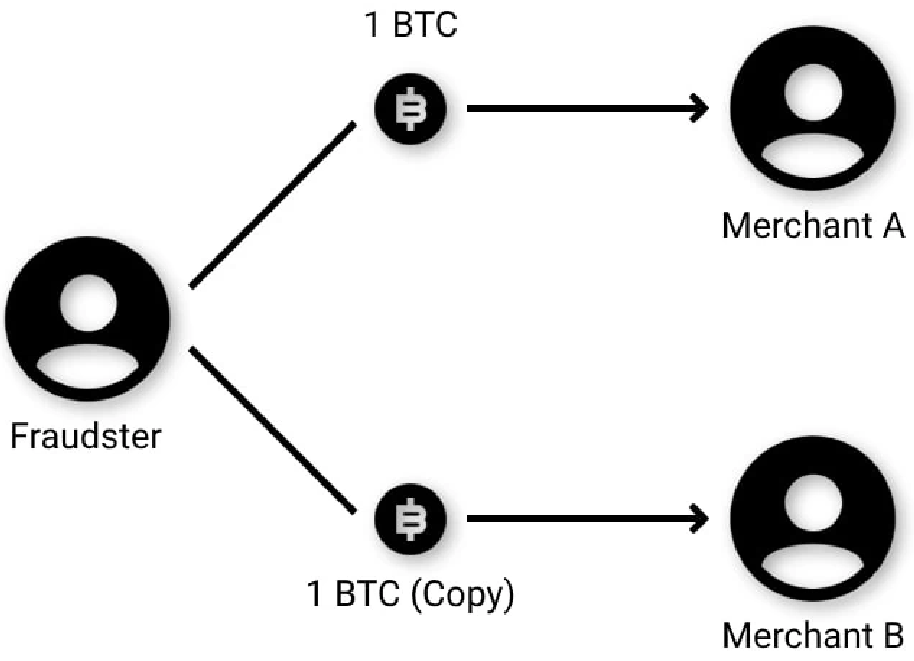
Coins and Transactions

- Coins and transfers
  - Coin: chain of digital signatures (hash pointer list data-structure)
  - Transfer C from A to B: C' = sign(A, hash(C,B))



Double Spending

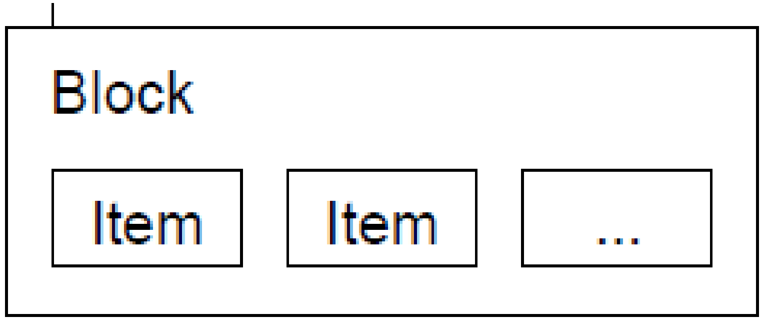
So far nothing to stop a fraudster to send the same coin to two parties



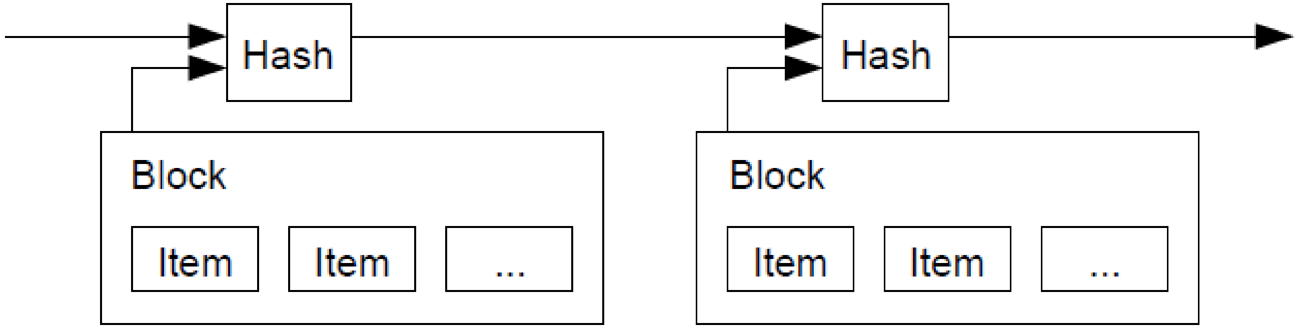
Solution: enforce with a very high probability, a total order on transactions

Timestamp Servers: Blocks and Total Orders

- Put each transaction in a totally-ordered block of transactions
  - The only way transactions get communicated



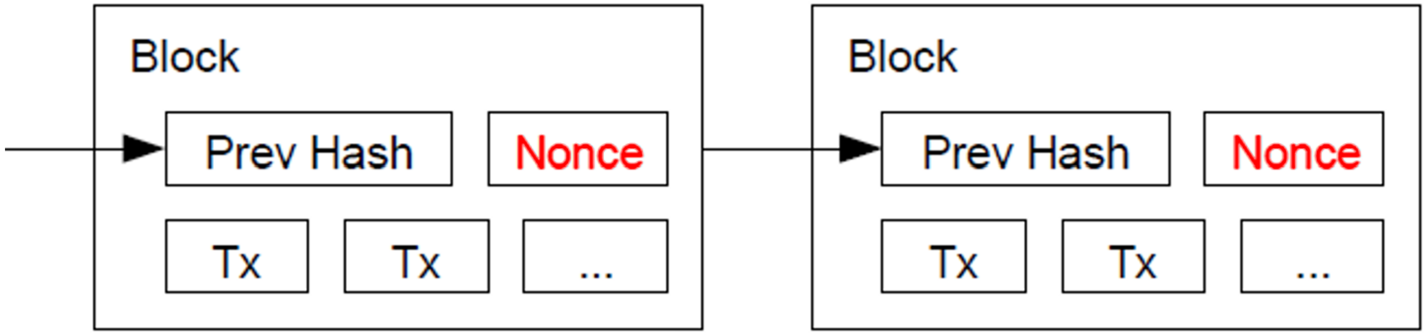
- Linked list of hash pointers to enforce total order on blocks:



- Problem of divergence remains, but lifted from transactions to blocks
- Need to enforce total order on blocks; in a decentralized way

Proof-of-Work

- Idea
  - Have only one block proposer at a time, but who?
  - Whoever solves a puzzle generation challenge first (recall previous lecture)
  - Requires computational power, so solving it is … **proof of work**



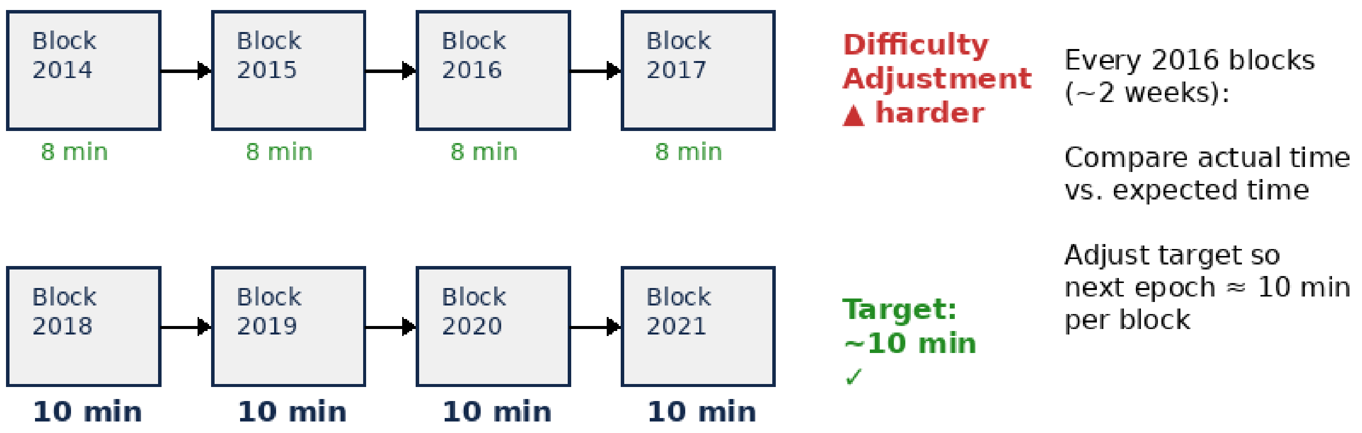
- Specifically, the puzzle challenge is to find **Nonce** in the block such that the hash of the block is smaller than an epsilon (hash starting with zeros)
- Easy to verify that such Nonce is indeed correct, so all nodes update chain
- Problem of divergence still exists in theory; but nodes pick longest chain

Decentralized Network

- Nodes can join and leave. They all execute the same protocol:
    1. New transactions are broadcast to all nodes
    2. Each node collects new transactions into a block
    3. Each node works on finding a difficult proof-of-work for its block
    4. When a node finds a proof-of-work, it broadcasts the block to all nodes
    5. Nodes accept the block only if all transactions in it are valid and not already spent
    6. Nodes express their acceptance of the block by working on creating the next block in the chain, using the hash of the accepted block as the previous hash
- Nodes always consider the **longest chain**
- Economic incentive:
    - First transaction in the block (also called coinbase), is a new coin owned by the block creator / proposer / miner

Mining Economics: Difficulty Adjustment

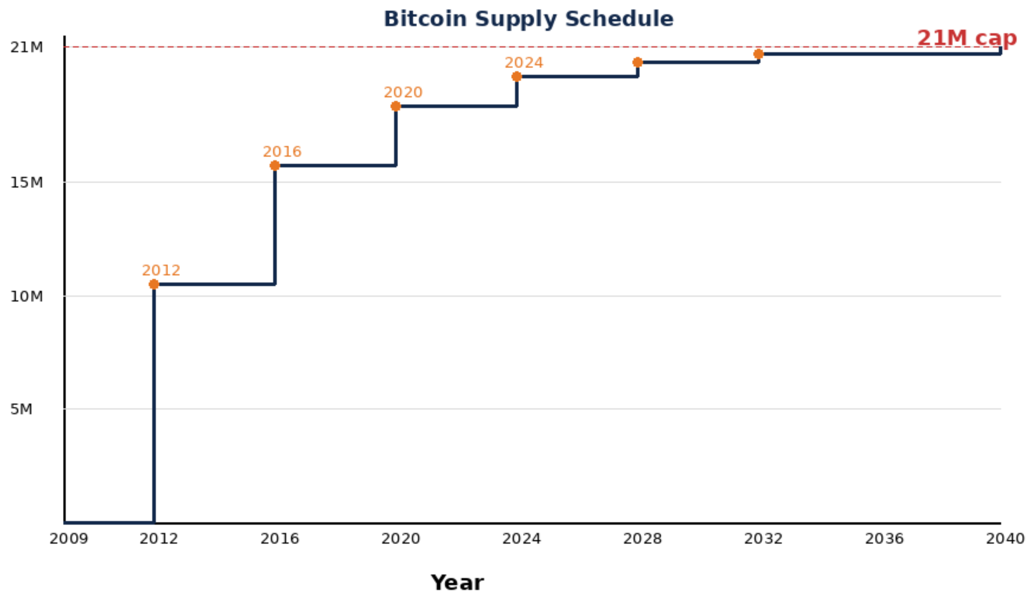
- Target: one block every ~10 minutes
  - Too fast? Mining is too easy. Too slow? Network unusable



- Difficulty adjusts every 2016 blocks (~2 weeks)
  - Compare actual time vs. expected time (2016 × 10 min)
  - If blocks came too fast → increase difficulty (smaller target)
  - If blocks came too slow → decrease difficulty (larger target)
- Self-regulating feedback loop
  - Works regardless of total hash power in the network

Bitcoin Supply and Halving

- Fixed supply cap: 21 million BTC
- Block reward halves every 210,000 blocks (~4 years)
  - 2009: 50 BTC → 2012: 25 → 2016: 12.5 → 2020: 6.25 → 2024: 3.125

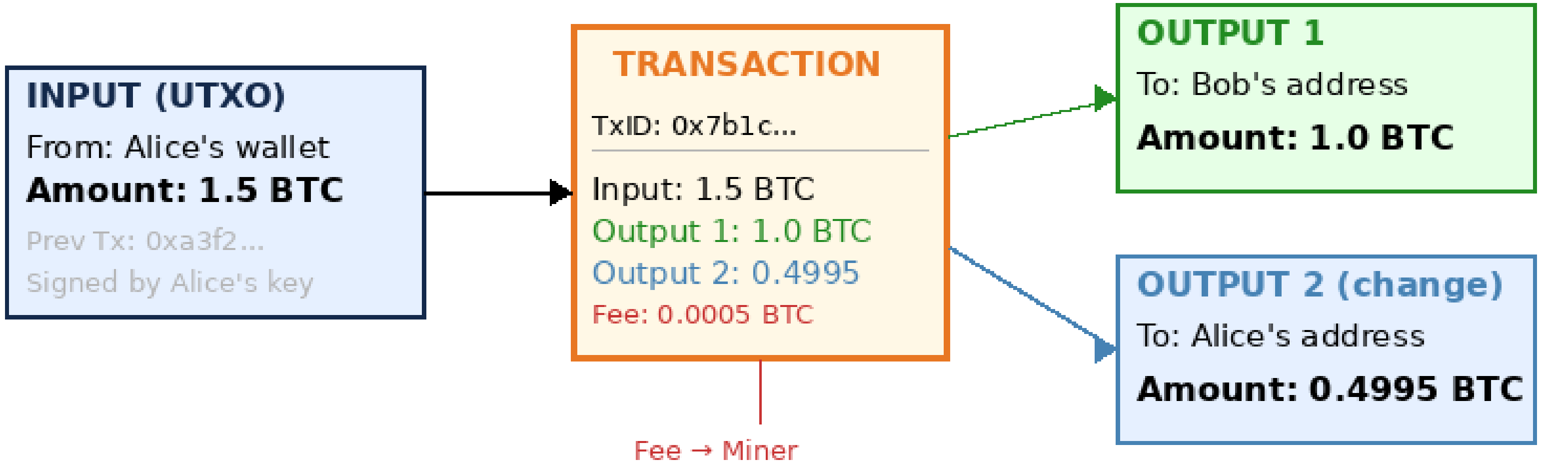


- Last bitcoin mined in year ~2140
- Deflationary by design
  - After all BTC mined, miners earn only transaction fees
- Contrast with fiat: central banks can print unlimited money

A Concrete Transaction: The UTXO Model

Bitcoin does not use account balances

- It uses Unspent Transaction Outputs (UTXOs)
- Like paying with bills: you hand over a $20, get change back

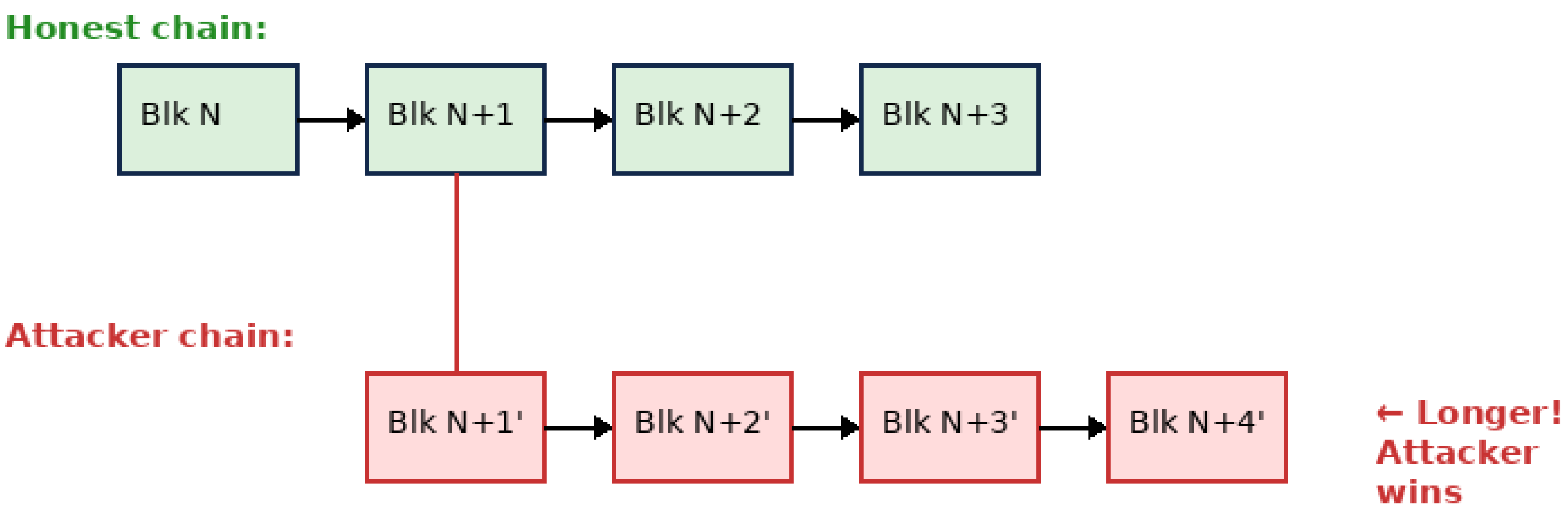


UTXO Model (Unspent Transaction Output)

- Coins are not accounts with balances — they are unspent outputs from previous transactions
- To spend, you consume entire UTXOs as inputs and create new UTXOs as outputs
- Change goes back to yourself as a new UTXO (like getting change from a $20 bill)
- Transaction fee = sum(inputs) - sum(outputs) — goes to the miner who includes the tx

Security: The 51% Attack

- What if an attacker controls >50% of the network hash power?
  - Can build a longer chain, rewriting history and **reversing transactions**
- Defense: wait for multiple confirmations before trusting a transaction



With >50% hash power, attacker can build a longer chain, reversing confirmed transactions

At 10% hash power: 0.02% chance to reverse 6 confirmations | At 30%: 17.7% | At 50%+: certain

This is why merchants wait for multiple confirmations (typically 6 blocks ≈ 1 hour)

Genesis Block and Bitcoin Milestones

- Satoshi Nakamoto mined the first block on January 3, 2009
- The coinbase parameter contained a message that became legendary:

The Times **Chancellor on brink of second bailout for banks**
03/Jan/2009

Genesis Block (Block #0)

Hash: 000000000019d6689c085ae165831e934ff763ae46a2a6c172b3f1b60a8ce26f

Timestamp: 2009-01-03 18:15:05 UTC

Coinbase: "The Times 03/Jan/2009 Chancellor on brink of second bailout for banks"

Reward: 50 BTC (unspendable — hardcoded special case)

Bitcoin Milestones

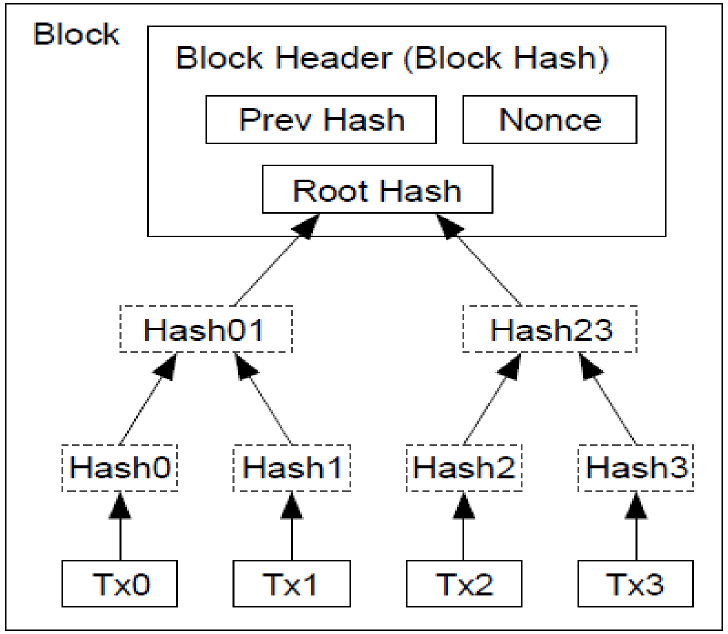
▯ May 22, 2010: Laszlo Hanyecz pays 10,000 BTC for two pizzas (~$41 at the time)

▯ First known Bitcoin price: $0.0009 (Oct 2009, New Liberty Standard)

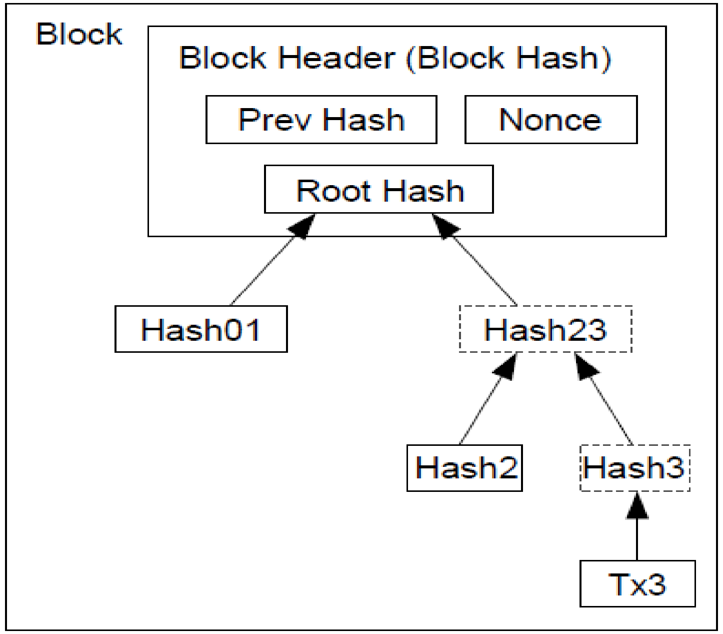
▯ Satoshi Nakamoto's identity remains unknown — last public message: April 2011

Reclaiming Space

- Transactions need not be stored on-chain permanently
- Stored in Merkle tree, whose root hash is in the block header
- Once buried under enough blocks, spent transactions can be pruned
  - Only the Merkle root is needed to verify the block’s integrity
  - A block header is only ~80 bytes vs. ~1 MB for a full block



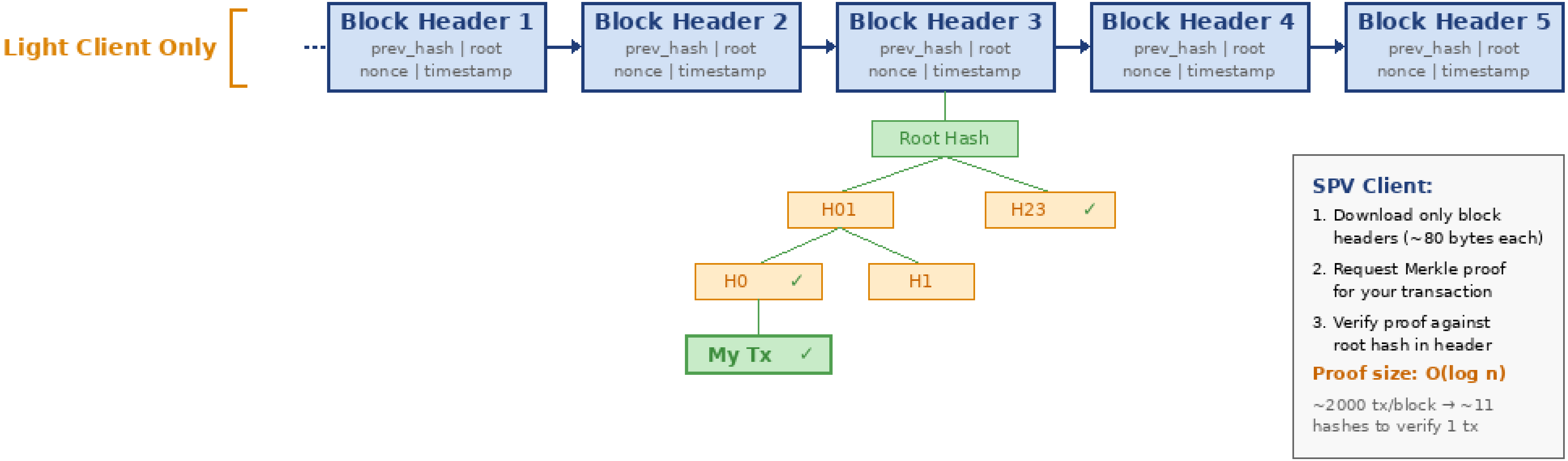
Transactions Hashed in a Merkle Tree



After Pruning Tx0-2 from the Block

Simplified Payment Verification (SPV)

- Lightweight clients verify payments without downloading full blockchain
  - Download only block headers (~80 bytes each, vs ~1 MB per full block)
  - Request a Merkle proof for the specific transaction from a full node
- Proof size is O(log n) – only ~11 hashes to verify 1 tx among ~2000



How It All Fits Together

From a signed transaction to an immutable, decentralized ledger:

Transactions → **Mempool** → **Block Assembly** → **Proof-of-Work** → **Broadcast**

- **Transactions**: Alice signs & sends to Bob (UTXO model)
- **Mempool**: Unconfirmed transactions wait here
- **Block Assembly**: Miner collects txs + coinbase reward
- **Proof-of-Work**: Find nonce: H(block) < target (~10 min)
- **Broadcast**: Winning block sent to all nodes

The Blockchain:

Block #0 → **Block #1** → **Block #2** → **Block #3** → **Block #4**

Each block — Header: prev_hash, merkle_root, nonce | time

Security Properties

- **Immutability**: Changing old blocks requires redoing all subsequent PoW
- **Decentralized**: No single point of failure; any node can verify
- **Double-Spend Resistant**: Longest chain wins; attacker needs >50% power
- **Incentive-Aligned**: Miners profit from honest behavior (block reward + fees)